Contributions à l'extraction et à la représentation des règles d'association par graphes et arbres hiérarchiques

Parfait Bemarisika

Université de Toamasina
Ecole doctorale thématique Science, Culture, Société et Développement (SCSD)
Equipe d'Accueil Mathématiques, Informatique et Applications (MIA)

MEMOIRE DE SYNTHESE

en vue de l'obtention de

l'Habilitation à Diriger des Recherches

Spécialité Mathématiques et Informatique

par

Docteur Bemarisika Parfait

Contributions à l'extraction et à la représentation des règles d'association par graphes et arbres hiérarchiques

Soutenu publiquement le 17 avril 2023

devant le jury :

RAKOTONDRABE Daniela Tovonirina	Professeur, Université de Toamasina	Président
RÉGNIER Jean-Claude	Professeur Emérite, Université de Lyon 2	Rapporteur externe
COUTURIER Raphaël	Professeur, Université de Franche-Comté	Rapporteur externe
MAHATODY Thomas	Maître de Conférences HDR, Université de Fianarantsoa	Rapporteur interne
RAVELONIRINA Hanitriniaina Sammy Grégoire	Professeur, Université d'Antananarivo	Examinateur
RAKOTOARIVELO Rivo Andry	Professeur, Université de Fianarantsoa	Examinateur
RAHERINIRINA Angelo Fulgence	Professeur, Université de Fianarantsoa	Examinateur
TOTOHASINA André	Professeur Titulaire, Université d'Antsiranana	Garant scientifique

Remerciements

Au terme de ce parcours en vue de l'obtention de l'Habilitation à Diriger des Recherches (HDR), mes remerciements vont tout d'abord à mon Garant scientifique, Monsieur TOTOHASINA André, Professeur titulaire à l'Université d'Antsiranana (Madagascar) qui me fait confiance sur la maturité scientifique au regard de la qualité de mes travaux sur l'aptitude à maîtriser une stratégie de recherche et sur la capacité à encadrer de jeunes chercheurs. Il a été très disponible pour m'accompagner dans différents projets de recherche, dans le co-encadrement des thèses et dans la lecture des premières versions de ce mémoire. Je lui remercie également de m'avoir guidé dans la préparation de mon dossier et dans le suivi des démarches administratives pour ma soutenance.

Je remercie chaleureusement Monsieur RAKOTONDRABE Daniela Tovonirina, Professeur à l'Université de Toamasina (Madagascar), Directeur de l'Ecole Doctorale Thématique SCSD pour m'avoir fait l'honneur de présider la soutenance de ce mémoire.

J'adresse mes vifs remerciements à mes rapporteurs, Monsieur RÉGNIER Jean-Claude, Professeur émerite à l'Université Lumière de Lyon 2 (France); Monsieur COUTURIER Raphaël, Professeur à l'Université de Franche-Comté, Institut Femto-ST (France); et Monsieur MAHATODY Thomas, Maître de Conférences HDR à l'Université de Fianarantsoa (Madagascar). Leurs commentaires précieux me permettront de mener à bien la suite de ces travaux.

Je tiens à exprimer mes chaleureux remerciements à Monsieur RAVELONIRINA Hanitriniaina Sammy Grégoire, Professeur à l'Université d'Antananarivo (Madagascar); Monsieur RAKOTOARIVELO Rivo Andry, Professeur à l'Université de Fianarantsoa (Madagascar); et Monsieur RAHERINIRINA Agelo Fulgence, Professeur à l'Université de Fianarantsoa (Madagascar), qui ont accepté d'être membres du jury en tant qu'examinateurs. Leurs critiques constructives très précieuses me permettront également de mener à bien la suite de ces travaux.

Je tiens à remercier chaleureusement toutes les personnes avec qui j'ai eu l'occasion de collaborer. Je citerai en particulier Docteur RAMANANTSOA Harrimann, Maître de Conférence à l'Institut Supérieur de Technologies d'Antsiranana, avec les discussions passionnées sur l'aspect algorithmique de l'extraction des motifs fréquents; Monsieur JERSON Aurélien pour les échanges fructueux sur l'algorithme d'arbres hiérarchiques.

Je souhaite également remercier de tout cœur tous les étudiants avec qui j'ai eu l'honneur de travailler. Pour le passé, merci Dewars, Alain, François, Elrish, Dolphe et Debon; pour le présent, merci Sarah Maëva, Justin et Rufin.

En dernier lieu dans cette longue liste de remerciements, mais ce n'est pas le moindre, je tiens à remercier mes parents pour m'avoir offert la possibilité de poursuivre des études supérieures jusqu'à l'obtention de doctorat. Cette HDR, qui n'a bien sûr pas la même valeur symbolique à leurs yeux, est néanmoins le fruit de leur travail quotidien.

Résumé

Les travaux présentés concernent l'extraction de connaissances dans une base de données binaire $\mathcal{D}$. Nous y présentons différentes contributions. Dans un premier temps, nous abordons une nouvelle méthode permettant l'extraction simultanée des itemsets fermés fréquents, maximaux fréquents, et leurs générateurs minimaux de $\mathcal{D}$. Dans un second temps, des techniques originales permettant la génération des règles d'association valides sont également développées. Des expériences sur des données bien connues de la littérature confirment leur potentiel. Enfin, nous présentons des approches permettant la représentation de ces règles valides par des graphe et arbre. Des expérimentations menées sur des données de référence, à des résolutions différentes, montrent l'efficacité de nos approches tant en fouille qu'en classification de données.

Mots clés: Motif fréquent, Règle d'association, Classification, Graphe implicatif, Arbre hiérarchique.

Abstract

The presented work concerns knowledge discovery in binary database $\mathcal{D}$. We then present different contributions. First, we present a new method for mining simultaneously the frequent closed itemsets, the frequent maximal itemsets, and their minimal generators in $\mathcal{D}$. Secondly, original techniques for generating the valid association rules are also developed. Experiments on reference database show the potential of these new techniques. Finally, we present the approaches for representing these valid association rules by graphs and trees. Experiments, at different resolutions, show the effectiveness of our approaches both on data mining and on classification.

Keywords: Frequent itemset, Association rule, Classification, Implicative graph, Hierarchy tree.

Famintinana

Ny asa naroso dia mahakasika ny fitrandrahana hairaha avy amin'ny antontanisa roalafy, antsoina hoe $\mathcal{D}$. Anjara biriky maro samihafa no natolotra. Voalohany, ny teknika vaovao enti-mitrandraka ny singa mateti-pitranga ao anaty $\mathcal{D}$. Faharoa, ny fomba fijery vaovao momba ny fitrandrahana ny fitsipi-pikambanan'ny singa samy hafa izay tsara indrindra araka ny refin-kalitao. Ny fanandramana nampiharana ny antontanisa vitsivitsy, fampiasa matetika eo anivon'ny literatiora, dia manamafy fa mahomby tokoa izany teknika vaovao izany. Farany, natolotra etoana ihany koa ny modely vaovao hanehoana ara-tsary, atao hoe grafim-panjotra sy hazo maro rantsana, ny vondron'ny fitsipi-pikambanana raikitra. Ny andrana natao tamin'ny antontanisa teo aloha sy ny hafa, ka nanomezana tolo-kevitra samihafa ny amin'ny kalitaon'ny vokatra azo, dia mampiseho hatrany ny fahombiazan'ny voka-pikarohana naroso, fa indrindra eo amin'ny sehatry ny fitrandrahana antontanisa na koa fanasokajiana azy.

Teny mafonja: Singa mateti-pitranga, Fitsipi-pikambanana, Fanasokajiana, Grafy, Hazo maro-rantsana

Equipe d'Accueil:	Mathématiques, Informatique et Applications, Univ. Toamasina (Madagascar)
Auteur:	Bemarisika Parfait, Maître de Conférences, Université d'Antsiranana (Madagascar)
Contacts personnels:	Tel. +261 (0)32 41 231 43, E-mail: bemarisikap7@yahoo.fr
Liste tab/fig/algo:	Liste des tableaux: 09, Liste des figures: 16, Liste des algorithmes: 18
Garant scientifique:	Totohasina André, Professeur Titulaire, Université d'Antsiranana (Madagascar).

Table des matières

Liste des tableaux

Table des figures

Liste des algorithmes

Travaux présentés dans ce mémoire

Les travaux que je vais présenter dans cette notes de synthèse se déclinent dans le cadre de collaborations académiques et d'encadrements de masters recherche (ou indifférencié). D'autres travaux moins en rapport avec les thèmes de ce manuscrit ne sont pas présentés dans cette liste, ni dans le manuscrit mais indiqués dans l'autre liste pour d'autres travaux de l'autreur ci-dessous. La liste respecte l'ordre chronologique de publication des articles mais pas celui de leurs réalisations. La numérotation de la bibliographie générale est aussi rappelée pour chacun des articles.

Revues internationales avec comité de lecture

1. P. Bemarisika, et A. Totohasina, *Visualisation interactive des graphes implicatifs*. REVUT Scientific Journal (RSJ'2020), Vol. 3, DOI : https ://doi.org/10.46857/rsj.2021.3 [BT20f].
2. P. Bemarisika, et A. Totohasina, *An Efficient Method for Mining Informative Association Rules in Knowledge Extraction*. MAKE, pp.227-247, Springer 2020 [BT20e].
3. P. Bemarisika and A. Totohasina, *Elimination of Redundant Association Rules*. Information Systems Architecture and Technology (ISAT), pp.208-218, Springer 2019 [BT19b].
4. P. Bemarisika, H. Ramanantsoa, and A. Totohasina, *An Efficient Approach for Extraction Positive and Negative Association Rules from Big Data*. MAKE, pp.79-97, 2018 [BRT18].

Conférences internationales avec comité de lecture

5. P. Bemarisika, and A. Totohasina, CONCISE *: An Algorithm for Mining Positive and Negative Non-Redundant Association Rules*, In Pal R., and K.P. Shukla (eds), SCRS Conference Proceedings on Intelligent Systems, pp. 13-34, 2021 [BT21a].
6. P. Bemarisika, and A. Totohasina, *Generating a Condensed Representation for Positive and Negative Association Rules*, In International Conference on Business Information Systems (BIS), pp. 175-186, Springer, 2021 [BT21b].
7. P. Bemarisika, R. Couturier, et A. Totohasina, *Une Construction condensée et interactive d'un graphe implicatif*, In J-C. Régnier et al.(Eds), ASI'2021, pp. 199-224, 2021 [BCT21].

8. P. Bemarisika, et A. Totohasina, *Concise Representations for Positive and Negative Association Rules*, In Intnl Conf. on Fuzzy Systems and Knowledge Discovery (FSKD), 2020 [BT20a].
9. P. Bemarisika, et A. Totohasina, *Consise Extraction for Informative Positive and Negative Association Rules*, ACM International Conference on Information Integration and Web-based Applications & Services (iiWAS'20) [BT20b].
10. P. Bemarisika, et A. Totohasina, *An Efficient Method for Mining Informative Association Rules in Knowledge Extraction*, In Intnal Conf. on CD-MAKE, pp. 227-247, 2020 [BT20c].
11. P. Bemarisika, et A. Totohasina, *NONRED, An efficient algorithm for mining non-redundant rules in Big Data*, In Intnl Conf. on Big Data and KD, DaWaK 2020 [BT20d].
12. P. Bemarisika, and A. Totohasina, *An Informative Base of Positive and Negative Association Rules on Big Data*, In IEEE Int. Conf. on Big Data, pp. 2428-2437, Springer, 2019 [BT19a].
13. P. Bemarisika, et A. Totohasina, *Nouvelles bases des règles d'association non-redondantes*, Conf. Int. sur Société Francophone de Classification (SFC), pp. 77-82, 2019 [BT19c].
14. P. Bemarisika, and A. Totohasina, *ERAPN, An Algorithm for Extraction Positive and Negative Association Rules in Big Data*, Big Data and KD, pp. 329-344, Springer, 2018 [BT18].
15. P. Bemarisika, and A. Totohasina, *Optimized Mining of Potential Positive and Negative Association Rules*, In Intnl Conf on Big Data and KD, pp. 424-432, Springer, 2017 [BT17].
16. P. Bemarisika, et A. Totohasina, *Optimisation de l'extraction des règles d'association positives et négatives*, Société Francophone de Classification (SFC), pp. 25-28, 2017 [BT17].

Workshop International avec comité de lecture

17. P. Bemarisika, et A. Totohasina, *Visualisation interactive des graphes d'une règle d'association informative*, Workshop VIF (Visualisation d'informations, Interactions et Fouille de données), en conjonction avec *EGC'2020*, pp. 5-6, 2020 [BT20e].

Articles soumis et/ou en préparation

18. P. Bemarisika, A. Jerison, et A. Totohasina, *Une méthode de construction d'arbres de classification hiérarchique implicative*. Soumis et accepté à la Conf Intle sur SFC'2022 [BJT22a].
19. P. Bemarisika, A. Jerson, et A. Totohasina, *Une méthode simultannée pour la compression, le partitionnement et la construction d'un graphe implicatif*. Soumis et accepté à la Conférence Internationale sur Société Francophone de Classification (SFC'2022) [BJT22b].
20. P. Bemarisika, H. Ramanantsoa, et A. Totohasina, *Passage à l'échelle de l'algorithme CMG*. Preprint au Laboratoire Mathématiques et Informatique, ENSET-Université d'Antsiranana.

Développement d'outils logiciels

21. Package `rchicmgk` : En collaboration avec Raphaël COUTURIER (Université de Franche-Comté, France), nous avons développé une nouvelle bibliothèque `rchicmgk`, permettant d'implémenter des graphes implicatifs et des arbres hiérarchiques. Cet outil est développé sous logiciel R, et a été préparé à l'occasion des articles [BT20e, BT20f, BCT21, BJT22a, BJT22b].

Autres travaux de l'auteur

Publications

22. Rabenantenaina, T., P. Bemarisika, et A. Totohasina, *Une stratégie d'estimation d'une série temporelle*, REVUT Scientific Journal, DOI : https ://doi.org/10.46857/rsj.2021.3 [RBT20b].
23. Rabenantenaina, T., P. Bemarsika, et A. Totohasina, *Une approche d'estimation de paramètres d'un processus temporel*. Journées de Recherche des ISTs (Institut Supérieur de Technologie) et leurs partenaires internationaux, 2020 [RBT20a].
24. P. Bemarisika, *Extraction des règles d'association selon le couple support-M_{GK} : Graphes implicatifs, et Applications en didactique des mathématiques*. Thèse de doctorat de l'Université d'Antananarivo (Madagascar), préparée au Laboratoire d'Informatique et de Mathématiques (LIM) de l'Université de La Réunion (France), soutenue 2016 [Bem16].
25. P. Bemarisika, et A. Totohasina, *EOMF, Un Algorithme d'Extraction Optimisée des Motifs Fréquents*, Société Francophone de Classification (SFC), pp. 198-203, 2016 [BT16].
26. P. Bemarisika, et A. Totohasina, *Apport des règles négatives à l'extraction des règles d'association*, Société Francophone de Classification (SFC), pp. 99-104, 2014 [BT14a].
27. P. Bemarisika and A. Totohasina, *A Novel Algorithm for Mining Negative and Positive Association Rules*. Journal of Computers and Information Techlgy, pp. 792-798, 2014 [BT14c].
28. P. Bemarisika and A. Totohasina A, *Elaboration of implicative graph according to measure M_{GK}*. Journal of Computer Science Issues, pp. 52-59, 2014 [BT14b].

Publications en didactique

29. P. Bemarisika, H. Ramanantsoa, L. Ramifidisoa, et A. Totohasina, *Résolution d'équations polynomiales par l'utilisation des TICs*. Colq Intnal de TICs, ENS-Antananarivo, 2012 [BRTR12].
30. H. Ramanantsoa, **P. Bemarisika**, L. Ramifidisoa, et A. Totohasina, *Enseignement des limites par l'utilisation des TICs*. Colloque intnal des TICs, ENS d'Antananarivo, 2012 [RBTR12].

Introduction

Le présent mémoire d'Habilitation à Diriger des Recherches présente mes travaux de recherche que j'ai menés ces 7 dernières années après ma thèse de doctorat [Bem16] préparée conjoitement au sein du laboratoire d'équipe d'accueil Didactique des Mathématiques et de l'Informatique (EDMI) de l'ENS-Université d'Antananarivo (Madagascar) et du Laboratoire d'Informatique et de Mathématiques (LIM) de l'Université de La Réunion (France), soutenue le 20 avril 2016 à l'Université d'Antsiranana (Madagascar) alors que j'étais déjà Assistant d'ESR (Enseignement Supérieur et de Recherche) depuis 4 ans. Les travaux synthétisés s'insèrent principalement dans deux axes : l'extraction des meilleures règles d'association, et la représentation de ces meilleures règles d'association par des graphes implicatifs et des arbres hiérarchiques. Ils sont fédérés autour d'un objectif commun, celui de la fouille de données, et partagent une préoccupation sous-jacente pour l'aide à la décision et pour le développement d'applications. Ces recherches ont été menées :

- à l'ENSET (Ecole Normale Supérieure pour l'Enseignement Technique) de l'Université d'Antsiranana où j'ai été nommé Maître de Conférences en novembre 2019, puis élu Responsable de la Mention EADIMI (Education, Apprentissage, Didactique et Ingénierie en Mathématiques et Informatique) depuis 2016 jusqu'en 2022, et où j'ai effectué mes principales tâches d'enseignement et d'encadrement de plusieurs mémoires d'étudiants en masters indifférenciés ;
- à l'Equipe d'accueil MIA (Mathématiques, Informatique et Applications) de l'Université de Toamasina (Madagascar) où j'ai co-dirigé avec André Totohasina (Professeur à l'Université d'Antsiranana) 2 thésards : Théophile Rabenantenaina et Ionja Lalaina Narisoa Iandriah, qui travaillent respectivement sur le problème d'estimation de séries temporelles et celui de Contrôle optimal stochastique dirigé par un mouvement brownien fractionnaire ;
- à l'Ecole doctorale PE2Di (Problématique de l'Education et Didactiques des Disciplines) de l'Université d'Antananarivo (Madagascar) en tant que Responsable du laboratoire EDMI.

Dans cette introduction, je souhaite tout d'abord en présenter brièvement la genèse par mon cheminement d'enseignant-chercheur en Modélisations Mathématiques et Informatique.

Cheminement d'enseignant-chercheur

J'ai commencé à enseigner en tant que vacataire au sein de l'ENSET, entre 2008 et 2011. Mes premières activités traitaient des travaux pratiques (TP) d'Analyse multivariée sous R, et

d'Algorithmique & Programmation sous Matlab. Après avoir obtenu le diplôme de Master de Mathématiques et Applications, spécialité Statistique et Econométrie de l'Université de Toulouse I (France) en 2010 où j'ai été initié à la recherche par Professeur Yves Aragon sur le problème de Séries temporelles, j'ai été nommé Assistant d'ESR à l'ENSET-Université d'Antsiranana (Madagascar) en septembre 2011 lequel j'ai débuté ma carrière d'enseignant-chercheur. L'année suivante (2012-2013), j'ai suivi le Master 2-Recherche de Mathématiques et Applications, spécialité Mathématiques Approfondies à finalité Recherche de l'Université Franche-Comté (France), ce qui m'a permis d'approfondir, avec Professeur Youri Kabanov, la théorie de Processus stochastiques. Depuis 2012 jusqu'à 2015, j'ai été nommé Responsable de Licence première année de l'ENSET.

A cette même période 2012-2015, je me suis inscrit en thèse à l'école doctorale PE2Di dans l'équipe d'accueil Education et Didactique des Mathématiques et Informatique (EDMI) sous les directions du Professeur André Totohasina (Université d'Antsiranana) et du Professeur Jean Diatta (Université de La Réunion). J'ai beaucoup appris auprès de mes deux directeurs qui m'ont permis de vivre la recherche. Mon sujet de thèse portait sur la fouille des règles d'association. Les domaines d'applications qui intéressaient Totohasina André étaient à la fois l'Analyse Statistique Implicative (ASI) et la didactique des mathématiques, et j'ai pu naturellement évaluer l'impact des mes travaux sur le problème de didactique de la statistique. Ma thèse de doctorat a été financée par l'Agence Universitaire de la Francophonie (AUF) dans le cadre de projet Horizons francophones Sciences Fondamentales **Informatique et Mathématiques**, et j'ai pu rejoindre mon second directeur pour poursuivre en alternance (4 mois par an) mes activités de recherche au sein du LIM (Univ. La réunion). Cela m'a permis de renforcer des compétences sur le problème de fouille de données à partir duquel j'ai avancé mes recherches tant sur l'approche théorique que sur l'approche algorithmique. En fait, le premier aspect de mes travaux s'appuyait sur la formalisation théorique astucieuse de la fouille des motifs fréquents et celle de la génération des meilleures règles d'association d'une base de données. Le second aspect, quant à lui, s'est focalisé sur le développement algorithmique permettant d'implémenter ces approches théoriques ainsi formulées. Le troisième aspect concernait la représentation de ces règles associatives valides par des graphes implicatifs.

Dans le cadre de cette thèse, nous avons développé un algorithme, appelé EOMF [BT16], pour l'extraction des motifs fréquents dans un contexte binaire. Nous avons également développé un algorithme GenPNR [Bem16], qui prolonge nos travaux [BT14a, BT14c], permettant la génération des meilleures règles d'association à partir de ces motifs fréquents. Nous avons développé un algorithme dénommé ImplicativeGraph, qui étend nos travaux [BT14b], pour les graphes implicatifs au sens de la mesure M_{GK}. Nous avons ensuite développé sous le logiciel R un nouvel outil dénommé *CHIC-*M_{GK} implantant cet algorithme ImplicativeGraph. Avec l'aide de cet outil, nous avons proposé une démarche pour l'identification d'un problème qui freine l'enseignement/apprentissage de la Statistique à Madagascar [Bem16]. Nous y avons identifié des difficultés de nos étudiants en L1 lors de la résolution d'exercices portant sur le problème de tests de comparaison des paramètres statistiques. Aussi, les réflexions menées sur l'enseignement/apprentissage des mathématiques aux Collège et Lycée ont abouti à deux articles de didactique des mathématiques [BRTR12, RBTR12].

Motivations

Plusieurs pistes de recherche ont été relevées dans la conclusion de ma thèse de doctorat. Elles gravitent en particulier autour de trois problématiques : (*i*) Extraction des motifs fréquents dans un contexte binaire ; (*ii*) Génération de l'ensemble des meilleures règles d'association à partir de ces motifs fréquents ; et (*iii*) Représentation de cet ensemble des règles d'association valides par des

graphes et des arbres hiérarchiques. Cependant, la volumétrie engendrée par la combinatoire sur des grandes masses de données complexifie les coûts de l'extraction des motifs fréquents. Cela peut entraver les capacités de filtrage des meilleures règles d'association, et par conséquent la lisibilité des graphes et des arbres qu'elles engendrent. C'est dans ces contextes que j'ai poursuivi mes travaux de recherche postdoctorale en élargissant au fur et à mesure les modèles considérés.

Face à ces problèmes, plusieurs travaux ont été déjà proposés dans la littérature. Certains sont basés sur le concept des motifs fermés fréquents [PTB+05, HYN11, Ngu12, OLL+16, Maa17], et d'autres sur les motifs maximaux fréquents [MT97, DQ13, LLWH14, MLLL16, Maa17] dans le but de réduire le coût de l'extraction. Afin d'élaguer les règles d'association redondantes, les approches existantes [PTB+05, GYNS06, HYN11, LHH12] se sont focalisées sur le concept des bases des règles d'association. Cependant, malgré leurs intérêts incontestables, l'extraction et la représentation des meilleures règles d'association restent encore un défi majeur tel que discuté ci-après.

Au niveau de l'extraction des motifs fréquents, ces approches précitées sont limitées sur la définition traditionnelle d'un support des motifs [AS94], qui nécessite l'accès systématique dans un jeu de données qui peut être grand et dense. La question de l'efficacité de ces approches en termes de performances reste discutable dû au calcul des fermetures qui nécessite à des parcours exhaustifs d'un jeu de données. De plus, ces approches ne sont pas autonomes en terme de fouille des motifs fermés fréquents, des motifs maximaux fréquents et leurs générateurs d'un tel jeu de données.

Au niveau de génération des règles d'association, ces approches sont insuffisantes, car elles ommettent les règles négatives du type $X \rightarrow \overline{Y}$, $\overline{X} \rightarrow Y$ et $\overline{X} \rightarrow \overline{Y}$ en n'étudiant que les règles positives $X \rightarrow Y$. Or, les règles positives à elles seules ne suffisent pas pour couvrir tous les besoins de la fouille des règles d'association, il faut aussi des règles négatives. Par ailleurs, ces approches existantes sélectionnent facilement les règles d'association inintéressantes à cause d'utilisation de la mesure moins sélective, *confiance* [AS94]. On entend d'une règle inintéressante que sa confiance est inférieure ou égale à la valeur de référence *support de la conclusion de la règle*. Cette limite s'avère très handicapante sur leurs résultats dans la question de génération des règles approximatives.

Dans le cadre de représentation des règles d'association par des graphes et arbres hiérarchiques, une approche pionnière, basée sur la mesure *intensité d'implication*, a été introduite dans [GAB+96, GKB03] et étendue dans [GRMG13]. Bien qu'elles soient efficaces, ces aproches présentent encore des limites remarquables. Elles n'étudient que des règles d'association positives, et ne proposent aucune technique pour l'analyse des règles d'association négatives. Par ailleurs, les graphes et arbres de ces approches sont souvent très grands à cause des règles d'association redondantes, et leur stockage peut s'avérer coûteux en espace mémoire. Au niveau de la démarche adoptée, elles ne proposent aucune technique de partitionnement des arcs de façon plus appropriée, étant donné que celui-ci apparait très central du fait qu'il structure les graphes et arbres de classification.

Pour surmonter ces notables limites, nous présentons, dans ce mémoire d'HDR, nos principales contributions comme résumées dans les trois pistes de recherche complémentaires suivantes :

- Proposition d'un nouvel algorithme autonome permettant l'extraction simultanée trois classes des motifs fréquents telles que fermés, maximaux, et leurs générateurs minimaux ;
- Proposition d'approches théoriques et algorithmiques pour l'extraction des bases des règles d'association positives et négatives les plus concises que celles définies dans la littérature ;
- Proposition d'approches théoriques et algorithmiques pour la représentation de l'ensemble des meilleures règles d'association par des graphes implicatifs et des arbres hiérarchiques.

Organisation du mémoire

Ce mémoire s'organise essentiellement autour de chacune des thématiques de recherche que nous l'avons parlées ci-dessus. Il se décline principalement en trois chapitres complémentaires.

Le chapitre 1 synthétise mes travaux autour de l'extraction des motifs fréquents à laquelle je me suis familiarisé pendant ma thèse. J'y aborde principalement la question des motifs fermés fréquents, des maximaux fréquents et leurs générateurs minimaux. Ces travaux sont partis du constat de l'absence d'un algorithme autonome permettant l'extraction simultannée de ces trois classes des motifs. De façon à assurer une continuité avec ce qui précède, la section 1.1 introduit les concepts de base pour l'extraction des motifs fréquents. Dans la section 1.2, nous exposons une nouvelle approche permettant l'extraction de l'ensemble des motifs fréquents [BT17, BT17, BT18, BT20c], composé de ces trois classes des motifs précitées sur laquelle sont proposées de nouvelles techniques de comptage des supports et celles d'élagage de l'espace de recherche. Enfin, dans la section 1.3, nous présentons un nouvel algorithme autonome, appelé CMG (*Closed Maximal and Generator*), permettant d'implémenter cette nouvelle approche précédemment développée [BT21a, BT21b].

Le chapitre 2 recense mes travaux relatifs à la génération des meilleures règles d'association. C'est une deuxième partie des recherches que j'ai entreprises depuis le début de ma thèse. Nous y décrivons quatre travaux. Le premier concerne la conception d'une nouvelle mesure statistique mgk, et en discute les propriétés sous-jacentes [BT20a, BT20e]. Nous présentons ensuite une nouvelle méthode d'élimination des règles d'association redondantes sur laquelle sont proposées de nouvelles approches pour la réduction de l'espace de recherche induit par l'ensemble des règles [BT19a, BT20e] et pour l'élaboration des bases des règles non-redondantes [BT19a, BT19b]. Basées sur ces formalisations, nous avons conçu des nouveaux algorithmes implémentant celles-ci [BT21a, BT21b]. Enfin, nous présentons nos expérimentations menées sur quelques bases de données, expérimentations visant à évaluer notre approche, comparée aux approches existantes sémantiquement proches.

Le chapitre 3 regroupe principalement nos contributions récentes, et concerne la représentation des meilleures règles d'association par des graphe implicatif et arbre hiérarchique. Il est à noter que je me suis déjà intéressé à des questions de ce graphe implicatif depuis la fin de ma thèse [Bem16], mais avec la mesure de qualité classique M_{GK} [Gui00, TR05]. Malheureusement, M_{GK} ne prend pas compte la question du nombre de contre-exemples $n_{X\overline{Y}}$ d'une règle $X \rightarrow Y$ sur un échantillon de transactions d'une base de données où sont extraites les meilleures règles d'association. Nous avons ensuite proposé une nouvelle mesure statistique mgk [BJT22a], dans un premier temps. Nous y avons repris les fondements théoriques de M_{GK} et adapté à la quantification d'invraisemblance de la faiblesse de tel nombre de contre-exemples $n_{X\overline{Y}}$. Les sections 3.3 et 3.4 se focalisent sur la mise en place des différents algorithmes pour les graphe implicatif [BCT21, BJT22a] et arbre hiérarchique [BJT22b]. Afin d'expérimenter et d'évaluer nos approches, nous avons conçu un nouvel outil `rchicmgk`, implémenté sous forme d'un package du logiciel R. Plus précisément, le package `rchicmgk` est la suite de l'outil *CHIC-*M_{GK} que nous avons développé vers la fin de ma thèse. Il est le fruit de collaboration avec Raphaël Couturier (Université de Franche-Comté). Dans le cadre de classification, nous avons développé certains programmes informatiques pour l'arbre hiérarchique au sens d'une nouvelle mesure *cohmgk*. Actuellement, le package `rchicmgk` inclut à la fois le graphe implicatif et l'arbre hiérachique. Il a été préparé à l'occasion des publications [BT20e, BT20f, BCT21, BJT22a, BJT22b]. Il offre, entre autres, un moyen visuel et interactif très puissant aux experts en fouille de données, et permet d'aider également ces experts dans leur prise de décision.

Le mémoire se termine par une conclusion qui résume l'ensemble des travaux de recherche, et présente les principales perspectives liées à des différents résultats présentés dans ce manuscrit.

Chapitre 1

Contributions à l'extraction des motifs fréquents

On présente dans ce chapitre les travaux relatifs à l'extraction des motifs fréquents : motifs fermés et maximaux, ainsi que leurs générateurs minimaux. Certaines preuves des propriétés mathématiques sont omises pour conserver l'unité de ce document. Après avoir rappelé quelques concepts préliminaires (section 1.1), nous nous appuyons sur deux idées principales : l'élagage de l'espace de recherche de ces trois classes des motifs fréquents (section 1.2), et la définition d'un algorithme par niveau, dénommé CMG (*Closed-maximal-Generators*), permettant l'extraction simultanée de tels trois sous-ensembles des motifs fréquents à partir d'un contexte d'extraction (section 1.3). Les publications liées à ce chapitre sont : [BT17, BT17, BT18, BT20c, BT20b, BT21a, BT21b].

1.1 Définitions et notations

Nous allons commencer par formaliser le concept d'une base de données (ou plus généralement un contexte formel). Un tel contexte peut être formellement représenté sous la forme d'un triplet $\mathcal{D} = (\mathcal{T}, \mathcal{I}, \mathcal{R})$ décrivant un ensemble fini $\mathcal{T}$ de transactions (avec un identificateur appelé TID (Tuple IDentifier)), un ensemble fini $\mathcal{I}$ d'items (ou de motifs, ou d'attributs), et une relation binaire $\mathcal{R} \subseteq \mathcal{I} \times \mathcal{T}$ entre $\mathcal{I}$ et $\mathcal{T}$, telle que $i\mathcal{R}t$ signifie que l'item i est présent dans la transaction t (i.e. $t[i] = 1$), et par $\neg i\mathcal{R}t$ sinon (i.e. $t[i] = 0$). Le tableau 1.1 ci-dessous représente une base de données $\mathcal{D}$ ayant 5 items $\{A, B, C, D, E\}$ et 6 transactions. Un sous-ensemble X de $\mathcal{I}$ (i.e. $X \subseteq \mathcal{I}$), tel que

Table 1.1 – Un exemple d'une base de données $\mathcal{D}$ à 5 items et 6 transactions

TID	A	B	C	D	E
t_1	1	0	1	1	0
t_2	0	1	1	0	1
t_3	1	1	1	0	1
t_4	0	1	0	0	1
t_5	1	1	1	0	1
t_6	0	1	1	0	1

$|X| = k$, est appelé un motif de longueur k (ou un k-motif). Par exemple, AB [1] est un 2-motif

1. Nous utilisons une forme sans séparateur pour les itemsets : par exemple, AB représente l'ensemble $\{A, B\}$.

de $\mathcal{I}$. Pour tout $X \subseteq \mathcal{I}$, $\overline{X} = \{t \in \mathcal{T} | \exists i \in X : (i,t) \notin \mathcal{R}\}$ est le complementaire de X dans $\mathcal{I}$ (i.e. $\mathcal{I} \backslash X$). Par exemple, nous avons, dans le contexte $\mathcal{D}$ du tableau 1.1, $AB = \{t_3, t_5\}$ alors que $\overline{AB} = \{t_1, t_2, t_4, t_6\}$. Le nombre de transactions de $\mathcal{D}$ contenant un motif X, noté $X' = \{t \in \mathcal{T} | i\mathcal{R}t, \forall i \in X\}$, est appelé extension de X. Le support relatif [AS94] de X est le rapport entre le cardinal de son extension X' et le nombre total $|\mathcal{T}|$ de transactions de la base de données $\mathcal{D}$:

$$supp(X) = P(X') = \frac{|X'|}{|\mathcal{T}|}, \tag{1.1}$$

où P est la probabilité discrète uniforme sur $(\mathcal{I}, \mathcal{P}(\mathcal{I}))$. Soit $minsup \in]0,1[$. Un motif X est dit fréquent si $supp(X) \geq minsup$. On notera par $\mathcal{F}$ l'ensemble des motifs fréquents dans $\mathcal{D}$:

$$\mathcal{F} = \{X \subseteq \mathcal{I} | X \neq \emptyset \ \wedge \ supp(X) \geq minsup\} \tag{1.2}$$

Définition 1 (Support absolu disjonctif)**.** *Soit I un motif de $\mathcal{I}$. Le support disjonctif [GS96, CG06, Fen07] absolu de I que l'on note $supp_{abs}(\vee I)$ est défini par :*

$$supp_{abs}(\vee I) = \sum_{J \subseteq I, J \neq \emptyset} (-1)^{|J|-1} supp_{abs}(J)$$

$$supp_{abs}(\overline{I}) = |\mathcal{T}| - supp_{abs}(I)$$

Du tableau 1.1, on a $supp_{abs}(\vee AD) = |\{t_1, t_3, t_5\}| = 3$, et $supp_{abs}(\overline{AD}) = |\{t_2, t_4, t_6\}| = 3$.

Soit $2^{\mathcal{I}}$ (resp. $2^{\mathcal{T}}$) l'ensemble des parties de $\mathcal{I}$ (resp. de $\mathcal{T}$). Pour $I \subseteq \mathcal{I}$ et $T \subseteq \mathcal{T}$, nous définissons deux fonctions ϕ et ψ qui définissent la connexion de Galois [GW99] entre $2^{\mathcal{I}}$ et $2^{\mathcal{T}}$:

$$\begin{array}{rccll} \phi: & 2^{\mathcal{I}} & \rightarrow & 2^{\mathcal{T}} \text{ (extension)} & \\ & I & \mapsto & \phi(I) & = I' = \{t \in \mathcal{T} | i\mathcal{R}t, \forall i \in I\} \\ \psi: & 2^{\mathcal{T}} & \rightarrow & 2^{\mathcal{I}} \text{ (intension)} & \\ & T & \mapsto & \psi(T) & = T' = \{i \in \mathcal{I} | i\mathcal{R}t, \forall t \in T\}. \end{array}$$

Autrement dit, $\phi(I)$ (resp. $\psi(T)$) dénote l'ensemble des transactions partageant les mêmes items $i \in \mathcal{I}$, appelé aussi extension de I (resp. l'ensemble des items communs à un groupe de transactions $t \in \mathcal{T}$, appelé aussi intension de T). Le couple d'applications (ϕ, ψ) est une connection de Galois entre les ordres partiels ou treillis $(2^{\mathcal{I}}, \subseteq)$ et $(2^{\mathcal{T}}, \subseteq)$. Pour tous $I_1, I_2 \subseteq \mathcal{I}$ et tous $T_1, T_2 \subseteq \mathcal{T}$, on a :

$$I_1 \subseteq I_2 \Rightarrow \phi(I_1) \supseteq \phi(I_2) \text{ et } T_1 \subseteq T_2 \Rightarrow \psi(T_1) \supseteq \psi(T_2) \tag{1.3}$$

Proposition 1 ([Fen07])**.** *Soit (ϕ, ψ) une correspondance de Galois, on a : $\phi o \psi o \phi = \phi$ et $\psi o \phi o \psi = \psi$. Cette fois-ci, la correspondance de Galois (ϕ, ψ) sera dit couple involutif.*

Les opérateurs $\gamma = \psi o \phi$ dans $2^{\mathcal{I}}$ et $\gamma' = \phi o \psi$ dans $2^{\mathcal{T}}$ sont appelés opérateurs de fermeture de Galois. Ainsi, $X \subseteq \mathcal{I}$ est fermé si son image par γ égale lui-même, i.e. $\gamma(X) = X$.

Exemple 1. *Etant donné que A et C, du tableau 1.1, apparaissent simultanément dans les transactions t_1, t_3 et t_5, on a alors $\phi(AC) = \{t_1, t_3, t_5\}$. D'autre part, t_1, t_3 et t_5 partagent en commun les motifs A et C, on a ensuite $\psi(\{t_1, t_3, t_5\}) = \{AC\}$. Il en résulte que $\gamma(AC) = \psi o \phi(AC) = \psi(\phi(AC)) = \psi(\{t_1, t_3, t_5\}) = \{AC\}$. Ainsi, $\gamma(AC) = \{AC\}$. Autrement dit, AC est un fermé.*

L'opérateur γ, tout comme l'opérateur γ', est caractérisé par le fait qu'il est :
- Isotonie : pour tous $X, Y \subseteq \mathcal{I}$, on a $X \subseteq Y \Rightarrow \gamma(X) \subseteq \gamma(Y)$;
- Extensivité : pour tout $X \subseteq \mathcal{I}$, on a $X \subseteq \gamma(X)$;
- Idempotence : pour tout $X \subseteq \mathcal{I}$, on a $\gamma(\gamma(X)) = \gamma(X)$.

Proposition 2. *Pour tout $X \subseteq \mathcal{I}$ dans un contexte $\mathcal{D}$, on a $supp(X) = supp(\gamma(X))$.*

Démonstration. Soit $X \subseteq \mathcal{I}$ et $\gamma(X)$ sa fermeture. Comme $X \subseteq \mathcal{I}$, on a par extensivité $X \subseteq \gamma(X) \Rightarrow \phi(X) \supseteq \phi(\gamma(X))$ (par anti-monotonie de ϕ). Par involutivité du couple (ϕ, ψ) (cf. Proposition 1), on a $\phi o \psi o \phi(X) = \gamma'(\phi(X)) = \phi(X)$. Par suite, on obtient : $supp(\gamma(X)) = \frac{|\phi(\gamma(X))|}{|\mathcal{T}|} = \frac{|\phi o \psi o \phi(X))|}{|\mathcal{T}|} = \frac{|\phi o \psi(\phi(X))|}{|\mathcal{T}|} = \frac{|\gamma'(\phi(X))|}{|\mathcal{T}|} = \frac{|\phi(X)|}{|\mathcal{T}|} = supp(X)$. Ainsi, $supp(X) = supp(\gamma(X))$. □

1.2 Une nouvelle méthode d'extraction des motifs fréquents

Une limite très souvent formulée dans la littérature à propos de l'extraction des motifs fréquents repose sur le coût de l'extraction ($2^{|\mathcal{I}|}$ au pire des cas). Ainsi, plusieurs approches [PTB+05, Gay09, HYN11, DQ13, LLWH14, Maa17], basées sur le concept de fermeture, ont été proposées. Cependant, ces approches, comme mentionné, présentent certaines limites notables dont la principale est associée au calcul des fermetures basé sur deux correspondances ϕ et ψ, étant donné qu'une seule correspondance nécessite à elle seule du parcours exhaustif d'un jeu de données. Une autre lacune induite par ces approches réside sur le calcul des supports. En effet, elles sont limitées sur la définition traditionnelle d'un support selon Agrawal [AS94] qui nécessite des accès répétitifs au contexte. Ces contextes sont abordés dans [BT18, BT20c, BT21b, BCT21] et synthétisés ci-dessous.

Dans [BT18], nous avons proposé une nouvelle technique d'extraction des motifs fréquents, dénommée *reduce-access-database*, où nous avons proposé de nouvelles définitions (fondées sur le support) d'un ensemble des fermés fréquents (def. 3), des maximaux (def. 4) et des générateurs (def. 5), permettant de remédier à l'inconvénient de la définition originelle [MT97, PTB+05]. Ces nouvelles définitions dépendent fortement de la définition d'une classe d'équivalence (cf. def 2).

Définition 2 (Classe d'équivalence)**.** *Deux motifs I et J de $\mathcal{I}$ sont dits équivalents, dénotés $I \cong J$, si et seulement s'ils sont comparables (i.e., $I \subseteq J$ ou $I \supseteq J$) et $supp(I) = supp(J)$. La classe d'équivalence de I, notée $[I]$, est définie par $[I] = \{J \subseteq \mathcal{I} \mid I \cong J\}$.*

Un motif le plus grand (au sens de $\subseteq$) dans sa classe d'équivalence est appelé motif fermé.

Définition 3 (Fermé fréquent)**.** *Soit $\mathcal{F}$ l'ensemble des motifs fréquents. Un motif I est fermé si aucun sur-ensemble (comparable) n'est fréquent. L'ensemble des fermés fréquents, noté $\mathcal{FC}$, est :*

$$\mathcal{FC} = \{I \in \mathcal{F} \mid \nexists J \in \mathcal{F} \text{ tel que } J \supset I \ \wedge \ supp(J) = supp(I)\} \tag{1.4}$$

Exemple 2. *Prenons le tableau 1.1, soit $minsup = 2/6$. Le motif C est fréquent et n'a pas de sur-ensemble ayant le même support, donc femé et générateur à la fois. Le motif A est fréquent mais pas fermé, car $AC \supset A$ et fréquent (i.e., $supp(A) = supp(AC) = 3/6 > 2/6$). De même, les motifs B et E sont fréquents mais pas femés car $A, E \subset BE$ et BE est fréquent (i.e., $supp(B) = supp(E) = supp(BE) = 5/6 > 2/6$). Les motifs BC et CE sont fréquents mais pas femés, car ils sont inclus dans BCE qui est fréquent (i.e., $supp(BC) = supp(CE) = supp(BCE) = 4/6 > 2/6$). Les motifs AB et AE quant à eux sont fréquents mais pas femés, car $AB, AE \subset ABCE$ et $ABCE$*

est fréquent (i.e., $supp(AB) = supp(AE) = supp(ABCE) = 2/6$). Ainsi, l'ensemble des fermés induit par cette petite base de données du tableau 1.1 est $\mathcal{FC} = \{C, AC, BE, BCE, ABCE\}$.

Définition 4 (Maximal fréquent)**.** *Soit $\mathcal{FC}$ l'ensemble des motifs fermés fréquents. I est maximal si aucun de ses sur-ensembles n'est fréquent. L'ensemble $\mathcal{FM}$ des maximaux fréquents est :*

$$\mathcal{FM} = \{I \in \mathcal{FC} \mid \nexists J \in \mathcal{FC} \text{ tel que } J \supset I \ \wedge \ supp(J) = supp(I)\} \tag{1.5}$$

Exemple 3. *Prenons la même base de données du tableau 1.1, soit $minsup = 2/6$. Le motif BCE est fréquent mais pas maximal, car son sur-ensemble $ABCE$ est fréquent. Et, $ABCE$ est maximal fréquent, car aucun ses sur-ensembles n'est fréquent. Donc, l'ensemble des motifs maximaux fréquents dans cette petite base de données pour $minsup = 2/6$ est $\mathcal{FM} = \{ABCE\}$.*

Proposition 3. *Tout motif maximal (fréquent) est nécessairement fermé (fréquent).*

Dans [BCT21], nous avons montré qu'un motif minimal infréquent d'une base de données peut être dérivé d'un motif maximal fréquent, tel que présenté dans la Proposition 4 ci-après.

Proposition 4. *Soit $h \in \mathcal{FM}$, $\forall l \notin h$, $\nexists \widetilde{l} \subset l$ tel que $\widetilde{l} \notin \mathcal{F}$, alors l est un motif minimal infréquent.*

Définition 5 (Générateur minimal)**.** *Soit $\mathcal{F}$ l'ensemble des motifs fréquents. Un motif G est générateur minimal[2] d'un certain fermé $\mathcal{C}$ (i.e., $\gamma(G) = \mathcal{C}$) s'il n'existe pas un sous-ensemble $g \subset G$ tel que $supp(g) = supp(G)$. L'ensemble des générateurs minimaux d'un fermé $\mathcal{C}$, noté $\mathcal{G}_\mathcal{C}$, est :*

$$\mathcal{G}_\mathcal{C} = \{G \in \mathcal{F} \mid \gamma(G) = \mathcal{C} \ \wedge \ \nexists g \subset G \text{ tel que } supp(g) = supp(G)\} \tag{1.6}$$

Exemple 4. *Prenons aussi le tableau 1.1, soit $minsup = 2/6$. De façon duale avec l'exemple 2, l'ensemble des générateurs de cette petite base de données est $\mathcal{G} = \{A, B, C, E, AB, AE, BE, CE\}$.*

Le défaut induit par le calcul des fermetures peut être résolu par la proposition 5 suivante :

Proposition 5. *Soient I_1 et I_2 deux motifs de $\mathcal{I}$. Si $supp(I_1) = supp(I_2)$, alors $\gamma(I_1) = \gamma(I_2)$.*

Cette proposition 5 est centrale pour identifier les motifs fermés et leurs générateurs, pouvant être dérivés de leur support, c-à-d qu'il s'avère inutile de passer au calcul des fermetures, car deux motifs appartenant à une même classe d'équivalence ont le même support (i.e., même fermeture).

L'efficacité de notre stratégie d'élagage de l'espace de recherche découle par suite du comptage plus économique de support d'un motif candidat comme donné dans le théorème 1 ci-dessous.

Théorème 1 ([BT17, BT18, BT20c, BT20e, BT20d])**.** *Pour tout X un k-itemset ($k \geq 3$) non-générateur et x un $(k-1)$-sous-ensemble de X, on a :*

$$supp(X) = min(\{supp(x)\} \mid x \subset X) \tag{1.7}$$

Démonstration. Soit $\mathcal{I}_X$ l'ensemble des motifs inclus strictement dans X. Soit x un motif de $\mathcal{I}_X$ qui minimise le support dans cet ensemble. Du fait de la propriété d'antimonotonie de support, on a :

$$x \subset X \Rightarrow \phi(X) \subset \phi(x) \text{ donc } supp(X) < supp(x)$$

Par ailleurs, X est non-générateur, il existe donc un motif $Y \in \mathcal{I}_X$ tel que $supp(Y) = supp(X)$. Or, $supp(x)$ est minimal dans $\mathcal{I}_X$, donc $supp(x) \leq supp(Y)$. Finalement $supp(x) = supp(X)$. □

2. appelé aussi itemset clé dans [STB+02] et itemset libre dans [BBR03].

L'originalité de ce Théorème 1 repose sur le fait qu'il permet d'éviter l'accès répétitif à la base de données lors du calcul des supports candidats. Plus concrètement, le support d'un k-non-générateur ($k \geq 3$) peut être déduit de celui de ses $(k-1)$-sous-ensembles, sans passer à la base de données. De l'exemple ci-dessus du tableau 1.1, on a trouvé que le 2-motif AB est générateur. Cela garantit que son sur-ensemble ABC n'est pas générateur. Ainsi, le support de ABC est dérivable de ses sous-ensembles AB, AC et BC sans passer au contexte d'extraction, tel que donné comme suit :

$$supp(ABC) = min\big(supp(AB), supp(AC), supp(BC)\big) = min(2/6, 3/6, 4/6) = 2/6.$$

Corollaire 1. *Un motif X est générateur si et seulement si :*

$$supp(X) < min\big(\{supp(x)\} \mid x \subset X\big) \tag{1.8}$$

Démonstration. Posons $\ell = min\big(\{supp(x)\} \mid x \subset X\big)$. Si X est générateur, alors :

$$\forall x \subseteq \mathcal{I} \text{ tel que } x \subset X \Rightarrow \phi(X) \subset \phi(x) \text{ donc } supp(X) < supp(x)$$

Le passage au minimum sur l'ensemble fini des minorants x de X donne $supp(X) < \ell$. Réciproquement, si $supp(X) < \ell$ alors $supp(X) \neq \ell$. La contraposée du théorème 1 implique que X est générateur. □

A noter que le Théorème 1 n'est utilisé qu'à un itemset (non-générateur) de longueur supérieure ou égale à 3. En effet, les 1-motifs n'ont pas de sous-ensembles, et les 2-itemsets ont exactement 2 sous-ensembles de taille 1 pouvant être générés à partir des 1-motifs fréquents, donc forcément fréquents (grâce à l'anti-monotonie de support), inutile d'utiliser le support estimé de l'équation (1.7). Dans ce cas, notre approche utilise le support classique d'Agrawal [AS94] tel que défini dans l'équation (1.1). Elle s'en distingue cependant au niveau de la structure de données utilisée. Nous utilisons une nouvelle structure de données, appelée MatriceSupport [BT16, Bem16], telle que présentée dans la figure 1.1 ci-dessous, pour stocker les supports des 1 et 2-motifs. Pour cela, nous considé-

	$v_1 \ldots$	v_j	$\ldots v_m$
t_1			
$\vdots$			
t_i	$\ldots$	t_{ij}	$\ldots$
$\vdots$			
t_n			

Scanner le contexte $\mathcal{D}$

	$v_1 \ldots$	v_j	$\ldots v_m$
v_1			
$\vdots$	$\ddots$		
v_i		v_{ij}	$\ldots$
$\vdots$			$\ddots$
v_m			

Figure 1.1 – MatriceSupport (droite) sur un contexte formel $\mathcal{D}$ (gauche)

rons une base de données très générale (Figure 1.1-gauche) de m attributs $\mathcal{I} = \{v_j | 1 \leq j \leq m\}$ et de n transactions $\mathcal{T} = \{t_i | 1 \leq i \leq n\}$. L'intersection de la i^{e} ligne et de la j^{e} colonne de la base de données $\mathcal{D}$ est la quantité $t_{ij} = v_j(t_i)$ qui est une valeur observée de v_j sur la transaction élémentaire t_i. La MatriceSupport associée (Figure 1.1-droite) est la projection de cette base $\mathcal{D}$ par rapport à ses attributs de telle sorte que chaque attribut correspond à une cellule de la matrice créée laquelle associe le nombre de fois, noté $v_{ij} = v_j(v_i)$, que l'attribut v_j apparaît dans la ligne

v_i. Cette valeur v_{ij} est ensuite utilisée pour déterminer le support (relatif) d'un itemset candidat :

$$supp(v_j) = v_{ij}/|\mathcal{D}|. \tag{1.9}$$

En pratique, comme cette distribution matricielle est symétrique, seule une demi matrice de support doit être stockée par souci d'éviter les redondances, et nous considérons à cet effet la partie supérieure. L'originalité de cette matrice de support MatriceSupport réside du fait qu'elle permet d'identifier simultanément les supports des 1 et 2-motifs en une seule passe à la base de données (au lieu de 2 passes pour les existants). Précisément, les supports des singletons (1-motifs) peuvent être obtenus via les éléments diagonaux, alors que les 2-motifs par les éléments supérieurs de cette matrice. Nous considérons à cet égard une illustration très simplifiée où on considère la petite base $\mathcal{D}$ du tableau 1.1. Grâce à cette matrice, on obtient, à titre d'exemples, les supports :

TID	A	B	C	D	E
t_1	1	0	1	1	0
t_2	0	1	1	0	1
t_3	1	1	1	0	1
t_4	0	1	0	0	1
t_5	1	1	1	0	1
t_6	0	1	1	0	1

Scanner la base $\mathcal{D}$

MatriceSupport					
	A	B	C	D	E
A	3	2	3	1	2
B	-	5	4	0	5
C	-	-	5	1	4
D	-	-	-	1	0
E	-	-	-	-	5

Figure 1.2 – Formalisme de la MatriceSupport (droite) sur la petite base $\mathcal{D}$ (gauche)

$$supp(A) = 3/6, supp(B) = supp(C) = supp(E) = 5/6, supp(AB) = 2/6 \text{ et } supp(BC) = 4/6.$$

L'intérêt de cette méthode est étudié dans le théorème 3, combiné à celui d'Agrawal [AS94].

Théorème 2 (Principe d'Apriori [AS94])**.** *(1) Tout sous-ensemble d'un motif fréquent est fréquent, et (2) tout sur-ensemble d'un motif non-fréquent est non-fréquent.*

Démonstration. (1) Soient $X, Y \subseteq \mathcal{I}$ tels que $X \in \mathcal{F}$ et $Y \subseteq X$. Comme $Y \subseteq X$, on a (Propriété 1.3) $\phi(Y) \supseteq \phi(X) \Rightarrow supp(Y) \geqslant supp(X) \geqslant minsup \Rightarrow Y \in \mathcal{F}$. (2) Soient $X, Y \subseteq \mathcal{I}$ t.q. $X \notin \mathcal{F}$ et $Y \supseteq X$. Comme $Y \supseteq X$, on a $\phi(Y) \subseteq \phi(X) \Rightarrow supp(Y) \leqslant supp(X) \leqslant minsup \Rightarrow Y \notin \mathcal{F}$. □

Ce qui relève que si un motif candidat est fréquent, il est inutile d'étudier ses sous-ensembles qui sont aussi fréquents grâce à la monotonie de support. Inversement, s'il est infréquent, il est inutile de générer ses sur-ensembles qui sont aussi non-fréquents (anti-monotonie de support).

Théorème 3 ([BT20e, BT21a])**.** *Tout sous-ensemble d'un générateur est aussi générateur.*

Démonstration. Soient $X, Y \subseteq \mathcal{I}$ tels que $X \subset Y$. Il existe un itemset Z non vide et disjoint de X tel que $Y = X \cup Z$. Supposons que X est non-générateur, alors X admet un sous-ensemble propre x qui lui est équivalent : $x \subset X$ et $x \cong X$. De $x \cong X$, on a $x \cup Z \cong X \cup Z$. De plus, $X \cap Z = \emptyset$, on a donc $x \cup Z \subset X \cup Z = Y$ implique que Y est non générateur. La contraposée donne le résultat. □

Corollaire 2. *Tout sur-ensemble d'un non-générateur est aussi non-générateur.*

Il en résulte que si un motif candidat est générateur, il est inutile de générer ses sous-ensembles qui sont aussi générateurs. Inversement, aucun accès à la base de données n'est effectué si un motif candidat est non-générateur, puisqu'il est dérivable de ses sous-ensembles grâce au théorème 1.

Il est intéressant de noter que ces deux théorèmes nous permettent d'élaguer l'espace de recherche des motifs fréquents. C'est une situation idéale pour la conception de l'algorithme CMG.

1.3 Algorithme CMG

Comme signalé, la principale motivation de notre approche repose sur l'absence dans la littérature d'un algorithme autonome permettant l'extraction simultanée les ensembles $\mathcal{FC}$ d'itemsets fermés fréquents, $\mathcal{FM}$ d'itemsets maximaux fréquents, et $\mathcal{G}$ des générateurs minimaux. Cet aspect est abordé dans [BT21a, BT21b] et présenté synthétiquement dans les algorithmes ci-dessous, dont le principal est l'Algorithme 1. Ce dernier prend en entrée une base de données $\mathcal{D}$ et un seuil

Algorithm 1 Algorithme CMG

Require: Database $\mathcal{D}$, minimum support threshold $minsup \in]0,1[$.
Ensure: $\mathcal{FCMG} = \langle closed, maximal, generator, supp \rangle$

$\mathcal{FCMG} \leftarrow \emptyset$; $\mathcal{FC} \leftarrow \emptyset$; $\mathcal{FM} \leftarrow \emptyset$; $\mathcal{FG} \leftarrow \emptyset$; $\mathcal{FCMG}.supp \leftarrow 0$;
$\mathcal{F} \leftarrow EOMF(\mathcal{D}, minsup)$ /* $\mathcal{F} = \{\mathcal{F}_1, \ldots, \mathcal{F}_\ell\}$, ℓ is the size of largest frequent itemset */
for (each itemset $h \in \mathcal{F}_1$) **do**
  $h.gen \leftarrow true$; $h.closed \leftarrow true$;
end for
for all $(k \leftarrow 2; k \leq \ell; k++)$ **do**
  if $(\mathcal{F}_k \neq \emptyset)$ **then**
    for (each itemset $h \in \mathcal{F}_k$) **do**
      $h.gen \leftarrow true$; $h.closed \leftarrow true$;
      for all (subset $\widetilde{h} \in \mathcal{F}_{k-1}$ of h) **do**
        if $(supp(\widetilde{h}) == supp(h))$ **then**
          $h.gen \leftarrow false$; $\widetilde{h}.closed \leftarrow false$;
        end if
      end for
    end for
    $\mathcal{FC}_{k-1} \leftarrow \{h \in \mathcal{F}_{k-1} \mid h.closed = true\}$;
    GenMaximal$(\mathcal{FC}_{k-1}, \mathcal{F}_k)$;
    GenGenerators$(\mathcal{FC}_{k-1}, \mathcal{F}_k)$;
  else
    $\mathcal{FC}_{k-1} \leftarrow \{h \in \mathcal{F}_{k-1} \mid h.closed = true\}$; /* Frequent closed itemset of size $k-1$ */
    GenMaximal$(\mathcal{FC}_{k-1})$;
    GenGenerators$(\mathcal{FC}_{k-1})$;
  end if
end for
$\mathcal{FC}_k \leftarrow \mathcal{F}_k$;
GenMaximal$(\mathcal{FC}_k)$;
GenGenerators$(\mathcal{FC}_k)$;
$\mathcal{FCMG} \leftarrow \bigcup_{j=1}^{k} \{\mathcal{FCMG}_j.generator, \mathcal{FCMG}_j.closed, \mathcal{FCMG}_j.maximal, \mathcal{FCMG}_j.supp\}$;

minimum de support $minsup$ fixé. Il retourne l'ensemble des fermés fréquents, des maximaux et

de leurs générateurs en faisant appel à deux procédures secondaires (lignes 17 et 18). L'algorithme CMG est un algorithme par niveau pour l'espace de recherche. Il démarre par un appel à la fonction EOMF [BT16, Bem16] qui détermine l'ensemble des itemsets fréquents dans $\mathcal{D}$, dont le pseudo-code est décrit par l'Algorithme 17 dans l'annexe B. Il vérifie ensuite, pour chaque k-itemset fréquent ($k \geq 2$), s'il s'agit d'un fermé en examinant les supports de tous ses sous-ensembles de longueur $k-1$. Deux variables booléennes *gen* et *closed* sont alors utilisées afin d'identifier si un itemset est un générateur ou un fermé fréquent. Si $\mathcal{F}_k$ est vide et $\mathcal{F}_{k-1}$ n'est pas vide, les éléments de $\mathcal{F}_{k-1}$ sont fermés, et la variable *gen* est un générateur (lignes 16 et 18). Inversement, si $\mathcal{F}_k$ est non vide et $\mathcal{F}_{k-1}$ est vide, tous les motifs de $\mathcal{F}_k$ sont des générateurs, et aucune étape supplémentaire n'est nécessaire, puisque tous les motifs sont initialement marqués comme générateurs minimaux.

Un motif c est identifié comme générateur durant les étapes 8-15 de l'algorithme CMG. Si le support de c est identique que celui de l'un de ses sous-ensembles de longueur $k-1$ dans $\mathcal{F}_{k-1}$, alors c n'est pas un générateur et, inversement, il n'est pas un fermé. Aux étapes 16 et 20, tous les itemsets fermés de longueur $k-1$ sont ajoutés à l'ensemble $\mathcal{FC}_{k-1}$. L'étape 25 découvre l'ensemble des fermés de longueur maximale. Aux étapes 17, 21 et 26, la procédure GenMaximal (Algo 2) est appelée afin de générer les maximaux. Elle prend l'ensemble $\mathcal{FC}_k$ en entrée. Pour chaque fermé

Algorithm 2 Procedure GenMaximal($\mathcal{FC}_k$)

Require: $\mathcal{FC}_k$ /* Frequent closed itemset of size k */
Ensure: Assign the maximal to each closed itemsets of $\mathcal{FC}_k$.
1: **for** (each itemset $h \in \mathcal{FC}_k$) **do**
2: **if** ($\nexists \widetilde{h} \supset h \mid \widetilde{h}.maximal = true$) **then**
3: $\mathcal{FM} \leftarrow \mathcal{FM} \cup \{h\}$;
4: **end if**
5: **end for**

$h \in \mathcal{FC}_k$, elle vérifie s'il n'existe aucun autre fermé $\widetilde{h}$ contenant h tel que $\widetilde{h}$ est maximal (ligne 2 de GenMaximal). Si c'est le cas, le fermé h est maximal, puis ajouté aussi dans l'ensemble des maximaux $\mathcal{FM}$. Aux étapes 18, 22 et 27, la procédure GenGenerators (Algo 3) est appelée afin de mettre à jour la liste globale des générateurs et d'affecter ces générateurs aux ensembles des fermés (ou maximaux) respectifs. Elle prend l'ensemble $\mathcal{FC}_k$ en entrée. Pour chaque fermé $c \in \mathcal{FC}_k$,

Algorithm 3 Procedure GenGenerators($\mathcal{FC}_k$)

Require: $\mathcal{FC}_k$ /* Frequent closed itemset of size k */
Ensure: Assign the generators to each closed itemsets of $\mathcal{FC}_k$.
1: **for** (each itemset $c \in \mathcal{FC}_k$) **do**
2: **for all** (subset $\widetilde{c} \in \mathcal{FG}$ of c) **do**
3: add $\widetilde{c}$ in c.generator ;
4: **end for**
5: **end for**
6: $\mathcal{FG} \leftarrow \mathcal{FG} \cup \{h \in \mathcal{F}_k \mid h.generator = true \wedge h.closed = false \wedge h.maximal = false\}$;

ses sous-ensembles propres dans l'ensemble global des générateurs $\mathcal{FG}$ sont alors retirés puis ajoutés aux les générateurs de c (étapes 1-5 de la procédure GenGenerators). Cette procédure met à jour l'ensemble global de générateurs $\mathcal{FG}$ par les itemsets, qui ne sont pas fermés mais qui sont

des générateurs avant le début de la prochaine itération. Si l'ensemble des générateurs d'un motif fermé donné est vide, ce fermé est alors le générateur de lui-même (c'est donc l'unique motif dans sa classe d'équivalence). Enfin, la liste $\mathcal{FCMG}$ classe les k-motifs fréquents (fermés, maximaux et leurs générateurs), ainsi que leurs supports selon leur ordre de sélection (ligne 28).

Théorème 4. *Soit $\mathcal{D}$ une base de données avec m items et n transactions. Soit $\mathcal{F}_k$ (resp. C_k) l'ensemble des motifs fréquents (resp. des motifs candidats) de taille k de $\mathcal{D}$. La complexité de l'algorithme CMG (Algorithme 1) est $\mathcal{O}(n \times 2^m)$ dans le pire des cas.*

Démonstration. Soient $m = |\mathcal{I}|$ et $n = |\mathcal{T}|$. Tout d'abord, CMG calcule les motifs fréquents de $\mathcal{D}$ via l'algorithme EOMF (ligne 2). Dans ce cas, le coût du test de fréquence est en $\mathcal{O}(|C_k|)$ et le coût de la génération des candidats du niveau $(k+1)$ est en $\mathcal{O}(k|C_k|)$ dans le pire des cas. Puisqu'une base de données est en général grande, c'est le coût du calcul de la fréquence qui domine et la complexité de ces opérations est $\mathcal{O}(|\mathcal{T}| \times |C_k|) = \mathcal{O}(n \times 2^m)$ dans le pire des cas. Générer les itemsets fermés et maximaux ainsi que leurs générateurs (lignes 3-28) sur l'ensemble $\mathcal{F}_k$ des motifs fréquents se fait en $\mathcal{O}(|\mathcal{F}_k|)$ dans le pire des cas. Toutefois, ce dernier est dominé par $\mathcal{O}(n \times 2^m)$. Finalement, la complexité globale de l'algorithme CMG est en $\mathcal{O}(n \times 2^m)$ dans le pire des cas. □

La figure 1.3 ci-après illustre l'exécution de CMG mené à la petite base du Tableau 1.1 et à un $minsup = 2/6$. Par suite, nous avons 5 classes d'équivalence pour les motifs fréquents tels

Scan $\mathcal{D}$ $\longrightarrow$

$\mathcal{FCMG}_1$

generator	closed	maximal	Supp
A	AC	-	3/6
B	BE	-	5/6
C	C	-	5/6
D	ACD	-	1/6
E	BE	-	5/6

Suppr. infréqnts $\longrightarrow$

$\mathcal{FCMG}_1$

generator	closed	maximal	supp
A	AC	-	3/6
B	BE	-	5/6
C	C	-	5/6
E	BE	-	5/6

$\mathcal{FC}_2$ $\longrightarrow$

$\mathcal{FCMG}_2$

generator	closed	maximal	supp
AB	ABCE	ABCE	2/6
AE	ABCE	ABCE	2/6
BC	BCE	-	4/6
CE	BCE	-	4/6

Supprimer infréqts $\longrightarrow$

$\mathcal{FCMG}_2$

generator	closed	maximal	supp
AB	ABCE	ABCE	2/6
AE	ABCE	ABCE	2/6
BC	BCE	-	4/6
CE	BCE	-	4/6

Figure 1.3 – Liste des motifs fermés, maximaux et générateurs fréquents, avec $minsup = 2/6$

que $[C] = \{C\}$, $[AC] = \{A, AC\}$, $[BE] = \{B, E, BE\}$, $[BCE] = \{BC, CE, BCE\}$, et $[ABCE] = \{AE, AB, ABCE\}$. Il en résulte que l'item C est à la fois générateur et fermé, alors que l'itemset $ABCE$ est à la fois fermé et maximal de cette petite base de données $\mathcal{D}$, et on obtient :

$$\mathcal{FCMG} = \Big\{\{C, 5/6\}, \{A, AC, 3/6\}, \{B, E, BE, 5/6\}, \{BC, CE, BCE, 4/6\}, \{AE, AB, ABCE, 2/6\}\Big\}$$

A signaler que ces opérations sont faites en une seule fois dans la base de données $\mathcal{D}$ à l'aide de l'algorithme CMG. Ce n'est pas le cas pour les existants (entre autres, [AS94, PTB+05, HYN11]), ceux-ci en font 4 passes. Autrement dit, notre algorithme CMG permet de restreindre significativement l'accès systématique à la donnée $\mathcal{D}$, ce qui rend en pratique une complexité beaucoup plus raisonnable, et lui rend utilisable même si avec une base de données très volumineuse et/ou dense.

Chapitre 2

Contributions à la génération des meilleures règles d'association

Il est question dans ce chapitre des résultats synthétisant les articles [BT18, BRT18, BT19a, BT19b, BT19c, BT20a, BT20e, BT20d, BT21a, BT21b] qui relèvent de la génération des meilleures règles positives et négatives. Comme dans le précédent chapitre 1, certaines preuves des propriétés mathématiques ne sont pas développées (voire omises) en raison du manque d'espace. Après avoir rappelé quelques concepts préliminaires sur les règles d'association (section 2.1), nous synthétiserons comment chacun de ces articles a contribué à la conception d'une nouvelle mesure statistique (section 2.2), à la collection des règles d'association non-redondantes au sens de cette nouvelle mesure statistique (section 2.3), à la mise en œuvre des nouveaux algorithmes permettant la génération des règles non-redondantes (section 2.4), et à l'évaluation expérimentale (section 2.5).

2.1 Terminologie et notations

Dans cette partie, nous nous interessons à la génération des règles d'association telles qu'elles ont été popularisées par Agrawal *et al.* [AIS93, AS94], au moyen des mesures de qualité.

Définition 6. *Une règle d'association est un couple d'attributs noté $X \to Y$, où X et Y sont des motifs disjoints, appelés respectivement prémisse (ou antécédent) et conclusion (ou conséquent).*

Une telle règle d'association $X \to Y$ est aussi appelée *règle d'association positive.*

Définition 7. *Une règle d'association est dite négative, si l'un au moins de deux motifs est négatif, de la forme : $X \to \overline{Y}, \overline{X} \to Y, \overline{X} \to \overline{Y}$. Ce sont des règles qui représentent les relations cachées.*

Les règles d'association, tout comme les motifs fréquents, sont aussi apprises à partir d'une base de données $\mathcal{D} = (\mathcal{T}, \mathcal{I}, \mathcal{R})$ telle que définie dans le chapitre 1, qui comporte $n = |\mathcal{T}|$ transactions décrites par un ensemble $\mathcal{I}$ de m items (ou attributs). On note $n_X = |X'|$ (resp. $n_Y = |Y'|$, $n_{XY} = |X' \cap Y'|$, $n_{X\overline{Y}} = |X' \cap \overline{Y'}|$) le nombre de transactions qui réalisent X (resp. Y, $X \cap Y$ et $X \cap \overline{Y}$). En fait, n_{XY} comptabilise les exemples $X \cap Y$ de $X \to Y$, c-à-d ceux qui vérifient conjointement la prémisse et la conclusion. Tandis que $n_{X\overline{Y}}$ comptabilise les contre-exemples $X \cap \overline{Y}$ de $X \to Y$, ceux qui vérifient la prémisse mais pas la conclusion. On note N_{XY} la variable aléatoire qui génère n_{XY}, alors que $N_{X\overline{Y}}$ celle qui génère $n_{X\overline{Y}}$. Pour évaluer la qualité d'une règle d'association, deux mesures de qualité sont couramment utilisées : le support et la confiance d'Agrawal [AIS93] :

$$supp(X \to Y) = supp(X \cup Y) = P(X' \cap Y') = \frac{n_{XY}}{n} \tag{2.1}$$

$$conf(X \to Y) = P(Y'|X') = \frac{P(X' \cap Y')}{P(X')} = \frac{n_{XY}}{n_X} \tag{2.2}$$

Ces deux mesures fondent la stratégie utilisée dans les principaux algorithmes d'extraction, à la suite d'Apriori [AS94]. En effet, une règle $X \to Y$ sera valide selon support/confiance si $supp(X \to Y) \geq minsup$ et $conf(X \to Y) \geq minconf$, où $minsup$ et $minconf$ sont des seuils préalablement fixés dans $[0; 1]$. Toutefois, le couple support/confiance, malgré sa popularité courante, présente des défauts notables. Entre autres, il produit des règles trop nombreuses dont plusieurs sont inintéressantes. Plus précisément, les propriétés sémantiques de la confiance ne prennent pas compte de la dépendance négative et de l'indépendance entre prémisse et conclusion. En conséquence, des règles inintéressantes demeurent encore parasiter les résultats de ce couple support/confiance.

Définition 8 ([BRT18])**.** *Une règle $X \to Y$ est dite inintéressante si X et Y sont (i) statistiquement indépendants (i.e. $P(Y'|X') = P(Y')$), ou (ii) négativement dépendants (i.e. $P(Y'|X') < P(Y')$).*

Théorème 5. *Pour tous $X, Y \subseteq \mathcal{I}$, aucune des règles $X \to Y$, $X \to \overline{Y}$, $\overline{X} \to Y$ et $\overline{X} \to \overline{Y}$ ne peut être intéressante si X et Y sont statistiquement indépendants.*

Démonstration. Cf. Proposition 3 (resp. Propriété 4) dans [BRT18] (resp. [BT20b, BT20d, BT20e]). □

Définition 9 ([BRT18])**.** *Pour tous deux motifs disjoints X et Y, une règle $X \to Y$ est dite intéressante si $P(Y'|X') > P(Y')$, c'est-à-dire lorsqu'il y a dépendance positive entre X et Y.*

Autrement dit, une règle d'association $X \to Y$ est intéressante lorsque sa Confiance $P(Y'|X')$ dépasse une valeur de référence $P(Y')$ (Probabilité de conclusion) de telle règle d'association.

A ces hypothèses, nous considérons deux exemples fictifs du tableau 2.1 ci-dessous menés à deux couples (A, B) et (C, D). Pour le couple (A, B), on a $supp(A \cup B) = 0.72$ et $conf(A \to B) = 0.9$.

Table 2.1 – Table de contigence des couples fictifs (A, B) et (C, D)

	B	$\neg B$	$\sum$		D	$\neg D$	$\sum$
A	72	8	80	C	20	5	25
$\neg A$	18	2	20	$\neg C$	70	5	75
$\sum$	90	10	100	$\sum$	90	10	100

Ces valeurs raisonnablement élevées nous invitent à penser que la règle $A \to B$ est potentiellement pertinente. Pourtant, $P(B'|A') = P(B') = 0.9$ signifie que B est indépendant de A, ainsi la règle d'association $A \to B$ n'est pas non plus intéresante. Pour (C, D), on obtient $supp(C \cup D) = 0.2$ et $conf(C \to D) = 0.8$. Or, $P(D'|C') = 0.8 < 0.9 = P(D')$ implique que C et D sont négativement dépendants, donc la règle d'association $C \to D$ n'est pas non plus intéressante.

Il est donc nécessaire de disposer d'autres mesures plus sélectives prenant en compte de l'écart de la confiance à sa valeur de référence (probabilité de conclusion de la règle), qui s'appelle souvent *l'écart à l'indépendance*. Ainsi définie, entre autres, la mesure M_{GK}, introduite initialement par

Guillaume [Gui00] puis raffinée par Totohasina *et al.* (cf. [TR05, TF08]) :

$$M_{GK}(X \to Y) = \begin{cases} \frac{P(Y'|X') - P(Y')}{1 - P(Y')}, \text{si } P(Y'|X') > P(Y') \\ \frac{P(Y'|X') - P(Y')}{P(Y')}, \text{si } P(Y'|X') \leq P(Y') \end{cases} \tag{2.3}$$

Dans la relation (2.3), la première composante (resp. deuxième composante) est aussi appelée *composante favorisante* (resp. *composante défavorisante*) de M_{GK}, et notée respectivement par M^f_{GK} et M^d_{GK}. Dans le reste du rapport, nous utiliserons la notation M_{GK} pour désigner M^f_{GK} ou M^d_{GK} selon le cas. La mesure M_{GK} varie dans $[-1; 1]$. Plus elle s'éloigne de 0, plus le couple (X, Y) est fortement dépendant (positivement ou négativement). L'originalité de M_{GK} réside sur le fait qu'elle permet d'éviter systématiquement les règles inintéressantes. En effet, avec les mêmes exemples du tableau 2.1, on a :$P(B'|A') = P(B') \Rightarrow M_{GK}(A \to B) = 0$, et $P(D'|C') < P(D') \Rightarrow M_{GK}(C \to D) < 0$. Ces aspects prouvent que $A \to B$ et $C \to D$ sont inintéressantes selon M_{GK}.

Pour décider une règle d'association avec M_{GK}, Totohasina *et al.* [TR05, DRT07] ont défini une valeur critique d'une certaine règle d'association $X \to Y$, notée $\eta_\alpha(X \to Y)$, et définie par :

$$\eta_\alpha(X \to Y) = \sqrt{\frac{1}{n} \frac{n - n_X}{n_X} \frac{n_Y}{n - n_Y} \chi^2(\alpha)}, \tag{2.4}$$

où χ^2 est une statistique de Khi-deux à un degré de liberté, qui dépend à un seuil critique $\alpha \in [0, 1[$. Ainsi, une telle règle $X \to Y$ sera valide si $supp(X \cup Y) \geq minsup$ et $M_{GK}(X \to Y) \geq \eta_\alpha(X \to Y)$.

2.2 Conception d'une nouvelle mesure de qualité

D'un côté, nous avons discuté, dans la section 2.1, l'intérêt remarquable d'une mesure qui tient compte de l'écart à l'indépendance. D'autre côté, il arrive bien souvent que les transactions décrites dans une base de données à partir desquelles sont extraites les règles d'association ne sont qu'un échantillon plus vaste. Entre ces deux aspects, il est naturel de disposer d'une mesure qui permet de prendre en compte à la fois de l'écart à l'indépendance et de la taille de l'échantillon. Une mesure usuellement utilisée pour cela est *l'intensité d'implication* de Gras [GAB+96]. Cette mesure, initialement appliquée à la didatique des mathématiques, a ensuite été utilisée en fouille de données [Fle96]. L'intensité d'implication φ d'une règle d'association $X \to Y$ est définie par :

$$\varphi(X, Y) = P(N_{X\overline{Y}} \geq n_{X\overline{Y}} | H_0) \tag{2.5}$$

Basée sur l'indice d'implication $q(X, \overline{Y}) = \frac{n_{X\overline{Y}} - \frac{n_X n_{\overline{Y}}}{n}}{\sqrt{\frac{n_X n_{\overline{Y}}}{n}}}$, la mesure φ quantifie l'invraisemblance de la faiblesse du nombre de contre-exemples observés $n_{X\overline{Y}}$ sous l'hypothèse H_0 d'indépendance entre X et Y. Cette fois-ci, la variable $N_{X\overline{Y}}$ est une variable poissonienne [Ler81]. En notant Φ la fonction de répartition de la loi normale standardisée $\mathcal{N}(0, 1)$, on a $P(N_{X\overline{Y}} \leq n_{X\overline{Y}} | H_0) \cong \Phi(q(X, \overline{Y}))$, et l'intensité φ peut se réécrire :

$$\varphi(X, Y) = 1 - \Phi(q(X, \overline{Y})) \tag{2.6}$$

Toutefois, depuis ces dernières années, φ suscite plusieurs critiques [CG15, CP15, Hah17, DLL17], malgré son intérêt notable. En effet, lorsque n est grand, en statistique tout particu-

lièrement, le moindre écart à l'indépendance devient très significatif, et la valeur de φ reste très collée à la valeur maximale 1. Pour y faire face, nous proposons une nouvelle mesure statistique plus discriminante, notée mgk, une adaptation de la mesure M_{GK} (cf. Eq. (2.3)). La mesure mgk a été abordée dans [BT20a, BT20b, BT20e, BT21b] et présentée comme suit. Nous avons adopté le même procédé que pour l'intensité d'implication φ. Tout d'abord, on détermine la loi de la variable aléatoire $N_{X\overline{Y}}$ sous l'hypothèse H_0 d'indépendance entre X et Y. Associons ensuite aux motifs X et Y deux autres motifs Z et T de $\mathcal{I}$, tirés aléatoirement et indépendamment de mêmes cardinaux respectifs que X et Y (i.e. $|Z'| = n_X$ et $|T'| = n_Y$), et on obtient $N_{X\overline{Y}} = |\phi(Z \cup \overline{T})|$, qui suit une loi de Poisson de paramètre $\lambda = \frac{n_X n_{\overline{Y}}}{n}$ [Ler81]. Il est important de rappeler que seule la composante favorisante M^f_{GK} de M_{GK} est implicative, et sera active dans la modélisation. Nous définissons ensuite sa valeur observée pour une règle d'association $X \to Y$, notée $\widetilde{mgk}(X,Y)$, et définie par :

$$\widetilde{mgk}(X,Y) = \frac{\frac{n_X n_{\overline{Y}}}{n} - n_{X\overline{Y}}}{\frac{n_X n_{\overline{Y}}}{n}} = -\frac{n_{X\overline{Y}} - \frac{n_X n_{\overline{Y}}}{n}}{\frac{n_X n_{\overline{Y}}}{n}} = -\widetilde{mgk}(X,\overline{Y}) = \frac{-q(X,\overline{Y})}{\sqrt{\lambda}} \tag{2.7}$$

On détermine la p-value de cette valeur observée $\widetilde{mgk}(X,Y)$. Cette p-value correspond à la probabilité $P(N_{X\overline{Y}} \leq n_{X\overline{Y}}|H_0)$. En centrant et réduisant la variable aléatoire $N_{X\overline{Y}}$ sous H_0, on a :

$$\begin{aligned} P(N_{X\overline{Y}} \leq n_{X\overline{Y}}|H_0) &= P\left(\frac{N_{X\overline{Y}} - \lambda}{\sqrt{\lambda}} \leq \frac{n_{X\overline{Y}} - \lambda}{\sqrt{\lambda}}\right) \\ &= P\left(\frac{q(X,\overline{Y})}{\sqrt{\lambda}} \leq \widetilde{mgk}(X,\overline{Y})\right). \end{aligned}$$

Pour $\lambda \geq 5$, la variable notée $Q(X,\overline{Y}) = \dfrac{q(X,\overline{Y})}{\sqrt{\lambda}}$ suit approximativement la loi normale standardisée $\mathcal{N}(0,1)$, et la p-value $P(N_{X\overline{Y}} \leq n_{X\overline{Y}}|H_0)$ peut s'écrire en conséquence comme :

$$P(N_{X\overline{Y}} \leq n_{X\overline{Y}}|H_0) = P\left(Q(X,\overline{Y}) \leq \widetilde{mgk}(X,\overline{Y})\right) \cong \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\widetilde{mgk}(X,\overline{Y})} e^{-\frac{t^2}{2}}\, \mathrm{d}t \tag{2.8}$$

Insistons sur le sens de cette intégrale, celle-ci représente la probabilité gaussienne pour que le nombre de transactions observées satifaisant la règle d'association $X \to Y$ soit supérieur à celui qui serait attendu sous l'hypothèse H_0 d'indépendance entre les motifs fréquents X et Y dans la base de données $\mathcal{D}$. Autrement dit, la probabilité $P\left(Q(X,\overline{Y}) \leq \widetilde{mgk}(X,\overline{Y})\right)$ représente la p-value du test visant à refuter l'hypothèse d'indépendance de tels motifs fréquents X et Y dans $\mathcal{D}$.

Définition 10 ([BT21b])**.** *Partant de deux motifs disjoints X et Y tels que $n_Y \neq n$, la mesure statistique d'une règle $X \to Y$, notée $mgk(X,Y)$, est approximativement donnée par :*

$$mgk(X,Y) = 1 - P\left(Q(X,\overline{Y}) \leq \widetilde{mgk}(X,\overline{Y})\right) \cong 1 - \Phi\left(\widetilde{mgk}(X,\overline{Y})\right) \tag{2.9}$$

La mesure mgk est une mesure à valeurs dans $[0,1]$. Pour tous $X, Y \subseteq \mathcal{I}$, mgk quantifie alors la dépendance *positive* entre X et Y. Elle est égale à 0 en cas d'indépendance entre X et Y (i.e. $n_{X\overline{Y}} = \frac{n_X n_{\overline{Y}}}{n}$), passe à 1/2 à l'équilibre (i.e. $n_{X\overline{Y}} = n_{XY}$), et atteint 1 à l'implication logique (i.e. $n_{X\overline{Y}} = 0$). Son expression qualitative sous certain seuil critique α tel que $0 \leq \alpha < 1$ devient :

Définition 11 ([BT21a, BT21b]). *Soit $X, Y \subseteq \mathcal{I}$ tels que $X \cap Y = \emptyset$. Une règle d'association $X \rightarrow Y$ est dite valide au niveau de confiance $1-\alpha$, notée aussi $(1-\alpha)$-valide, si et seulement si :*

$$mgk(X,Y) = 1 - \Phi\left(\widetilde{mgk}(X,\overline{Y})\right) \geq 1-\alpha \tag{2.10}$$

Définition 12. *Une règle d'association $X \rightarrow Y$ est dite* exacte *si $mgk(X,Y) = 1$ (i.e. $n_{X\overline{Y}} = 0$), et* approximative *si $mgk(X,Y) < 1$ (i.e. $n_{X\overline{Y}} \neq 0$).*

Autrement dit, une règle d'association exacte est de la forme $X \rightarrow Y \backslash X$ telle que $X \cong Y$. Une telle règle d'association sera dite approximative si $supp(X) \neq supp(Y)$ (i.e. $mgk(X,Y) \neq 1$).

Corollaire 3. *Soit $X \rightarrow Y \cup \{l\}$ une règle d'association exacte dans $\mathcal{D}$, pour tout itemset fréquent l. Alors, la règle $X \rightarrow Y$ est aussi une règle d'association exacte de la base de données $\mathcal{D}$.*

Proposition 6 ([BT19a]). *Pour 2 motifs disjoints X et Y, on a $P(Y'|X') = 1 \Leftrightarrow mgk(X,Y) = 1$.*

Les théorèmes 6, 7, 8 et 9 ci-après montrent que notre approche possède un comportement efficace, tant en terme d'espace de recherche que de temps de calcul. Cette efficacité découle de la propriété dichotomique de M_{GK}, héritée de la mesure statistique mgk comme le lemme 1 l'explique :

Lemme 1 ([BT21a, BT21b, BCT21]). *Pour tous $X, Y \subseteq \mathcal{I}$, on a $mgk(X,Y) = -mgk(X,\overline{Y})$.*

Idée de preuve : Dans [Fen07, Tot08, Bem16, BT19a, BT20e], nous avons démontré que, $\forall X, Y \subseteq \mathcal{I}$, on a $M_{GK}(X \rightarrow Y) = -M_{GK}(X \rightarrow \overline{Y})$ équivalent à $mgk(X,Y) = -mgk(X,\overline{Y})$. Cette dichotomie s'explique par le fait que deux règles contraires sont contra-variantes, puisque les exemples de l'une sont les contre-exemples de l'autre, et vice-versa. Si l'une possède plus d'exemples et donc moins de contre-exemples qu'à l'indépendance, alors l'autre possède moins d'exemples et plus de contre-exemples qu'à l'indépendance, et vice-versa. Dans les deux cas, on aura toujours à considérer une *dépendance positive* de deux motifs comme expliqué dans la section 2.1.

Théorème 6. *Pour tous $X, Y \subseteq \mathcal{I}$ tels que $P(Y'|X') > P(Y')$, on a $mgk(X,Y) = mgk(\overline{Y},\overline{X})$.*

Idée de preuve : Dans [BT19b, BT20e], nous avons démontré que si X favorise Y (i.e., $P(Y'|X') > P(Y')$), on a $M_{GK}(X \rightarrow Y) = M_{GK}(\overline{Y} \rightarrow \overline{X})$, d'où $mgk(X,Y) = mgk(\overline{Y},\overline{X})$. Il en résulte que la mesure M_{GK} est implicative, et il en est de même pour la mesure statistique mgk.

Théorème 7. *$\forall X, Y \subseteq \mathcal{I}$ tels que $X \subset Y$ et $P(Y'|X') > P(Y')$, on a $mgk(X,Y) < mgk(\overline{X},\overline{Y})$.*

Idée de preuve : Comme $P(Y'|X') > P(Y')$, on a $M_{GK}(Y \rightarrow X) = \frac{P(X')}{P(\overline{X'})}\frac{P(\overline{Y'})}{P(Y')}M_{GK}(X \rightarrow Y)$ [BT19a, BT20e]. Puisque $X \subset Y$, on a $P(X') > P(Y') \Leftrightarrow P(\overline{X'}) < P(\overline{Y'}) \Leftrightarrow P(X')P(\overline{Y'}) > P(\overline{X'})P(Y')$ implique $M_{GK}(X \rightarrow Y) < M_{GK}(Y \rightarrow X) = M_{GK}(\overline{X} \rightarrow \overline{Y})$. Ainsi, $M_{GK}(X \rightarrow Y) < M_{GK}(\overline{X} \rightarrow \overline{Y})$, d'où $mgk(X,Y) < mgk(\overline{X},\overline{Y})$.

Théorème 8. *Soient $X, Y \subseteq \mathcal{I}$ tels que $P(Y'|X') < P(Y')$, on a (i) $mgk(X,\overline{Y}) = mgk(Y,\overline{X})$, et (ii) $mgk(\overline{X},Y) = mgk(\overline{Y},X)$.*

Schéma de preuve : Puisque $P(Y'|X') < P(Y')$, on obtient, d'après le lemme 1 : (a) $P(\overline{Y'}|X') > P(\overline{Y'})$, (b) $P(\overline{X'}|Y') > P(\overline{X'})$, (c) $P(Y'|\overline{X'}) > P(Y')$, et (d) $P(X'|\overline{Y'}) > P(X')$. Le reste de la preuve découle de la propriété implicative de la mesure M_{GK}. (i) Comme $P(\overline{Y'}|X') > P(\overline{Y'})$ et $P(\overline{X'}|Y') > P(\overline{X'})$, on obtient, grâce à la propriété implicative de la mesure M_{GK}, $M_{GK}(X \rightarrow \overline{Y}) = M_{GK}(Y \rightarrow \overline{X})$ équivaut à $mgk(X,\overline{Y}) = mgk(Y,\overline{X})$. (ii), Comme $P(Y'|\overline{X'}) > P(Y')$ et $P(X'|\overline{Y'}) > P(X')$, on obtient, toujours à l'aide de la propriété implicative de la mesure M_{GK}, $M_{GK}(\overline{X} \rightarrow Y) = M_{GK}(\overline{Y} \rightarrow X)$ équivaut à $mgk(\overline{X},Y) = mgk(\overline{Y},X)$.

Théorème 9. *Soient $X, Y \subseteq \mathcal{I}$ tels que $X \subset Y$ et $P(Y'|X') < P(Y')$: $mgk(X, \overline{Y}) < mgk(\overline{X}, Y)$.*

Schéma de preuve : Comme $P(Y'|X') < P(Y')$, on a $P(\overline{Y'}|X') > P(\overline{Y'})$ (cf. Lemme 1) implique que le nombre de contre-exemples est supérieur au nombre d'exemples. De $P(\overline{Y'}|X') > P(\overline{Y'})$, on a $M_{GK}(X \rightarrow \overline{Y}) = \frac{P(\overline{X'})P(\overline{Y'})}{P(X')P(Y')} M_{GK}(\overline{X} \rightarrow Y)$ [BT19a, BT20e], et il faut que $P(\overline{X'})P(\overline{Y'}) < P(X')P(Y')$. Ainsi, $M_{GK}(X \rightarrow \overline{Y}) < M_{GK}(\overline{X} \rightarrow Y)$, d'où $mgk(X, \overline{Y}) < mgk(\overline{X}, Y)$.

Théorème 10. *Soient $X_1 \rightarrow Y \backslash X_1$ et $X_2 \rightarrow Y \backslash X_2$ deux règles quelconques d'une base de données telles que $X_1 \cap Y = \emptyset$, $X_2 \cap Y = \emptyset$, et $X_1 \subseteq X_2 \subseteq Y$, on a $mgk(X_1, Y) \leqslant mgk(X_2, Y)$.*

Démonstration. Puisque $X_1 \subseteq X_2$, on obtient $\phi(X_2) \subseteq \phi(X_1) \Rightarrow \frac{|\phi(X_2)|}{\mathcal{T}} \leqslant \frac{|\phi(X_1)|}{\mathcal{T}} \Leftrightarrow supp(X_2) \leqslant supp(X_1)$ (i.e. $P(X_2') \leqslant P(X_1')$. D'autre part, si $X_1 \subseteq X_2 \subseteq Y$, on a alors $P(Y'|X_1') = \frac{|\phi(X_1 \cup Y)|}{\phi(X_1)} = \frac{|\phi(X_1) \cap \phi(Y)|}{|\phi(X_1)|} = \frac{|\phi(Y)|}{|\phi(X_1)|}$. Comme $\phi(X_2) \subseteq \phi(X_1)$, on a $\frac{|\phi(Y)|}{|\phi(X_1)|} \leqslant \frac{|\phi(Y)|}{|\phi(X_2)|} \Leftrightarrow \frac{P(Y')}{P(X_1')} \leqslant \frac{P(Y')}{P(X_2')} \Leftrightarrow P(Y'|X_1') \leqslant P(Y'|X_2') \Leftrightarrow P(Y'|X_1') - P(Y') \leqslant P(Y'|X_2') - P(Y')$. Pour $P(Y') \neq 1$, on a $\frac{P(Y'|X_1') - P(Y')}{1 - P(Y')} \leqslant \frac{P(Y'|X_2') - P(Y')}{1 - P(Y')} \Leftrightarrow M_{GK}(X_1 \rightarrow Y \backslash X_1) \leqslant M_{GK}(X_2 \rightarrow Y \backslash X_2)$ d'où $mgk(X_1, Y) \leqslant mgk(X_2, Y)$. □

Il en résulte que mgk est antimonotone selon l'inclusion '$\subseteq$' des attributs, c'est-à-dire que plus on fait passer d'attributs de gauche à droite, plus mgk diminue. Par exemple, pour un 4-itemset fréquent $ABCD$, on a $mgk(ABC \rightarrow D) \geqslant mgk(AB \rightarrow CD) \geqslant mgk(A \rightarrow BCD)$ signifie que si la règle $A \rightarrow BCD$ est valide, alors les règles $ABC \rightarrow D$ et $AB \rightarrow CD$ sont aussi valides. Inversement, si la règle $ABC \rightarrow D$ n'est pas valide, alors les règles $AB \rightarrow CD$ et $A \rightarrow BCD$ ne sont pas non plus valides. Ce théorème est central pour la génération des règles d'association approximatives, et permet d'éliminer les règles non-génératrices (i.e. redondantes) dans un jeu de données.

2.3 Elimination des règles d'association redondantes

Cette section s'inscrit dans une double problématique : (*i*) Elagage de l'espace de recherche des meilleures règles (sous-section 2.3.1), et (*ii*) Définition de nouvelles bases des règles (sous-sec. 2.3.2).

2.3.1 Elagage de l'espace de recherche des meilleures règles

La complexité de la génération des règles d'association, comme la fouille des motifs, est aussi exponentielle. Elle est dramatique lorsqu'on considère les règles négatives, comme ces types sont très subtils. Ainsi, pour un motif fréquent I, le nombre de règles positives et négatives qui peuvent être enregistrées est égal à $5^{|I|} - 2 \times 3^{|I|} + 1$, au lieu de $3^{|I|} - 2^{|I|+1} + 1$ pour les positives uniquement. Par rapport à ces dimensions, il est naturel de ne pas parcourir exhaustivement l'espace de recherche.

Dans [BT19a, BT20e], nous avons proposé une méthode *reduce-rules-space* permettant la réduction efficace de l'espace de recherche des meilleures règles d'association d'un contexte $\mathcal{D}$. La pierre angulaire de cette méthode repose sur le lemme 1 et les théorèmes 6, 7, 8 et 9 ci-dessus. Nous expliquons cette restriction en exploitant le concept de dépendance positive entre les motifs.

Soit $\mathscr{C} = \{X \rightarrow Y, Y \rightarrow X, \overline{X} \rightarrow \overline{Y}, \overline{Y} \rightarrow \overline{X}, X \rightarrow \overline{Y}, \overline{X} \rightarrow Y, \overline{Y} \rightarrow X, Y \rightarrow \overline{X}\}$ un ensemble de toutes les règles positives/négatives candidates. Le lemme 1 montre que si X favorise Y (i.e. $P(Y'|X') > P(Y')$), alors X défavorise $\overline{Y}$ (i.e. $P(\overline{Y'}|X') < P(\overline{Y'})$). Ce qui relève que si X favorise Y, ce sont les 4 règles $X \rightarrow Y$, $Y \rightarrow X$, $\overline{X} \rightarrow \overline{Y}$ et $\overline{Y} \rightarrow \overline{X}$ qui sont étudiées. Inversement, si X défavorise Y (i.e. X favorise $\overline{Y}$), ce sont alors les 4 règles contraires $X \rightarrow \overline{Y}$, $\overline{X} \rightarrow Y$, $\overline{Y} \rightarrow X$ et

$Y \rightarrow \overline{X}$ qui sont étudiées. Cela partitionne l'ensemble $\mathscr{C}$ en 2 sous-ensembles disjoints, que l'on note par $\mathscr{C}_1 = \{X \rightarrow Y, Y \rightarrow X, \overline{X} \rightarrow \overline{Y}, \overline{Y} \rightarrow \overline{X}\}$ et $\mathscr{C}_2 = \{X \rightarrow \overline{Y}, \overline{X} \rightarrow Y, \overline{Y} \rightarrow X, Y \rightarrow \overline{X}\}$. Nous expliquons par suite la restriction de l'espace de recherche pour chacun des $\mathscr{C}_1$ et $\mathscr{C}_2$.

Au niveau de $\mathscr{C}_1$, nous avons établi, grâce au théorème 6, 2 relations d'équivalence de deux règles contraposées : $mgk(X,Y) = mgk(\overline{Y},\overline{X})$ et $mgk(Y,X) = mgk(\overline{X},\overline{Y})$, $\forall X, Y \subseteq \mathcal{I}$, s'agissant que si $X \rightarrow Y$ et $\overline{X} \rightarrow \overline{Y}$ sont valides, alors $\overline{Y} \rightarrow \overline{X}$ et $Y \rightarrow X$ sont aussi valides, et vice-versa. Autrement dit, $\overline{Y} \rightarrow \overline{X}$ (resp. $Y \rightarrow X$) est dérivable de $X \rightarrow Y$ (resp. $\overline{X} \rightarrow \overline{Y}$), et réciproquement. Grâce au théorème 7, nous avons la relation $mgk(X,Y) < mgk(Y,X)$, $\forall X \subseteq Y$, exprimant que si $X \rightarrow Y$ est valide au sens de mgk, alors $Y \rightarrow X$ l'est aussi. Disons que la règle $Y \rightarrow X$ est dérivable de $X \rightarrow Y$. En conclusion, ces 3 relations nous revèlent que l'intérêt des 3 règles $\overline{X} \rightarrow \overline{Y}$, $\overline{Y} \rightarrow \overline{X}$ et $Y \rightarrow X$ peut être déduit de celui de la règle $X \rightarrow Y$. Autrement dit, tous les éléments de $\mathscr{C}_1$ peuvent être dérivés d'une seule règle $X \rightarrow Y$, soit 3/4 de réduction d'espace de recherche.

Au niveau de $\mathscr{C}_2$, on obtient, grâce au théorème 8, 2 relations : $mgk(X,\overline{Y}) = mgk(Y,\overline{X})$ et $mgk(\overline{X},Y) = mgk(\overline{Y},X)$, signifiant que $Y \rightarrow \overline{X}$ (resp. $\overline{Y} \rightarrow X$) est redondante si $X \rightarrow \overline{Y}$ (resp. $\overline{X} \rightarrow Y$) est valide, et réciproquement. Nous ne conservons à cet effet que $X \rightarrow \overline{Y}$ et $\overline{X} \rightarrow Y$ au détriment des $Y \rightarrow \overline{X}$ et $\overline{Y} \rightarrow X$. De plus, nous avons établi, grâce au théorème 9, une relation $mgk(X,\overline{Y}) \leq mgk(\overline{X},Y)$, $\forall X \subset Y$, indiquant quant à elle que si $X \rightarrow \overline{Y}$ est valide, alors $\overline{X} \rightarrow Y$ le sera également. En conclusion, grâce à ces trois relations, il en résulte que toutes les règles de $\mathscr{C}_2$ peuvent être dérivées d'une seule règle $X \rightarrow \overline{Y}$, soit 3/4 de réduction de l'espace de recherche.

En bref, nous obtenons dans l'ensemble $\mathscr{C}$ une famille des 4 règles, à savoir $X \rightarrow Y$, $X \rightarrow \overline{Y}$, $\overline{X} \rightarrow Y$ et $\overline{X} \rightarrow \overline{Y}$. Nous n'étudions dans cette nouvelle famille que 2 types, à savoir $X \rightarrow Y$ et $X \rightarrow \overline{Y}$, donc 1/4 de l'ensemble $\mathscr{C}$, soit une restriction de 75% de l'espace de recherche.

2.3.2 Définition de nouvelles bases des règles d'association

Comme signalé dans la sous-section 2.3.1 précédente, le nombre de règles d'association, pouvant être extraites dans une base de données, est souvent élevé, dont la majorité sont redondantes. En conséquence, l'utilisateur rencontre des difficultés pour effectuer convenablement l'interprétation de ses résultats. Pour y faire face, diverses approches [GYNS06, HYN11, LHH12], fondées sur le concept des bases de règles, ont été proposées. Une base de règles d'association est un ensemble minimal des règles non-redondantes à partir duquel on peut dériver les autres règles valides. De façon classique, une base des règles est cette fois constituée par 2 sous-bases : base des règles exactes et base des règles approximatives. Comme signalé, ces approches précitées ne parviennent pas à traiter les règles d'association négatives, et ce, avec le couple moins sélectif support/confiance. De plus, elles encaissent aussi les règles d'association inintéressantes au sens de la définition 8.

La première tentative de solution à ces limites remarquables, particulièrement celles relevées dans [PTB+05, GYNS06], a été proposée dans [FDT06, FDT07, Fen07, Tot08], où ces derniers définissent, selon la mesure plus sélective M_{GK}, une base des règles positives exactes (soit BRPE), une base des règles positives approximatives (BRPA), une base des règles négatives exactes (BRNE), et une base des règles négatives approximatives (BRNA). Ces propositions ont été raffinées dans [Ram16], où l'auteur redéfinit ces 4 bases de règles. Cependant, cinq lacunes surgissent :

1. Ces approches correctives [FDT07, Ram16] manquent de l'autonomie en terme de génération des bases des règles, à cause de l'absence d'un modèle permettant l'extraction des motifs (fermés, maximaux et générateurs) fréquents dans une base de données, étant donnés que ces types des motifs jouent un rôle crucial à l'élaboration de ces bases des règles. Cela rend ainsi difficile (ou quasi-impossible) leur réalisation pratique (i.e., leur passage à l'échelle).

2. Elles souffrent du temps de calcul très grand surtout pour des grandes bases de données, à cause de la valeur critique η_α (Eq. (2.4)) utilisée pour décider une règle. En effet, chaque règle d'association candidate dispose de sa propre valeur critique η_α en fonction de χ^2, calculée intermédiairement à partir d'une base de données pouvant être grande et/ou dense. Etant donnée qu'une règle d'association est constituée au moins de deux motifs, et le coût de calcul d'un motif ℓ de $\mathcal{I}$ à partir d'un tableau de contigence pour le χ^2 atteint $4\binom{|\mathcal{I}|}{|\ell|}2^{|\ell|}$.
3. Elles souffrent de la perte de meilleures règles d'association (règles proches de l'implication logique, i.e., proches de valeur maximale 1) d'une part, et d'autre part, de la génération de mauvaises règles d'association (règles proches de l'indépendance statistique, i.e., proches de 0), à cause de cette valeur critique η_α. Autrement dit, elles ignorent parfois les règles dont M_{GK} est proche de 1 mais inférieure à η_α. A l'inverse, elles acceptent souvent les règles dont M_{GK} est proche de 0 mais supérieure à η_α. Par conséquent, certaines propriétés sémantiques de M_{GK} développées par ces mêmes auteurs [DRT07, FDT07, Ram16] permettant d'éviter systématiquement les règles d'association dont prémisse et conséquent sont *proches de l'indépendance* ne sont pas respectées lors de réalisation pratique. Ces lacunes sont très handicapantes leurs résultats pour la génération des règles d'association approximatives.
4. Les bases des règles BRPE, BRPA, BRNE et BRNA définies dans [FDT06, FDT07, Ram16] ne parviennent pas à améliorer correctement la volumétrie de l'ensemble des règles valides. Plus précisément, elles ne sont pas minimales du fait qu'elles génèrent les règles non-génératrices.
5. Elles ne proposent aucune approche formelle pour la génération des règles d'association dérivées. En conséquence, l'ensemble des règles restituées par ces approches n'est pas complet.

Dans ce qui suit, nous allons essayer de porter quelques éléments d'amélioration à ces notables défauts. Le premier problème est déjà résolu dans [BT20a], où nous avons défini un algorithme autonome permettant l'extraction simultanée des motifs (fermés, maximaux et générateurs) fréquents tel que présenté dans le chapitre 1. Les 2e et 3e problèmes sont résolus dans [BT21a, BT21b], où nous avons défini un nouveau modèle d'élagage des règles valides, tel qu'abordé dans la section 2.2. A la différence des approches de [FDT06, FDT07, Ram16], il n'y a pas non plus des calculs à part pour la contrainte d'élagage des règles d'association valides. En effet, nous avons utilisé la p-value (cf. éq. (2.8)) du test visant à refuter l'hypothèse H_0 d'indépendance des X et Y d'une règle $X \to Y$. Pour certain $\alpha \in [0, 1[$, H_0 est acceptée si p-value est faible, i.e. si $P(N_{X\overline{Y}} \leq n_{X\overline{Y}}|H_0) < \alpha$, et rejetée sinon. Autrement dit, la règle $X \to Y$ est dite valide si $1 - P(N_{X\overline{Y}} \leq n_{X\overline{Y}}|H_0) \geq 1 - \alpha$. En fait, α étant arbitraire dans $[0, 1[$, et ses variations n'affectent pas des calculs intermédiaires pour décider statistiquement une telle règle $X \to Y$. Dans ce cas, on peut choisir des α faibles, et on aura de bonnes règles (sans perte de connaissances). A l'inverse, on peut prendre des α relativement grands, on aura alors des règles moins robustes, de telle sorte que le nombre de contre-exemples reste encore faible pour éviter les mauvaises règles pouvant être encaissées par [FDT06, FDT07, Ram16].

A l'égard des autres lacunes relevées tant pour les approches [GYNS06, HYN11, LHH12] que pour [FDT06, FDT07, Ram16], nous proposons de nouvelles formulations pour les bases des règles d'association en utilisant la nouvelle mesure *mgk* que nous avons proposée dans [BT20a, BT20e, BT21b]. De ce fait, en notant par $\mathcal{AR}$ l'ensemble de toutes les règles valides, notre approche consiste à trouver une base des règles non-redondantes qui fournit un ensemble de taille plus petite que $|\mathcal{AR}|$ et de qualité maximale. Nous nous intéressons pour cela à la base minimale des règles.

Définition 13 (Base minimale)**.** *Soient $\mathcal{AR}$ l'ensemble de règles valides, et $\mathcal{B}$ une base de l'ensemble $\mathcal{AR}$. $\mathcal{B}$ est dit minimal s'il n'existe pas d'ensemble des règles $\mathcal{B}' \subset \mathcal{B}$ tel que $\mathcal{B}'$ est une base de $\mathcal{AR}$.*

Nous définissons dans ce qui suit une règle d'association redondante d'une base de données $\mathcal{D}$.

Définition 14. *Une règle $r_1 : X_1 \rightarrow Y_1$ est dite redondante s'il existe une règle $r_2 : X_2 \rightarrow Y_2$ telle que : (i) $X_2 \subseteq X_1 \ \wedge \ Y_2 \supset Y_1$, (ii) $supp(r_1) = supp(r_2) \ \wedge \ mgk(r_1) = mgk(r_2)$.*

Dans cette définition 14, le premier critère consiste à réduire les redondances, car il ne sélectionne que de règles d'association informatives (i.e., règles d'association n'ayant que de règles d'association génératrices). Une règle d'association la plus informative est celle qui, à partir de la plus petite quantité d'information, fournit la plus grande quantité d'information. Autrement dit, une règle d'association $X \rightarrow Y$ est informative si X (resp. Y) est minimal (resp. maximal) au sens de l'inclusion '$\subset$'. Une telle règle d'association est dite génératrice si elle n'est couverte par aucune règle d'association dans un contexte $\mathcal{D}$, c'est-à-dire qu'il n'existe pas d'une règle d'association $T \rightarrow Z$ telle que $X \supseteq T$ et $Z \supseteq Y$. Le deuxième critère permet de garantir la qualité des règles d'association. C'est surtout le premier critère qui nous amène à définir ci-dessous (Définition 15) le concept d'une relation de couverture muni d'une relation d'ordre (partiel) que l'on note par $\prec$.

Définition 15. *Soit $\mathcal{AR}$ l'ensemble de toutes les règles valides dans un contexte $\mathcal{D}$. Soient $r_1 : X_1 \rightarrow Y_1$ et $r_2 : X_2 \rightarrow Y_2$ deux règles valides de $\mathcal{AR}$. La règle r_1 couvre la règle r_2 (ou d'une manière équivalente, r_2 est redondante par rapport à r_1), notée $r_2 \prec r_1$, si $X_2 \supseteq X_1$ et $Y_2 \subset Y_1$.*

Proposition 7. *Soit $\mathcal{AR}$ l'ensemble de toutes les règles valides dans un contexte $\mathcal{D}$. La relation d'ordre $\prec$ sur cet ensemble $\mathcal{AR}$, notée $(\mathcal{AR}, \prec)$, vérifie les prpriétés suivantes :*

- *réflexive : pour toute règle $r \in \mathcal{AR}$, on a $r \prec r$*
- *antisymétrique : pour toutes règles $r_1, r_2 \in \mathcal{AR}$, on a $r_1 \prec r_2 \wedge r_2 \prec r_1 \Rightarrow r_1 \equiv r_2$*
- *transitive : pour toutes règles $r_1, r_2, r_3 \in \mathcal{AR}$, on a $r_1 \prec r_2 \wedge r_2 \prec r_3 \Rightarrow r_1 \prec r_3$*

Rappelons que nos bases des règles d'association, comme dans [FDT07, Ram16], ne concernent que des règles d'association du type $X \rightarrow Y$ et $X \rightarrow \overline{Y}$, retenues grâce à la stratégie d'élagage de l'espace de recherche que nous l'avons présentée ci-dessus. Nous présentons ci-dessous nos résultats pour chacun de ces deux types de règles retenus, puis montrons à l'aide des théorèmes 11, 12, 13 et 14 ci-après que ces nouvelles bases sont non-redondantes. Les autres types de règles à savoir $\overline{X} \rightarrow Y$ et $\overline{X} \rightarrow \overline{Y}$ seront obtenus par dérivation comme nous présenterons dans la section 2.4.

Nous commençons par présenter nos résultats pour la base de règles positives exactes, notée $\mathcal{BE}^+$, afin de pallier la limite principale de la base bien connue BRPE [FDT06, Fen07, FDT07, Tot08, Ram16]. Un fait marquant pour la base BRPE est surtout qu'elle n'est pas minimale.

Définition 16. *Soient $\mathcal{RE}^+$ l'ensemble de toutes les règles positives exactes, $\mathcal{FC}$ l'ensemble des fermés, et $\mathcal{G}_{\mathcal{C}}$ l'ensemble des générateurs d'un fermé $\mathcal{C}$ de $\mathcal{FC}$. La base $\mathcal{BE}^+$ est définie par :*

$$\mathcal{BE}^+ = \{r : G \rightarrow \mathcal{C} \backslash G \mid G \in \mathcal{G}_{\mathcal{C}}, \ \mathcal{C} \in \mathcal{FC} \ , \ G \neq \mathcal{C}, \ \wedge \ (\nexists r' \in \mathcal{RE}^+ \text{ tel que } r \prec r')\} \tag{2.11}$$

Comme signalé, l'originalité de cette nouvelle base $\mathcal{BE}^+$ réside sur le fait qu'elle offre la possibilité d'élaguer les règles non-génératrices (i.e. redondantes) qui pourraient encaisser par les aproches précitées, grâce à la propriété d'élagage fondée sur la relation d'ordre '$\prec$' telle que définie dans la définition 15 ci-dessus. Cela garantit la minimalié que nous présentons dans le corollaire 4.

Les preuves des théorèmes 11 et 12 ci-dessous sont fortement liées au lemme 2 ci-après.

Lemme 2 ([BT20c]). *Soit $X, Y, T, Z \subseteq \mathcal{I}$ tels que $P(Y'|X') > P(Y')$ et $P(Z'|T') > P(Z')$. Si $X \subset T \subseteq \gamma(X)$, $Z \subset Y \subseteq \gamma(Z)$, on a alors $supp(X \cup Y) = supp(T \cup Z) \Rightarrow mgk(X, Y) = mgk(T, Z)$.*

En plus de [BT20c], la preuve de ce Lemme 2 est aussi rappelée dans l'annexe C.

Théorème 11. *Soit $\mathcal{RE}^+$ l'ensemble de toutes les règles positives exactes valides. (i) Toute règle de $\mathcal{RE}^+$ peut être dérivée de $\mathcal{BE}^+$, et (ii) toute règle de $\mathcal{BE}^+$ est une règle non-redondante.*

Démonstration. (*i*) Soit $r : G \rightarrow \mathcal{C}\backslash G \in \mathcal{BE}^+$ telle que $G \subset \mathcal{C}$ et $supp(G) = supp(\mathcal{C})$ (i.e., $|\phi(G)| = |\phi(\mathcal{C})|$). Puisque $r \in \mathcal{BE}^+$, r est alors génératrice, et on peut trouver au moins une règle $r_1 : X_1 \rightarrow Y_1\backslash X_1$ de $\mathcal{RE}^+$ couverte par r tel que $X_1 \subset Y_1$. De $G \subset \mathcal{C}$, on a $\phi(G) \supseteq \phi(\mathcal{C})$. De $\phi(G) \supseteq \phi(\mathcal{C})$ et $|\phi(G)| = |\phi(\mathcal{C})|$, on obtient $\phi(G) = \phi(\mathcal{C}) \Rightarrow \psi o \phi(G) = \psi o \phi(\mathcal{C})$, i.e. $\gamma(G) = \gamma(\mathcal{C}) = \mathcal{C}$. D'autre part, on obtient $mgk(r) = 1 \Leftrightarrow P(\mathcal{C}'|G') = 1 \Leftrightarrow \frac{supp(G \cup \mathcal{C})}{supp(G)} = 1 \Rightarrow supp(G \cup \mathcal{C}) = supp(G) = supp(\mathcal{C})$. De façon analogue, on obtient pour r_1, $supp(X_1 \cup Y_1) = supp(X_1) = supp(Y_1)$. Montrons maintenant que r_1 peut être dérivée de r. Par définition, $\mathcal{BE}^+$ est un sous-ensemble de $\mathcal{RE}^+$ (i.e. $\mathcal{BE}^+ \subset \mathcal{RE}^+$), alors $supp(X_1) = supp(G) = supp(Y_1) = supp(\mathcal{C}) \Rightarrow supp(X_1 \cup Y_1) = supp(G \cup \mathcal{C})$ (i.e. $supp(r_1) = supp(r)$) implique $mgk(r_1) = mgk(r)$ (cf. Lemme 2), ce qui prouve que r_1 est dérivable de r. Autrement dit, toute règle de $\mathcal{RE}^+$ peut être dérivée de la base $\mathcal{BE}^+$.

(*ii*) Soit $r_1 : G \rightarrow \mathcal{C}\backslash G$ une règle de $\mathcal{BE}^+$ telle que $G \subset \gamma(G) = \mathcal{C}$. Montrons qu'il n'existe aucune règle $r_2 : X_2 \rightarrow Y_2\backslash X_2 \in \mathcal{RE}^+\backslash\mathcal{BE}^+$ qui couvre r_1 tel que $supp(r_2) = supp(r_1)$, $mgk(r_2) = mgk(r_1)$. Si r_2 couvre r_1 (i.e. $r_1 \prec r_2$), alors on a : $X_2 \subseteq G$ et $\mathcal{C} \subseteq Y_2$. Comme $X_2 \subseteq G$, on a par définition d'un générateur (définition 5) $\gamma(X_2) \subseteq \gamma(G) = \mathcal{C} \Rightarrow X_2 \notin \mathcal{G}_\mathcal{C} \Rightarrow r_2 \notin \mathcal{BE}^+$. Comme $\mathcal{C} \subseteq Y_2$, on a par définition d'un fermé (définition 3), $\mathcal{C} = \gamma(\mathcal{C}) = \gamma(G) \subset Y_2 = \gamma(Y_2)$. On en déduit, grâce à la définition 5, que $G \notin \mathcal{G}_{Y_2}$ et conclut que $r_2 \notin \mathcal{BE}^+$. Dans tous les cas, la règle r_1 n'est pas couverte par la règle r_2, elle ne satisfait pas donc à la définition 15. Autrement dit, r_1 n'est couverte par aucune règle de $\mathcal{RE}^+\backslash\mathcal{BE}^+$. Cela conclut que $\mathcal{BE}^+$ est une base non-redondante. □

Corollaire 4. *Soit $\mathcal{RE}^+$ l'ensemble de règles positives exactes. La base $\mathcal{BE}^+$ est minimale.*

Démonstration. Soit $\mathcal{BE}^+$ une base de $\mathcal{RE}^+$. Supposons que $\mathcal{BE}^+$ n'est pas minimale, alors il existe $\widetilde{\mathcal{BE}}^+ \subset \mathcal{BE}^+$ tel que $\widetilde{\mathcal{BE}}^+$ est une base de $\mathcal{RE}^+$. Soit $r : g \rightarrow c\backslash g \in \mathcal{BE}^+$, $\exists r' : G \rightarrow \mathcal{C}\backslash G \in \widetilde{\mathcal{BE}}^+$ tel que $g \supseteq G$, $c\backslash g \subseteq \mathcal{C}\backslash G$ et $g \subseteq (c\backslash\mathcal{C}) \cup G$. De $\mathcal{C}\backslash G \supseteq c\backslash g$, on a $\mathcal{C} \supseteq \mathcal{C}\backslash G \supseteq c\backslash g$, donc $\mathcal{C} \supseteq c\backslash g$. Comme $r \in \mathcal{BE}^+$, on a $g \subset c$, ce qui implique évidemment $((c\backslash g) \cup g = c$ et $(c\backslash g) \cap g = \emptyset)$, donc $c\backslash(c\backslash g) = g$. Puisque $\mathcal{C} \supseteq c\backslash g$, on a alors $c\backslash\mathcal{C} \subseteq c\backslash(c\backslash g) = g$, ce qui fait $c\backslash\mathcal{C} \subseteq g$. De $g \supseteq G$ et $c\backslash\mathcal{C} \subseteq g$, on a $(c\backslash\mathcal{C}) \cup G \subseteq g \cup G = g$, c'est-à-dire $(c\backslash\mathcal{C}) \cup G \subseteq g$, contradiction avec $g \subseteq (c\backslash\mathcal{C}) \cup G)$, i.e. r n'est couverte par aucune règle de $\widetilde{\mathcal{BE}}^+$. Ce qui relève que $\mathcal{BE}^+$ est minimale. □

La définition 17, comme définition 16, qui définit la base des règles positives approximatives de notre approche, notée $\mathcal{BA}^+$, est aussi une extension des définitions 13 et 14 ci-dessus.

Définition 17. *Soient $\mathcal{RA}^+$ l'ensemble de toutes les règles positives approximatives, $\mathcal{FC}$ celui des fermés, $\mathcal{G}$ celui de générateurs, et α un seuil critique tq. $0 \leq \alpha < 1$. La base $\mathcal{BA}^+$ est définie par :*

$$\mathcal{BA}^+(\alpha) = \{r : g \rightarrow c\backslash g \mid (g,c) \in \mathcal{G} \times \mathcal{FC}, \gamma(g) \subset c, P(c'|g') > P(c'), mgk(g,c) \geq 1-\alpha, \\ \wedge\ (\nexists r' \in \mathcal{RA}^+ \textit{ tel que } r \prec r')\} \quad (2.12)$$

Cette base a deux avantages principaux. Le premier, à l'aide de la métrique $P(\mathcal{C}'|G') > P(\mathcal{C}')$, étant la possibilité d'élaguer les règles inintéressantes telles que définies dans la définition 8 (limites des approches classiques [GYNS06, HYN11, LHH12]). Le second, grâce à la contrainte d'élagage $mgk(G,\mathcal{C}) \geq 1-\alpha$, est l'obtention des meilleures règles associatives valides (limites des approches correctives [FDT06, FDT07, Ram16] basées sur la valeur critique). Concrètement, la contrainte d'élagage $mgk(G,\mathcal{C}) \geq 1-\alpha$ permet de garantir simultanément l'élagage des règles d'association

proches de l'indépendance qui pourraient être encaissées par ces approches correctives [FDT06, FDT07, Ram16], et la récupération des règles fortes naïvement écartées par celles-ci.

Lemme 3. *Soient $\mathcal{FC}$ un ensemble des motifs fermés fréquents, $\mathcal{G}$ l'ensemble des générateurs minimaux des motifs fermés dans $\mathcal{FC}$. Si $\exists \mathcal{C} \in \mathcal{FC}$ et $\exists G \in \mathcal{G}$ tels que $\gamma(G) \subset \mathcal{C}$, $g \subset G$, $g \subseteq ((c\backslash\mathcal{C}) \cup G)$ et $mgk(g \to c\backslash g) < mgk(G \to \mathcal{C}\backslash G)$, alors $G \to \mathcal{C}\backslash G$ est dérivable de $g \to c\backslash g$.*

Démonstration. Soit $Q = c\backslash\mathcal{C}$ tel que $c \supseteq Q \cup \mathcal{C}$ et $Q \cap \mathcal{C} = \emptyset$. On obtient ainsi :

$$c\backslash((c\backslash\mathcal{C}) \cup G) \supseteq (Q \cup \mathcal{C})\backslash(Q \cup G).$$

Comme $Q \cap \mathcal{C} = \emptyset$ et $G \subseteq \mathcal{C}$, alors $Q \cap G = \emptyset$, et obtient par suite :

$$c\backslash((c\backslash\mathcal{C}) \cup G) \supseteq (Q \cup \mathcal{C})\backslash(Q \cup G) = ((Q \cup \mathcal{C})\backslash Q)\backslash G = \mathcal{C}\backslash G, \text{ ainsi } c\backslash((c\backslash\mathcal{C}) \cup G) \supseteq \mathcal{C}\backslash G.$$

Comme $g \subseteq ((c\backslash\mathcal{C}) \cup G)$, on a $c\backslash g \supseteq c\backslash((c\backslash\mathcal{C}) \cup G) \supseteq \mathcal{C}\backslash G$, i.e. $c\backslash g \supseteq \mathcal{C}\backslash G$. De $c\backslash g \supseteq \mathcal{C}\backslash G$ et $g \subset G$, on a $mgk(g \to c\backslash g) < mgk(G \to \mathcal{C}\backslash G)$ (i.e., $G \to \mathcal{C}\backslash G \prec g \to c\backslash g$), et conclut que $G \to \mathcal{C}\backslash G$ est redondante par rapport à $g \to c\backslash g$, car elle ne satisfait pas à la propriété (i) de la définition 14. □

Corollaire 5. *Soient $\mathcal{FC}$ l'ensemble des itemsets fermés, $\mathcal{G}$ l'ensemble des générateurs minimaux des itemsets fermés dans $\mathcal{FC}$. Soit $c \in \mathcal{FC}$ et $g \in \mathcal{G}$ tels que $\gamma(g) \subset c$. Si $g \to c\backslash g$ est redondante, alors $\forall \mathcal{C} \in \mathcal{FC}$, $\forall G \in \mathcal{G}$ tels que $G \subseteq g$ et $c\backslash g \subset \mathcal{C}\backslash G$, on a $(g \to c\backslash g) \prec (G \to \mathcal{C}\backslash G)$.*

Démonstration. Puisque $G \subseteq g$, alors $G \supset ((\mathcal{C}\backslash c) \cup g)$ n'est pas vrai, c'est-à-dire que :

(i) $G \subseteq (\mathcal{C}\backslash c) \cup g)$, ou (ii) $G \cap ((\mathcal{C}\backslash c) \cup g) = \emptyset$, ou (iii) $(G \cap (\mathcal{C}\backslash c) \cup g) \subset ((\mathcal{C}\backslash c) \cup g) \wedge (G \cap (\mathcal{C}\backslash c) \cup g) \subset G$.

Dans tous les cas, supposons que la règle approximative $g \to c\backslash g$ soit non-redondante.

(i) Si $G \subseteq (\mathcal{C}\backslash c) \cup g)$ est vrai et $g \to c\backslash g$ est supposée non-redondante, alors $\exists \mathcal{C} \in \mathcal{FC}, \exists G \in \mathcal{G}, \gamma(G) \subseteq \mathcal{C}$ (i.e. $G \subset \mathcal{C}$) tels que $G \supseteq g$ et $\mathcal{C}\backslash G \subset c\backslash g$. De $\mathcal{C}\backslash G \subset c\backslash g$ et $G \supseteq g$, on a $\mathcal{C}\backslash G \subset c\backslash g \subseteq c$. Comme $\gamma(G) \subseteq \mathcal{C}$ (i.e. $G \subset \mathcal{C}$), on a évidemment $(\mathcal{C}\backslash G) \cup G = \mathcal{C}$ et $(\mathcal{C}\backslash G) \cap G = \emptyset$, ainsi $\mathcal{C}\backslash(\mathcal{C}\backslash G) = G$. Puisque $\mathcal{C}\backslash G \subseteq c$, alors $\mathcal{C}\backslash c \subseteq G$. De $\mathcal{C}\backslash c \subseteq G$ et $G \supseteq g$, on a $(\mathcal{C}\backslash c) \cup g \subseteq G \cup g = G$, ainsi $(\mathcal{C}\backslash c) \cup g \subseteq G$, contradictoire à $G \subseteq (\mathcal{C}\backslash c) \cup g)$, et prouve que $g \to c\backslash g$ est redondante.

(ii) Si $G \cap (\mathcal{C}\backslash c) \cup g = \emptyset$ est vrai, alors $g \cap G = \emptyset$, donc $G \supseteq g$ est toujours faux. Ainsi, par la définition 14, la règle approximative $g \to c\backslash g$ est redondante par rapport à la règle $G \to \mathcal{C}\backslash G$.

(iii) Si $(G \cap (\mathcal{C}\backslash c) \cup g) \subset ((\mathcal{C}\backslash c) \cup g) \wedge (G \cap (\mathcal{C}\backslash c) \cup g) \subset G$ est vrai, alors il existe z tel que $z \in (\mathcal{C}\backslash c)$ et $z \notin g$ (ou $z \in G$ et $z \notin g$). Cela implique que $c\backslash g \supseteq \mathcal{C}\backslash G$ et $g \supseteq G$ sont tous faux. Donc, par la définition 14, la règle approximative $g \to c\backslash g$ est redondante par rapport à $G \to \mathcal{C}\backslash G$. □

Théorème 12. *Soit $\mathcal{RA}^+$ l'ensemble de règles positives approximatives. (i) Toute règle de $\mathcal{RA}^+$ peut être dérivée de $\mathcal{BA}^+$, et (ii) toute règle de $\mathcal{BA}^+$ est une règle non-redondante.*

Démonstration. La preuve de ce théorème découle du lemme 3 et du corollaire 5 ci-dessus. □

Corollaire 6. *Soit $\mathcal{RA}^+$ l'ensemble de règles d'association positives approximatives. Une base de règles positives approximatives $\mathcal{BA}^+$ est minimale.*

Démonstration. La preuve de ce corollaire 6 est analogue à celle du corollaire 4. □

La définition 18 ci-dessous définit une base de règles négatives exactes, que l'on note par $\mathcal{BE}^-$.

Définition 18. *Soient $\mathcal{RE}^-$ l'ensemble de toutes les règles négatives exactes, $\mathcal{FM}$ l'ensemble des maximaux fréquents, et minsup un support minimum tel que $0 < minsup < 1$. Pour chaque maximal h, $\mathcal{G}_h$ désigne l'ensemble de ses générateurs. La base $\mathcal{BE}^-$ de $\mathcal{RE}^-$ est définie par :*

$$\mathcal{BE}^- = \{r : G \rightarrow \overline{z} \mid G \in \mathcal{G}_h, h \in \mathcal{FM}, (\forall z \in \mathcal{I}\backslash\{h\} \text{ tel que } supp(z) < minsup), \\ \wedge\ (\nexists r' \in \mathcal{RE}^- \text{ tel que } r \prec r')\} \quad (2.13)$$

Cette base a deux avantages. Le premier repose sur la précision du conséquent $\overline{z}$. Ce dernier est sélectionné à partir d'un ensemble des motifs minimaux infréquents associés à des itemsets maximaux fréquents dans $\mathcal{D}$ (limite des approches correctives [FDT06, FDT07, Ram16]). Le second, grâce à la relation de couverture telle que données dans la définition 15, étant la possibilité d'élaguer les règles d'association non génératrices (deuxième limite de ces approches correctives). Cela garantit la minimalité de cette base $\mathcal{BE}^-$ telle que démontrée à l'aide du corollaire 7 ci-dessous.

Le lemme 4 ci-après est le premier pas vers les preuves des théorèmes 13 et 14 ci-dessous.

Lemme 4. *$\forall X, Y \subseteq \mathcal{I}$ t.q. $supp(X) \neq 0$ et $supp(Y) \neq 0$: $mgk(X, \overline{Y}) = 1 \Leftrightarrow supp(X \cup Y) = 0$.*

Idée de preuve. Puisque $mgk(X, \overline{Y}) = 1$, on a, grâce à la Proposition 6, $P(\overline{Y'}|X') = 1$ équivaut à $P(Y'|X') = 0$ équivaut à $\frac{supp(X \cup Y)}{supp(X)} = 0$ (pour $supp(X) \neq 0$), d'où $supp(X \cup Y) = 0$.

Théorème 13. *Soit $\mathcal{RE}^-$ l'ensemble de toutes les règles négatives exactes. (i) Toute règle de $\mathcal{RE}^-$ peut être dérivée de $\mathcal{BE}^-$, et (ii) toute règle de $\mathcal{BE}^-$ est une règle non-redondante.*

Démonstration. (*i*) Soit $r : X \rightarrow \overline{Y}\backslash X$ une règle de $\mathcal{RE}^-$. Par définition, il existe au moins une règle $r' : G \rightarrow \overline{y}\backslash G \in \mathcal{BE}^-$ qui couvre r. Montrons que r est dérivable de r'. En effet, puisque r est une règle de $\mathcal{RE}^-$, on a alors $mgk(r) = 1 \Leftrightarrow P(\overline{Y'}|X') = 1 \Leftrightarrow \frac{supp(X \cup \overline{Y})}{supp(\overline{Y})} = 1 \Rightarrow supp(X \cup \overline{Y}) = supp(\overline{Y})$. D'autre part, comme $mgk(r) = 1$, on a, par le lemme 4, $supp(X \cup Y) = 0 \Rightarrow supp(X \cup \overline{Y}) = supp(X)$. De $supp(X \cup \overline{Y}) = supp(X)$ et $supp(X \cup \overline{Y}) = supp(\overline{Y})$, on obtient $supp(X \cup \overline{Y}) = supp(X) = supp(\overline{Y})$. De façon analogue, nous avons pour r' : $supp(G \cup \overline{y}) = supp(G) = supp(\overline{y})$. Comme r_1 couvre r, on a : $G \subset X$ et $\overline{Y} \subset \overline{y}$ impliquent $\gamma(G) \subset \gamma(X)$ et $\gamma(\overline{Y}) \subset \gamma(\overline{y})$. D'autre part, $\mathcal{BE}^-$ est un sous-ensemble de $\mathcal{RE}^-$ (i.e. $\mathcal{BE}^- \subset \mathcal{RE}^-$), on a $\gamma(G) = \gamma(X)$ et $\gamma(\overline{Y}) = \gamma(\overline{y})$ équivaut à $supp(G) = supp(X)$ et $supp(\overline{Y}) = supp(\overline{y})$ impliquent $supp(G \cup \overline{y}) = supp(X \cup \overline{Y})$ (i.e. $supp(r') = supp(r)$) implique $mgk(r') = mgk(r)$, ce qui relève que la règle r est dérivable de la règle r'. Autrement dit, toutes règles de l'ensemble $\mathcal{RE}^-$ peuvent être dérivées de la base $\mathcal{BE}^-$.

(*ii*) Soit $r : g \rightarrow \overline{y}\backslash g$ une règle de $\mathcal{BE}^-$ telle que $g \subset \gamma(g)$ et $y \notin \gamma(g)$. Supposons que r soit redondante, i.e. $\forall G \subset \gamma(G)$, $\forall x \notin \gamma(G)$, $\exists r' : G \rightarrow \overline{x}\backslash G \in \mathcal{RE}^-$ telle que $g \supseteq G$ et $\overline{y}\backslash g \subseteq \overline{x}\backslash G$. Comme $g \supseteq G$, on obtient d'après le corollaire 5, $g \subseteq (\overline{y}\backslash\overline{x}) \cup G$. De $\overline{x}\backslash G \supseteq \overline{y}\backslash g$, on obtient $\overline{x} \supseteq \overline{x}\backslash G \supseteq \overline{y}\backslash g$, donc $\overline{x} \supseteq \overline{y}\backslash g$. Comme $g \subseteq \gamma(g)$ et $y \notin \gamma(g)$, on a évidemment $(\overline{y}\backslash g) \cup g = \overline{y}$ et $(\overline{y}\backslash g) \cap g = \emptyset$, donc $\overline{y}\backslash(\overline{y}\backslash g) = g$. Puisque $\overline{x} \supseteq \overline{y}\backslash g$, on obtient $\overline{y}\backslash\overline{x} \subseteq \overline{y}\backslash(\overline{y}\backslash g) = g$, c'est-à-dire $\overline{y}\backslash\overline{x} \subseteq g$. De $\overline{y}\backslash\overline{x} \subseteq g$ et $g \supseteq G$, on obtient $(\overline{y}\backslash\overline{x}) \cup G \subseteq g$, contradiction du fait que $(\overline{y}\backslash\overline{x}) \cup G \supseteq g$. Donc, la règle $g \rightarrow \overline{y}\backslash g$ est non redondante, autrement dit la règle r n'est couverte par aucune règle de $\mathcal{RE}^-$ (cf. définition 15). Ainsi, la base $\mathcal{BE}^-$ est une base non-redondante. □

Corollaire 7. *Soit $\mathcal{RE}^-$ l'ensemble de règles négatives exactes valides. La base $\mathcal{BE}^-$ est minimale.*

Démonstration. Soit $\mathcal{BE}^-$ une base de l'ensemble $\mathcal{RE}^-$. Supposons que $\mathcal{BE}^-$ n'est pas minimale, alors il existe $\widehat{\mathcal{BE}}^- \subset \mathcal{BE}^-$ tel que $\widehat{\mathcal{BE}}^-$ est une base de $\mathcal{RE}^-$. La suite de la preuve découle de la propriété (*ii*) du théorème 13 ci-dessus, et analogue à ce qu'on a fait avec le corollaire 4 ci-dessus. □

Enfin, la définition 19 définit la base de règles négatives approximatives, notée $\mathcal{BA}^-$.

Définition 19. *Soient $\mathcal{RA}^-$ l'ensemble des règles négatives approximatives, $\mathcal{G}$ l'ensemble des générateurs, et α un certain seuil critique tel que $0 \leq \alpha < 1$. La base $\mathcal{BA}^-$ est définie par :*

$$\mathcal{BA}^-(\alpha) = \{r : G \rightarrow \overline{g}\backslash G \mid (G,g) \in \mathcal{G}_{\gamma(G)} \times \mathcal{G}_{\gamma(g)}, \gamma(G) \subsetneq \gamma(g), P(g'|G') < P(g'), mgk(r) \geq 1-\alpha, \\ \wedge \ (\nexists r' \in \mathcal{RA}^- \text{ tel que } r \prec r')\} \quad (2.14)$$

La base $\mathcal{BA}^-$ a essentiellement triple avantages. Le premier étant la possibilité d'élaguer les règles inintéressantes telles que données dans la définition 8, grâce à la contrainte d'élagage $P(g')|G') < P(g')$ (i.e., $P(\overline{g'}|G') > P(\overline{g'})$) (limite des approches [GYNS06, HYN11, LHH12]). Le second, grâce à la contrainte d'élagage $mgk(G,\mathcal{C}) \geq 1-\alpha$ (cf. définition 11), est la possiblité d'élaguer les règles proches de l'indépendance qui peuvent être encaissées par les approches correctives [FDT06, FDT07, Ram16] d'une part, et de récupérer les règles fortes naïvement écartées par celles-ci d'autre part. Le troisième, grâce à la relation de couverture telle que données dans la définition 15, étant la possibilité d'élaguer les règles non génératrices, qui garantit la minimalité de la base $\mathcal{BA}^-$.

Théorème 14. *Soit $\mathcal{RA}^-$ l'ensemble des règles négatives approximatives. (i) Toute règle de $\mathcal{RA}^-$ peut être dérivée de $\mathcal{BA}^-$, et (ii) toute règle de $\mathcal{BA}^-$ est une règle non-redondante.*

Démonstration. La preuve de la partie (i) de ce théorème découle du lemme 3. Il reste à montrer la partie (ii). Soit $r : g \rightarrow \overline{y}\backslash g$ une règle de $\mathcal{BA}^-$, $\forall g \in \mathcal{G}_{\gamma(g)}$, $\forall y \in \mathcal{G}_{\gamma(y)}$ tels que $\gamma(g) \subsetneq \gamma(y)$ et $P(\overline{y'}|g') > P(\overline{y'})$. Supposons que r soit redondante, i.e. $\exists r' : G \rightarrow \overline{x}\backslash G \in \mathcal{RA}^-$, où $\forall G \in \mathcal{G}_{\gamma(G)}$, $\forall x \in \mathcal{G}_{\gamma(x)}$ tels que $\gamma(G) \subsetneq \gamma(x)$, $P(\overline{x'}|G') > P(\overline{x'})$, $g \supseteq G$ et $\overline{y}\backslash g \subseteq \overline{x}\backslash G$. Comme $g \supseteq G$, on obtient d'après le corollaire 5 $g \subseteq (\overline{y}\backslash\overline{x}) \cup G$. De $\overline{x}\backslash G \supseteq \overline{y}\backslash g$, on obtient $\overline{x} \supseteq \overline{x}\backslash G \supseteq \overline{y}\backslash g$, donc $\overline{x} \supseteq \overline{y}\backslash g$. Comme $g \rightarrow \overline{y}\backslash g \in \mathcal{BA}^-$ tel que $g \subseteq \gamma(g)$ (par définition), on obtient évidemment $((\overline{y}\backslash g) \cup g$ et $(\overline{y}\backslash g) \cap g = \emptyset)$, donc $\overline{y}\backslash(\overline{y}\backslash g) = g$. Puisque $\overline{x} \supseteq \overline{y}\backslash g$, on obtient alors $\overline{y}\backslash\overline{x} \subseteq \overline{y}\backslash(\overline{y}\backslash g) = g$, c'est-à-dire $\overline{y}\backslash\overline{x} \subseteq g$. De $\overline{y}\backslash\overline{x} \subseteq g$ et $g \supseteq G$, on obtient $(\overline{y}\backslash\overline{x}) \cup G \subseteq g$, ce qui contredit du fait que $(\overline{y}\backslash\overline{x}) \cup G \supseteq g$. Plus précisément, la règle $g \rightarrow \overline{y}\backslash g$ est une règle non redondante, car elle n'est couverte par aucune règle de la base $\mathcal{RA}^-$ (cf. définition 15). Donc, la base $\mathcal{BA}^-$ est une base non-redondante. □

Corollaire 8. *Soit $\mathcal{RA}^-$ l'ensemble de règles négatives approximatives. La base $\mathcal{BA}^-$ est minimale.*

Démonstration. Soit $\mathcal{BA}^-$ une base de $\mathcal{RA}^-$. Supposons que $\mathcal{BA}^-$ n'est pas minimale, alors il existe $\widetilde{\mathcal{BA}}^- \subset \mathcal{BA}^-$ tel que $\widetilde{\mathcal{BA}}^-$ est une base de $\mathcal{RA}^-$. La suite de la preuve découle de la propriété (ii) du théorème 14 ci-dessus, et analogue à celle du corollaire 4. □

2.4 Détails algorithmiques de la génération des règles valides

Comme signalé, la complexité de la génération des règles d'association étant exponentielle, et serait dramatique en présence de données volumineuses et/ou denses. Il est ainsi quasi-impossible d'analyser celles-ci sans l'aide des outils informatiques. Différentes techniques ont été proposées dont nous en citerons quelques-unes. Dans [GYNS06, HYN11, LHH12], des algorithmes pour la génération des bases des règles positives, basés sur le couple support/confiance, sont proposés. Dans [FDT06, FDT07, Ram16], les auteurs étendent ces approches précitées, où ils rédéfinissent des algorithmes pour les bases des règles positives, et ajoutent ceux pour les règles négatives à l'aide du couple support/M_{GK}. Cependant, comme discuté, ces diverses approches proposées présentent

des limites remarquables. Pour cela, nous avons proposé dans [BT20b, BT20d, BT21a, BT21b] des nouveaux algorithmes d'extraction des bases des règles (sous-section 2.4.1), et des nouveaux algorithmes pour les règles dérivées (sous-section 2.4.2), à l'aide d'un nouveau couple support/mgk.

2.4.1 Algorithme d'extraction des bases des règles valides

Dans [BT21a], nous avons proposé un algorithme, appelé CONCISE, qui améliore l'algorithme NONRED [BT20d]. En effet, l'algorithme CONCISE se divise essentiellement en deux étapes : (i) l'extraction des motifs fréquents (Algo. 1), et (ii) la génération des règles d'association positives et négatives non-redondantes (Algo. 4). En fait, l'étape (ii) s'effectue à l'aide d'une procédure principale, appelée CBNR (Concise base of non-redundant rules). CBNR (Algo. 4), qui prend

Algorithm 4 CBNR (Concise base of non-redundant rules)

Require: $\mathcal{FCMG} = \langle closed, maximal, generator, support \rangle$.
Ensure: Concise base of non-redundant rules.
1: $\mathcal{BE}^+(\mathcal{FCMG})$;
2: $\mathcal{BA}^+(\mathcal{FCMG})$;
3: $\mathcal{BE}^-(\mathcal{FCMG})$;
4: $\mathcal{BA}^-(\mathcal{FCMG})$;

en entrée l'ensemble $\mathcal{FCMG}$ composé de quatre champs $\langle closed, maximal, generator, support \rangle$ de l'algorithme CMG (Algo. 1), retourne en sortie les bases des règles non-redondantes en faisant appel à quatre procédures sécondaires $\mathcal{BE}^+$, $\mathcal{BA}^+$, $\mathcal{BE}^-$ et $\mathcal{BA}^-$. Ce choix de décomposition est motivé par la parallélisation de ces quatre procédures lors de l'implémentation pratique de celles-ci.

La procédure $\mathcal{BE}^+$ (Algorithme 5), qui prend en entrée l'ensemble $\mathcal{FCMG}$ des motifs (fermés, maximaux et générateurs) fréquents, retourne en sortie la base des règles d'association positives exactes. Elle est initialisée à vide (ligne 1), et examine ensuite chaque élément de $\mathcal{FCMG}$ dans

Algorithm 5 Procedure $\mathcal{BE}^+$

Require: $\mathcal{FCMG} = \langle closed, maximal, generator, support \rangle$.
Ensure: $\mathcal{BE}^+$, Concise base of exact positives rules.
1: $\mathcal{BE}^+ = \emptyset$;
2: **for** ($k = 1$ to ℓ, where ℓ is the size of largest frequent itemset in $\mathcal{FCMG}$) **do**
3: **for all** (k-maximal h of $\mathcal{FCMG}_k.maximal$) **do**
4: **for all** (k-generator $g \in \mathcal{FCMG}_k.generator$ of h) **do**
5: **if** ($g \neq h \ \wedge \ \nexists$ certain rule $(a \rightarrow b) \mid (g \rightarrow h) \prec (a \rightarrow b)$) **then**
6: $\mathcal{BE}^+ \leftarrow \mathcal{BE}^+ \cup \{g \rightarrow h \backslash g, g.supp\}$;
7: **end if**
8: **end for**
9: **end for**
10: **end for**

l'ordre croissant de k (lignes 2-10). Pour chaque k-générateur $g \in \mathcal{FCMG}$ du fermé h, la procédure $\mathcal{BE}^+$ vérifie si g n'est pas un unique élément dans sa classe d'équivalence (ligne 4). Si c'est le cas, la règle d'association $g \rightarrow h \backslash g$ est alors valide, et ajoutée dans la liste $\mathcal{BE}^+$ (ligne 5).

La procédure $\mathcal{BA}^+$ (algorithme 6), qui prend en entrée l'ensemble $\mathcal{FCMG}$, donne en sortie la base des règles d'association approximatives. Elle est initialisée à vide (ligne 1), et examine ensuite

Algorithm 6 Procedure $\mathcal{BA}^+$

Require: $\mathcal{FCMG} = \langle closed, maximal, generator, support \rangle$, a real $\alpha \in [0, 1[$.
Ensure: $\mathcal{BA}^+$, Concise base of approximate positive rules.
1: $\mathcal{BA}^+ = \emptyset$;
2: **for** ($k = 1$ to ℓ, where ℓ is the size of largest frequent itemsets in $\mathcal{FCMG}$) **do**
3: **for all** (k-generator $g \in \mathcal{FCMG}_k.generator$) **do**
4: **for all** (frequent closed itemset $c \in \mathcal{FCMG}_{j>k} \mid c \supset \gamma(g)$) **do**
5: **if** $(mgk(g \rightarrow c \backslash g) \geq 1 - \alpha \ \wedge \ \nexists (G \rightarrow \mathcal{C} \backslash G) \mid (g \rightarrow c \backslash g) \prec (G \rightarrow \mathcal{C} \backslash G))$ **then**
6: $\mathcal{BA}^+ \leftarrow \mathcal{BA}^+ \cup \{r : g \rightarrow c \backslash g, r.mgk, r.supp\}$;
7: **end if**
8: **end for**
9: **end for**
10: **end for**
11: **return** $\mathcal{BA}^+$

l'ensemble $\mathcal{FCMG}$ selon l'ordre croissant de k (lignes 2-10), où k est la longueur d'un itemset fréquent. Pour chaque k-générateur $g \in \mathcal{FCMG}_k.generator$, la procédure $\mathcal{BA}^+$ considère un fermé c contenant le fermé $\gamma(g)$ (lignes 4-9). Ensuite, elle vérifie si la règle approximative $g \rightarrow c$ est valide (ligne 5). Si c'est le cas, une telle règle $g \rightarrow c \backslash g$ est alors ajoutée dans la liste $\mathcal{BA}^+$ (ligne 6).

La procédure $\mathcal{BE}^-$ est résumée dans l'algorithme 7. Elle prend en entrée l'ensemble $\mathcal{FCMG}$ des motifs fréquents, de 4 champs tels que closed, maximal, generator et support. Elle retourne en sortie la base des règles d'association négatives exactes. Elle est initialisée à vide (ligne 1), et

Algorithm 7 Procedure $\mathcal{BE}^-$

Require: $\mathcal{FCMG} = \langle closed, maximal, generator, support \rangle$, $minsup$.
Ensure: $\mathcal{BE}^-$, A concise base of exact negative rules.
1: $\mathcal{BE}^- = \emptyset$;
2: **for** ($k = 1$ to ℓ, where ℓ is the size of largest frequent itemsets in $\mathcal{FCMG}$) **do**
3: **for** (each k-maximal $h \in \mathcal{FCMG}.maximal$) **do**
4: **for** (each k-generator $g \in \mathcal{FCMG}.generator$ of h) **do**
5: **if** $(\exists z \in \mathcal{I} \backslash \{h\} \mid supp(z) < minsup \ \wedge \nexists (a \rightarrow \overline{b})$ tq. $(g \rightarrow \overline{z}) \prec (a \rightarrow \overline{b}))$ **then**
6: $\mathcal{BE}^- \leftarrow \mathcal{BE}^- \cup \{r : g \rightarrow \overline{z} \backslash g, r.supp\}$;
7: **end if**
8: **end for**
9: **end for**
10: **end for**

examine ensuite l'ensemble $\mathcal{FCMG}$ suivant l'ordre croissant de k (lignes 2-8), où k est la nongueur d'un itemset fréquent. Elle vérifie, pour chaque k-générateur $g \in \mathcal{FCMG}_k.generator$ d'un maximal $h \in \mathcal{FCMG}_k.maximal$, s'il existe un motif minimal infréquent, dual du maximal h. Si c'est le cas, la règle $g \rightarrow \overline{z} \backslash g$ est alors une règle négative exacte, et ajoutée dans l'ensemble $\mathcal{BE}^-$ (ligne 5).

L'algorithme 8 ci-après résume la procédure $\mathcal{BA}^-$. Il prend en entrée l'ensemble $\mathcal{FCMG}$ des

motifs fréquents et un seuil critique $\alpha \in [0, 1[$, et retourne en sortie la base $\mathcal{BA}^-$. Il est initialisé à vide

Algorithm 8 Procedure $\mathcal{BA}^-$

Require: $\mathcal{FCMG} = \langle closed, maximal, generator, support \rangle$, and a real $\alpha \in [0, 1[$.
Ensure: $\mathcal{BA}^-$, Concise base of approximate negative rules.
1: $\mathcal{BA}^- = \emptyset$;
2: **for** ($k = 1$ to ℓ, where ℓ is the size of largest frequent itemset in $\mathcal{FCMG}$) **do**
3: **for** (each k-generator $G \in \mathcal{FCMG}_k.generator$) **do**
4: **for** (each k-generator $g \in \mathcal{FCMG}_{j \neq k}.generator \mid \gamma(G) \subsetneq \gamma(g) \ \wedge \ P(\gamma(g)'|\gamma(G)') < P(\gamma(g)')$) **do**
5: **if** ($mgk(G, \overline{g}) \geqslant 1-\alpha, \nexists (Z \to \overline{z} \backslash Z) \mid (G \to \overline{g} \backslash G) \prec (Z \to \overline{z} \backslash Z)$, où $\forall (Z, z) \in \mathcal{G}_{\gamma(Z)} \times \mathcal{G}_{\gamma(z)}, \gamma(Z) \subsetneq \gamma(z)$, et $P(\gamma(z)'|\gamma(Z)') < P(\gamma(z)')$) **then**
6: $\mathcal{BA}^- \leftarrow \mathcal{BA}^- \cup \{G \to \overline{g} \backslash G\}$;
7: **end if**
8: **end for**
9: **end for**
10: **end for**

(ligne 1), et examine ensuite l'ensemble $\mathcal{FCMG}$ suivant l'ordre croissant de k (lignes 2-10). Il vérifie simultanément, pour deux k-générateurs $G \in \mathcal{FCMG}_k.generator$ et $g \in \mathcal{FCMG}_{j \neq k}.generator$ tels que $\gamma(G) \subsetneq \gamma(g)$ et $P(\gamma(g)'|\gamma(G)') < P(\gamma(g)')$, si la règle $G \to \overline{g}$ est valide, et n'existe pas $Z \to \overline{z} \backslash Z$ qui couvre $G \to \overline{g} \backslash G$, où $(Z, z) \in \mathcal{G}_{\gamma(Z)} \times \mathcal{G}_{\gamma(z)}$, $\gamma(Z) \subsetneq \gamma(z)$ et $P(\gamma(z)'|\gamma(Z)') < P(\gamma(z)')$ (ligne 5). Si c'est le cas, ladite $G \to \overline{g} \backslash G$ est alors valide, et ajoutée dans $\mathcal{BE}^-$ (ligne 6).

2.4.2 Algorithmes de génération des règles dérivées

Dans cette sous-section, nous allons présenter nos algorithmes permettant de reconstruire toutes les règles d'association positives et négatives exactes et approximatives qui ne sont pas présents dans les approches variantes/correctives [FDT06, FDT07, Ram16]. Leur génération découle surtout du théorème 10 que nous avons développé dans [BT21a] et présenté dans le chapitre 2.

L'algorithme 9, qui prend en entrée la base $\mathcal{BE}^+$, retourne toutes les règles positives et négatives exactes. Il est initialisé à vide en ligne 1, et considère la règle $r_1 : X \to Y$, avec $|Y| > 1$, suivant

Algorithm 9 Deriving All Exact Positive and Negative Rules

Require: $\mathcal{BE}^+$.
Ensure: $\mathcal{RE}^{+-}$, *All exact positives and negatives rules.*
1: $\mathcal{RE}^{+-} = \emptyset$;
2: **for all** ($\{r_1 : X \to Y \backslash X, r_1.mgk\} \in \mathcal{BE}^+ \mid X \in \mathcal{G}_Y \wedge |Y| > 1$) **do**
3: **if** ($supp(\overline{X} \cup \overline{Y}) \geq minsup$) **then**
4: $\mathcal{RE}^{+-} \leftarrow \mathcal{RE}^{+-} \cup \{r_2 : \overline{X} \to \overline{Y} \backslash \overline{X}, r_1.mgk\}$;
5: **end if**
6: **for** (each other generator Z of $\mathcal{G}_Y$) **do**
7: $\mathcal{RE}^{+-} \leftarrow \mathcal{RE}^{+-} \cup \{r_3 : Z \to Y \backslash Z, r_4 : \overline{Z} \to \overline{Y} \backslash \overline{Z}, r_1.mgk\}$;
8: **end for**
9: **end for**

l'ordre croissant de leur conséquent (lignes 2-9). Il vérifie si l'itemset $\overline{X} \cup \overline{Y}$ est fréquent (ligne 3). Si c'est le cas, alors la règle $\overline{X} \to \overline{Y} \backslash \overline{X}$ est valide, et ajoutée dans l'ensemble $\mathcal{RE}^{+-}$ (ligne 4). Cette

règle a les mêmes support et mgk que la règle r_1. Pour chaque générateur $Z \in \mathcal{G}_Y$, l'algorithme 9 génère les règles du type $Z \rightarrow Y \backslash Z$ et $\overline{Z} \rightarrow \overline{Y} \backslash \overline{Z}$ qui ont les mêmes support et mgk que r_1.

L'algorithme 10, qui prend en entrée la base $\mathcal{BA}^+$ des règles positives approximatives, retourne l'ensemble $\mathcal{RA}^+$ des règles d'association positives approximatives. Il est initialisé à $\mathcal{BA}^+$ en ligne 1.

Algorithm 10 Deriving All Approximate Positive Rules

Require: $\mathcal{BA}^+$.
Ensure: $\mathcal{RA}^+$, *All approximate positive rules.*
1: $\mathcal{RA}^+ = \mathcal{BA}^+$;
2: **for** ($k = 1$ to ℓ, where ℓ is the size of largest frequent itemsets) **do**
3: **for all** ($\{r_1 : X \rightarrow Y \backslash X, r_1.supp, r_1.mgk\} \in \mathcal{BA}^+ \mid X \in \mathcal{G}_{\gamma(X)}, \gamma(X) \subset Y$ et $|Y| > k$) **do**
4: **if** ($supp(\overline{X} \cup \overline{Y}) \geq minsup$) **then**
5: $\mathcal{RA}^+ \leftarrow \mathcal{RA}^+ \cup \{r_2 : \overline{X} \rightarrow \overline{Y} \backslash \overline{X}, r_2.supp, r_2.mgk\}$;
6: **end if**
7: **for** (each other generator $Z \in \mathcal{G}_{\gamma(X)}$) **do**
8: $\mathcal{RA}^+ \leftarrow \mathcal{RA}^+ \cup \{r_3 : Z \rightarrow Y \backslash Z, r_1.supp, r_1.mgk\}$;
9: **if** ($supp(\overline{Z} \cup \overline{Y}) \geq minsup$) **then**
10: $\mathcal{RA}^+ \leftarrow \mathcal{RA}^+ \cup \{r_4 : \overline{Z} \rightarrow \overline{Y} \backslash \overline{Z}, r_2.supp, r_2.mgk\}$;
11: **end if**
12: **end for**
13: **end for**
14: **end for**

Pour chaque règle $X \rightarrow Y$ telle que $|Y| > 1$, l'algorithme 10 génère tout d'abord les règles négatives de la forme $\overline{X} \rightarrow \overline{Y} \backslash \overline{X}$ (lignes 4-6). Pour chaque autre générateur Z de même classe d'équivalence que X, l'algorithme 10 génère aussi les règles des types $Z \rightarrow Y \backslash Z$ et $\overline{Z} \rightarrow \overline{Y} \backslash \overline{Z}$ (lignes 7-12).

L'algorithme 11, qui prend en entrée la base $\mathcal{BE}^-$ des règles négatives exactes, retourne en sortie l'ensemble $\mathcal{RE}^-$ de toutes les règles négatives exactes. Il est initialisé à vide en ligne 1, et considère

Algorithm 11 Deriving All Exact Negative Rules

Require: $\mathcal{BE}^-$.
Ensure: $\mathcal{RE}^-$, *All exact negative rules.*
1: $\mathcal{RE}^- = \emptyset$;
2: **for** ($i = 1$ to ℓ, where ℓ is the size of largest frequent itemsets) **do**
3: **for all** ($\{r_1 : X \rightarrow \overline{y} \backslash X, r_1.mgk\} \in \mathcal{BE}^- \mid X \in \mathcal{G}_{\gamma(X)} \wedge (y \notin \gamma(X), y \notin \mathcal{F}, |\overline{y}| > i)$) **do**
4: **if** ($supp(\overline{X} \cup y) \geq minsup$) **then**
5: $\mathcal{RE}^- \leftarrow \mathcal{RE}^- \cup \{r_2 : \overline{X} \rightarrow y \backslash \overline{X}, r_1.mgk\}$;
6: **end if**
7: **for** (each other generator $Z \in \mathcal{G}_{\gamma(X)}$) **do**
8: $\mathcal{RA}^+ \leftarrow \mathcal{RA}^+ \cup \{r_3 : Z \rightarrow \overline{y} \backslash Z \wedge r_4 : \overline{Z} \rightarrow y \backslash \overline{Z}, r_1.mgk\}$;
9: **end for**
10: **end for**
11: **end for**

ensuite toutes les règles $X \rightarrow \overline{y}$, avec $|\overline{y}| > 1$, suivant l'ordre croissant de la taille de leur conséquent

(lignes 2-11). Pour chaque règle $X \rightarrow Y$ telle que $|Y| > 1$, l'algorithme 11 génère tout d'abord les règles de la forme $\overline{X} \rightarrow \overline{y} \backslash \overline{X}$ (lignes 4-6). Puis, pour chaque autre générateur Z de même classe d'équivalence que X, l'algorithme 10 génère aussi les règles d'association des types $Z \rightarrow \overline{y} \backslash Z$ et $\overline{Z} \rightarrow y \backslash \overline{Z}$ (lignes 7-9). Ces règles d'association ont le même mgk que la règle r_1.

L'algorithme 12, qui prend en entrée la base $\mathcal{BA}^-$, retourne l'ensemble $\mathcal{RA}^-$ de toutes les règles négatives approximatives. Il est initialisé à vide en ligne 1. Pour toute règle du type $G \rightarrow \overline{g}$ de $\mathcal{BA}^-$

Algorithm 12 Deriving All Approximate Negative Rules

Require: $\mathcal{BA}^-$.
Ensure: $\mathcal{RA}^-$, *All approximate negative rules.*
1: $\mathcal{RA}^- = \emptyset$;
2: **for** ($i = 1$ to $s - 1$, where s is the size of largest frequent generator itemset) **do**
3: **for all** $(\{G \rightarrow \overline{g}\} \in \mathcal{BA}^- \mid |g| > i \wedge ((G, g) \in \mathcal{G}_{\gamma(G)} \times \mathcal{G}_{\gamma(g)}$ et $\gamma(G) \subsetneq \gamma(g)))$ **do**
4: **if** $(supp(\overline{\gamma(G)} \cup \gamma(g)) \geq minsup)$ **then**
5: $\mathcal{RA}^- \leftarrow \mathcal{RA}^- \cup \{\overline{\gamma(G)} \rightarrow \gamma(g) \backslash \overline{\gamma(G)}\}$;
6: **for** (each other generator $h \in \mathcal{G}_{\gamma(g)}$) **do**
7: $\mathcal{RA}^- \leftarrow \mathcal{RA}^- \cup \{G \rightarrow \overline{h} \backslash G, \overline{\gamma(G)} \rightarrow \gamma(h) \backslash \overline{\gamma(G)}\}$;
8: **end for**
9: **end if**
10: **for** (each other generator $Z \in \mathcal{G}_{\gamma(G)}$) **do**
11: $\mathcal{RA}^- \leftarrow \mathcal{RA}^- \cup \{Z \rightarrow \overline{g} \backslash Z\}$;
12: **if** $(supp(\overline{\gamma(Z)} \cup \gamma(g)) \geq minsup)$ **then**
13: $\mathcal{RA}^- \leftarrow \mathcal{RA}^- \cup \{\overline{\gamma(Z)} \rightarrow \gamma(g) \backslash \overline{\gamma(Z)}\}$;
14: **end if**
15: **end for**
16: **end for**
17: **end for**

telle que $|\overline{g}| > 1$, l'algorithme 12 génère, après avoir vérifié si l'itemset $\overline{\gamma(G)} \cup \gamma(g)$ est fréquent, les règles dérivées des types $\overline{\gamma(G)} \rightarrow \gamma(g) \backslash \overline{\gamma(G)}$, et $G \rightarrow \overline{h} \backslash G$ ainsi que $\overline{\gamma(G)} \rightarrow \gamma(h) \backslash \overline{\gamma(G)}$ (lignes 5-9), où h est un autre générateur d'une même classe d'équivalence que g (i.e., $h \in \mathcal{G}_{\gamma(g)}$). Puis, il génère, pour chaque autre générateur Z de $\mathcal{G}_{\gamma(G)}$, les règles des types $Z \rightarrow \overline{g} \backslash Z$ et $\overline{\gamma(Z)} \rightarrow \gamma(g) \backslash \overline{\gamma(Z)}$.

Théorème 15. *Soient $X \rightarrow Y$ et $X \rightarrow \overline{Y}$ deux principales règles générées, I un m-itemset fréquent dans $\mathcal{D}$, et l la longueur des prémisses d'une règle telle que $1 \leq l < m$. La complexité de la procédure principale (Algorithme 4) est, au pire des cas, $\mathcal{O}\left(\max\left((2^l 3^m - 3^m - 2^l - m), (3^m - 2m)\right) |\mathcal{FCMG}|^3\right)$.*

Démonstration. Les lignes 1 et 2 génèrent les règles positives du type $X \rightarrow Y \backslash X$. Par souci d'élagage des redondances, nous adoptons l'astuce suivante. Considérons une certaine règle $Z \rightarrow T \backslash Z$. Supposons que $X \rightarrow Y \backslash X$ soit non-redondante par rapport à $Z \rightarrow T \backslash Z$. Nous envisageons alors deux cas : $X \subseteq Z$ et $T \backslash Z \subset Y \backslash X$ (i.e. $|X| \leq |Z|$, $|T \backslash Z| < |Y \backslash X| \leq |I|$). Si $|X| < |Z|$, alors X, pour tout $k < l$, peut être sélectionné en $\binom{l}{k}$. Similairement, si $|Y \backslash X| < |I|$, alors Y peut être sélectionné en $\binom{m}{l}$. Du coup, le nombre de règles possibles telles que $|X| < |Z|$ et $|Y \backslash X| < |I|$ est exprimé :

$$\sum_{k=1}^{l-1} \binom{l}{k} \times \sum_{l=0}^{m-1} \binom{m}{l} = (2^l - 2)(2^m - 1) = 2^{m+l} - 2^{m+1} - 2^l + 2.$$

Par ailleurs, pour $|X| = |Z|$ et $|Y| = |I|$, on a $2^m - 2$ règles. Au total, on a :

$$2^{m+l} - 2^{m+1} - 2^l + 2 + 2^m - 2 = 2^{m+l} - 2^m - 2^l$$

Notant $|X \cup Y| = i$, alors $X \cup Y$ peut être sélectionné en $\binom{m}{i}$, et on a :

$$\begin{aligned}
m\text{-itemset} &\Rightarrow \binom{m}{m}(2^{m+l} - 2^m - 2^l) \\
(m-1)\text{-itemset} &\Rightarrow \binom{m}{m-1}(2^{(m-1)+l} - 2^{m-1} - 2^l) \\
&\dots \\
(2)\text{-itemset} &\Rightarrow \binom{m}{2}(2^{2+l} - 2^2 - 2^l)
\end{aligned}$$

En faisant la somme, on obtient :

$$\begin{aligned}
\sum_{i=2}^{m}\binom{m}{i}(2^{i+l} - 2^i - 2^l) &= 2^l\sum_{i=2}^{m}\binom{m}{i}2^i - \sum_{i=2}^{m}\binom{m}{i} - 2^l\sum_{i=2}^{m}\binom{m}{i} \\
&= 2^l\left[\sum_{i=0}^{m}\binom{m}{i}2^i - 1 - 2m\right] - \left[\sum_{i=0}^{m}\binom{m}{i}2^i - 1 - 2m\right] - 2^l\left[\sum_{i=0}^{m}\binom{m}{i} - 1 - m\right] \\
&= (2^l - 1)\left[\sum_{i=0}^{m}\binom{m}{i}2^i - 1 - 2m\right] - 2^l\left[\sum_{i=0}^{m}\binom{m}{i} - 1 - m\right]
\end{aligned}$$

D'après la formule binomiale, $\sum_{i=0}^{m}\binom{m}{i}x^i = (1+x)^m$, on obtient :

$$\begin{aligned}
\sum_{i=2}^{m}\binom{m}{i}(2^{i+l} - 2^i - 2^l) &= (2^l - 1)(3^m - 1 - 2m) - 2^l(2^m - 1 - m) \\
&= 2^l 3^m - 3^m - 2^l - m
\end{aligned}$$

Ainsi, au pire des cas, le coût des lignes 1 et 2 (i.e. Algo. 5 lignes 3-8 et Algo. 6 lignes 3-9) est :

$$\mathcal{O}\left(|\mathcal{FCMG}|^3(2^l 3^m - 3^m - 2^l - m)\right) \tag{2.15}$$

En fait, cette complexité (Equation (2.15)) vient de l'étude de la boucle **for** imbriquée des procédures respectives $\mathcal{BE}^+$ (Algorithme 5, lignes 2-8) et $\mathcal{BA}^+$ (Algorithme 6, lignes 2-9), qui parcourent pour chacun trois fois l'ensemble $\mathcal{FCMG}$, qui s'effectuent alors en $\mathcal{O}(|\mathcal{FCMG}|^3)$.

Les lignes 3 et 4, qui génèrent quant à elles les règles du type $X \rightarrow \overline{Y}$, sont calculées comme

suit. Soit $|X \cup \overline{Y}| = i$, alors $X \cup \overline{Y}$ peut être sélectionné en $\binom{m}{i}$, et on a :

$$\begin{aligned} m\text{-itemset} &\Rightarrow \binom{m}{m} 2^{m+1} \\ (m-1)\text{-itemset} &\Rightarrow \binom{m}{m-1} 2^{m} \\ &\dots \\ (2)\text{-itemset} &\Rightarrow \binom{m}{2} 2^{2+1} \end{aligned}$$

En faisant la somme, on obtient :

$$\begin{aligned} \sum_{i=2}^{m} \binom{m}{i} (2^{i+1}) &= 2 \sum_{i=2}^{m} \binom{m}{i} 2^{i} \\ &= 2 \left[\sum_{i=0}^{m} \binom{m}{i} 2^{i} - (1+2m) \right] \\ &= 2(3^m - 2m - 1) \end{aligned}$$

Ainsi, au pire des cas, les lignes 3 et 4 (Algo. 11 lignes 2-10 et Algo. 12 lignes 2-10) donnent :

$$\mathcal{O}(|\mathcal{FCMG}|^3(3^m - 2m)) \tag{2.16}$$

Comme précédent, cette complexité s'obtient en étudiant la boucle **for** imbriquée des procédures $\mathcal{BE}^-$ (Algo. 11, lignes 2-10) et $\mathcal{BA}^+$ (Algo. 12, lignes 2-10), qui parcourent trois fois à l'ensemble $\mathcal{FCMG}$ des motifs fréquents, et s'effectuent en $\mathcal{O}(|\mathcal{FCMG}|^3)$ dans le pire de cas.

Au final, la complexité au pire des cas de l'algo. 4 (i.e. (2.15)+(2.16)), $\forall 1 \leq l < m$, est en

$$\begin{aligned} &\mathcal{O}\left(|\mathcal{FCMG}|^3(2^l 3^m - 3^m - 2^l - m)\right) + \mathcal{O}\left(|\mathcal{FCMG}|^3(3^m - 2m)\right) \\ &\qquad = \mathcal{O}\left(\max\left((2^l 3^m - 3^m - 2^l - m), (3^m - 2m)\right) |\mathcal{FCMG}|^3\right). \end{aligned}$$

□

2.5 Evaluation expérimentale

Dans cette section, nous évaluons notre algorithme CONCISE [BT21a] comparé à deux approches sémantiquement proches, à savoir l'algorithme PRINCE [HYN11] et approche de Ramanantsoa [Ram16], mené sur quelques bases de données de FIMI [1]. Nous commençons par décrire le protocole expérimental (sous-sec. 2.5.1). Puis, nous discutons des résultats obtenus (sous-sec. 2.5.2).

1. http ://fimi.ua.ac.be/data/

2.5.1 Protocole expérimental

L'approche a été implémentée sous R. Tous les tests ont été effectués sur un PC Core i3-2350M avec 2,3 GHz (4Go RAM). Nous avons sélectionné quelques données bien connues de la littérature que le tableau 2.2 ci-après l'expose. Précisément, ce tableau 2.2 résume, pour chaque base de

Table 2.2 – Caractéristiques des bases d'expérimentations

Database	$\lvert\mathcal{T}\rvert$	$\lvert\mathcal{I}\rvert$	$\lvert\widehat{\mathcal{T}}\rvert$	ρ	Type de données	Taille
Chess	3 196	75	37	49%	game steps	239 700
Connect	67 557	129	43	33%	game steps	8 714 853
T40I10D100K	100 000	1 000	40	4%	synthetic dataset	100 000 000
Pumsb	49 046	7 117	74	1%	census data	349 060 382

données, le nombre de transactions $|\mathcal{T}|$, le nombre de motifs $|\mathcal{I}|$, la taille moyenne de transactions $|\widehat{\mathcal{T}}|$, la densité $\rho = |\widehat{\mathcal{T}}|/|\mathcal{I}|$ de données, et la taille de données (i.e., $|\mathcal{T}| \times |\mathcal{I}|$). Le choix des bases est motivé par la variété de leur nombre de transactions, nombre de motifs et la densité. Certaines bases comme `Chess` et `Connect` sont très denses de densités respectives 49% et 33%, et d'autres bases comme `T40I10D100K` et `Pumsb` sont creuses (voire très creuses) de densités respectives 4% et 1%. Pour chaque approche, notons par E^+ (resp. E^-) la base des règles d'association positives exactes (resp. négatives exactes) extraite, alors que A^+ (resp. A^-) celle des règles positives approximatives (resp. négatives approximatives). Par suite, nous désignons par $|Algo(\mathcal{B})|$ la cardinalité de la base $\mathcal{B}$ pour l'algorithme Algo. Notons aussi par *confidence* le seuil commun pour chaque algorithme. Autrement dit, $confidence = 95\%$ correspond respectivement au risque $\alpha = 5\%$ pour CONCISE et l'approche de Ramanantsoa [Ram16], et $minconf = 95\%$ pour l'algorithme PRINCE [HYN11].

2.5.2 Résultats et discussions

Nous analysons les résultats obtenus tant en termes de cardinalités des bases des règles d'association extraites pour chacun des algorithmes que de temps de calcul associés.

Nous nous intéressons tout d'abord à l'étude de l'évolution de la cardinalité des bases des règles d'association positives exactes/approximatives pour chaque algorithme, et chaque base de données. Nous y avons fixé le seuil de support $minsup = 1\%$. La figure 2.1 ci-après reporte les résultats suite à la variation de confiance. L'analyse de cette figure 2.1 nous permet de relever ce qui suit. Une première remarque est qu'aucune règle d'association positive exacte n'est extraite dans la base de données T40I10D100K pour chacun des algorithmes. Cela est probablement dû au fait que les itemsets fermés fréquents et les générateurs minimaux de telle base de données sont confondus. Nous remarquons aussi que CONCISE et PRINCE produisent le même nombre des règles d'association exactes pour tous les cas. Ceci s'explique du fait que les mesures *confiance* et *mgk* sont équivalentes en terme de génération des règles d'association exactes. Et, on s'aperçoit que ce nombre des règles d'association exactes est, pour eux, très réduit par rapport à celui de Ramanantsoa. L'explication vient du fait que CONCISE et PRINCE ne sélectionnent que des règles génératrices. Cette différence des cardinalités atteint sa valeur maximale pour `Connect` à une $confiance = 95\%$. En effet, CONCISE et PRINCE restituent environ, pour chacun, 2500 règles exactes, alors que Ramanantsoa fait environ 52000. Autrement dit, CONCISE et PRINCE, versus Ramanantsoa, permettent pour chacun d'éliminer 95,2% des redondances dans cette base `Connect`.

Pour les bases des règles approximatives, CONCISE en offre de manière générale un nombre très

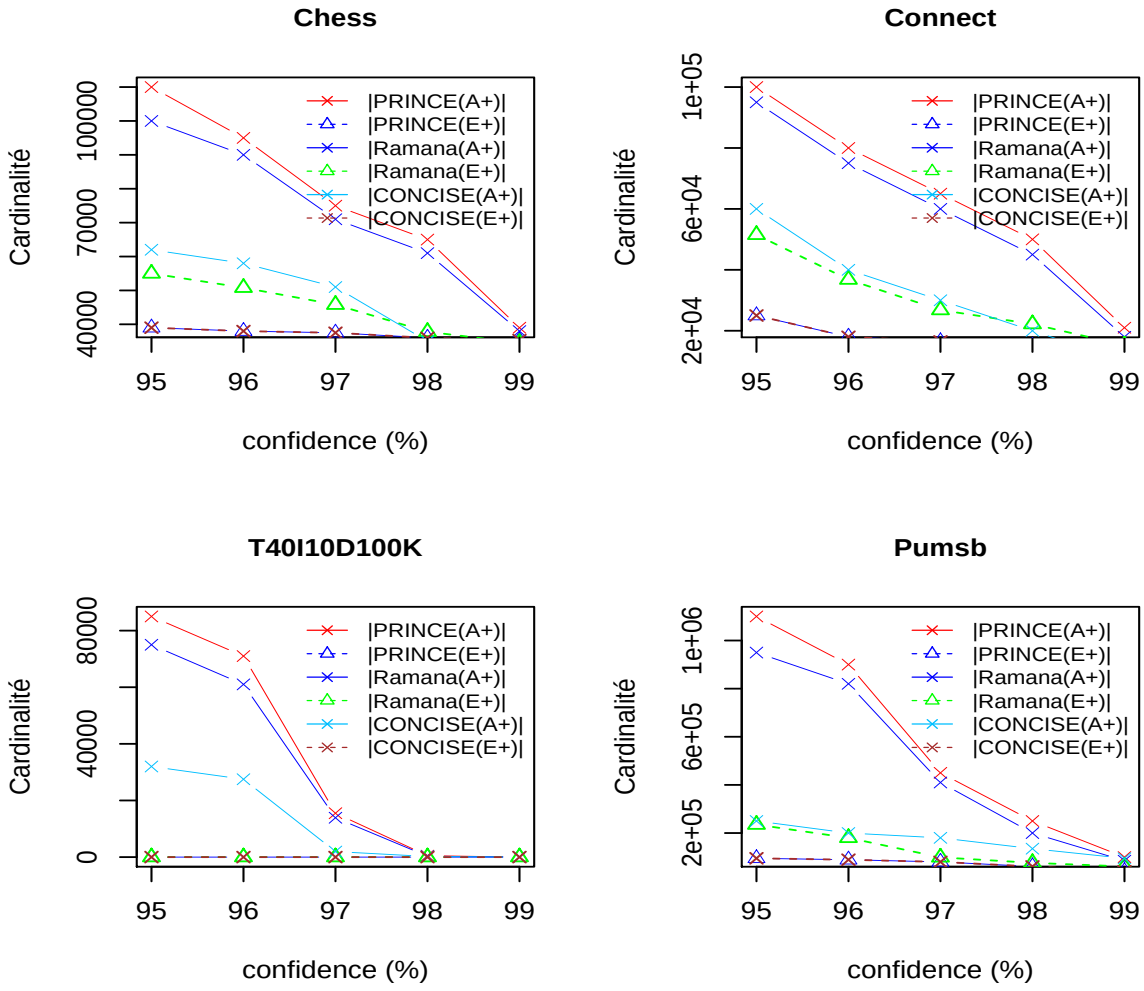

Figure 2.1 – Figure comparative de cardinalités des bases des règles positives exactes et approximatives

inférieur par rapport aux PRINCE et Ramanantsoa. Ceci n'est pas étonnant et s'explique du fait que CONCISE utilise une méthode d'élagage des règles inintéressantes (cf. définition 8), des règles proches de l'indépendance, et des règles non-génératrices, qui augmentent la cardinalité des règles extraites par l'approche de Ramanantsoa. Cet écart est très remarquable pour le jeu de données `Pumsb` avec une $confiance = 95\%$. En effet, CONCISE extrait environ 25000 règles approximatives, alors que PRINCE (resp. Ramanantsoa) fait environ 1500000 (resp. 950000) règles. Autrement dit, CONCISE permet d'éliminer 98,33% des règles inintéressantes versus PRINCE (resp. 97,37% des règles proches de l'indépendance et redondantes versus Ramanantsoa) dans la donnée `Pumsb`.

Au niveau des bases des règles négatives, notre étude a été effectuée sur les mêmes contraintes que l'étude des règles positives ci-dessus. Comme les règles négatives sont absentes dans l'algorithme PRINCE, nous limitons alors notre étude autour de l'algorithme CONCISE et l'approche de Ramanantsoa seulement. Quelles que soient les bases de données et les valeurs de confiance,

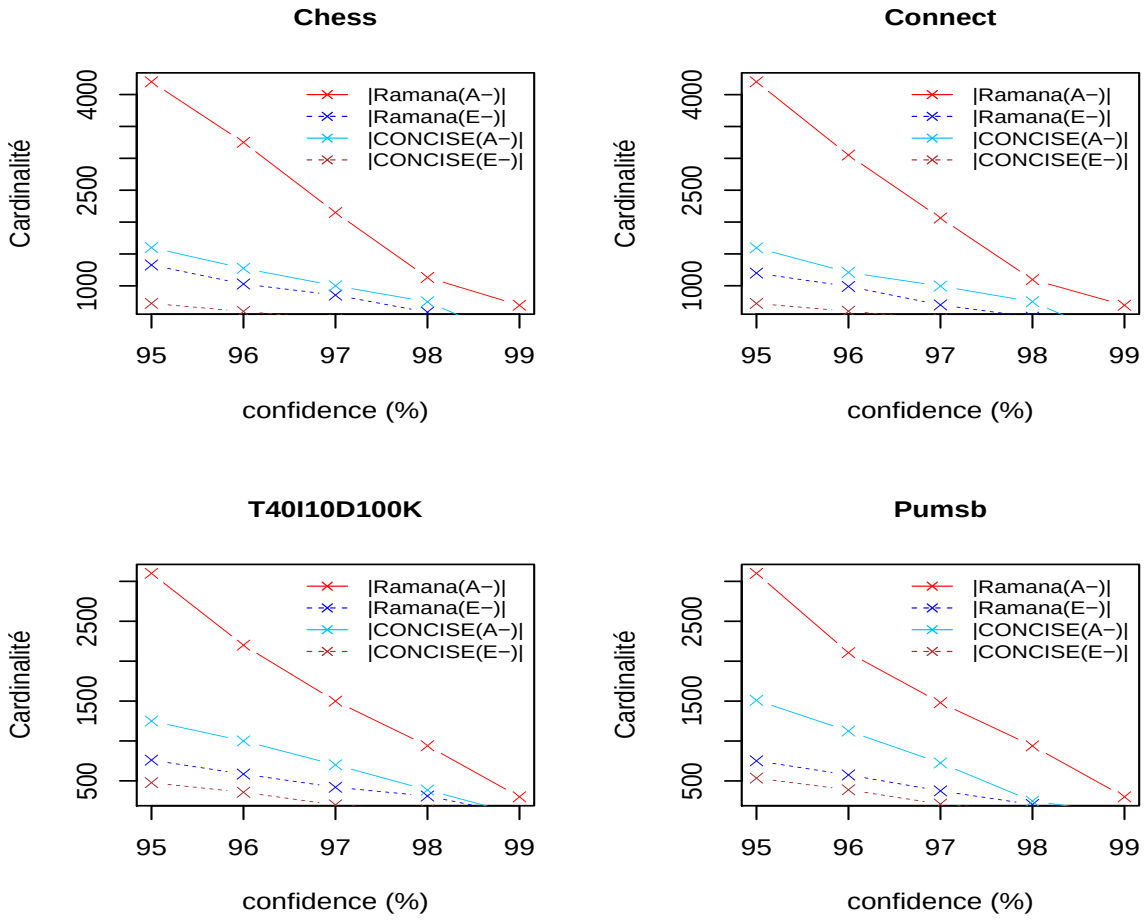

Figure 2.2 – Figure comparative des cardinalités des bases des règles négatives exactes et approximatives

CONCISE reste encore meilleur par rapport à l'approche de Ramanantsoa. La raison principale est toujours liée au fait que CONCISE ne sélectionne que des règles génératrices (cf. définition 16), afin d'éviter les redondances. Ce n'est pas le cas pour l'approche de Ramanantsoa. De plus, CONCISE utilise une technique efficace permettant d'éliminer les règles proches de l'indépendance (cf. définition 11), alors que l'approche de Ramanantsoa, quant à elle, utilise la valeur critique (Eq. (2.4)), qui n'est pas très sélective, car elle accepte très souvent des règles proches de l'indépendance (i.e., inintéressantes). Prenons par exemple la base `Chess`, avec $confiance = 95\%$, pour CONCISE il y a environ 1600 règles négatives approximatives (resp. 700 exactes), là où Ramanantsoa fait environ 4500 négatives approximatives (resp. 1500 exactes). Autrement dit, CONCISE permet d'éliminer 64,44% des règles négatives approximatives (resp. 53,33% des règles négatives exactes) redondantes et/ou proches de l'indépendance, encaissées par l'approche de Ramanantsoa, dans `Chess`.

Nous abordons maintenant l'étude du temps de calcul de CONCISE versus aux approches existantes. A signaler que l'approche de Ramanantsoa ne prend pas compte l'extraction des motifs fréquents d'une base de données, étant donnée que cette étape est une étape fondamentale (voire la

plus consommatrice en temps de calcul) lors de l'extraction des règles d'association. Cela nous va ainsi limiter notre étude autour de PRINCE et CONCISE uniquement. Les temps d'exécution de l'algorithme CONCISE comparés à l'algorithme PRINCE sur les mêmes contextes du tableau 2.2 sont présentés dans la figure 2.3 en faisant varier le *minsup* et fixer la $confiance = 0.6$. En plus des

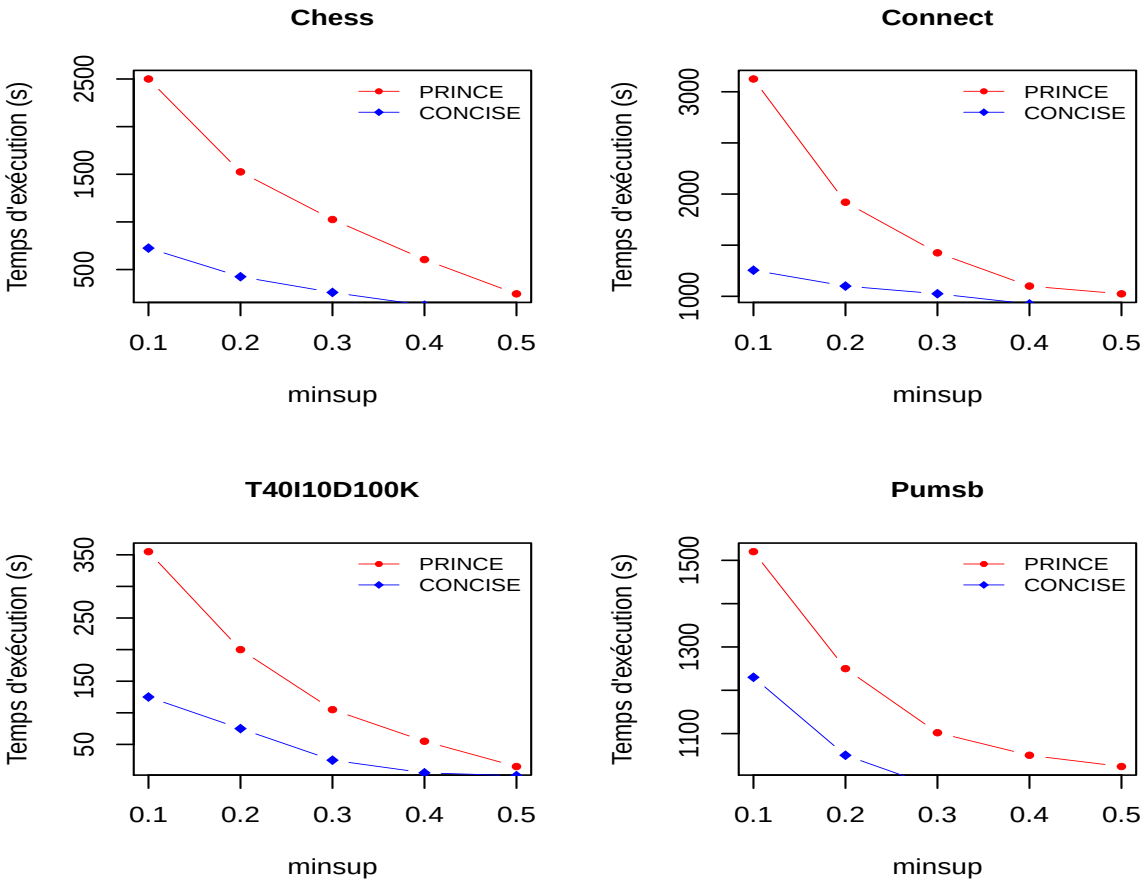

Figure 2.3 – Temps d'exécution de CONCISE versus PRINCE pour l'extraction des bases des règles

bases des règles d'association négatives exactes/approximatives que PRINCE ne peut pas traiter, CONCISE reste encore meilleur dans tous les cas. Cette différence des performances atteint son maximum pour les contextes `Chess` et `Connect`. L'explication vient du fait que ces deux bases de données sont de nature très dense et produisent un nombre important de motifs fréquents fermés, ce qui pénalise PRINCE dû au coût de la construction du treillis d'Iceberg pour des basses valeurs de *minsup*. Ainsi, l'espace de recherche pour PRINCE peut être parcouru dans sa globalité. En effet, l'algorithme CONCISE est environ 4 (resp. 3) fois plus rapide que PRINCE pour la base `Chess` (resp. `Connect`) avec un support de 0.1. Par ailleurs, sur les bases éparses telles que T40I10D100K et `Pumsb`, les temps de calcul pour les deux algorithmes restent raisonnables pour touts les *minsup*. Prenons par exemple la base T40I10D100K avec $minsup = 0.1$, l'extraction des bases des règles est réalisée en moins de 150 secondes (resp. 400 secondes) pour CONCISE (resp. PRINCE).

Chapitre 3

Contributions à la représentation des règles d'association par des graphe et arbre

Nous traiterons dans ce chapitre la synthèse des articles [BT20e, BT20f, BCT21, BJT22a, BJT22b]. Par souci du manque d'espace, certains aspects algorithmiques ne sont pas très détaillés. La section 3.1 introduit quelques bases préliminaires sur le problème des graphes et des arbres hiérarchiques. La section 3.2 définit quant à elle de nouvelles mesures de qualité induites par le graphe implicatif et l'arbre hiérarchique. Les sections 3.3 et 3.4 se focalisent sur la mise en place respective d'un nouvel algorithme d'élaboration de graphes implicatifs, et celui des arbres hiérachiques d'un ensemble des meilleures règles d'association valides. La section 3.5 s'articule sur la conception d'un nouvel outil `rchicmgk`. Enfin, la section 3.6 est dédiée à l'évaluation expérimentale.

3.1 Terminologie et notations

Définition 20 (Décomposition d'ensemble)**.** *Soient $\mathcal{E}$ un certain ensemble et $\Omega = \{\mathcal{E}_i\}_{1\leq i\leq p}$ un ensemble d'ensembles $\mathcal{E}_i$ tels que $\mathcal{E}_i \subseteq \mathcal{E}$. Ω est une décomposition de $\mathcal{E}$ ssi $\bigcup_{i=1}^{p} \mathcal{E}_i = \mathcal{E}$ et $\mathcal{E}_i \cap \mathcal{E}_j = \emptyset$.*

Définition 21 (Partition d'ensemble)**.** *Soient $\mathcal{E}$ un ensemble quelconque, et $\Omega = \{\mathcal{E}_i\}_{1\leq i\leq p}$ une décomposition de $\mathcal{E}$. Ω est une partition de $\mathcal{E}$ ssi $\mathcal{E}_i \cap \mathcal{E}_j = \emptyset$ et $\bigcup_{i=1}^{p} \mathcal{E}_i = \mathcal{E}$, $\forall i \neq j$.*

Définition 22 (Nombre de Bell, [BB04, QUE13, Ben18])**.** *Le p-ième nombre de Bell, B_p, est le nombre de partitions d'un ensemble à p éléments en sous-ensembles non-vides.*

Nous noterons Ω l'ensemble des partitions que l'on peut obtenir à partir d'un ensemble quelconque de p éléments. Le cardinal de Ω est donné par le nombre de Bell B_p ($p \geq 0$) :

$$B_{p+1} = \sum_{k=0}^{p} \binom{p}{k} B_k, \text{ avec } B_0 = B_1 = 1$$

Dans le cas (idéal) où l'on dispose d'une fonction f permettant d'associer un coût à chaque partition, le problème du partitionnement consiste à rechercher le meilleur coût :

$$argmin_{\Pi\in\Omega} f(\Pi)$$

où Π désigne une partition quelconque [1]. Cette formulation n'est toutefois que d'un intérêt limité

1. Nous utilisons ici la notation Π pour une certaine partition, la notation P étant réservée aux probabilités.

en pratique, car d'une part l'on ne dispose pas toujours d'une fonction de coût, et d'autre part le nombre de partitions possibles (nombre de Bell) est en général trop élevé pour qu'une recherche exhaustive soit menée. En pratique, seul un petit nombre de partitions est en général examiné.

Définition 23 (Classe)**.** *Soit* Π *une partition d'un certain ensemble* $\mathcal{E}$*. Un élément de* Π *est appelé une classe. Une classe correspond donc à un sous-ensemble de* $\mathcal{E}$*. Une classe d'une partition* Π *sera de plus notée* C_i*, où* $1 \leq i \leq |\Pi|$*.*

Il n'existe pas de définition claire de ce qu'est une classe *a priori*. Autrement dit, la problématique du partitionnement, et plus généralement celle de la classification d'un ensemble quelconque, est en partie subjective, les classes recherchées dépendent des applications et analyses visées.

Définition 24 (Paire d'éléments)**.** *Soit* $\mathcal{E}$ *un certain ensemble. On appelle paire d'éléments de* $\mathcal{E}$ *tout ensemble non-orienté de deux éléments* $\{u, v\}$ *tels que* $\{u, v\} \subseteq \mathcal{E}$*.*

Définition 25 (Couple d'éléments)**.** *Soient* $\mathcal{E}$ *un ensemble, et* u *et* v *deux éléments de* $\mathcal{E}$*. On appelle couple d'éléments* u *et* v*, la donnée de* u *et* v *dans un ordre déterminée, notée* (u, v)*.*

Tout système qui consiste en un ensemble discret d'états appelés sommets (nœuds) entre lesquels des liens appelés arcs (arêtes) traduisent une certaine relation, peut être modélisé par un graphe.

Définition 26 (Graphe)**.** *Un graphe* $G = (V, E)$ *est donné par un ensemble* V *de sommets (**vertices**) et un ensemble d'arêtes (**edges**)* $E \subseteq V \times V$*. Deux sommets reliés par une arête sont dits **voisins**, et une arête reliant deux sommets* u *et* v *est dite **incidente** à* u *et* v*, et est notée* (u, v)*.*

Définition 27 (Ordre et taille d'un graphe)**.** *Soit* $G = (V, E)$ *un graphe. On appelle ordre (resp. taille) du graphe, le nombre de sommets* $|V|$ *(resp. d'arêtes* $|E|$*) dans le graphe* G*.*

Les arêtes peuvent être orientées ou non. Une arête orientée s'appelle aussi un arc. On peut par suite distinguer deux graphes, à savoir graphe non-orienté et graphe orienté.

Définition 28 (Graphe non-orienté)**.** *Soient* V *et* E *deux ensembles de sommets (resp. d'arêtes), tels que* $E \subseteq \{\{u, v\} | u \in V, v \in V\}$*. On appelle graphe non-orienté le couple défini par* $G = (V, E)$*.*

Définition 29 (Graphe orienté)**.** *Soient* V *et* E *deux ensembles de sommets (resp. d'arêtes), tels que* $E \subseteq \{(u, v) | u \in V, v \in V\}$*. On appelle graphe orienté le couple défini par* $G = (V, E)$*.*

Soit $a = (u, v)$ un arc (arête orienté), les sommets u et v sont appelés les extrémités de l'arc a.

Définition 30 (Boucle)**.** *Soient* $G = (V, E)$ *un graphe orienté, et* $a = (u, v)$ *un arc (arête orientée) de* E *tel que* $u, v \in V$*. L'arc* a *est une boucle si et seulement si* $u = v$*. Autrement dit, une boucle est un arc qui relie un sommet à lui-même.*

Définition 31 (Chemin orienté, [QUE13])**.** *Soit* $G = (V, E)$ *un graphe orienté. On appelle chemin orienté dans* G *une séquence* $(v_1, a_1, v_2, a_2, \cdots, a_{l-1}, v_l)$ *avec :*
- $\forall i \in [1, l], v_i \in V$ *;*
- $\forall i \in [1, l-1], a_i = (v_i, v_{i+1}) \in E$ *;*
- $\forall i, j \in [1, l]$*,* $i \neq j$*,* $v_i \neq v_j$ *et* $a_i \neq a_j$*.*

Définition 32 (Graphe connexe)**.** *Soit* $G = (V, E)$ *un graphe. Le graphe* G *est connexe s'il existe un chemin entre toute paire de sommets dans* G*.*

Définition 33 (Arbre). *Soient V et E deux ensembles de sommets (resp. d'arêtes). On dit que $\mathcal{A} = (V, E)$ est un arbre ssi $\mathcal{A}$ est un graphe acyclique (sans boucle) et connexe.*

Définition 34 (Sous-arbre). *Soit $\mathcal{A} = (V, E)$ un arbre. Tout sous-graphe connexe de $\mathcal{A}$ est appelé un sous-arbre de $\mathcal{A}$.*

Définition 35 (Arbre enraciné, [QUE13]). *Soit $\mathcal{A} = (V, E)$ un arbre orienté. On dit que $\mathcal{A}$ est un arbre enraciné en un nœud $u \in V$ (appelé racine) si et seulement s'il existe un unique chemin orienté de u à tout autre nœud de $\mathcal{A}$.*

Définition 36 (Hauteur, [QUE13]). *Soit $\mathcal{A} = (V, E)$ un arbre enraciné en u. La hauteur d'un nœud $v \in V$ notée $h(u, v)$ correspond à la distance entre u et v.*

Définition 37 (Niveau, [QUE13]). *Soit $\mathcal{A} = (V, E)$ un arbre enraciné en u. Le i^e niveau de $\mathcal{A}$ est l'ensemble de feuilles dans le sous-arbre induit par l'ensemble $\{v \in V, h(u, v) \leq i\}$.*

3.2 Mesures de qualité des graphe et arbre hiérarchique

Dans cette partie, nous restons toujours à l'analyse de données transactionnelles. Comme cela avait été présenté dans la figure 1.1, les données peuvent souvent être représentées à l'aide d'un tableau comme rappelé du tableau 3.1 ci-après, où les colonnes sont indexées par les motifs (appelés aussi sommets), alors que les lignes par les transactions. Notons par m le nombre de sommets et n

Table 3.1 – Une base de données transactionnelles

	$v_1 \ldots$	v_j	$\ldots v_m$
t_1 $\vdots$			
t_i	$\ldots$	t_{ij}	$\ldots$
$\vdots$ t_n			

celui de transactions. Par suite, on peut noter par $V = \{v_j | 1 \leq j \leq m\}$ l'ensemble des sommets, et $\mathcal{T} = \{t_i | 1 \leq i \leq n\}$ celui de transactions. L'intersection de la i^e ligne et de la j^e colonne est $t_{ij} = v_j(t_i)$ qui est une valeur observée de sommet v_j sur la transaction élémentaire t_i.

Bien que la recherche de structures d'une telle base de données représente une aide essentielle en fouille de données. La représentation par des graphe implicatif et arbre hiérarchique permet de répondre à cette problématique dont l'objectif affiché est comme mentionné d'assister graphiquement l'utilisateur à explorer dans ses données les meilleures règles d'association telles que celles par exemple $v_k \to v_\ell$ (resp. $v_k \to (v_\ell \to v_s)$), où v_k, v_ℓ et v_s sont les sommets (resp. feuilles) de graphes (resp. nœuds d'arbres). Une telle règle d'association $v_k \to v_\ell$ est modélisée par un *arc* (resp. une *classe de cardinal 2*) dans le cadre de graphes implicatifs (resp. d'arbres hiérarchiques). Soit $a_{v_k v_\ell} = (v_k, v_\ell)$ un arc, on appelle le sommet v_k (resp. v_ℓ) la source (resp. la destination) de l'arc $a_{v_k v_\ell}$. On dit aussi que v_ℓ est le *successeur* de v_k, et v_k le *prédécesseur* de v_ℓ. Dans ce graphe, un sommet est un motif, et une arête est valuée par la valeur d'une mesure de qualité.

Ainsi, une étape ultime de cela est celle de la définition d'une mesure de qualité (appelée aussi mesure de similarité) afin de quantifier la similarité des couples d'éléments de l'ensemble

V des sommets. Il existe pour cela plusieurs mesures de qualité, la plus utilisée étant l'*intensité d'implication* [GAB+96] telle que définie dans l'équation (2.6) et rappelée ci-après :

$$\varphi(v_k, v_\ell) = 1 - \Phi(q(v_k, \overline{v_\ell})), \text{ où } q(v_k, \overline{v_\ell}) = \frac{n_{v_k\overline{v_\ell}} - \frac{n_{v_k} n_{\overline{v_\ell}}}{n}}{\sqrt{\frac{n_{v_k} n_{\overline{v_\ell}}}{n}}}$$

Toutefois, comme mentionné dans le chapitre 2 précédent (section 2.2), l'intensité d'implication φ présente plusieurs défauts remarquables. Pour y faire face, nous avons proposé dans [BT21a, BT21b] une autre mesure plus sélective telle que définie dans l'équation (2.9) et rappelée ci-après :

$$mgk(v_k, v_\ell) = 1 - \Phi(\widetilde{mgk}(v_k, \overline{v_\ell})), \text{ où } \widetilde{mgk}(v_k, \overline{v_\ell}) = \frac{n_{v_k\overline{v_\ell}} - \frac{n_{v_k} n_{\overline{v_\ell}}}{n}}{\frac{n_{v_k} n_{\overline{v_\ell}}}{n}}$$

Il en résulte que $\frac{\partial \widetilde{mgk}}{\partial n_{v_k\overline{v_\ell}}} = \frac{1}{\frac{n_{v_k} n_{\overline{v_\ell}}}{n}} = \frac{1}{\frac{n_{v_k}(n-n_{v_k})}{n}} > 0$ et $\frac{\partial q}{\partial n_{v_k\overline{v_\ell}}} = \frac{1}{\sqrt{\frac{n_{v_k} n_{\overline{v_\ell}}}{n}}} = \frac{1}{\sqrt{\frac{n_{v_k}(n-n_{v_k})}{n}}} > 0$, ce qui relève que les indices de qualité $\widetilde{mgk}$ et q décroissent quand $n_{v_k\overline{v_\ell}}$ augmente et d'autant plus vite que $\frac{n_{v_k} n_{\overline{v_\ell}}}{n}$ est faible. Autrement dit, les mesures statistiques mgk et φ sont tous sensibles aux occurrences de $n_{v_k\overline{v_\ell}}$. Par ailleurs, on obtient que $\frac{\partial \widetilde{mgk}}{\partial n_{v_k\overline{v_\ell}}} < \frac{\partial q}{\partial n_{v_k\overline{v_\ell}}}$, ce qui bien affirme que φ croît plus vite vers la valeur maximale 1 que mgk, et que ce dernier est plus discriminante que φ. Une étude beaucoup plus poussée sur les variations de la mesure mgk a été effectuée dans l'annexe D.

Dans le cadre de classification, une mesure la plus classique est celle basée sur l'intensité d'implication φ. Pour tous motifs v_k et v_ℓ, la cohésion d'une classe (v_k, v_ℓ) de degré 2 est définie, à partir de l'entropie au sens de Shannon entre chrochets, est donée par la formule (3.1) ci-après :

$$coh\varphi(v_k, v_\ell) = \begin{cases} \sqrt{1 - [-\varphi \log_2(\varphi) - (1-\varphi)\log_2(1-\varphi)]^2} & \text{si } \varphi \geq 1/2 \\ 0 & \text{si } \varphi < 1/2 \end{cases} \tag{3.1}$$

Toutefois, $coh\varphi$ hérite les défauts de φ. Ainsi, nous avons proposé dans [BJT22a] une nouvelle mesure de cohésion plus sélective, notée $cohmgk(v_k, v_\ell)$, et définie dans l'équation (3.2). Dans ce cas, nous reprenons la formule (3.1) mais dotée d'une autre métrique $\vartheta = mgk(v_k, v_\ell)$:

$$cohmgk(v_k, v_\ell) = \begin{cases} \sqrt{1 - [-\vartheta \log_2(\vartheta) - (1-\vartheta)\log_2(1-\vartheta)]^2} & \text{si } \vartheta \geq 1/2 \\ 0 & \text{si } \vartheta < 1/2 \end{cases} \tag{3.2}$$

La mesure $cohmgk$, tout comme $coh\varphi$, prend la valeur dans $[0; 1]$ telle que $0 \log_2(0) = \log_2(1) = 0$. Le cas où $cohmgk$ est proche de 1 (resp. égale à 1) correspond à l'implication logique où la cohésion d'une classe est proche de 1 (resp. égale à 1). De la même manière, pour $cohmgk = 0$, la mesure mgk est inférieure à $1/2$ où le nombre d'exemples est inférieur à celui de contre-exemples.

Définition 38 ([BCT21, BJT22b]). *Soient $V = \{v_1, \cdots, v_m\}$ l'ensemble de sommets, et $E = \{(v_k, v_\ell) | v_k \in V, v_\ell \in V\}$ celui d'arcs. Un graphe implicatif est un couple :*

$$G_\alpha = (V, E) \text{ ssi } mgk(v_k, v_\ell) \geq 1 - \alpha,$$

où mgk est une mesure de qualité définie dans (2.9), et α un seuil tel que $0 \leq \alpha < 1$.

Soit $\mathcal{R}_\alpha$ une relation définie sur $V \times V$ par *mgk* à un seuil α. Nous définissons une relation $v_i \mathcal{R}_\alpha v_j$ si et seulement si $mgk(v_i, v_j) \geq 1 - \alpha$. La relation définie par l'implication statistique, si elle est réflexive et acyclique, n'est pas explicitement transitive. Il existe certes une fermeture transitive $v_i \mathcal{R}_\alpha v_k$ si l'assertion ($v_i \mathcal{R}_\alpha v_j$ et $v_j \mathcal{R}_\alpha v_k$ tels que $mgk(v_i, v_k) \geq 0.5$) est vérifiée.

Il est à signaler que des arcs du graphe implicatif G_α peuvent apparaître ou disparaître selon les variations du seuil critique α. Cela conduit à une partition en plusieurs sous-graphes.

Définition 39 ([BCT21, BJT22b]). *On appelle $G'_\alpha = (V', E')$ un sous-graphe de G_α si et seulement si $V' \subset V$ et $E' \subseteq E$ dont les sommets sont dans V'.*

Définition 40 ([BJT22a]). *Soient $V = \{v_1, \cdots, v_m\}$ l'ensemble de sommets, $E = \{(v_k, v_\ell) | v_k \in V, v_\ell \in V\}$ l'ensemble d'arcs. Un arbre hiérarchique est défini par le couple :*

$$\mathcal{H} = (V, E) \text{ si et seulement si } mgk(v_k, v_\ell) \geq 0.5$$

où mgk est une mesure statistique telle que donnée dans l'équation (2.9) du chapitre 2.

Définition 41 ([BJT22a]). *On appelle $\mathcal{H}' = (V', E')$ un sous-arbre de $\mathcal{H}$ tel que $V' \subset V$ et $E' \subseteq E$.*

En adaptant les mêmes techniques dans [GKB03, Ler08, GRMG13], nous obtenons les expressions ci-dessous pour la cohésion d'une classe et inter-classe de degré quelconque.

Définition 42 ([BJT22a]). *Soit $C = (X_1, \ldots, X_r)$ une classe de degré r dont l'ordre induit est $X_1 \subset \ldots \subset X_r$. On définit la cohésion de la classe C par la moyenne géométrique des valeurs de la cohmgk sur l'ensemble des couples (X_i, X_j) du graphe de la relation d'ordre, donnée par :*

$$cohmgk(C) = \left[\prod_{i=1,\ldots,r-1}^{j=2,\ldots,r, j>i} cohmgk(X_i, X_j) \right]^{\frac{2}{r(r-1)}} \tag{3.3}$$

Considérons maintenant deux classes $C_j = \{X_{j_l} | 1 \leq l \leq r\}$ et $C_{j^\star} = \{X_{j^\star_q} | 1 \leq q \leq s\}$ de taille respective r et s, présentes à un niveau donné de la hiérarchie implicative, avec $X_{j_1} \subset X_{j_2} \subset \ldots \subset X_{j_l} \subset \ldots \subset X_{j_r}$ et $X_{j^\star_1} \subset X_{j^\star_2} \subset \ldots \subset X_{j^\star_q} \subset \ldots \subset X_{j^\star_s}$. La cohésion entre deux classes (inter-classes) C_j et $C_{j^\star}$ est définie par l'équation (3.4) ci-après :

$$cohmgk(C_j, C_{j^\star}) = \left[cohmgk(C_j)^{\binom{l}{2}} cohmgk(C_{j^\star})^{\binom{q}{2}} \prod cohmgk(X_{j_l}, X_{j^\star_q}) | 1 \leq l \leq r, 1 \leq q \leq s \right]^{\frac{2}{\theta(\theta-1)}} \tag{3.4}$$

où $\theta = r + s$, et $\binom{\eta}{2}$ note pour η entier, un coefficient binomial qui vaut $\frac{\eta(\eta-1)}{2}$. Cette cohésion joue un rôle important dans le problème de fusion des classes d'un arbre hiérarchique orienté.

3.3 IMGRAPH, un nouvel algorithme de graphes implicatifs

L'objectif principal de l'algorithme ImGraph (*Implicative Graph*) [BJT22a] est de pallier les principales lacunes de l'approche classique de Gras [GRMG13], c'est-à-dire (*i*) d'offrir une possibilité d'aller au-delà de règles d'association positives, (*ii*) de proposer une technique de partitionnement

d'arcs qui structure l'ensemble des graphes, et (*iii*) de réduire le temps de calcul sur cet ensemble des graphes en réduisant le nombre de croisement au sein des partitions par l'élagage des redondances. L'algorithme ImGraph se divise essentiellement en 3 étapes développées ci-après.

3.3.1 Construction d'une matrice de similarité

Cette première étape consiste à construire une matrice de similarité. Cela permet de sélectionner les meilleures règles, à un certain seuil α, en utilisant l'algorithme CONCISE (cf. Chapitre 2). CONCISE consiste en 2 étapes : (*i*) Extraction des motifs fréquents en utilisant l'algorithme CMG (cf. Chapitre 1), (*ii*) Génération des meilleures règles via l'algorithme CBNR (Algorithme 4). CBNR propose une stratégie de filtrage très puissante qui permet d'éliminer les règles d'association redondantes en utilisant les concepts de bases des règles [BT19b, BT21a, BT21b], et de conserver celles qui sont meilleures. C'est une technique qui n'est pas présente dans la démarche initiale [GAB+96, GRMG13] de graphes implicatifs. L'objectif est non seulement de réduire la dimension de l'espace de recherche mais aussi d'obtenir des meilleurs graphes (i.e., graphes de taille raisonnable et moins denses). En fait, à la sortie de CONCISE, nous obtenons, pour deux motifs disjoints v_k et v_ℓ, 4 familles des règles d'association valides, à savoir famille des règles d'association positives exactes (soit $mgk(v_k, v_\ell) = 1$) et approximatives ($mgk(v_k, v_\ell) \neq 1$), famille des règles d'association négatives exactes ($mgk(v_k, \overline{v_\ell}) = 1$, $mgk(\overline{v_k}, v_\ell) = 1$ et $mgk(\overline{v_k}, \overline{v_\ell}) = 1$) et approximatives ($mgk(v_k, \overline{v_\ell}) \neq 1$, $mgk(\overline{v_k}, v_\ell) \neq 1$ et $mgk(\overline{v_k}, \overline{v_\ell}) \neq 1$). Ces ensembles de règles valides seront formatés sous la forme d'une matrice d'adjacence, notée $\mathcal{M}_{sim} = (\mu_{k\ell})_{1 \leq k,\ell \leq |V|-h}$, et définie par :

$$\mathcal{M}_{sim} = \begin{cases} mgk(v_k, v_\ell), & \text{si } mgk(v_k, v_\ell) \geq 1 - \alpha; \\ 0, & \text{sinon} \end{cases} \tag{3.5}$$

où h est le nombre des motifs infréquents supprimés. Le terme général $\mu_{k\ell}$ correspond à la valeur $mgk(v_k, v_\ell)$. Plus précisément, la matrice $\mathcal{M}_{sim}$ est une matrice carrée d'ordre $|V|-h$ qui rassemble les termes $\mu_{k\ell} = mgk(v_k, v_\ell)$ à un seuil critique $\alpha \in [0, 1[$ fixé par l'utilisateur. De façon plus générale, une telle matrice de similarité $\mathcal{M}_{sim}$ (à K classes) peut être reformulée comme suit :

$$\mathcal{M}_{sim} = \begin{pmatrix} \mu_{11} & \mu_{12} & \cdots & \mu_{1K} \\ \mu_{21} & \mu_{22} & \cdots & \mu_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ \mu_{K1} & \mu_{K2} & \cdots & \mu_{KK} \end{pmatrix}$$

où les lignes et les colonnes représentent pour chacune les motifs associés à cet ensembles des règles valides. Dans ce cas, les μ_{kk} sont nuls, car $\mu_{kk} = mgk(v_k, v_k)$ n'a pas de sens selon la propriété statistique d'une règle d'association. Les $\mu_{k\ell}$ égalent $mgk(v_k, v_\ell)$ s'ils sont valides, et 0 sinon.

3.3.2 Ordonnancement de la matrice de similarité

La deuxième étape consiste à déterminer l'ordre des sommets de $\mathcal{M}_{sim}$, c'est-à-dire que les sommets d'une même composante connexe doivent être consécutifs. Cela donc rend les sommets similaires dans une même classe, et de découper $\mathcal{M}_{sim}$ en blocs homogènes (puisque si v_k et v_ℓ ne sont pas dans la même composante connexe, alors $\mu_{k\ell} = 0$). Un bloc est une sous-matrice pour

laquelle les sommets sont connexes entre eux. Ce procédé est récursif pour chaque sommet de la matrice $\mathcal{M}_{sim}$, et s'arrête quand il n'y a pas des sommets non regroupés. Soit C_k la k^{e} communauté ainsi obtenue et η_k le nombre de sommets contenus dans cette communauté, nous obtenons une matrice de similarité ayant une forme diagonale par blocs, notée $\mathcal{M}_{sim}^{bloc}$, et définie par :

$$\mathcal{M}_{sim}^{bloc} = \begin{pmatrix} C_1 & O_{12} & \cdots & O_{1K} \\ O_{21} & C_2 & \cdots & O_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ O_{K1} & O_{K2} & \cdots & C_K \end{pmatrix}$$

Chacun des blocs $C_{k,\forall k\in\{1,\ldots,K\}}$ est une sous-matrice carrée de taille $\eta_k \times \eta_k$, associé à sous graphe $G'_{\alpha,k} = (V_{C_k}, E_k) \subset G_\alpha$ induit par la k^{e} communauté C_k de G_α. Les périphériques $(O_{k\ell})_{k,\ell\in\{1,\ldots,K\},k\neq\ell}$ sont des matrices nulles rectangulaires de taille $\eta_k \times \eta_\ell$. Par suite, on peut extraire, à un certain seuil α, une partition Π_α des communautés à partir de la matrice $\mathcal{M}_{sim}^{bloc}$, telle que donnée suivante :

$$\Pi_\alpha = \mathcal{M}_{sim}^{bloc}.C_{k,\forall k\in\{1,\ldots,K\}} = \{C_1, \ldots, C_K\} \tag{3.6}$$

A cet égard, nous considérons une illustration très simplifiée avec une petite base de données

$$\mathcal{D}' = \begin{array}{c} \\ t_1 \\ t_2 \\ t_3 \\ t_4 \\ t_5 \\ t_6 \end{array} \begin{array}{c} \begin{array}{ccccc} v_1 & v_2 & v_3 & v_4 & v_5 \end{array} \\ \begin{pmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 \end{pmatrix} \end{array}$$

Pour ce faire, nous fixons $minsup = 2/6$ et faisons varier α. Un α qui détermine la meilleure partition est appelé α *optimal*. On entend d'un α optimal lorsqu'on augmente sa valeur, il n'y aura pas de fusion dans les étapes suivantes qui donnerait une meilleure similarité. Après avoir varié α, on obtient $\alpha = 0.01$ comme un α optimal, et la matrice de similarité associée est donnée ci-après :

$$\mathcal{M}_{sim} = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_5 \end{array} \begin{array}{c} \begin{array}{cccc} v_1 & v_2 & v_3 & v_5 \end{array} \\ \begin{pmatrix} 0 & 0 & 1.00 & 0 \\ 0 & 0 & 0 & 1.00 \\ 0 & 0 & 0 & 0 \\ 0 & 1.00 & 0 & 0 \end{pmatrix} \end{array}$$

Nous observons que v_4 a été supprimé du fait qu'il n'est pas fréquent. A partir de la matrice de similarité $\mathcal{M}_{sim}$, nous construisons la matrice de similarité par blocs $\mathcal{M}_{sim}^{bloc}$ donnée par :

$$\mathcal{M}_{sim}^{bloc} = \begin{array}{c} \\ v_1 \\ v_3 \\ v_2 \\ v_5 \end{array} \begin{array}{c} \begin{array}{cccc} v_1 & v_3 & v_2 & v_5 \end{array} \\ \begin{pmatrix} 0 & 1.00 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.00 \\ 0 & 0 & 1.00 & 0 \end{pmatrix} \end{array} = \begin{pmatrix} \boxed{\begin{matrix} 0 & 1.00 \\ 0 & 0 \end{matrix}} & O \\ O & \boxed{\begin{matrix} 0 & 1.00 \\ 1.00 & 0 \end{matrix}} \end{pmatrix}$$

Ce résultat fait apparaître deux blocs homogènes, soit une partition $\Pi_\alpha = \{(v_1, v_3), (v_2, v_5)\}$.

Cette étape sera synthétisée à l'aide de l'algorithme 13, appelé ReSiMa (*Reorganizing Similarity Matrix*). Ainsi, notons par τ le compteur d'éléments pour chaque classe, et ℓ le nombre de lignes/colonnes supprimées. L'algorithme ReSiMa prend en entrée la matrice de similarité trouvée à l'étape 1, et donne en sortie une matrice diagonale par blocs d'ordres respectifs $\eta_1, \ldots, \eta_k$.

Algorithm 13 ReSiMa (Reorganizing Similarity Matrix)

Require: Une matrice de similarité $\mathcal{M}_{sim} = (\mu_{uv})_{p\times p}$.
Ensure: Une matrice diagonale par blocs $\mathcal{M}_{sim}^{bloc}$ d'ordres respectifs $\eta_1, \ldots, \eta_k$.

$\ell \leftarrow 0$; $i \leftarrow 1$; $j \leftarrow 2$; $\tau \leftarrow 1$; $C_1 \leftarrow \{\mu_1\}$; /* ℓ, i, j, τ variables intermédiaires */
repeat
  while $(\mu_{ij} + \mu_{ji} \neq 0 \;||\; \ell \geq p - j - 1)$ **do**
    $permut\,(\mu[, j], \mu[, p - \ell])$; $permut\,(\mu[j,], \mu[p - \ell,])$; $\ell \leftarrow \ell + 1$;
  end while
  if $(\mu_{ij} + \mu_{ji} \neq 0)$ **then**
    if $(C_k \cap \{\mu_j\} = \emptyset)$ **then**
      $\tau \leftarrow \tau + 1$; $j \leftarrow j + 1$; $C_k \leftarrow C_k \cup \{\mu_j\}$;
    end if
  else
    if $(i < \tau)$ **then**
      $i \leftarrow i + 1$; $j \leftarrow j + 1$; $\tau \leftarrow 0$;
    else
      if $(i < p - 3)$ **then**
        $\eta_k \leftarrow \tau$; $k \leftarrow k + 1$; $C_k \leftarrow \{\mu_{i+1}\}$; $\tau \leftarrow \tau + 1$; $i \leftarrow i + 1$; $j \leftarrow j + 1$;
      else
        if $(i < p - 2)$ **then**
          $\eta_k \leftarrow \tau$; $k \leftarrow k + 1$; $C_k \leftarrow \{\mu_{i+1}, \mu_{i+2}, \mu_{i+3}\}$; $\eta_k \leftarrow \tau + 3$; $j \leftarrow j + 1$;
        else
          $\eta_k \leftarrow \tau$; $k \leftarrow k + 1$; $C_k \leftarrow \{\mu_{i+1}, \mu_{i+2}\}$; $\eta_k \leftarrow \tau + 2$; $j \leftarrow j + 1$;
        end if
      end if
    end if
  end if
until $(j > p)$
$\zeta_1 \leftarrow \eta_1$;
for $(i = 2$ to $k)$ **do**
  $\zeta_i \leftarrow \eta_i - \eta_{i-1}$;
end for

3.3.3 Configuration algorithmique de graphes implicatifs

La troisième et dernière étape de l'algorithme ImGraph, configuration de graphes implicatifs $G_\alpha = (V, E)$ au sens de la définition 38, s'effectue au niveau de la matrice diagonale par blocs $\mathcal{M}_{sim}^{bloc}$ trouvée à l'étape 2 ci-dessus. Cela consiste à construire, pour chaque communauté $\{C_k\}_{k\in\{1,\ldots,K\}}$ de $\mathcal{M}_{sim}^{bloc}$, les sous graphes $G'_{\alpha,k} = (V_{C_k}, E_k)_{k\in\{1,\ldots,K\}}$ de G_α, où V_{C_k} (resp. E_k) est l'ensemble des sommets (resp. d'arcs) associés à C_k. Cette configuration est résumée dans l'algorithme 14, dénommé CImGraph. L'algorithme CImGraph prend en entrée la matrice de similarité $\mathcal{M}_{sim}^{bloc}$ par blocs, et retourne en sortie l'ensemble des graphes implicatifs constitué d'un enemble des sommets

V et celui d'arcs E. En particulier, soient $x \to y$ et $y \to z$ deux certaines règles, la fermeture transitive $x \to z$ est acceptée lorsque $mgk(x, z)$ est supérieur ou égal à la neutralité (i.e., $\geq 0,5$). Soit V_{C_k} un ensemble de sommets d'une communauté C_k. Pour chaque communauté de $\mathcal{M}_{sim}^{bloc}$, le

Algorithm 14 CImGraph (Constructing Implicative Graph)

Require: Une matrice $\mathcal{M}_{sim}^{bloc} = (v_{ij})$ diagonale par blocs d'ordres respectifs $\eta_1, \ldots, \eta_K$.
Ensure: Un graphe implicatif $G_\alpha = (V, E)$

```
E ← ∅ ; h ← 1 ; ℓ ← 0 ; p ← η_1 ;                    /* h, ℓ, p variables intermédiaires */
repeat
   for (i = ℓ + 1 to p) do
      for (j = ℓ + 1 to p) do
         for (k = ℓ + 1 to p) do
            if (v_ij ≠ 0) then
               if (v_ij.v_jk ≠ 0) then
                  if (i ≠ k ∧ v_ik ≥ 0.5) then
                     E ← E ∪ {(i, j), (j, k), (i, k)} ;
                  else
                     E ← E ∪ {(i, j), (j, k)} ;
                  end if
               else
                  E ← E ∪ {(i, j)} ;
               end if
            end if
         end for
      end for
   end for
   if (h < K) then
      ℓ ← ℓ + η_h ; h ← h + 1 ; p ← ℓ + η_h ;
   else
      h ← h + 1
   end if
until (h > K + 1)
```

processus de construction est récursif, et s'arrête lorqu'il n'y a pas de sommets non enchaînés.

3.4 CAHI, un algorithme d'arbres hiérarchiques orientés

A notre connaissance, la littérature s'est beaucoup intéressée au problème d'arbres hiérarchiques non-orientés (CAH, classification ascendante hiérarchique), très peu d'approches traitent le problème d'arbres hiérarchiques orientés. Nous pouvons néanmoins citer le modèle pionnier [Ler08, GRMG13] basé sur l'intensité d'implication [GAB+96]. Cependant, comme nous venons de constater que ce modèle pose plusieurs limites. Entre autres, il produit de très grand nombre des règles qui bruitent l'arbre, dont la majorité sont des redondances. Une autre lacune repose du fait qu'il ne propose aucune technique de partitionnement d'arcs, alors que celui-ci apparait très central en classification, car il structure l'arbre. Pour surmonter ces limites, nous avons proposé dans [BJT22a] un nouvel algorithme, appelé CAHI (*Classification Ascendante Hiérarchique Implicative*), permettant la construction d'arbres hiérarchiques orientés. CAHI fonctionne à peu

près avec le même principe que ImGraph. Il se divise aussi en 3 étapes. (*i*) Construction d'une matrice de cohésion en utilisant la nouvelle mesure de cohésion *cohmgk* définie dans l'équation (3.2). S'ajoute à cela une stratégie d'élagage des redondances. Il s'agit, à notre connaissance, de la première tentative d'éliminer les redondances dans le cadre d'arbres hiérarchiques orientés. (*ii*) Construction des classes homogènes à partir de la matrice de cohésion trouvée à l'étape 1. (*iii*) Configuration algorithmique d'arbres hiérarchiques suivant les classes trouvées à l'étape 2.

3.4.1 Construction d'une matrice de cohésion

La construction d'une matrice de cohésion consiste à calculer la cohésion pour tous les couples possibles des sommets de V. Cela permet de sélectionner les meilleures règles d'association. Ainsi, le calcul d'une matrice de cohésion, tout comme calcul de matrice de similarité, passe également à l'élagage des règles redondantes en faisant appel l'algorithme CONCISE [BT21a], particulièrement l'algorithme CBNR (algo. 4, chapitre 2). On est donc ramené au résultat de l'étape 1 de l'algorithme ImGraph. Précisément, on obtient le même ensemble de règles d'association positives et négatives qui a été stocké sous forme matricielle $\mathcal{M}$ de l'équation (3.5) ci-dessus. Par suite, la matrice de cohésion, notée $\mathcal{M}_{coh} = (\mu_{k\ell})_{1\leq k,\ell\leq |V|-h'}$, s'obtient en se ramenant tous les coefficients non nuls de la matrice $\mathcal{M}_{sim}$ au coefficients cohésions via la mesure *cohmgk* de (3.2), ce qui peut se ramener à

$$\mathcal{M}_{coh} = \begin{cases} cohmgk(v_k, v_\ell), & \text{si } mgk(v_k, v_\ell) \geq 0.5; \\ 0, & \text{sinon} \end{cases} \tag{3.7}$$

où h' est le nombre des motifs (i.e. lignes et colonnes) supprimés lors de la transformation.

3.4.2 Ordonnancement d'une matrice de cohésion

Très souvent, l'ordre initial de la matrice de cohésion $\mathcal{M}_{coh}$ ne donne pas une bonne classification, cette deuxième étape permet de retrouver un ordre des sommets qui soit compatible avec les structures trouvées à l'étape 1. Cela consiste à regrouper les lignes et colonnes de $\mathcal{M}_{coh}$, c'est-à-dire que les sommets d'une même composante connexe doivent être consécutifs. Donc, cela permet de découper la matrice de cohésion $\mathcal{M}_{coh}$ en blocs homogènes. Cette technique est récursive pour chaque sommet associé à $\mathcal{M}_{coh}$, et s'arrête quand il n'y a pas des sommets non-regroupés. On entend d'un bloc une sous-matrice pour laquelle les sommets associés sont connexes entre eux. Soit C_k la k^{e} communauté ainsi obtenue et η_k le nombre de sommets contenus dans cette communauté. La matrice $\mathcal{M}_{coh}$ devient alors en une matrice diagonale par blocs, notée $\mathcal{M}_{coh}^{bloc}$, et donnée par :

$$\mathcal{M}_{coh}^{bloc} = \begin{pmatrix} C_1 & O_{12} & \cdots & O_{1K} \\ O_{21} & C_2 & \cdots & O_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ O_{K1} & O_{K2} & \cdots & C_K \end{pmatrix}.$$

Chacun des communautés $C_{k,\forall k\in\{1,\ldots,K\}}$ est une sous-matrice carrée de taille $\eta_k \times \eta_k$, associé à sous-arbre $\mathcal{H}'_k = (V_{C_k}, E_k) \subset \mathcal{H}$ induit par la k^{e} classe C_k de $\mathcal{M}_{coh}^{bloc}$. Les périphériques $(O_{k\ell})_{k,\ell\in\{1,\ldots,K\},k\neq\ell}$ quant à elles sont des matrices nulles rectangulaires de taille $\eta_k \times \eta_\ell$. On obtient ensuite, à un seuil

α, une collection Π_α de partitions d'arbres tirées de $\mathcal{M}_{coh}^{bloc}$ telle que donnée dans la relation (3.8)

$$\Pi_\alpha = \mathcal{M}_{coh}^{bloc}.C_{k,\forall k\in\{1,\ldots,K\}} = \{C_1,\ldots,C_K\} \tag{3.8}$$

si et seulement si les assertions $((i)\ C_k \neq \emptyset, \forall k \in \{1,\ldots,K\}$, $(ii)\ \bigcup_{k\in\{1,\ldots,K\}} C_k = V$ et $(iii)\ C_k \cap C_{k'} = \emptyset, \forall k \neq k'$, où V est l'ensemble des sommets associés à $\mathcal{M}_{coh}^{bloc})$ sont vérifiées. La mise en œuvre de cette étape s'obtient en adaptant l'algorithme ReSiMa (Algorithme 13) ci-dessus.

3.4.3 Configuration algorithmique d'arbres hiérarchiques

La troisième et dernière étape de CAHI, configuration algorithmique d'arbres hiérarchiques, s'effectue au niveau de la matrice $\mathcal{M}_{coh}^{bloc}$ trouvée à l'étape 2. Cela consiste à construire les sous-arbres $\mathcal{H}_k = (V_{C_k}, E_k)_{k\in\{1,\ldots,K\}}$, associés à K communautés $\{C_k\}_{k\in\{1,\ldots,K\}}$. Cette étape est synthétisée à l'aide de l'algorithme 15, appelé IHTree (*Implicative Hierarchy Tree*). Cet algorithme prend en entrée la matrice $\mathcal{M}_{coh}^{bloc}$, et donne en sortie les sous-arbres pour chaque communauté possible, en faisant appel à une procédure secondaire FINDMAX (Algorithme 16) qui recherche, au processus de parcours d'une certaine communauté dans $\mathcal{M}_{coh}^{bloc}$, le coefficient maximal. A cette fin, il procède le principe suivant. Soit $C_k = (c_{ij})_{1\leq i,j\leq \eta_k}$ une k^{e} communauté de $\mathcal{M}_{coh}^{bloc}$, et V_{C_k} un ensemble de

Algorithm 15 Algorithme IHTree

Require: Matrice de cohésion $\mathcal{M}_{coh}^{bloc} = \{coh(v_i, v_j)\}_{1\leq i,j\leq n}$
Ensure: Arbre hiérarchique orienté $\mathcal{H}$

$\mathcal{H} = \emptyset$
for ($k = 1$ to K) **do**
  $\mathcal{M} \leftarrow \mathcal{M}_{coh}^{bloc}.C_k$; $\ell \leftarrow 0$; $\mathcal{H}_k \leftarrow \emptyset$; `/* ℓ variable intermédiaire */`
  for ($i = 1$ to η_k) **do**
    $C_i^{(0)} \leftarrow \{v_i\}$ `/* Générer des classes singletons */`
  end for
  repeat
    $\Theta \leftarrow FINDMAX(\mathcal{M})$
    $C_1^{(\ell+1)} \leftarrow \left(C^{\ell}_{\Theta.l_{\max}}, C^{\ell}_{\Theta.c_{\max}}\right)$
    if $(\mathcal{H}_k \cap C_1^{\ell+1} = \emptyset)$ **then**
      $\mathcal{H}_k \leftarrow \left(\mathcal{H}_k \cup C_1^{\ell+1}\right)$;
    else
      $\mathcal{H}_k \leftarrow \left(\mathcal{H}_k \backslash (\mathcal{H}_k \cap C_1^{\ell+1}) \cup C_1^{\ell+1}\right)$;
    end if
    reorganize $\mathcal{M}$ `/* Réorganisation de M après certaines combinaisons des classes */`
    $\ell \leftarrow \ell + 1$;
  until ($\ell \geq n - 2 \vee \Theta.\max$ est faible)
  $\mathcal{H} \leftarrow \mathcal{H} \cup \mathcal{H}_k$
end for
return $\mathcal{H}$

sommets de C_k. Pour chaque v_i, v_j de V_{C_k}, IHTree compare toutes les cohésions des arrangements 2 à 2 de V_{C_k}, et conserve celle, notée $\mathscr{C}_1$, qui correspond au meilleur coefficient, c-à-d que si c_{ij} est le plus grand élément de $\{c_{il}; l \neq i\}$, alors IHTree conserve $\mathscr{C}_1 = (v_i, v_j)$. Il compare ensuite toutes les cohésions à 2 éléments à celles des 3 éléments du type $(v_k, (v_i, v_j))$ et $((v_i, v_j), v_k)$, et conserve celle, notée $\mathscr{C}_2$, correspondant au maximum retenu, et ainsi de suite. En repétant ce processus

jusqu'à ce que les sommets de V_{C_k} soient enchainés. La condition d'arrêt porte à la fois sur le nombre d'itérations et la stabilité des classes. Il est parfois difficile, à cause des problèmes d'arrondi informatique, d'imposer que la partition obtenue soit stable. Partant d'un autre bloc, IHTree construit comme précédemment les sous-arbres, et s'arrête s'il n'y a pas de bloc non visités.

La procédure FINDMAX (Algorithme 16) prend en entrée une communauté de $\mathcal{M}_{coh}^{bloc}$, et retourne en sortie le triplet Θ composé de 3 champs, à savoir max le coefficient maximal de telle communauté, et ses coordonnées $l_{\max}$ et $c_{\max}$ qui correspondent respectivement aux indices de ligne et de colonne du coefficient max, donc spécifient la cellule contenant le coefficient max.

Algorithm 16 Procedure FINDMAX

Require: Matrice carrée $\mathcal{M} = (\mu_{ij})_{n\times n}$
Ensure: La donnée $\Theta = \langle \max, l_{\max}, c_{\max} \rangle$ `/*` max `coefficient maximal de` $\mathcal{M}$`,` $l_{\max}$ `(resp.` $c_{\max}$`) ligne (resp. colonne) correpondant à` max `*/`

$v_0 \leftarrow \mu_{11}$; $i \leftarrow 1$; $j \leftarrow 2$;
repeat
  repeat
    if $(v_0 < \mu_{ij})$ **then**
      $v_0 \leftarrow \mu_{ij}$; $\Theta \leftarrow (v_0, i, j)$;
    end if
    $j \leftarrow j + 1$;
  until $(j = n + 1)$
  $i \leftarrow i + 1$; $j \leftarrow 1$
until $(i = n + 1)$
return Θ

3.5 Package `rchicmgk`

Le package `rchicmgk` est le *fruit informatique* des travaux à la fois sur la fouille de données et sur l'analyse statistique implicative (ASI). Il s'inscrit dans le domaine de visualisation d'informations, notamment visualisation graphique d'un ensemble des règles d'association. Il se base ainsi sur des techniques de représentations visuelles et interactives pour permettre à l'utilisateur d'explorer de grandes quantités d'informations dans ses données. Dans la littérature, différentes techniques ont été proposées. Nous citons entre autres, le package `arules` [Bor03], basé sur la mesure confiance $Conf$ d'Agrawal [AIS93, AS94] ; le logiciel CHIC [GKB03, GRMG13], basé sur l'Intensité d'implication φ [GAB+96], et logiciel RCHIC [CG15, CP15] qui est une extension de CHIC sous R en ajoutant la mesure implifiance [GCG15]. Précisément, l'implifiance est la combinaison des φ et $Conf$. Etant données les limites respectives des $Conf$ et φ (chap. 1 et 2) que l'implifiance perçoit, nous proposons un nouveau package, nommé `rchicmgk` [BJT22a, BJT22b] à l'aide des nouvelles mesures plus sélectives *mgk* (équation (2.9)) et *cohmgk* (équation (3.2)), présentées dans tels chapitres 1 et 2.

Le package `rchicmgk` est une extension du prototype *CHIC-*M_{GK} que nous avons développé dans ma thèse de doctorat [Bem16]. Précisément, `rchicmgk` a initialement été conçu pour le graphe implicatif selon M_{GK}. Il est actuellement amélioré à l'instar de la nouvelle mesure statistique *mgk* [BT20c]. Par ailleurs, nous avons conçu de nouveaux programmes permettant la construction d'arbres hiérarchiques, en collaboration avec Raphaël Couturier (Université de Franche-Comté,

France). Concrètement, `rchicmgk` est une bibliothèque d'analyse et de visualisation graphique d'un ensemble des règles valides en utilisant plusieurs algorithmes tels que CMG [BT21a] pour l'extraction des motifs fréquents, CONCISE [BT21a] pour la génération des règles, ImGraph [BJT22b] pour la construction de graphes, et CAHI [BJT22a] pour l'élaboration d'arbres hiérarchiques.

Développée au fin de représentation graphique d'un ensemble des règles valides, `rchicmgk` traite des données binaires. Il s'articule alors autour de 2 phases : préparation et traitement de données.

3.5.1 Préparation de données

Ce module est consacré à la collecte et au prétraitement des données qui seront ensuite utilisées aux processus d'analyse et de représentation graphique des connaissances véhiculées dans ces données. En général, les données sont décrites par un tableau numérique où chaque colonne correspond à un motif de $\mathcal{I}$, chaque ligne correspond à une transaction de $\mathcal{T}$. Afin de les ramener en binaire, on note par 1 si le motif i est présent dans la transaction t (i.e., $t[i] = 1$) et 0 sinon. Une binarisation permet alors d'obtenir des attributs qui s'organisent en contexte $\mathcal{D} = (\mathcal{T}, \mathcal{I}, \mathcal{R})$ où chaque transaction $t \in \mathcal{T}$ est en relation selon $\mathcal{R}$ avec un ensemble d'attributs $\mathcal{I}$. Les variables-intervalles peuvent être découpées en différents intervalles à partir d'un nombre d'intervalles choisi par l'utilisateur. Le package `rchicmgk` prend en charge des fichiers au format *csv*. Notons d'ailleurs qu'il est possible de générer des données en respectant ce format grâce aux outils quelconques du logiciel R [2].

3.5.2 Traitement de données

Nous commençons par charger tout d'abord la bibliothèque `rchicmgk` dans un shell R (Rstudio par exemple) en utilisant la commande `library(rchicmgk)`. Comme prévu, `rchicmgk` propose deux volets graphiques, à savoir graphe implicatif et arbre hiérarchique. En exécutant la commande `rchicmgk()`, une boîte à outils s'affiche pour que l'utilisateur choisisse l'option pour le graphe implicatif ou pour l'arbre hiérarchique. Nous allons maintenant les présenter dans cet ordre.

Graphe implicatif. Dès que `rchicmgk` a généré l'ensemble des règles valides, il est possible de construire un graphe implicatif à partir de cet ensemble des règles valides. Dans la pratique, 4 seuils sont disponibles et `rchicmgk` propose des couleurs différentes pour les identifier. L'utilisateur peut disposer les valeurs comme il le souhaite. La figure 3.1 représente un exemple de graphes implicatifs

Figure 3.1 – Un exemple de graphes implicatifs à une certaine base de données

mené à une certaine base de données, aux seuils $\alpha = 1\%$ (arcs rouges), 5% (arcs verts), 10% (arc bleu) et 15% (arcs bleux ciels). Comme le nombre de règles peut être important, l'utilisateur a la

2. http ://cran.r-project.org/

possibilité de sélectionner uniquement certaines variables lui semblent utiles pour son interprétation. Dans ce cas, il peut supprimer temporairement les variables désirées grâce à une boîte de dialogue prévu à cela. Ensuite, `rchicmgk` met à jour à nouveau l'ensemble de graphes, sans faire de calculs, car il les calcule une fois pour toute au début de chaque nouveau graphe, et les mémorise. Autrement dit, même si l'utilisateur sélectionne ou désélectionne certains motifs, puis change le seuil d'affichage des règles, le package `rchicmgk` met à jour les graphes sans aucun calcul supplémentaire. Cela permet à l'utilisateur de mettre en évidence les caractéristiques importantes de ses données.

Il est possible de sauvegarder l'état d'un graphe, i.e. la disposition des motifs, les seuils d'implication, la sélection ou non de chaque motif. Ainsi, l'utilisateur peut reprendre un graphe qu'il avait organisé soigneusement lors d'une précédente session. De plus, il est possible de sauvegarder plusieurs états sur le même graphe, et ainsi mettre en évidence différentes parties du graphe.

Arbre hiérarchique orienté. L'arbre hiérarchique orienté, tout comme le graphe implicatif, s'obtient aussi à partir d'un ensemble des règles d'association valides selon les paramètres choisis par l'utilisateur. Cet arbre peut s'apparenter à une méthode de classification orientée. Comme nous l'avons formalisé, une règle d'association est aussi appelée *classe*, elle agrège alors deux variables dans sa forme la plus simple. À chaque niveau de la classification, le package `rchicmgk` choisit la meilleure classe (i.e., classe qui possède la plus grande cohésion). Ensuite, à chaque étape, le package `rchicmgk` calcule un ensemble de nouvelles classes à partir des classes présentes dans la hiérarchie. Pour créer une nouvelle classe, on agrège une classe existante avec soit une variable qui n'a pas été agrégée pour l'instant, soit avec une autre classe de la hiérarchie. Néanmoins, chaque couple de variables lors de l'agrégation de deux classes doit avoir une cohésion valide. Par exemple, la formation de la classe $((a, b), c)$ nécessite que les classes (a, c) et (b, c) soient valides. La classe $((a, b), c)$ représente la règle $(a \to b) \to c$ telle que la classe (a, b) soit cohésive et que celle-ci soit cohésive à c. L'exemple de la figure 3.2 ci-après représente la hiérarchie orientée obtenue avec le même jeu de données que le graphe implicatif de la figureé 3.1 ci-dessus. Au

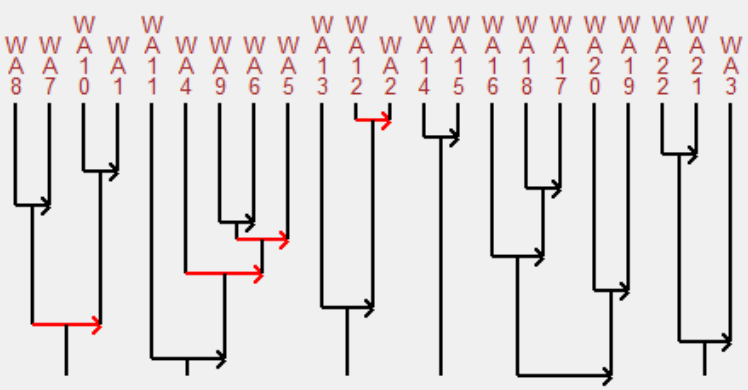

Figure 3.2 – Un exemple d'arbres hiérarchiques orientés

premier niveau de la hiérarchie, on remarque que la classe (WA12,WA2) est créée. Elle représente la règle d'association $WA12 \to WA2$ avec une cohésion plus forte que tous les autres couples. Ce premier niveau de la hiérarchie est d'ailleurs significatif comme l'indique la flèche rouge. Plus loin de cette hiéarchie, la classe $(WA13, (WA12, WA2))$ est formée, et représente la règle d'association $WA13 \to (WA12 \to WA2)$, c'est-à-dire $WA13 \to WA12 \wedge WA2$. Par construction, ce processus est récursif pour chaque communauté et s'arrête dès que la cohésion des classes candidates est faible.

3.6 Evaluation expérimentale

Dans cette section, nous évaluons nos algorithmes ImGraph et CAHI pour les deux types de représentation graphique que nous proposons, comparés à ceux de Gras *et al.* [GRMG13] et menés sur quelques bases de données de la littérature, en commençant par décrire le protocole expérimental (sous-section 3.6.1), puis discutant les résultats obtenus de l'approche (sous-section 3.6.2).

3.6.1 Protocole expérimental

Nos algorithmes ImGraph et CAHI ont été implémentés sous le package `rchicmgk`, alors que [GRMG13] sous l'outil `rchic` [CG15]. L'outil `rchic` est une bibliothèque implémentée sous R qui intègre aussi le graphe implicatif et l'arbre hiérarchique, basés sur l'intensité d'implication φ de Gras. L'objectif de nos expériences est de quantifier les bénéfices apportés par nos algorithmes comparés à ceux de [GRMG13] tant sur le plan de partitionnement que des performances.

Sur le plan de partitionnement d'arcs, nous avons sélectionné deux petites bases (moins denses et ni volumineuses), à savoir `iris` et `car` de l'UCI[3], qui décrivent respectivement de propriétés des plantes et automobiles dont les caractéristiques sont données dans le tableau 3.2. Précisément, ce

Table 3.2 – Caractéristiques des bases `iris` et `car`

Database	$\lvert\mathcal{T}\rvert$	$\lvert\mathcal{I}\rvert$	Type de données	Taille	cl
iris	150	15	real dataset	2250	3
car	1081	19	real dataset	20539	2

tableau 3.2 résume le nombre de transactions $|\mathcal{T}|$, le nombre de motifs $|\mathcal{I}|$, la taille de données (i.e., $|T| \times |\mathcal{I}|$), et cl le nombre de classes initiales. Le choix de ces bases est motivé par leur volumétrie très limitée pour bien comparer de manière visible les résultats avec notre compétiteur [GRMG13].

Au niveau des performances, nous avons sélectionné les mêmes bases de données que nous avons utilisées dans le chapitre 3 (cf. tableau 2.2), qui sont rappelées ci-après. Comme cela avait été décrit,

Table 2.2 - Caractéristiques des bases d'expérimentations

Database	$\lvert\mathcal{T}\rvert$	$\lvert\mathcal{I}\rvert$	$\lvert\widehat{\mathcal{T}}\rvert$	ρ	Type de données	Taille
Chess	3 196	75	37	49%	game steps	239 700
Connect	67 557	129	43	33%	game steps	8 714 853
T40I10D100K	100 000	1 000	40	4%	synthetic dataset	100 000 000
Pumsb	49 046	7 117	74	1%	census data	349 060 382

ce tableau 2.2 résume pour chaque base le nombre de transactions $|\mathcal{T}|$, le nombre de motifs $|\mathcal{I}|$, la taille moyenne de transactions $|\widehat{\mathcal{T}}|$, la densité $\rho = |\widehat{\mathcal{T}}|/|\mathcal{I}|$ de données, et la taille de données (i.e., $|T| \times |\mathcal{I}|$). Le choix de ces bases de données est motivé par leurs caractéristiques très denses et/ou volumineuses afin d'identifier les performances des algorithmes. Certaines bases comme `Chess` et `Connect` sont très denses, et d'autres comme `T40I10D100K` et `Pumsb` sont moins denses mais très volumineuses de tailles respectives 100 000 000 et 349 060 382. Tous les tests ont été effectués sur le même PC précédemment, en fixant le $minsup = 0.02$ et faisant varier le niveau de risque α.

3. http ://archive.ics.uci.edu/ml/

3.6.2 Résultats et discussions

Evaluation de partitionnement. Les figures 3.3 et 3.4 décrivent les graphes implicatifs pour ImGraph (haut) et [GRMG13] (bas) extraits dans `iris` et `car` à un seuil critique $\alpha = 0.5$ qui est un seuil à partir duquel une règle d'association commence à être significative. On remarque tout

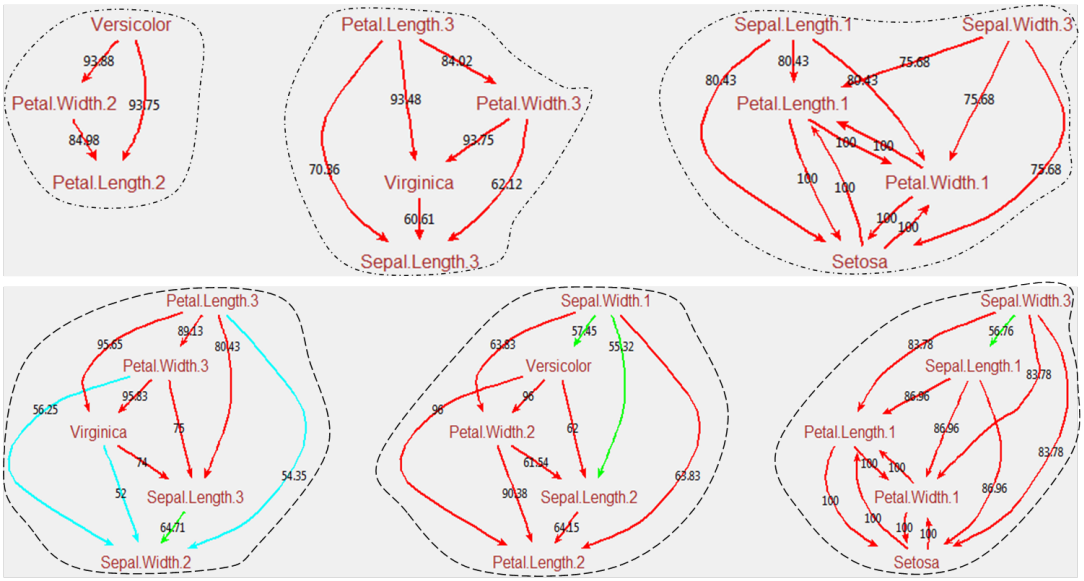

Figure 3.3 – Graphes implicatifs pour ImGraph (haut) et [GRMG13] (bas) avec `iris` et $\alpha = 0.5$.

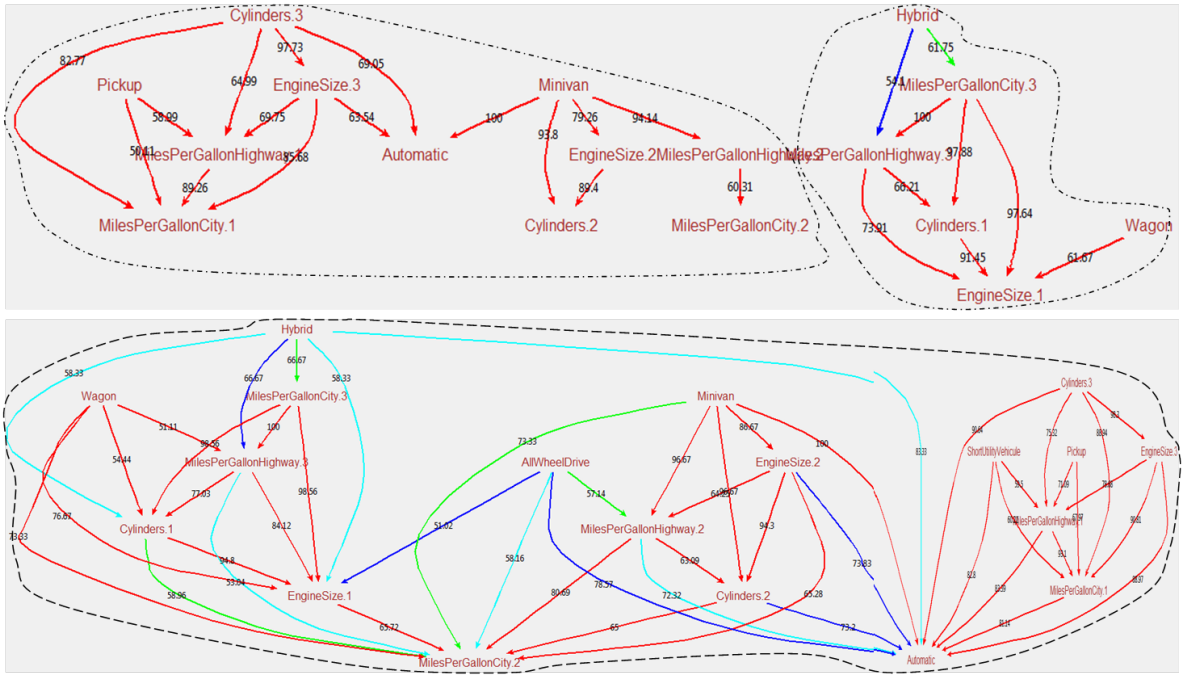

Figure 3.4 – Graphes implicatifs pour ImGraph (haut) et [GRMG13] (bas) avec `car` et $\alpha = 0.5$.

d'abord que quand α diminue (c'est-à-dire quand le critère de validation devient plus strict), les classes deviennent de plus en plus séparées, et de moins en moins denses. Sur la figure 3.3, nous observons que les deux approches restituent, pour chacune, 3 partitions. Elles fournissent donc un partitionnement qui suit les communautés existantes dans la base de données `iris`. Pour estimer le bénéfice de l'ImGraph versus [GRMG13], nous mesurons le nombre d'arcs échangés entre les partitions retenues. Nous observons que notre algorithme ImGraph réduit considérablement le nombre d'arcs pour chaque partition. Cela s'explique du fait que cet algorithme ImGraph utilise une technique de partitionnement très puissante permettant d'élaguer des arcs redondants que l'approche de Gras [GRMG13] ne fait pas. En effet, 21 arcs sont nécessaires pour analyser la base de données `iris` pour notre approche, tandis que 33 sont transmis avec l'approche de Gras [GRMG13], soit une baisse de 37% des règles restituées. Sur la figure 3.4, notre algorithme ImGraph engendre 2 classes, alors que [GRMG13] fait une seule classe couvrant la totalité des arcs significatifs du graphes, au seuil de risque $\alpha = 0.5$. Cette différence du nombre de classes valide notre intuition selon laquelle ImGraph présente un bon nombre de partitions. L'approche de Gras [GRMG13] quant à elle ne parvient pas à retrouver des classes de façon appropriée, et engendre souvent de classes très denses comme nous voyons sur la figure 3.4 (bas). De cette figure 3.4, l'algorithme ImGraph engendre 24 arcs pour la base `car`, alors que [GRMG13] fait 48, soit une baisse de 50%.

Au delà du partitionnement, les figures 3.3 et 3.4 mettent en évidence le pouvoir discriminant de l'ImGraph versus [GRMG13]. Cela est lié à la mesure de qualité utilisée. En effet, ImGraph est basé sur la mesure plus sélective *mgk*. [GRMG13] quant à lui est basé sur l'intensité d'implication φ qui est très collée à la valeur maximale 1, ce qui conduit à des classes trompeuses qui vont bruiter le graphe. Autrement dit, φ est toujours supérieure à *mgk*, pour les deux données. Cette différence se voit pour toutes les communautés des figures 3.3 et 3.4 lorsqu'on compare les poids des arcs induits par ces deux approches. Par exemple, avec les mêmes communautés précédemment considérées, $mgk(Versicolor, Petal.Width.2) = 0.93$ alors que $\varphi(Versicolor, Petal.Width.2) = 0.96$. Clairement, l'algorithme ImGraph est plus discriminant que l'approche de [GRMG13].

Nous continuons notre analyse sur l'arbre hiérarchique orienté. Nous comparons notre algorithme CAHI avec [GRMG13]. Les résultats, extraits avec les mêmes bases `iris` et `car` précédemment, sont représentés sur les figures 3.5 et 3.6 ci-après dont CAHI à gauche et [GRMG13] à droite. Les dégrés de cohésion sont résumés dans les tableaux 3.3 et 3.4 ci-dessous lesquels la cohésion plus

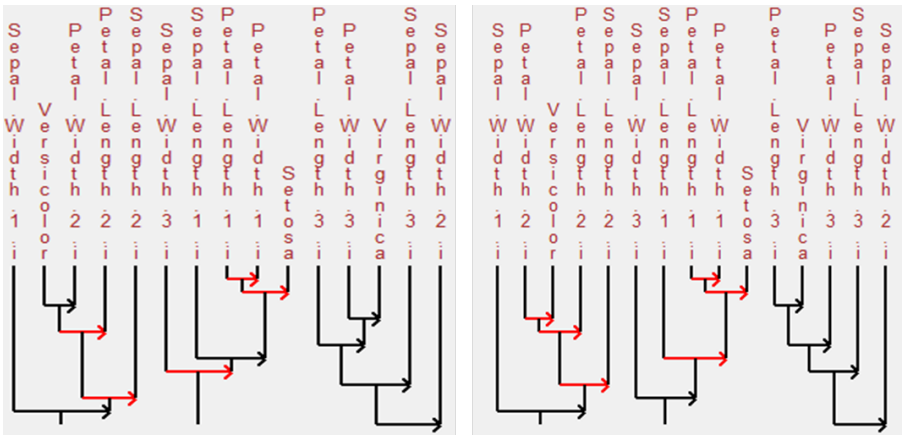

Figure 3.5 – Classification pour CAHI (gauche) et [GRMG13] (droite) avec `iris`

significative (resp. moins significative) correspond à un niveau d'arbre plus haut (resp. très bas).

Nous remarquons que quelques partitions engendrent des motifs qui n'apparaissent pas dans le

Niveau	1	2	3	4	5	6	7	8	9	10	11	12
Cohmgk	1	1	0.99	0.99	0.98	0.98	0.98	0.94	0.94	0.86	0.81	0.65
Cohφ	1	1	1	1	1	1	1	1	0.99	0.99	0.99	0.88

Table 3.3 – Cohésions par niveau d'arbres pour la base de données `iris`

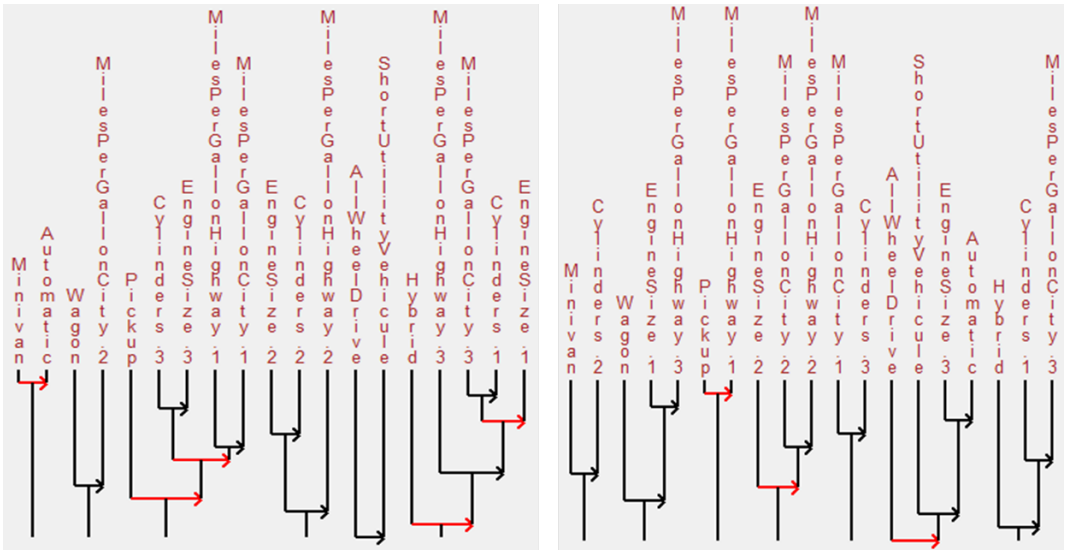

Figure 3.6 – Classification pour CAHI (gauche) et [GRMG13] (droite) avec `car`

Niveau	1	2	3	4	5	6	7	8	9	10	11	12	13
Cohmgk	0.99	0.99	0.99	0.99	0.99	0.98	0.96	0.91	0.90	0.87	0.80	0.76	0.73
Cohφ	1	1	1	1	1	1	1	1	0.99	0.99	0.90	0.79	-

Table 3.4 – Cohésions par niveau d'arbres pour la base de données `car`

graphe. Par exemple, si l'on considère la dernière partition (extrême gauche, fig. 3.5), des nouveaux motifs *Petal.Length.2* et *Sepal.Length.2* apparaissent, mais non dans le graphe. Cela s'explique du fait que l'arbre prend en compte les motifs faibles si la cohésion reste encore significative, mais il ne tient pas compte un seuil a priori. Dans la figure 3.5, ImGraph et [GRMG13] restituent, pour chacun, 3 partitions en 12 classes. Ils s'en distinguent cependant au niveau de sélection des variables intra-classes. En effet, toujours de la partition 3 de la figure 3.5, l'algorithme CAHI sélectionne, sur le premier niveau, le couple *(Petal.Width.3, Virginica)*, tandis que l'approche de Gras [GRMG13] considère la classe *(Petal.Length.3, Virginica)*. Dans la figure 3.6, CAHI engendre 6 partitions alors que [GRMG13] fait 7. Etant donné qu'un grand nombre de partitions conduit à un grand nombre de classes répliquées. Autrement dit, CAHI obtient encore de meilleures partitions versus [GRMG13].

En terme de discrimination (ou de précision), nous retrouvons les mêmes observations que précédemment : CAHI est meilleur que [GRMG13]. Cela est encore lié à la mesure de qualité utilisée. Comme nous l'avons signalé, CAHI est basé sur la mesure plus discriminante *cohmgk*, tandis que l'approche de Gras [GRMG13] sur *cohφ*. Nous observons, d'après les tableaux 3.3 et 3.4, que *cohφ* est toujours supérieure à *cohmgk* quel que soit le niveau d'arbres, et reste très collée à

la valeur maximale 1 même si le niveau d'arbres est déjà bas, ce qui conduit donc à des confusions de classes en terme de classification. Par exemple, $coh\varphi$ reste égal à 1 des niveaux 1 à 8, pour les 2 jeux de données. Cela s'explique du fait que $coh\varphi$ hérite les mêmes défauts que sa primitive φ.

Evaluation de performances Cette partie expérimentale a été menée de manière à (i) mésurer le comportement de nos mesures de qualité *mgk* et *cohmgk* comparé respectivement à celui de φ et $coh\varphi$ de Gras *et al.* [GRMG13], à (ii) mesurer la taille de classes pour différentes partitions induites par nos algorithmes ImGraph et CAHI versus [GRMG13], à (iii) mesurer la précision de classification par nos algorithmes ImGraph et CAHI par rapport à celle de [GRMG13], et enfin à (iv) évaluer les temps d'exécution de nos deux algorithmes comparés à ceux de [GRMG13].

(i) L'analyse comportementale de nos mesures de qualité s'effectue au niveau du nombre de classes dans différentes partitions induites par celles-ci versus ceux de [GRMG13]. Il existe bon nombre d'indices de validité d'une classification pour comparer les résultats d'un partitionnement. Nous pouvons citer, le facteur de mérite [MD09], la performance [For10], l'influence globale [GL10], et la mesure de surprise [AM11]. Cependant, la plus classique et sémantiquement appropriée à notre analyse étant la modularité selon Newman [NG04, New06]. Le principe consiste à rassembler les nœuds de telle manière qu'il y ait le maximum d'arêtes à l'intérieur d'une communauté et le minimum d'arêtes entre les communautés. Formellement, étant donné un graphe $G(V, E)$ d'une paire des ensembles de sommets V et d'arêtes E, la modularité d'une communauté C est définie :

$$q(C) = \frac{n_C}{|E|} - \left(\frac{d_C}{2|E|}\right)^2 \tag{3.9}$$

où n_C est le nombre d'arêtes dans C, d_C est la somme des degrés des nœuds appartenant à C et $|E| = \frac{1}{2}\sum_C d_C$ est le nombre total d'arêtes dans $G(V, E)$. Etant donnée $\Pi = \{C_1, \cdots, C_k\}$ une partition de $G(V, E)$, la modularité de Π est la somme des modularités de communautés, soit

$$Q(\Pi) = \sum_{C\in\Pi}\left(\frac{n_C}{|E|} - \left(\frac{d_C}{2|E|}\right)^2\right) \tag{3.10}$$

Dans le cas d'une partition ayant une seule communauté C couvrant la totalité du graphe, on a $n_C = |E|$ et $d_C = 2|E|$, soit une modularité nulle. Dans le meilleur des cas, la partition est formée de k-cliques non connectées entre elles. Dans ce cas, la modularité est égale à $\frac{1}{|E|} - \left(\frac{1}{2|E|}\right)^2$. Dans le cas plus défavorable d'une partition singleton avec un seul nœud par communauté, on a $n_C = 0$ pour toute communauté, soit une modularité négative. Pour un graphe de 2 nœuds reliés par une arête, chaque nœud étant dans une communauté, la modularité est minimale et vaut $-1/2$.

Il est possible de reformuler la modularité en prenant en compte la matrice d'adjacence Γ. Comme nous nous inscrivons dans un graphe pondéré, la formule (3.10) ci-dessus devient :

$$Q(\Pi) = \frac{1}{2|E|}\sum_{i,j}\left(\gamma_{ij} - \frac{k_i k_j}{2|E|}\right)\delta(C_i, C_j) \tag{3.11}$$

où γ_{ij} représente le poids des arcs entre i et j, $k_l = \sum_\ell \gamma_{l\ell}$ est la somme de poids des arcs attachés au nœud l, C_l est la communauté à laquelle appartient le nœud l, $2|E| = \sum_{i,j}\gamma_{ij}$, et la fonction de Kronecker $\delta(u, v)$ est égale à 1 si $u = v$ et 0 sinon. Plus précisément, la modularité de l'équation (3.11) est la somme pondérée pour toutes les communautés de la différence entre les arêtes observées à l'intérieur de la communauté (terme γ_{ij}) et la probabilité de ces arêtes (terme $\frac{k_i k_j}{2|E|}$).

A cette fin, nous avons utilisé les données du tableau 2.2, et considéré 250 partitions produites par chacune des mesures comparatives. Les résultats sont reportés dans la figure 3.7 ci-après. La

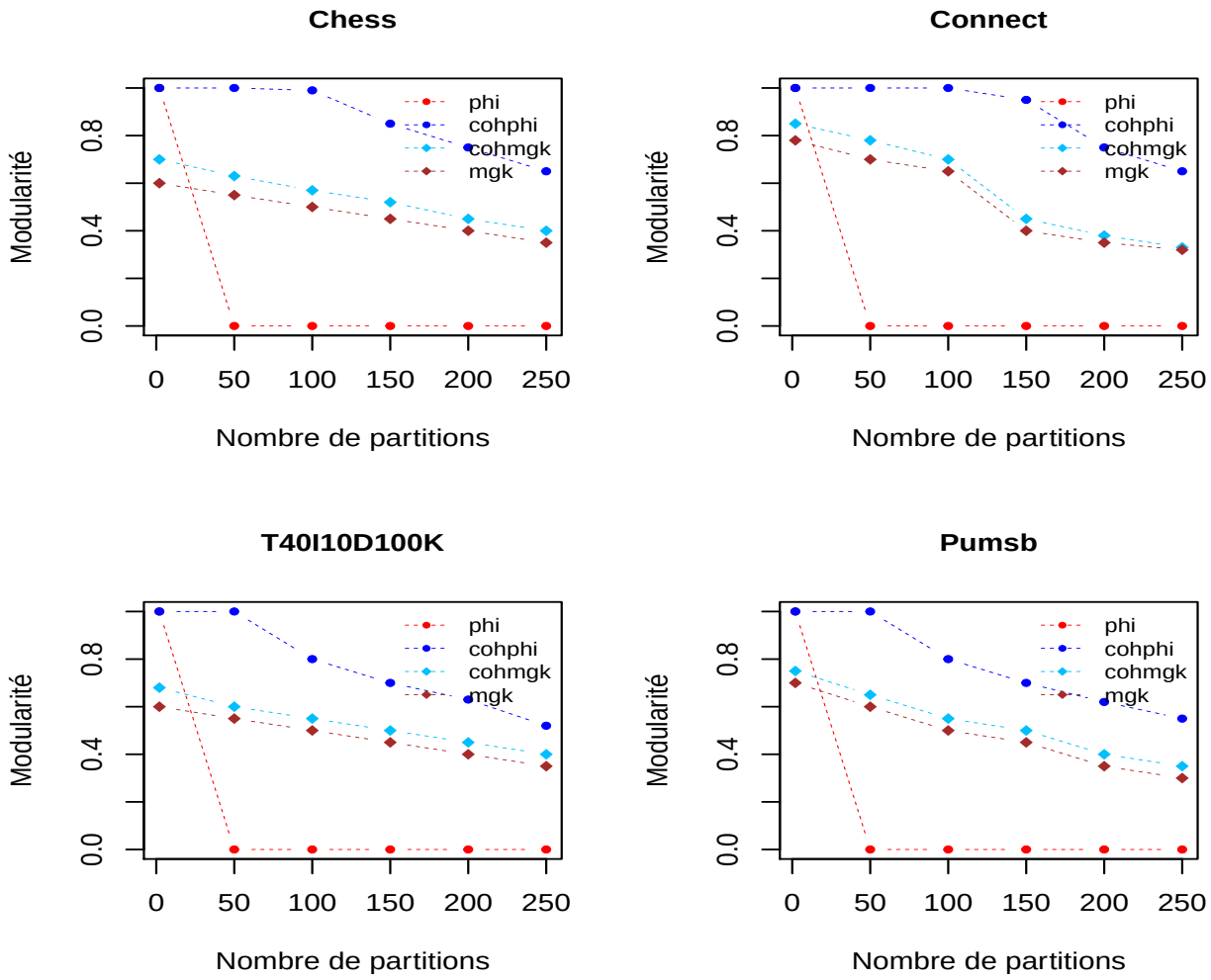

Figure 3.7 – Modularité en fonction du nombre de classes produites par *mgk*, *cohmgk*, φ et *coh*φ

modularité montre une forte dépendance avec le nombre de classes dans la partition, quelles que soient les données. De façon globale, la modularité présente des valeurs quasiment différentes pour ces indices de qualité. Cependant, nous observons que les valeurs associées à φ et *coh*φ sont les plus élevées pour certaines partitions. Ceci est attendu, car elles présentent de très grosses classes, et donc le nombre de règles intra-classes est très important. Cela s'explique du fait que φ et *coh*φ ne sont pas conçus pour l'élagage des règles redondantes, et qu'ils sont moins discriminants. Au delà des valeurs élevées, la mesure φ présente de très faibles variations, et rejoingne rapidement vers zéro. Cela conduit à une partition d'une seule classe couvrant la totalité du graphe. La mesure *coh*φ, quant à elle, présente un nombre très important de classes de grosses tailles. Nos méthodes de partitionnement, basées sur nos mesures de qualité *mgk* et *cohmgk*, présentent un bon équilibre

de point de vue structure des classes (i.e., absence de très grosses classes et faibles proportion de classes de petite taille). Cela montre que notre stratégie de partitionnement, capable de discriminer une classe parmi d'autres et d'élaguer les redonndances, dépasse comme attendue la méthode de Gras *et al.* [GRMG13]. D'autre part, cette expérience illustre implicitement le bénéfice de notre approche pour la génération de règles d'association négatives que [GRMG13] ne peut pas traiter. De façon implicite, les règles négatives obtenues ne font pas baisser la qualité de notre classification.

(*ii*) Sur l'évaluation de cardinalité de classes induites par nos algorithmes ImGraph et CAHI versus [GRMG13], nous considérons 250 classes. La figure 3.8 reporte les résultats menés sur les mêmes bases de données du tableau 2.2. Nous retrouvons les mêmes remarques que l'expérience

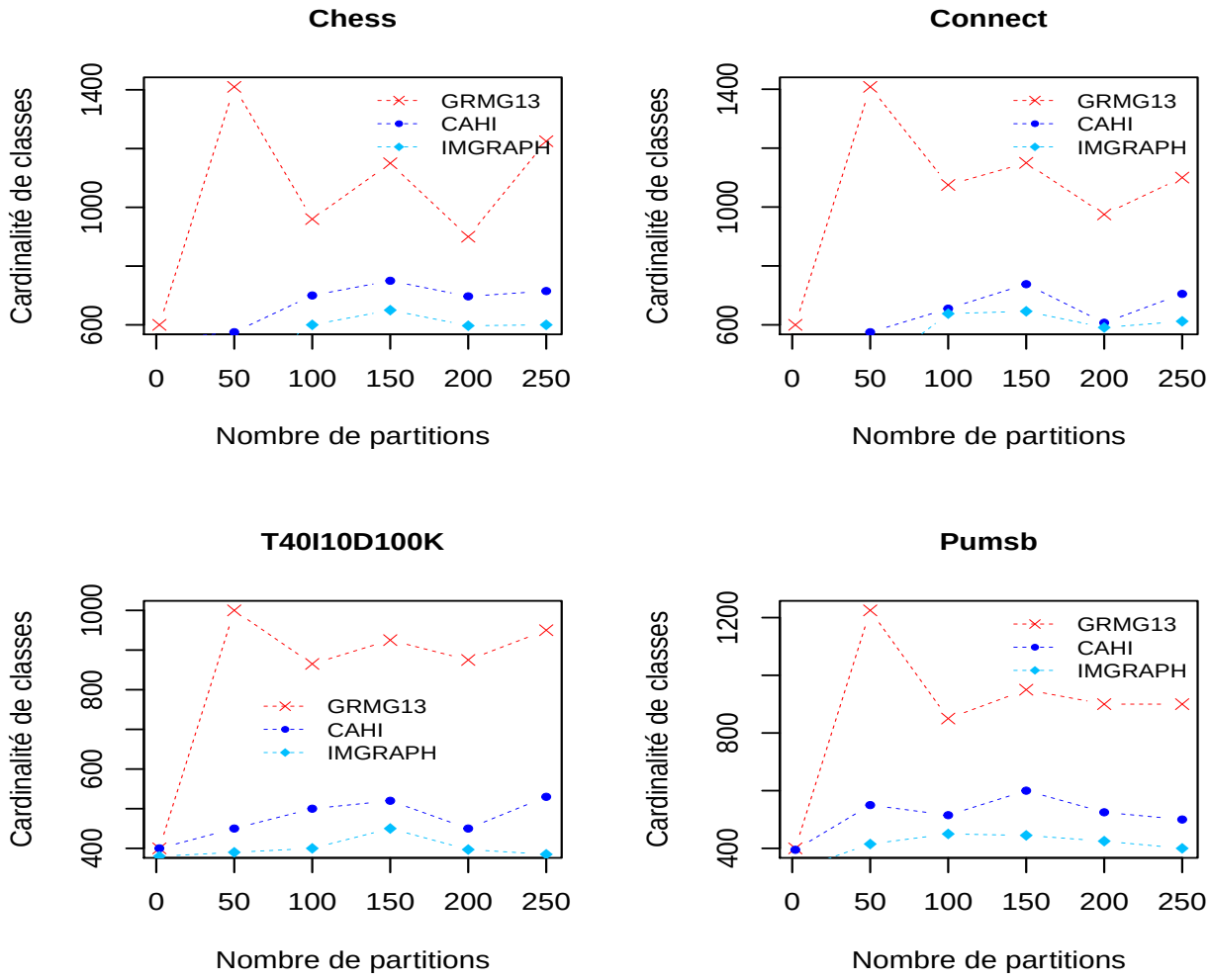

Figure 3.8 – Cardinalité de classes pour ImGraph et CAHI versus [GRMG13]

précédente. Nos deux algorithmes (ImGraph & CAHI) obtiennent de façon globale la meilleure classification. Ils ne produisent pas de classes de très grande taille, ni de petite taille. L'approche [GRMG13] obtient quant à elle de classes de très grosse taille pour toutes les bases de données, donc

d'hétérogénéité à l'intérieur des classes. Cela s'explique du fait qu'elle ne propose aucune technique d'élagage des règles d'association redondantes. Cet écart est très visible pour les jeux de données denses `Chess` et `Connect` avec la classe 50. Dans ce cas, CAHI (resp. ImGraph) restitue environ 500 (resp. 400) règles contre 1400 pour [GRMG13], soit une baisse de 64.29% (resp. 71.43%).

(*iii*) Nous procédons dans ce qui suit à l'évaluation de précision de classification pour nos deux algorithmes ImGraph et CAHI. Nous avons utilisé la courbe de précision moyenne assortie des intervalles de confiance à 95% pour quelques partitions induites par nos algorithmes ImGraph et CAHI, puis par [GRMG13]. Les résultats menés sur les deux bases de données plus denses (`Chess` et `Connect`) du tableau 2.2 sont rapportés dans la figure 3.9 ci-après. De façon globale, nos algorithmes

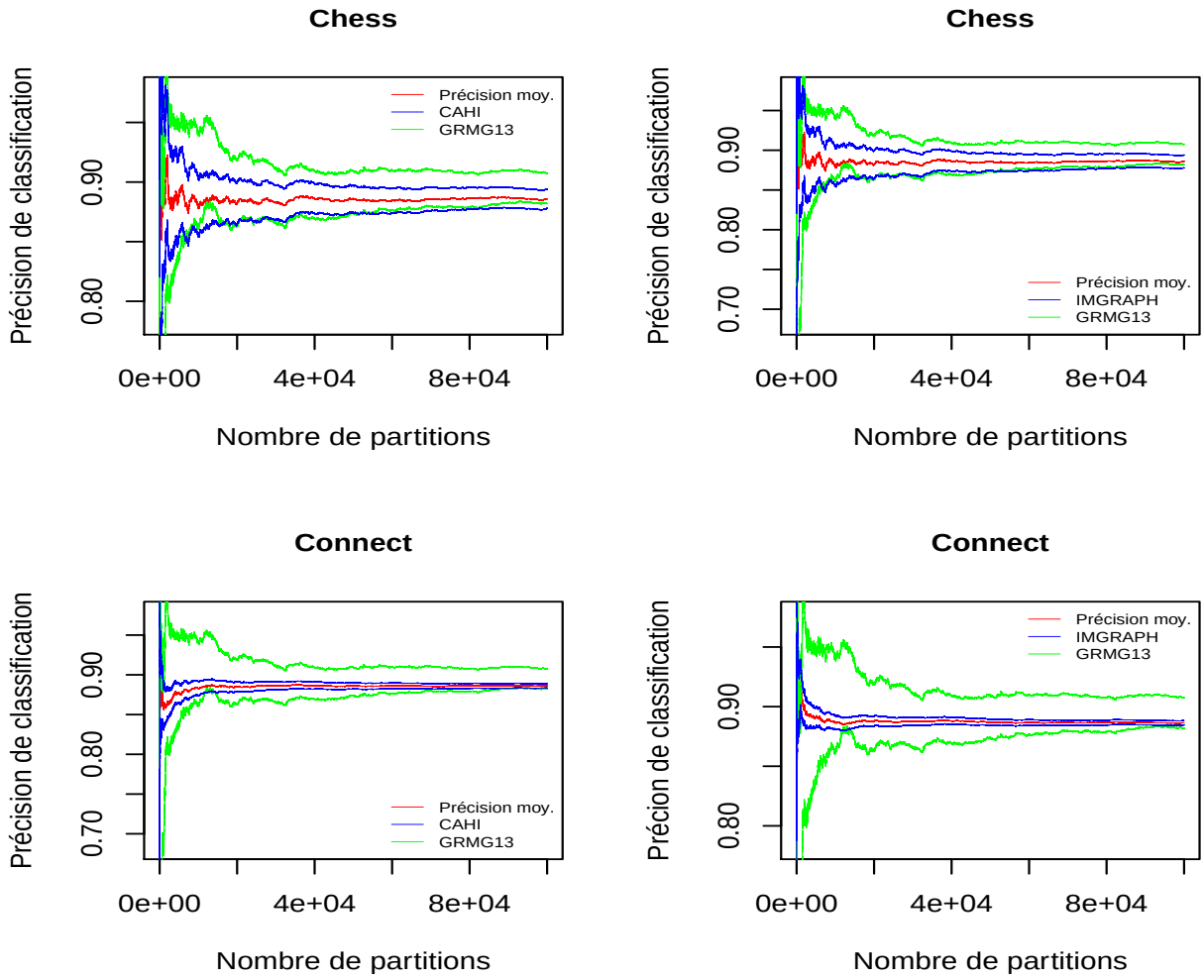

Figure 3.9 – Précision de classification selon ImGraph et CAHI *vs* [GRMG13]

de partitionnement affichent une meilleure précision moyenne par rapport à l'approche de Gras [GRMG13]. Comme nous voyons sur cette figure 3.9, les courbes de précision de nos méthodes (bleu) convergent beaucoup plus rapide vers la précision moyenne (rouge), et les intervalles de

confiance restent plus réduits et plus stables par rapport à ceux de [GRMG13] (courbes vertes), pour les deux données, ce qui valide donc nos modèles de classification. Pour la donnée `Chess`, nos méthodes engendrent cependant un pic de précision assez haut (en début de simulations) que celui de [GRMG13], mais converge rapidement de façon monotone vers le score moyen de classification. Cela s'explique du fait que la base de données `Chess` est trop dense pouvant contenir plusieurs motifs fréquents (donc, plusieurs règles valides). Ce résultat reste néanmoins prometteur, car nous avons considéré deux classes des règles (positives et négatives) à la fois, alors que [GRMG13] n'étudie qu'un seul type des règles (i.e., règles positives) qui est beaucoup plus moins subtile.

(*iv*) Nous évaluons maintenant les temps de calcul de nos algorithmes menés à des bases de données du tableau 2.2, et comparés à [GRMG13] en fonction du nombre de partitions. Pour la génération, nous considérons aussi 250 partitions. Nous constatons que les temps d'exécution aug-

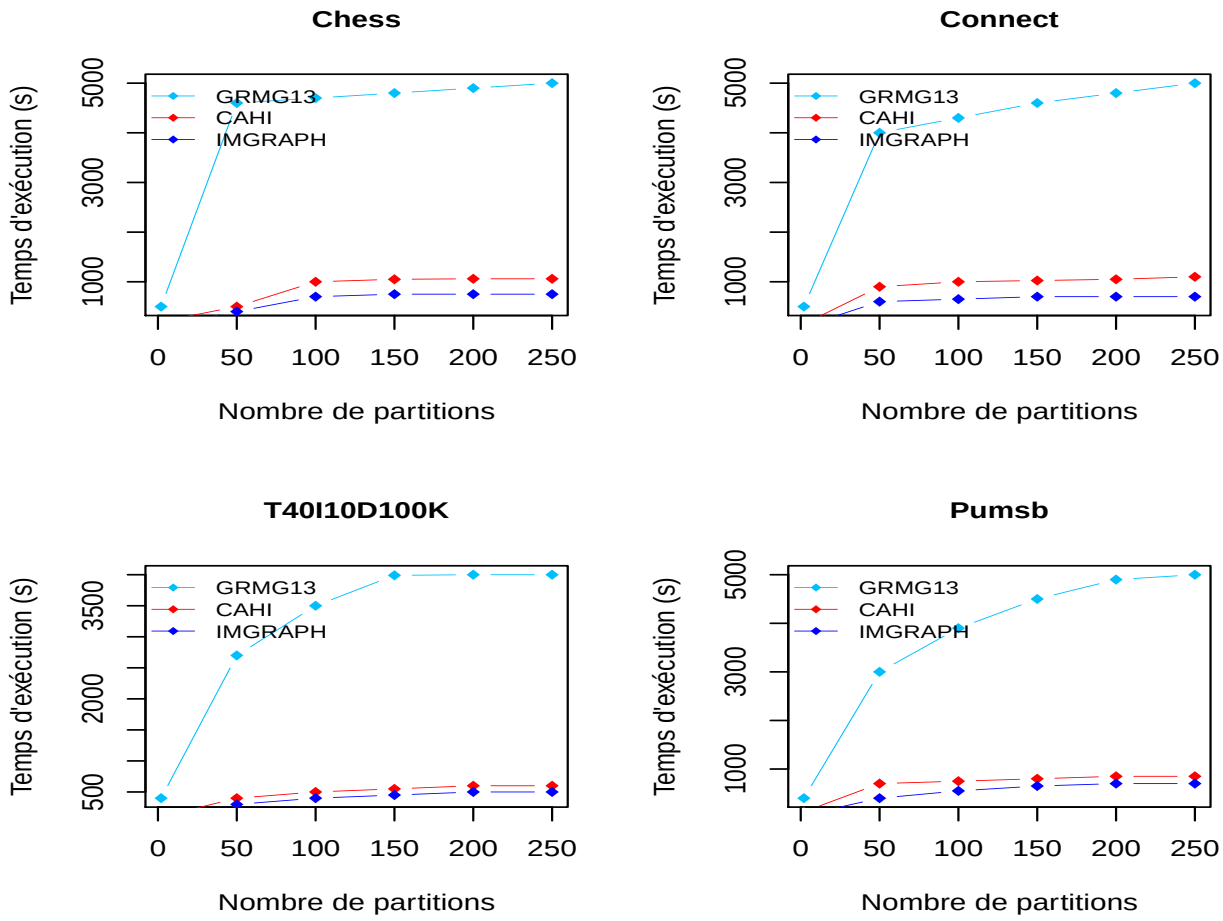

Figure 3.10 – Temps d'exécution en fonction du nombre de partitions pour ImGraph et CAHI *vs* [GRMG13]

mentent au fur et à mesure que le nombre de partitions augmentent. De façon globale, nous voyons que nos algorithmes (CAHI & ImGraph) sont quasiment similaires, et fournissent de meilleurs temps d'exécution par rapport à [GRMG13] au cours du calcul de graphes et d'arbres. Bien qu'il

y ait plus de règles d'association étudiées pour nos deux modèles CAHI et ImGraph que pour celles de [GRMG13], leurs temps d'exécution en fonction du nombre de partitions restent encore meilleurs. Il n'est pas surprenant que CAHI et ImGraph obtiennent les meilleurs résultats car ils ont été conçus particulièrement pour améliorer la qualité des graphes et des arbre hiérarchiques dans $\mathcal{D}$. Cela peut être expliqué en deux raisons principales. La première est dû au fait que nos algorithmes utilisent une technique *reduce-acces-database* permettant la réduction d'accès répétitif aux données pour l'extraction des motifs fréquents (cf. chapitre 1). A cet effet, une donnée d'expérimentation n'est parcourue qu'en partie, ce qui réduit considérablement l'espace de recherche. La deuxième s'explique du fait qu'ils permettent d'élaguer les redondances dans $\mathcal{D}$, ce qui réduit aussi l'espace de recherche de façon considérable. D'après la figure 3.10 (pour toutes les données), l'approche [GRMG13] obtient les moins bonnes performances en raison de l'absence d'une technique permettant l'extraction des motifs fréquents. Par conséquent, l'espace de recherche peut être parcourue dans sa globalité ($2^{|\mathcal{I}|}$ dans le pire des cas), ce qui augmente le temps de calcul pour la classification. Il n'y a pas non plus de stratégie d'élagage des règles d'association redondantes dans $\mathcal{D}$. En conséquence, l'espace de recherche des règles valides peut être aussi parcouru de façon exhaustive, ce qui augmente aussi le temps d'exécution. A titre d'exemple, si l'on considère la donnée dense `Chess` avec la partition 50, ImGraph (resp. CAHI) fait environ 400 (resp. 500) secondes contre environ 4600 secondes pour [GRMG13], soit 11 (resp. 9) fois plus rapide que [GRMG13].

Conclusion et Perspectives

Dans cette partie, je présente un bilan des principales contributions, et donne pour chacune un bref panorama de mes perspectives de recherche.

Bilan des contributions

Nos principales contributions peuvent être ventilées en quatre volets, qui s'articulent naturellement selon l'organisation de ce manuscrit : l'extraction des motifs fréquents, la génération de l'ensemble des meilleures règles d'association, la représentation de cet ensemble des meilleures règles d'association valides par des graphe et arbre, ainsi que la conception du package `rchicmgk`.

Dans le chapitre 1, j'ai présenté mes travaux de recherche et d'encadrement de la recherche autour de l'extraction des motifs fréquents dans une base de données. Nous y avons proposé une nouvelle approche (formelle) permettant l'extraction simultanée des motifs fermés fréquents, motifs maximaux fréquents, et motifs générateurs minimaux associés, sur lesquels sont proposées de nouvelles techniques de comptage des supports *reduce-acces-database*, et d'élagage pour l'espace de recherche des motifs fréquents [BT17, BT17, BT18, BT20c]. Une deuxième contribution consiste en la définition d'un nouvel algorithme autonome, appelé CMG (*Closed-Maximal-Generator*), qui consiste à implémenter automatique cette nouvelle approche précédemment développée [BT21a, BT21b]. L'orginalité de notre approche réside non seulement à la réduction de l'espace de recherche des motifs fréquents mais également à la réduction du temps d'exécution.

Dans le chapitre 2, j'ai présenté mes travaux portant sur la génération des meilleures règles d'association positives et négatives. Nous avons proposé une nouvelle mesure statistique plus sélective, notée *mgk*, capable de générer efficacement les meilleures règles d'association positives et négatives [BT20a, BT20e]. Nous avons également proposé une nouvelle stratégie permettant de restreindre l'espace de recherche des meilleures règles d'association [BT19a, BT20e]. Nous avons développé une nouvelle méthode d'élimination des redondances, sur laquelle sont définies quatre nouvelles bases (équations (2.11), (2.12), (2.13) et (2.14)) des règles non-redondantes [BT19a, BT19b], ce qui conduit aussi à réduire l'espace de recherche. L'orginalité de nos modèles repose non seulement à la réduction de l'espace de recherche, mais également à la réduction du temps de traitement. Basées à ces formalisations, nous avons également développé un nouvel algorithme, bâtisé CONCISE,

faisant appel à deux algorithmes sécondaires CMG (algorithme 1) et CBNR (algorithme 4), et permet d'implémenter celles-ci et toutes les règles d'association dérivées [BT21a, BT21b].

Dans le chapitre 3, j'ai présenté mes travaux de recherche sur la représentation d'un ensemble des règles d'association valides par des graphes implicatifs et des arbres hiérarchiques. Nous avons conçu une nouvelle mesure de qualité *cohmgk* [BJT22a], extension de la mesure statistique *mgk*, permettant la construction d'arbres de classification hiérarchique. Les travaux sur les mesures de qualité nous ont amené à définir formellement c'est qu'un *graphe implicatif* (Définition 38) et un *arbre hiérarchique* (Définition 40). Nous avons proposé de nouvelles méthodes de partitionnement des graphes et arbres hiérarchiques dans un objectif de classification. L'originalité de notre méthode est d'exploiter non seulement la réduction de la taille de l'espace de recherche procurée par la construction de ces graphes et arbres, mais également la réduction du temps de calcul. Basées à ces formalisations, nous avons développé deux algorithmes successifs : ImGraph pour le graphe implicatif [BCT21, BJT22a], et CAHI pour l'arbre hiérarchique [BJT22b]. Par la suite, en plus des algorithmes développés dans les chapitres 1 et 2, ces deux récents algorithmes (ImGraph et CAHI) serviront de base à l'élaboration d'un package `rchicmgk` offrant de nouvelles stratégies de construction des graphe implicatif et arbre hiérarchique orienté. Le package `rchicmgk` est une extension de notre outil *CHIC-*M_{GK} [Bem16], à l'instar des nouvelles mesures *mgk* et *cohmgk*. Ainsi, certain nombre de fonctions et de programmes pour le graphe (resp. arbre) ont été édités (resp. développés). Actuellement, le package `rchicmgk` inclut à la fois le graphe implicatif et l'arbre hiérachique [BCT21, BJT22a, BJT22b]. Il offre, entre autres, un moyen visuel et interactif très puissant aux experts en fouille de données, et permet d'aider ces experts dans leur prise de décision.

Perspectives

Parmi les axes de recherche abrodés, certains problèmes méritent d'être prolongés, et motivent encore des travaux de recherche et d'encadrement de recherche (thèse de doctorat ou mémoire de masters) actuels et à venir. Sans trop revenir en détail sur les problématiques signalées dans différents chapitres, j'aimerais conclure en généralisant ces perspectives selon les 4 axes ci-après.

Extraction de motifs fréquents. Malgré leurs intérêts incontestables, nos contributions pour l'extraction des motifs fréquents restent ouverte à diverses perspectives. Une première perspective intéressante serait de poursuivre le passage à l'échelle de notre algorithme CMG [BT21a, BT21b] comparé aux existants sur les masses de données. Ce travail fait l'objet des travaux de recherche en cours. D'un point de vue aspect algorithmique, l'algorithme CMG est contraignant des boucles **for** imbriquées. De ce fait, l'optimisation de cet algorithme CMG est indispensable. Une stratégie de parcours en profondeur pourrait être une solution efficace. A présent, l'algorithme CMG est limité à des motifs classiques, son extension à des motifs généralisés laisse envisager des perspectives intéressantes. Ce travail fait l'objet d'un mémoire de master en cours, sous mon encadrement.

Génération de meilleures règles d'association. Nos résultats sont très encourageants dans le sens où nous avons pu dépasser certaines limites des approches existantes. Toutefois, ils lèvent également plusieurs questions que nous discutons dans ce qui suit. En terme d'expérience, une perspective utile serait d'utiliser les règles d'association négatives obtenues dans une tâche de classification en faisant montrer que ces règles négatives ne font pas baisser la performance de classification sur des masses de données. C'est un travail en cours. Notre algorithme CONCISE

[BT21a, BT21b] a été conçu pour l'extraction des meilleures règles d'association positives/négatives classiques. Nous souhaitons étendre son principe à d'autres types de règles d'association telles que les règles d'association temporelles. Ce travail fait partie d'une thèse en cours. Par ailleurs, il serait aussi intéressant d'étendre cet algorithme CONCISE pour les règles d'association généralisées (règles contenant de conjonction des motifs positifs et négatifs) sur lequel des nouvelles bases des règles d'association généralisées pourront être définies. Une fois l'algorithme établi, des analyses expérimentales sur des masses de données pourront être réalisées. Un cadre applicatif serait alors l'analyse de données épidémiologiques et/ou génomiques (intéraction protéine-protéine).

Représentation graphique d'un ensemble des règles d'association valides. Dans ce cadre, nous avons développé une méthode simultanée pour la compression et le partitionnement des graphe implicatif et arbre hiérarchique orienté d'un ensemble des règles d'association valides dans une base de données. Nous y avons développé deux algorithmes successifs ImGraph [BCT21, BJT22a] et CAHI [BJT22b] permettant la construction des graphe implicatif et arbre hiérarchique respectivement. Actuellement, notre approche de classification d'arbres hiérarchiques tente, de plus en plus, d'être tolérant aux règles faibles. Ce mécanisme permet d'engendrer des classes conséquentes lorsqu'on est surtout en présence d'une base de données dense. Ainsi, une perspective intéressante serait de proposer une approche qui prend en compte des seuils d'élagage appropriés. Une autre perspective pertinente serait d'intégrer un aspect algorithmique beaucoup plus efficace pour la réorganisation d'une matrice de cohésion (souvent vaste) après d'éventuelles combinaisons possibles des classes. L'idée consisterait à limiter l'espace de recherche depuis cette matrice de cohésion. Ce travail fait partie d'un mémoire de master en cours. Comme nous l'avons signalé, l'extraction des règles d'association généralisées reste encore dans l'ombre. Une perspective pertinente serait alors d'élargir nos algorithmes (ImGraph & CAHI) dans une tâche de classification basée sur ce type des règles d'association généralisées. Une fois les algorithmes établis, une perspective naturelle serait d'envisager des analyses expérimentales sur des données réelles ou synthétiques.

Outil informatique. Afin d'aider l'utilisateur à explorer les meilleures connaissances dans ses données, nous avons conçu et implémenté un package, `rchicmgk`, suivant une approche exploratoire et de classification. Cette version `rchicmgk` liée aux approches basées sur les mesures *mgk* et *cohmgk* étant très récentes. Ainsi, plusieurs aspects informatiques sont actuellement à l'étude :

- Par construction, la librairie `rchicmgk` implémente pour l'instant 2 volets graphiques, à savoir *graphe implicatif* et *arbre hiérarchique*, qui intègrent pour chacun des règles positives et négatives à la fois. Cependant, l'étiquetage de ces deux types de règles n'est pas encore disponible pour l'instant, mais pourrait très bien s'automatiser. Dans ce sens, une perspective intéressante serait alors d'intégrer un calcul informatique d'étiquetage de ces types des règles.
- Lorsqu'on est en présence d'un grand volume des règles valides, la taille des graphe et arbre dépasse parfois la dimension de l'écran, alors que ceux-ci sont sauvés pour l'instant à l'aide d'une capture d'écran. Une perspective intéressante serait alors d'intégrer une technique de calibrage par rapport à la surface appropriée. Il serait aussi intéressant d'intégrer une stratégie d'enregistrement sous des formats classiques comme exemple `pdf, bmp, png`, etc.
- Pour l'instant, le package `rchicmgk` ne concerne que des graphes et arbres des règles d'association classiques. Nous pourrons à l'avenir éditer un calcul informatique permettant la représentation par des graphes et arbres de l'ensemble des règles d'association généralisées.
- La version actuelle de `rchicmgk` est non encore disponible sur le site CRAN de R. Une version plus complète, est actuellement en préparation et sera prochainement déposée sur CRAN.

Annexe A

Curriculum vitae

Etat civil

Parfait Bemarisika,	né le 01/01/1977 à Mangindrano (Madagascar)
`Nationalité:`	Malagasy, 3 enfants
`Téléphone:`	+261 32 41 231 43
`E-mail:`	bemarisikap7@yahoo.fr
`Adresse professionnelle:`	Laboratoire de Mathématiques et Informatique ENSET, Université d'Antsiranana (Madagascar)

Cursus universitaires

2012-2016 | Thèse de Doctorat en Didactique des Mathématiques et de l'Informatique de l'Université d'Antananarivo (Madagascar),
Titre : **Extraction de règles d'association selon le couple support-M_{GK} : Graphes implicatifs et applications en didactique des mathématiques**, soutenue le 20 avril 2016, mention *Honorable*, devant le jury :

Pr. Judith RAZAFIMBELO (Univ. Antananarivo) :	Présidente
Pr. Dominique TOURNÈS (Univ. La Réunion) :	Rapporteur
Pr. Jean-Claude RÉGNIER (Univ. Lyon 2) :	Rapporteur
Pr. Victor HARISON (Univ. Antananarivo) :	Rapporteur
Pr. Jean Emile RAKOTOSON (Univ. Fianarantsoa) :	Examinateur
Dr. Daniel Rajaonasy FENO (Univ. Toamasina) :	Examinateur
Pr. Jean DIATTA (Univ. La Réunion) :	Co-directeur
Pr. André TOTOHASINA (Univ. Antsiranana) :	Directeur

2012-2013 | Master Mathématiques et Applications, spécialité Mathématiques Approfondies à finalité Recherche, Univ. Franche-Comté (France).
Cours suivis (extrait) : Calcul stoch., Chaînes de Markov, Calcul scientifique.

2008-2010	Master Mathématiques et Applications, spécialité Statistique et Econométrie, Université de Toulouse 1 (France), mention *Assez Bien*. Titre : *Modèles temporels à la prédiction d'une base de données*.
2004-2006	Diplômé de l'INFA (22e promotion), Institut National de Formation Administrative (INFA) d'Antananarivo (Madagascar), mention *Très Bien*.
1999-2004	CAPEN en Génie Maths & Info, ENSET-Univ. Antsiranana. Titre *Faisabilité de l'introduction des coniques et quadriques au Lycée*, mention *Bien*.

Déroulement de carrière

Depuis 2019	Maître de Conférences à l'Université d'Antsiranana (Madagascar) - Ecole Normale Supérieure pour l'Enseignement Technique (ENSET).
2016-2019	Assistant Docteur d'Enseignement Supérieur et de Recherche (ESR) à l'Université d'Antsiranana (Madagascar), permanent à l'ENSET.
2012-2015	Doctorant à l'Equipe d'accueil Education et Didactique des Mathématiques et de l'Informatique (EDMI) de l'ENSET - Université d'Antsiranana (Madagascar), en Collaboration avec Laboratoire d'Informatique et de Mathématiques (LIM), Université de La Réunion (France).
2011-2016	Assistant d'ESR à l'Université d'Antsiranana (Madagascar), ENSET.
2008-2011	Enseignant vacataire à l'Université d'Antsiranana - ENSET.

Responsabilités scientifiques

Depuis 2017	Responsable du Laboratoire de Recherche de l'équipe d'accueil Education et Didactique des Mathématiques et de l'Informatique (LREDMI) à l'ENSET, Université d'Antsiranana (Madagascar).
Depuis 2016	Co-organisateur du séminaire des Masters Modélisations Mathématiques de l'ENSET, Université d'Antsiranana (Madagascar).

2015-2022	Responsable Mention Education-Apprentissage-Didactique et Ingénierie en Maths & Info (EADIMI), ENSET - Université d'Antsiranana.
2012-2022	Membre du Conseil d'école de l'ENSET, Univ. Antsiranana (Madagascar)
Depuis 2012	Membre du Conseil scientifique, ENSET - Univ. Antsiranana (Madagascar)
Depuis 2011	Collège des Enseignants de l'ENSET, Univ. Antsiranana (Madagascar)

Responsabilités pédagogiques

Depuis 2016	ENSET (Université d'Antsiranana), Mention EADIMI : - Responsable des UE Algo. & Programmation, L2-MM (Modélisations Maths). - Responsable de l'UE Théorie de mesures et Sciences de données en L3-MM. - Responsable de l'UE Maths discrètes et Stat exploratoire en M1-MM. - Responsable des UE EDO & EDS en M2-MMS (MM Stochastiques). - Responsable des UE Modélisations Statistique & Stochastique en M2-MMS.
2012-2015	Responsable Licence 1re année de l'ENSET, Université d'Antsiranana.

Activités d'enseignement

Depuis 2016	Eadimi & Education-Apprentissage-Didactique et Ingénierie en Génie Civil (Eadgc) de l'ENSET, Université d'Antsiranana (Madagascar) : - Statistique descriptive, L1 ENSET, Etude théorique (ET)/TD (30h). - Initiation à l'Informatique, L1 ENSET, ET/TD/TP (30h). - Probabilités élémentaires, L2 Eadimi, ET/TD/TP sous R (45h). - Statistique inférentielle, L2 Eadimi, ET/TD/TP sous R (45h). - Initiation au Logiciel R, L2 Eadimi, ET/TP (30h). - Statistique mathématique, L3 Eadimi, ET/TD/TP sous R (45h). - Théorie de sondages, M1 Eadimi, ET/TD/TP sous R (30h). - Traitement de données massives, M1 Eadimi, ET/TD/TP sous R (30h). - Algorithmique de graphes, M1 Eadimi, ET/TD/TP sous R (30h). - Processus stochastiques discrets, M1 Eadimi, ET/TD/TP (45h). - Séries temporelles, M1 Eadimi, ET/TD/TP sous R (30h). - Méthode numérique stochastique, M1 Eadigc, ET/TD (30h). - Statistique non-paramétrique, M2 Eadimi, ET/TD/TP sous R (30h). - Procecus stochastiques continus, M2 Eadimi, ET/TD/TP sous R (45h).

2016-2019	Ecole Supérieure en Agronomie et Environnement, Univ. Antsiranana : - Mathématiques générales, Licence 1, ET/TD (45h). - Probabilités-Statistique, Licence 2, ET/TD (45h).
2016-2018	Faculté de Droit, Economie, Gestion et Sciences Politiques, Mention Sciences Economiques, Université d'Antsiranana (Madagascar) : - Statistique descriptive, Licence 1, ET/TD (45h). - Statistique appliquée en Economie, Licence 2, ET/TD (45h).
2015-2017	Faculté des Lettres et Sciences Humaines, Université d'Antsiranana - Statistique spatiale, L3 Géographie, ET/TD (30h).
2011-2016	ENSET, Mention Eadimi, Université d'Antsiranana (UNA) : - Statistique descriptive, L1 Tronc-commun, ET/TD (30h). - Initiation à l'Informatique, L1 Tronc-commun, ET/TD/TP (30h). - Probabilités élémentaires, L2 Eadimi, ET/TD/TP sous R (45h). - Statistique inférentielle, L2 Eadimi, ET/TD/TP sous R (45h). - Initiation au Logiciel R, L2 Eadimi, ET/TD/TP (30h).
2008-2018	Institut Supérieur en Administration d'Entreprise - UNA : - Mathématiques générales, Licence 1, ET/TD (50h). - Mathématiques appliquées, Licence 2, ET/TD (60h).
2008-2016	Institut Supérieur de Technologie (IST) d'Antsiranana : - Maths 1, L1 (Parcours Commerce et Banque), ET/TD (30h). - Stat. inférentielle, L2 (Parcours Commerce, Banque), ET/TD (30h). - Probabilités-Statistique, L3 (Parcours Finances), ET/TD (30h).

Encadrement de thèses de doctorat (En cours)

Depuis 2020	Co-encadrement avec André Totohasina (ENSET, Université d'Antsiranana, Madagascar) de la thèse de **T. Rabenantenaina**, Equipe d'accueil Mathématiques-Informatique et Applications (MIA), Université de Toamasina. Cette thèse aborde la question de *Stratégie d'estimation d'un processus temporel.* Elle a donné lieu à deux articles publiés : T. Rabenantenaina *et al.* [RBT20a, RBT20b].
Depuis 2021	Co-encadrement avec André Totohasina (ENSET, Univ. Antsiranana) de la thèse de **I.L.N. Iandriah**, Equipe d'accueil MIA. Cette thèse repose sur thème du *Contrôle optimal stochastique dirigé par un mouvement brownien fractionnaire.*

Encadrement de Masters

2021-2022 | Encadrement 11 Master's thesis Master 2 Indifférencié (M2I) Modélisations Mathématiques de l'ENSET, Université d'Antsiranana (Madagascar) :
- S.M.T. AHMAD, *Partitionnement des graphes et des arbres hiérarchiques.*
- A. Anchikidine, *Choix optimal du paramètre de lissage par moindres carrés.*
- R. Justin, *Extraction condensée des motifs fréquents d'une donnée binaire.*
- I-L. Rajaorimanana, *Choix optimal du nombre de classes par l'histogramme.*
- H.R. Randriamahavelona, *Choix du paramètre de lissage à fenêtre adaptative et illustrations à des exemples réels ou simulés.*
- J.M. Randriamanana, *Etudes des modèles SARMA/GARCH et applications.*
- L.W.F. Razafindravelo, *Choix de paramètre de lissage par la méthode de Parzen-Rosenblatt et illustrations à des exemples réels ou simulés.*
- K. Soanivoniaina, *Etude de l'algorithme K-means et applications.*
- G.A. Soavinkasina, *Etudes des modèles ARMA/GARCH et applications.*
- F. Tilahizandry, *Analyse géostatistique et illustrations à des exemples réels.*
- R. Toto, *Autour de la génération des bases des règles d'association.*

2019-2020 | Encadrement 11 Master's thesis M2I, Modélisations Mathématiques de l'ENSET, Université d'Antsiranana (Madagascar) :
- A. Attoumani, *Application bijective et ses applications.*
- Dolphe, *Prototype rchicmgk et applications en didactique de maths.*
- J.F. Honoré, *Complexité algorithmique et applications en secondaire.*
- C. Raharijaonina, *Etat de l'art sur le modèle GARCH*
- N.G.E. Rakotofiringa, *Prototype* `rchicmgk` *et applications sur l'apprentissage de la Statistique descriptive en terminale littéraire.*
- R.A. Randriampenomanana, *Tests d'adéquation par une loi de probabilités.*
- E. Randrianjara, *Méthode de Nadaraya-Watson et Applications empiriques.*
- I. Ravelondera, *Processus de Poisson et Applications empiriques.*
- A. Ravolahy, *Méthode de Parzen-Rosenblatt et Applications empiriques.*
- F. Tombomanana, *Elaboration des arbres implicatifs selon la mesure* M_{GK}.
- D. Tsirainana, *Fouille des motifs fréquents et Graphes implicatifs.*

2019-2020 | Encadrement 11 Master's thesis (M1 Modélisations Mathématiques) de l'ENSET, Université d'Antsiranana (Madagascar) :
- S.M.T. AHMAD, *Détermination de clusters cohésifs.*
- A. Anchikidine, *Simulation des processus markoviens sous R.*
- R. Justin, *Simulation des minimums locaux sous R.*
- I-L. Rajaorimanana, *Estimation par intervalle de confiance.*
- H.R. Randriamahavelona, *Robustesse des tests statistiques sous R.*
- J.M. Randriamanana, *Simulation du processus SARIMA sous R.*

- L.W.F. Razafindravelo, *Rapports de vraisemblance sous R.*
- K. Soanivoniaina, *Méthode d'Analyse en Composantes Principales.*
- G.A. Soavinkasina, *Simulation du modèle ARMA sous R.*
- F. Tilahizandry, *Simulation des méthodes de bootstrap sous R.*
- R. Toto, *Simulations des méthodes de bootstrap généralisé sous R.*

2018-2019 | Encadrement 7 Master's thesis (M2I Modélisations Mathématiques) de l'ENSET, Université d'Antsiranana (Madagascar) :
- F. Damien, *Modèles de durée de vie censurée et Applications.*
- J. Mitantsoa, *Estimation de la fonction de répartition et Applications.*
- D. Rajerisolo, *Une élaboration d'un graphe implicatif selon M_{GK}.*
- A. Randrianantenaina, *Analyse de variance à un et à deux facteurs.*
- N.B. Ratsaraefadahy, *Estimation de la fonction de densité par noyau.*
- R. Randrianasolo, *Représentation Concise des règles d'association.*
- V. Razafimanana, *Etudes des Méthodes de sondage et Applications.*

2018-2019 | Encadrement 10 Master's thesis (M1 Modélisations Mathématiques) de l'ENSET, Université d'Antsiranana (Madagascar) :
- Dolphe, *Etudes des modèles ARIMA et SARIMA et Applications.*
- J.F. Honoré, *Tests d'indépendance de deux caractères et Applications.*
- C. Raharijaonina, *Critères d'informations d'un processus temporel.*
- N.G.E. Rakotofiringa, *Test de la racine unitaire et Applications..*
- R.A. Randriampenomanana, *Sur quelques tests paramétriques.*
- E. Randrianjara, *Etudes du modèle ARMA et Applications.*
- I. Ravelondera, *Au tour des tests non-paramétriques et Applications.*
- A. Ravolahy, *Estimateur de Parzen-Rosenblatt et Applications.*
- F. Tombomanana, *Tests d'adéquation du modèle ARIMA faible.*
- D. Tsirainana, *Estimation de la fonction de densité et Applications.*

2017-2018 | Encadrement 5 Master's thesis M2I Modélisations Mathématiques de l'ENSET, Université d'Antsiranana (Madagascar) :
- A.A. Beanjara, *Méta-modélisation numérique et Applications.*
- A. Felix, *Etudes des méthodes de Monte-Carlo et Applications.*
- A. Rafalimanana, *Extraction de Règles d'Association non redondantes.*
- A. Rasoanantenaina, *Stratégies d'échantillonnage et Applications.*
- T. Rabenantenaina, *Stratégie d'estimation d'un processus temporel.*

2017-2018 | Encadrement 8 Master's thesis (M1 Modélisations Mathématiques) de l'ENSET, Université d'Antsiranana (Madagascar) :
- A. Randrianantenaina, *Simulation des risques actuariels sous R.*
- D. Rajerisolo, *Etudes des processus de Poisson temporels.*

	- F. Damien, *Etudes des modèles de durée de vie censurée.* - J. Mitantsoa, *Etudes des modèles markoviens.* - N.B. Ratsaraefadahy, *Choix optimal du paramètre de lissage.* - R. Randrianasolo, *Simulation des valeurs extrêmes sous R.* - V. Razafimanana, *Simulation des méthodes de Monte-Carlo sous R.* - C. Razafindrakalo, *Traitement de données comportementales sous R.*
2016-2017	Encadrement 3 Master's thesis M2I Modélisations Mathématiques de l'ENSET, Université d'Antsiranana (Madagascar) : - D. Madiohavana, *Extraction de règles d'association généralisées.* - V. Rajoro, *Bandes de confiances pour la méthode de Monte Carlo.* - B. Tsiatandra, *Modélisation probabiliste d'un réseau de télécom.*
2016-2017	Encadrement 5 Master's thesis (M1 Modélisations Mathématiques) de l'ENSET, Université d'Antsiranana (Madagascar) : - A.A. Beanjara, *Méta-modélisation numérique et Applications.* - A. Rafalimanana, *Extraction de Règles d'Association non redondantes.* - T. Rabenantenaina, *Stratégie d'estimation d'un processus temporel.* - A. Rasoanantenaina, *Stratégies d'échantillonnage et Applications.* - J.N. Todisoa, *Utilisation de R à la modélisation statistique des risques.*
2015-2016	Co-encadrement avec A. Totohasina, 2 Master's thesis M2I Modélisations mathématiques de l'ENSET, Université d'Antsiranana (Madagascar) : - Andriamanantena, *Optimisation stochastique appliquée à la gestion de stock.* - Soavavinirina, *Optimisation stoch. appliquée au problème de plus court chemin.*
2015-2016	Encadrement 4 Master's thesis (M1 Modélisations Mathématiques) de l'ENSET, Université d'Antsiranana (Madagascar) : - A. Felix, *Résolution numérique des EDS.* - D. Madiohavana, *Méthodes d'échantillonnage et Applications.* - V. Rajoro, *Méthodes de Monte Carlo et Applications en finances.* - O. Tongalaza, *Etudes d'un processus temporel et Applications.*

Activités de recherche

Elles sont à cheval entre **Sciences de données** & **Informatique**, qui s'articulent autour des :
- Fouille de règles d'association en grande dimension ;
- Classification de données en grande dimension ;
- Statistiques non-paramétriques ;
- Séries temporelles ;
- Processus stochastiques.

Séminaires scientifiques

1. P. Bemarisika & F. Damien, *Modèles de durée de vie censurée à travers des exemples pratiques*, Séminaire de Masters du Laboratoire des Mathématiques et Informatique de l'ENSET, *4e Edition*, Université d'Antsiranana, janvier 2021.
2. P. Bemarisika & J. Mitantsoa, *Une estimation récursive de la fonction de répartition*, Séminaire de Masters du Laboratoire des Mathématiques et Informatique de l'ENSET, *4e Edition*, Université d'Antsiranana, janvier 2021.
3. P. Bemarisika & D. Rajerisolo, *Une construction d'un graphe implicatif selon M_{GK}*, Séminaire de Masters du Laboratoire des Mathématiques et Informatique de l'ENSET, *4e Edition*, Université d'Antsiranana, janvier 2021.
4. P. Bemarisika & A. Randrianantenaina, *Une modélisation d'analyse de variance à un et à deux facteurs*, Séminaire de Masters du Laboratoire des Mathématiques et Informatique de l'ENSET, *4e Edition*, Univ. Antsiranana, janvier 2021.
5. P. Bemarisika & R. Randrianasolo, *Une représentation concise de règles d'association*, Séminaire de Masters du Laboratoire des Mathématiques et Informatique de l'ENSET, *4e Edition*, Université d'Antsiranana, janvier 2021.
6. P. Bemarisika & N.B. Ratsaraefadahy, *Un choix optimal du paramètre de lissage*, Séminaire de Masters du Laboratoire des Mathématiques et Informatique de l'ENSET, *4e Edition*, Université d'Antsiranana, janvier 2021.
7. P. Bemarisika & V. Razafimanana, *Une synthèse des méthodes de sondage et Applications*, Séminaire de Masters du Laboratoire des Mathématiques et Informatique de l'ENSET, *4e Edition*, Université d'Antsiranana, janvier 2021.
8. P. Bemarisika, T. Rabenantenaina & A. Totohasina, *Une approche optimale d'estimation de séries temporelles*, Journée de Recherche des ISTs, *5e Edition*, Institut Supérieur de Technologie (IST) d'Antsiranana, 02-04 décembre 2020.
9. P. Bemarisika & A.A. Beanjara, *Une stratégie de méta-modélisation numérique de données*, Séminaire de Masters du Laboratoire des Mathématiques et Informatique de l'ENSET, *3e Edition*, Université d'Antsiranana, 2019.
10. P. Bemarisika & A. Felix, *Une simulation des méthodes de Monte-Carlo sous R et Applications*, Séminaire de Masters du Laboratoire des Mathématiques et Informatique de l'ENSET, *3e Edition*, Université d'Antsiranana, 2019.
11. P. Bemarisika & A. Rafalimanana, *Une méthode d'extraction de règles d'association non redondantes*, Séminaire de Masters du Laboratoire des Mathématiques et Informatique de l'ENSET, *3e Edition*, Université d'Antsiranana, 2019.
12. P. Bemarisika & A. Rasoanantenaina, *Une stratégie d'échantillonnage et Simulation sous R*, Séminaire de Masters du Laboratoire des Mathématiques et Informatique de l'ENSET, *3e Edition*, Université d'Antsiranana, 2019.
13. P. Bemarisika & T. Rabenantenaina, *Stratégie d'estimation d'un processus temporel*, Séminaire de Masters du Laboratoire des Mathématiques et Informatique de l'ENSET, *3e Edition*, Université d'Antsiranana, 2019.
14. P. Bemarisika & D. Madiohavana, *Une méthode d'extraction de règles d'association généralisées*, Séminaire de Masters du Laboratoire des Mathématiques et Informatique de l'ENSET, *2e Edition*, Université d'Antsiranana, 2018.

15. P. Bemarisika & V. Rajoro, *Bandes de confiances pour la méthode de Monte Carlo*, Séminaire de Masters du Laboratoire des Mathématiques et Informatique de l'ENSET, *2e Edition*, Université d'Antsiranana, 2018.
16. P. Bemarisika & B. Tsiatandra, *Une modélisation probabiliste d'un réseau de télécommunication*, Séminaire de Masters du Laboratoire des Mathématiques et Informatique de l'ENSET, *2e Edition*, Université d'Antsiranana, 2018.
17. J.B. Andriamanantena, P. Bemarisika & A. Totohasina, *Optimisation stochastique appliquée à la gestion de stock*, Séminaire de Masters du Laboratoire des Mathématiques et Informatique de l'ENSET, *1re Edition*, Univ. Antsiranana, 2017.
18. P. Bemarisika, C. Soavavinirina & A. Totohasina, *Optimisation stochastique appliquée à la recherche d'un plus court chemin*, Séminaire de Masters du Laboratoire des Mathématiques et Informatique de l'ENSET, *1re Edition*, Université d'Antsiranana, 2017.
19. P. Bemarisika, *Une méthode d'extraction de règles d'association positives et négatives.* Journées des doctorants du Laboratoire d'Informatique et de Mathématiques (LIM), Université de La Réunion (France), 18-19 mars 2015.
20. P. Bemarisika, *`chicmgk`, Un outil pour les graphes implicatifs*, Séminaire des doctorants de l'Ecole doctorale PE2Di, Equipe d'Accueil Education et Didactique des Mathématiques et de l'Informatique de l'ENSET, Univ. Antsiranana, 2015.
21. Participation au séminaire du Laboratoire Dynamique des Structures et Interactions des Macromolécules Biologiques (LDSIMB), Université de La Réunion (France). *Décomposition monomorphe d'une structure relationnelle*, avril 2015.
22. Participation au séminaire du Laboratoire d'Energétique, d'Electronique et Procédés (L2EP), Université de La Réunion (France). *(1) Modélisation et contrôle d'un système pile à combustible ; (2) Optimisation et design d'une pile à combustible réversible*, avril 2015.
23. Participation au séminaire du Laboratoire d'Informatique et de Mathématiques (LIM), Université de La Réunion (France). *(1) Equilibre dans les graphes orientés ; (2) Découverte des liens entre attributs : Application en didactique des mathématiques*, mars 2015.
24. Participation aux journées des Recherches du Laboratoire d'Informatique et de Mathématiques (LIM), Université de La Réunion (France). Thématique de Mathématiques et Informatique, février 2015.
25. Participation au séminaire du LDSIMB, Université de La Réunion (France). *(1) Etudes de Silico de la production d'hydrogène par des systèmes synthétiques ; (2) Etudes des déterminants structuraux de l'affinité de ligands de la protéine FKBP12*, février 2015.
26. P. Bemarisika, *Une Génération des règles d'association selon la mesure M_{GK}*. Séminaire des doctorants de l'Agence Universitaire de la Francophonie-Bureau régional Océan Indien, Univ. Antananarivo (Madagascar), 13-17 octobre 2014.
27. Participation à l'exposé sur la thématique *Classification*, Laboratoire d'Informatique et de Mathématiques (LIM), Université de La Réunion (France), avril 2014.
28. Participation à l'exposé sur le *Logiciel Python*, Laboratoire d'Informatique et de Mathématiques (LIM), Université de La Réunion (France), avril 2014.
29. Participation à l'exposé sur la thématique *Traitement d'images*, Laboratoire d'Informatique et de Mathématiques (LIM), Université de La Réunion (France), avril 2014.

30. P. Bemarisika & A. Totohasina *Un outil pédagogique pour l'extraction de règles d'association.* Atelier Pédagogie universitaire, Université d'Antsiranana, 1er-02 décembre 2013.
31. P. Bemarisika & A. Totohasina *Une approche d'enseignement progressif de résolution d'équations polynomiales par l'utilisation des TICs.* Atelier Pédagogie universitaire, Université d'Antsiranana (Madagascar), 15-16 janvier 2013.
32. P. Bemarisika, *Avancées récentes en fouille de règles d'association*, Séminaire des doctorants de l'Agence Universitaire de la Francophonie-Bureau régional Afrique de l'Ouest, Ecole Supérieure Polytechnique de Yaoundé (Cameroun), 1er-05 juillet 2013.
33. Participation au séminaire des doctorants du Laboratoire d'Informatique et de Mathématiques (LIM), Université de La Réunion (France). Thématique de Mathématiques et Informatique, mai 2013.
34. Participation au séminaire des doctorants du Laboratoire d'Informatique et de Mathématiques (LIM), Université de La Réunion (France). Thématique de Mathématiques et Informatique, janvier 2013.
35. P. Bemarisika, *Etat de l'art sur l'extraction de règles d'association*, Séminaire des doctorants de l'Agence Universitaire de la Francophonie (AUF)-Bureau régional Océan Indien, Antananarivo (Madagascar), 24-26 octobre 2012.

Collaborations scientifiques

1. Collaboration avec Raphaël Couturier (FEMTO-ST département DISC, Université de Franche-Comté, France) depuis 2015. Cette collaboration est centrée sur l'élaboration de la bibliothèque `rchicmgk` pour les graphes implicatifs, et la classification ascendante hiérarchique implicative (CAHI). Cette collaboration a abouti à des résultats [BCT21, BJT22a, BJT22b].

2. Collaboration avec André Totohasina (ENSET, Université d'Antsiranana, Madagascar). Cette collaboration est centrée sur le problème de fouille de données, notamment celui de l'extraction des règles d'association en grande dimension, et a abouti à des résultats (entre autres, [BT17], [BT17], [BT18], [BT19a], [BT19b], [BT19c], [BT20a], [BT20c], [BT20e], [BT20f], [BT21a], [BT21b]).

3. Collaboration avec Harrimann Ramanantsoa (Institut Supérieur de Technologies de Diégo-ISTD, Madagascar). Cette collaboration porte surtout sur l'aspect algorithmique de l'extraction de règles d'association en grande dimension, ainsi qu'à la validation expérimentale de celui-ci. Elle a donné lieu de résultat [BRT18]. Actuellement, nous avons entrain d'écrire un article portant sur le passage à l'échelle de l'algorithme CMG, comparé aux existants.

4. Activités de recherche menées au sein du Laboratoire d'Informatique et de Mathématiques (LIM), Université de La Réunion (France). En tant que doctorant du LIM de 2012 à 2015, j'ai effectué plusieurs séjours scientifiques dans le cadre de la préparation de ma thèse de doctorat (**janvier-mai 2012, janvier-mai 2013, janvier-mai 2014, janvier-mai 2015**). Durant ces séjours, j'ai mené plusieurs activités de recherche relatifs à mes travaux de thèse. Ces dernières ont permis aussi de développer plusieurs collaborations scientifiques avec des chercheurs du LIM et ont donné lieu à des résultats [BT14a, BT14b, BT14c, BT16].

Annexe B

Annexes du chapitre 1

Quelques preuves et propositions supplémentaires

Définition 43. *Soient $\mathcal{F}$ l'ensemble des motifs fréquents, et $\mathcal{FM}$ celui des maximaux fréquents d'une base de données $\mathcal{D}$. L'ensemble des motifs minimaux infréquents, noté ζ, est défini par :*

$$\zeta = \{l \notin \mathcal{F} \mid (\nexists \ell \notin \mathcal{F}, l \supset \ell \neq \emptyset) \ \wedge \ (\forall h \in \mathcal{FM}, l \subsetneq h)\}$$

D'après l'exemple de la figure 1.3 à un $minsup = 2/6$, nous avons trouvé que $D \notin \mathcal{F}$ et il n'existe pas un sous-ensemble non vide $D' \subset D$ tel que $supp(D') \leq 2/6$, d'où $D \in \zeta$.

Preuve de la Proposition 4. Soit $X \in \mathcal{FM}$. Nous montrons tout d'abord que $X \in \mathcal{FM} \Leftrightarrow \overline{X} \in \zeta$. En effet, $X \in \mathcal{FM} \Leftrightarrow \forall Y \in \zeta, Y \subsetneq X \Leftrightarrow \forall Y \in \zeta, Y \cap \overline{X} \neq \emptyset \Rightarrow \overline{X} \in \zeta$. Nous montrons maintenant que X est un maximal fréquent équivaut à $\overline{X}$ est un minimal infréquent, i.e. $X \in \mathcal{FM} \Leftrightarrow \overline{X} \in \zeta$. Supposons qu'il ne soit pas minimal, i.e. $\exists Y$ tel que $Y \in \zeta$. De $Y \in \zeta$, on a $\overline{Y} \in \mathcal{FM} \Rightarrow X$ tel que $X \subset \overline{Y}$, ce qui contredit du fait que X est un maximal dans $\mathcal{FM}$. Enfin, nous montrons que $X \in \zeta \Leftrightarrow \overline{X} \in \mathcal{FM}$. Supposons que $\overline{X}$ ne soit pas maximal, i.e. $\exists Y \in \mathcal{FM}$ tel que $\overline{X} \subset Y$. De $\overline{X} \subset Y$, on a $\overline{Y} \subset X$, ce qui contredit du fait que X est un minimal infréquent dans ζ. □

Preuve du Corollaire 2. Autrement dit, si l n'est pas un générateur, alors $\forall h \supset l$, h est aussi un non-générateur. Prenons encore le contexte $\mathcal{D}$ du tableau 1.1, soit $minsup = 2/6$. On obtient, d'après le résultat de la figure 1.3, que BE est un non-générateur, alors ses sur-ensembles BCE et $ABCE$ ne sont pas aussi des générateurs de ce contexte $\mathcal{D}$. □

Proposition 8. *Pour tous itemset l et item x, si $supp(l) = supp(l \cup \{x\})$, alors $\forall h \supset l$, $supp(h) = supp(h \cup \{x\})$.*

Démonstration. Le fait que $supp(l) = supp(l \cup \{x\})$ implique que pour toute transaction t contenant l, t contient aussi x. Soit une transaction t contenant h, t contient l car $l \subset h$, ainsi t contient aussi x. D'où $supp(h) = supp(h \cup \{x\})$. □

A noter que pour tout itemset non générateur l, il existe un item $x \in l$ tel que l'itemset obtenu en retirant x de l ait le même support que l, c'est-à-dire $support(\ell) = support(l)$ avec $\ell = l \backslash \{x\}$. La raison en est que si aucun élément x de ce type n'existe, alors l doit être un générateur selon la définition 5. Dans ce cas, x est alors un élément redondant de l.

Algorithme EOMF

L'algorithme EOMF (Algorithme 17) parcourt en largeur l'espace de recherche. Nous ne développons pas trop l'algorithme EOMF, qui constitue déjà un chapitre de la thèse de l'auteur. Il prend en entrée une base de données $\mathcal{D}$, un support minimum $minsup$, et donne en sortie un ensemble $\mathcal{F}$ des motifs fréquents de $\mathcal{D}$. FG désigne l'ensemble des fréquents générateurs, FNG l'ensemble des fréquents non-générateurs, et C_k celui des k-itemsets candidats, et $\mathcal{F}_k$ celui des k-itemsets fréquents. La fonction EOMF-GEN (Algorithme 18) est appelée pour générer les candidats. Elle

Algorithm 17 EOMF (Extraction Optimisée des Motifs Fréquents)

Require: $\mathcal{D} = (\mathcal{T}, \mathcal{I}, \mathcal{R})$ une base de données, et $minsup$ un seuil de support minimum.
Ensure: $\mathcal{F}$ l'ensemble des itemsets fréquents de $\mathcal{D}$

$\mathcal{F} \leftarrow \emptyset$; $FG \leftarrow \emptyset$; $FNG \leftarrow \emptyset$;
$C_k \leftarrow$ MatriceSupport($\mathcal{D}$) ; `/* Calcul de supports absolus des 1 et 2-motifs */`
for (each itemset $c \in C_k$) **do**
  for all (subset $c' \in C_k$ of c) **do**
    calculate $supp(c)$; $supp(c')$; `/* Calcul de support relatif des 1 et 2-motifs */`
    if ($supp(c) == supp(c')$) **then**
      $FG \leftarrow FG \cup \{c' \in C_k | supp(c') \geq minsup\}$;
    else
      $FNG \leftarrow FNG \cup \{c \in C_k | supp(c) \geq minsup\}$;
    end if
  end for
end for
$\mathcal{F}_k \leftarrow FG \cup FNG$;
$\mathcal{F} \leftarrow \bigcup_{i=1}^{k} \{\mathcal{F}_i.generator, \mathcal{F}_i.nongenerator, \mathcal{F}_i.supp\}$;
for ($k = 3; \mathcal{F}_{k-1} \neq \emptyset; k++$) **do**
  $C_k \leftarrow$ EOMF-GEN($\mathcal{F}_{k-1}$) ; `/* Générer les motifs candidats */`
  for (each itemset $c \in C_k$) **do**
    for all (($k-1$) subset c' of c) **do**
      if ($c' \in \mathcal{F}_{k-1}.nongenerator$) **then**
        $supp(c) = min\{supp(c') | c' \subset c\}$;
      else
        $supp(c) = |\phi(c)|/|\mathcal{D}|$;
      end if
    end for
  end for
  $\mathcal{F}_k \leftarrow \{c \in C_k | supp(c) \geq minsup\}$;
end for
$\mathcal{F} \leftarrow \bigcup_{i=1}^{k} \{\mathcal{F}_i.generator, \mathcal{F}_i.nongenerator, \mathcal{F}_i.supp\}$;

procède de deux étapes : étape de génération et d'élagage. Pour l'étape de génération, on suppose qu'il existe un ordre $<$ sur les items. Si p est un item, on note $p[i]$ le i^e plus grand item (au sens de $<$) contenu dans $\mathcal{F}_{k-1}$. Lors de la première étape, l'ensemble C_k est construit. Il contient l'ensemble des itemsets obtenus en joignant deux itemsets de $\mathcal{F}_{k-1}$ qui ont les mêmes $k-1$ premiers items. La

fonction renvoie ensuite les itemsets c tels que tout sous-ensemble s de c de taille $k-1$ soit dans $\mathcal{F}_{k-1}$. Par exemple, si $\mathcal{F}_{k-1} = \{AB, AC, BC, BD\}$, alors C_k contiendra ABC (en joignant AB et AC) et BCD (en joignant BC et BD). Ensuite, BCD sera éliminé car CD n'est pas dans $\mathcal{F}_{k-1}$.

Algorithm 18 Procedure EOMF-GEN

Require: Ensemble $\mathcal{F}_{k-1}$ de $(k-1)$-itemsets fréquents
Ensure: Ensemble C_k de k-itemsets candidats
1: $C_k = \{p[1] = q[1], \ldots, p[k-2] = q[k-2] \wedge p[k-1] < q[k-1] \mid p, q \in \mathcal{F}_{k-1}\}$ `/* Jointure */`
2: **for all** $(c \in C_k)$ **do**
3: **for all** $(s \subset c$ tel que $|s| = |c| - 1)$ **do**
4: **if** $(s \notin \mathcal{F}_{k-1})$ **then**
5: $C_k \leftarrow C_k \backslash \{c\}$;
6: **end if**
7: **end for**
8: **end for**
9: **return** C_k

Complexité : Soient $m = |\mathcal{I}|$ et $n = |\mathcal{T}|$. Tout d'abord, l'algorithme EOMF génère les 1 et 2-motifs (lignes 1-14), ce qui requiert dans le pire des cas $\mathcal{O}(|\mathcal{T}| \times |C_k|) = \mathcal{O}(n \times 2^m)$. Ce calcul est fait en une seule fois dans $\mathcal{D}$. La deuxième étape (lignes 15-28) se fait en $\mathcal{O}(n \times 2^{m-2})$ dans le pire des cas. En faisant la somme, on a $\mathcal{O}(n \times 2^{m-2}) + \mathcal{O}(n \times 2^m) = \mathcal{O}(n \times 2^m)$. Ainsi, la complexité globale d'EOMF est en $\mathcal{O}(n \times 2^m)$ dans le pire des cas. En pratique, grâce aux différentes optimisations développées (entre autres, Théorèmes 1, 2 et 3), cette complexité sera beaucoup plus faible.

Le tableau B.1 présente l'exemple d'exécution d'EOMF sur $\mathcal{D}$ du tableau 1.1 au $minsup = 2/6$. La différence importante d'EOMF versus aux existants réside du fait qu'il permet de générer

Scanner $\mathcal{D}$ $\longrightarrow$

MatriceSupport	A	B	C	D	E
A	3	2	3	1	2
B	-	5	4	0	5
C	-	-	5	1	4
D	-	-	-	1	0
E	-	-	-	-	5

Générer $\mathcal{F}_1$ et $\mathcal{F}_2$ $\longrightarrow$

C_1	FG	$supp$	$\mathcal{F}_1$	C_2	FG	$supp$	$\mathcal{F}_2$
A	oui	3/6	A	AB	oui	2/6	AB
B	oui	5/6	B	AC	non	3/6	AC
C	oui	5/6	C	AE	oui	2/6	AE
D	oui	1/6	-	BC	oui	4/6	BC
E	oui	5/6	E	BE	non	5/6	BE
				CE	oui	4/6	CE

Générer C_3 $\longrightarrow$

C_3	FG	$supp$
ABC	non	min(2/6,3/6,4/6)=2/6
ABE	non	min(2/6,2/6,5/6)=2/6
ACE	non	min(3/6,2/6,4/6)=2/6
BCE	non	min(4/6,5/6,4/6)=4/6

Générer $\mathcal{F}_3$ $\longrightarrow$

$\mathcal{F}_3$	FG	$supp$
ABC	non	2/6
ABE	non	2/6
ACE	non	2/6
BCE	non	4/6

Générer C_4 $\longrightarrow$

C_4	FG	$supp$
ABCE	non	min(2/6,2/6,2/6,4/6)=2/6

Générer $\mathcal{F}_4$ $\longrightarrow$

$\mathcal{F}_4$	FG	$supp$
ABCE	non	2/6

Table B.1 – Exécution de l'algorithme EOMF

l'ensemble des motifs fréquents de $\mathcal{D}$ en une seule passe, ce n'est pas le cas pour les existants, ceux-ci en font 4.

Annexe C

Annexes du chapitre 2

Quelques preuves supplémentaires

Preuve du Lemme 1. Nous prouvons en 2 cas : (1) X favorise Y et (2) X défavorise Y.

(1) Cas où X favorise Y (i.e., $P(Y'|X') > P(Y')$) , on a X défavorise $\overline{Y}$ (i.e., $P(\overline{Y'}|X') < P(\overline{Y'})$). Dans ce cas, $M_{GK}(X \rightarrow \overline{Y}) = \frac{P(\overline{Y'}|X') - P(\overline{Y'}))}{P(\overline{Y'})} = \frac{1-P(Y'|X')-1+P(Y')}{1-P(Y')} = -\frac{P(Y'|X')-P(Y')}{1-P(Y')} = -M_{GK}(X \rightarrow Y)$. Ainsi, $M_{GK}(X \rightarrow \overline{Y}) = -M_{GK}(X \rightarrow Y)$, d'où $mgk(X, \overline{Y}) = -mgk(X, Y)$.

(2) Cas où X défavorise Y (i.e., $P(Y'|X') < P(Y')$) , on a X favorise $\overline{Y}$ (i.e., $P(\overline{Y'}|X') > P(\overline{Y'})$). Par suite, $M_{GK}(X \rightarrow \overline{Y}) = \frac{P(\overline{Y'}|X') - P(\overline{Y'}))}{1-P(\overline{Y'})} = \frac{1-P(Y'|X')-1+P(Y')}{1-P(Y')} = -\frac{P(Y'|X')-P(Y')}{P(Y')} = -M_{GK}(X \rightarrow Y)$. Ainsi, $M_{GK}(X \rightarrow \overline{Y}) = -M_{GK}(X \rightarrow Y)$, d'où $mgk(X, \overline{Y}) = -mgk(X, Y)$. □

Preuve du Lemme 2. $\forall X, Y, T, Z \subseteq \mathcal{I}$, on a : $supp(X \cup Y) = \frac{|\phi(X \cup Y)|}{|\mathcal{T}|} = \frac{|\phi(X) \cap \phi(Y)|}{|\mathcal{T}|}$ et $supp(T \cup Z) = \frac{|\phi(T \cup Z)|}{|\mathcal{T}|} = \frac{|\phi(T) \cap \phi(Z)|}{|\mathcal{T}|}$. Si $X \subset T \subseteq \gamma(X)$, $Z \subset Y \subseteq \gamma(Z)$, alors on a $X \cong T$ et $Y \cong Z \Rightarrow |\phi(X)| = |\phi(T)|$ et $|\phi(Y)| = |\phi(Z)|$ impliquent que $supp(X \cup Y) = \frac{|\phi(T) \cap \phi(Z)|}{|\mathcal{T}|} = \frac{|\phi(T \cup Z)|}{|\mathcal{T}|} = supp(T \cup Z)$. Puisque $|\phi(X)| = |\phi(T)|$ et $|\phi(Y)| = |\phi(Z)|$ (i.e. $P(X') = P(T')$ et $P(Y') = P(Z')$), on a $P(Y'|X') = P(Z'|T') \Leftrightarrow P(Y'|X') - P(Y') = P(Z'|T') - P(Z') \Leftrightarrow \frac{P(Y'|X')-P(Y')}{1-P(Y')} = \frac{P(Z'|T')-P(Z')}{1-P(Z')} \Leftrightarrow M_{GK}(X \rightarrow Y) = M_{GK}(T \rightarrow Z)$ d'où $mgk(X, Y) = mgk(T, Z)$. □

Il en résulte que la règle $T \rightarrow Z$ est dérivable de $X \rightarrow Y$, et qu'elle est redondante par rapport à la règle $X \rightarrow Y$ (cf. Definition 14), donc à élaguer du processus de l'extraction.

Caractérisation d'autres règles redondantes

Nous montrons maintenant, via les Propositions 9, 10 et 11, comment caractériser d'autres règles d'association positives et négatives redondantes d'une base de données $\mathcal{D}$.

Proposition 9. *Soient $X, Y, T, Z \subseteq \mathcal{I}$ tels que $P(Y'|X') < P(Y')$ et $P(Z'|T') < P(Z')$. Si $X \subset T \subseteq \gamma(X)$, $Z \subset Y \subseteq \gamma(Z)$, alors $supp(X \cup \overline{Y}) = supp(T \cup \overline{Z})$ et $mgk(X, \overline{Y}) = mgk(T, \overline{Z})$.*

Démonstration. $\forall X, Y, T, Z \subseteq \mathcal{I}$, $supp(X \cup \overline{Y}) = \frac{|\phi(X \cup \overline{Y})|}{|\mathcal{T}|} = \frac{|\phi(X) \cap \phi(\overline{Y})|}{|\mathcal{T}|}$ et $supp(T \cup \overline{Z}) = \frac{|\phi(T \cup \overline{Z})|}{|\mathcal{T}|} = \frac{|\phi(T) \cap \phi(\overline{Z})|}{|\mathcal{T}|}$. Si $X \subset T \subseteq \gamma(X)$ et $Y \subset Z \subseteq \gamma(Y)$, alors on a $X \cong T$ et $Y \cong Z \Rightarrow |\phi(X)| = |\phi(T)|$

et $|\phi(Y)| = |\phi(Z)|$ (i.e. $|\phi(\overline{Y})| = |\phi(\overline{Z})|$) $\Rightarrow supp(X \cup \overline{Y}) = \frac{|\phi(X) \cap \phi(\overline{Y})|}{|\mathcal{T}|} = \frac{|\phi(T) \cap \phi(\overline{Z})|}{|\mathcal{T}|} = \frac{|\phi(T \cup \overline{Z})|}{|\mathcal{T}|} = supp(T \cup \overline{Z})$. Comme $|\phi(\overline{Y})| = |\phi(\overline{Z})|$ (i.e. $supp(\overline{Y}) = supp(\overline{Z})$) et $supp(X \cup \overline{Y}) = supp(T \cup \overline{Z})$, on a $\frac{supp(X \cup \overline{Y})}{supp(\overline{Y})} = \frac{supp(T \cup \overline{Z})}{supp(\overline{Z})} \Leftrightarrow P(\overline{Y'}|X') = P(\overline{Z'}|T') \Leftrightarrow P(\overline{Y'}|X') - P(\overline{Y'}) = P(\overline{Z'}|T') - P(\overline{Z'})$. Comme X (resp. T) défavorise Y (resp. Z), alors X (resp. T) favorise $\overline{Y}$ (resp. $\overline{Z}$) [BT19b], ce qui implique $\frac{P(\overline{Y'}|X') - P(\overline{Y'})}{1 - P(\overline{Y'})} = \frac{P(\overline{Z'}|T') - P(\overline{Y'})}{1 - P(\overline{Z'})} \Leftrightarrow M_{GK}(X \rightarrow \overline{Y}) = M_{GK}(T \rightarrow \overline{Z})$ d'où $mgk(X, \overline{Y}) = mgk(T, \overline{Z})$. □

Cette Proposition 9 explique que $T \rightarrow \overline{Z}$ peut être dérivée de $X \rightarrow \overline{Y}$, et que cette règle est redondante par rapport à la règle $X \rightarrow \overline{Y}$, donc à supprimer du processus de l'extraction.

Proposition 10. *Soient $X, Y, T, Z \subseteq \mathcal{I}$ tels que $P(Y'|X') < P(Y')$ et $P(Z'|T') < P(Z')$. Si $X \subset T \subseteq \gamma(X)$, $Z \subset Y \subseteq \gamma(Z)$, alors $supp(\overline{X} \cup Y) = supp(\overline{T} \cup Z)$ et $mgk(\overline{X}, Y) = mgk(\overline{T}, Z)$.*

Démonstration. $\forall X, Y, T, Z \subseteq \mathcal{I}$, $supp(\overline{X} \cup Y) = \frac{|\phi(\overline{X} \cup Y)|}{|\mathcal{T}|} = \frac{|\phi(\overline{X}) \cap \phi(Y)|}{|\mathcal{T}|}$ et $supp(\overline{T} \cup Z) = \frac{|\phi(\overline{T} \cup Z)|}{|\mathcal{T}|} = \frac{|\phi(\overline{T}) \cap \phi(Z)|}{|\mathcal{T}|}$. Puisque $T \subset X \subseteq \gamma(T)$, $Z \subset Y \subseteq \gamma(Z)$, on a $X \cong T$ et $Y \cong Z \Rightarrow |\phi(X)| = |\phi(T)|$ et $|\phi(Y)| = |\phi(Z)|$ (i.e. $P(X') = P(T')$ et $P(Y') = P(Z')$). Comme $|\phi(X)| = |\phi(T)|$ (i.e. $|\phi(\overline{X})| = |\phi(\overline{T})|$) et $|\phi(Y)| = |\phi(Z)|$, on a $supp(\overline{X} \cup Y) = \frac{|\phi(\overline{X}) \cap \phi(Y)|}{|\mathcal{T}|} = \frac{|\phi(\overline{T}) \cap \phi(Z)|}{|\mathcal{T}|} = \frac{|\phi(\overline{T} \cup Z)|}{|\mathcal{T}|} = supp(\overline{T} \cup Z)$. Comme $supp(\overline{X}) = supp(\overline{T})$, $P(Y') = P(Z')$, et $supp(\overline{X} \cup Y) = supp(\overline{T} \cup Z)$, on a $\frac{supp(\overline{X} \cup Y)}{supp(\overline{X})} = \frac{supp(\overline{T} \cup Z)}{supp(\overline{T})} \Leftrightarrow P(Y'|\overline{X'}) = P(Z'|\overline{T'}) \Leftrightarrow P(Y'|\overline{X'}) - P(Y') = P(Z'|\overline{T'}) - P(Z')$. Comme X défavorise Y (resp. T défavorise Z)), on a $\overline{X}$ favorise Y (resp. $\overline{T}$ favorise Z) [BT19b]. On obtient ainsi $\frac{P(Y'|\overline{X'}) - P(Y')}{1 - P(Y')} = \frac{P(Z'|\overline{T'}) - P(Z')}{1 - P(Z')} \Leftrightarrow M_{GK}(\overline{X} \rightarrow Y) = M_{GK}(\overline{T} \rightarrow Z)$ d'où $mgk(\overline{X}, Y) = mgk(\overline{T}, Z)$. □

Ce qui relève que $\overline{T} \rightarrow Z$ est dérivable de $\overline{X} \rightarrow Y$, et qu'elle est redondante par rapport à la règle $\overline{X} \rightarrow Y$, donc à élaguer du processus de l'extraction.

Proposition 11. *Soient $X, Y, T, Z \subseteq \mathcal{I}$ tels que $P(Y'|X') > P(Y')$ et $P(Z'|T') > P(Z')$. Si $T \subset X \subseteq \gamma(T)$ et $Y \subset Z \subseteq \gamma(Y)$, alors $supp(\overline{X} \cup \overline{Y}) = supp(\overline{T} \cup \overline{Z})$ et $mgk(\overline{X}, \overline{Y}) = mgk(\overline{T}, \overline{Z})$.*

Démonstration. Pour tous $X, Y, T, Z \subseteq \mathcal{I}$, on a $supp(\overline{X} \cup \overline{Y}) = \frac{|\phi(\overline{X} \cup \overline{Y})|}{|\mathcal{T}|} = \frac{|\phi(\overline{X}) \cap \phi(\overline{Y})|}{|\mathcal{T}|}$ et $supp(\overline{T} \cup \overline{Z}) = \frac{|\phi(\overline{T} \cup \overline{Z})|}{|\mathcal{T}|} = \frac{|\phi(\overline{T}) \cap \phi(\overline{Z})|}{|\mathcal{T}|}$. Si $T \subset X \subseteq \gamma(T)$ et $Z \subset Y \subseteq \gamma(Z)$, alors $X \cong T$ et $Y \cong Z \Rightarrow |\phi(X)| = |\phi(T)|$ (i.e. $|\phi(\overline{X})| = |\phi(\overline{T})|$) et $|\phi(Y)| = |\phi(Z)|$ (i.e. $|\phi(\overline{Y})| = |\phi(\overline{Z})|$). Depuis $|\phi(\overline{X})| = |\phi(\overline{T})|$ et $|\phi(\overline{Y})| = |\phi(\overline{Z})|$, on obtient $supp(\overline{X} \cup \overline{Y}) = \frac{|\phi(\overline{X}) \cap \phi(\overline{Y})|}{|\mathcal{T}|} = \frac{|\phi(\overline{T}) \cap \phi(\overline{Z})|}{|\mathcal{T}|} = \frac{|\phi(\overline{T} \cup \overline{Z})|}{|\mathcal{T}|} = supp(\overline{T} \cup \overline{Z})$. Comme $|\phi(\overline{X})| = |\phi(\overline{T})|$ (i.e. $supp(\overline{X}) = supp(\overline{T})$) et $supp(\overline{X} \cup \overline{Y}) = supp(\overline{T} \cup \overline{Z})$, on obtient $\frac{supp(\overline{X} \cup \overline{Y})}{supp(\overline{X})} = \frac{supp(\overline{T} \cup \overline{Z})}{supp(\overline{T})} \Leftrightarrow P(\overline{Y'}|\overline{X'}) = P(\overline{Z'}|\overline{T'}) \Leftrightarrow P(\overline{Y'}|\overline{X'}) - P(\overline{Y'}) = P(\overline{Z'}|\overline{T'}) - P(\overline{Z'})$. Comme $|\phi(\overline{Y})| = |\phi(\overline{Z})|$ (i.e. $P(\overline{Y'}) = P(\overline{Z'})$), on obtient $\Leftrightarrow \frac{P(\overline{Y'}|\overline{X'}) - P(\overline{Y'})}{1 - P(\overline{Y'})} = \frac{P(\overline{Z'}|\overline{T'}) - P(\overline{Z'})}{1 - P(\overline{Z'})} \Leftrightarrow M_{GK}(\overline{X} \rightarrow \overline{Y}) = M_{GK}(\overline{T} \rightarrow \overline{Z}) \Leftrightarrow mgk(\overline{X}, \overline{Y}) = mgk(\overline{T}, \overline{Z})$. □

Cette Proposition 11 explique que $\overline{T} \rightarrow \overline{Z}$ est dérivable de $\overline{X} \rightarrow \overline{Y}$, et qu'elle est redondante par rapport à $\overline{X} \rightarrow \overline{Y}$, donc à élaguer du processus de l'extracttion.

Annexe D

Annexes du chapitre 3

Cette partie porte sur l'étude des variations de la mesure statistique mgk à partir de l'indice de qualité $\widetilde{mgk}$ (cf. équation (2.7), i.e. variable observée). Etant donnée une règle d'association de la forme $X \to Y$, le but est d'examiner la sensibilté de cet indice de qualité $\widetilde{mgk}$ à des paramètres en jeu de telle règle $X \to Y$. Plusieurs méthodes ont été proposées dans la littérature, nous retenons ici une méthode mathématique [LMV+04, Vai06, GDGB07]. Cela consiste à étudier les variations des paramètres en examinant leurs dérivées partielles. Notre analyse se divise essentiellement en deux volets, tels que étude des variations en fonction des cardinaux, et en fonction de fréquence.

Etude des variations en fonction des cardinaux

Etudier la sensibilité de l'indice $\widetilde{mgk}$, revient à examiner ses variations au voisinage des 4 valeurs entières observées $(n, n_X, n_Y, n_{X\overline{Y}})$ des paramètres de la règle $X \to Y$. Cela alors consiste à analyser la différentielle de $\widetilde{mgk}$ par rapport à ces variables et d'en conserver la restriction à ces paramètres. Pour ce faire, nous considérons ces variables comme nombres réels, et l'indice $\widetilde{mgk}$ comme une fonction continûment différentiable de telle sorte que $0 \leq n_X \leq n_Y$, $n_{X\overline{Y}} \leq \inf\{n_X, n_Y\}$ et $\sup\{n_X, n_Y\} \leq n$. La différentielle de tel indice $\widetilde{mgk}$ s'exprime de la façon suivante :

$$d\widetilde{mgk} = \frac{\partial \widetilde{mgk}}{\partial n} dn + \frac{\partial \widetilde{mgk}}{\partial n_X} dn_X + \frac{\partial \widetilde{mgk}}{\partial n_Y} dn_Y + \frac{\partial \widetilde{mgk}}{\partial n_{X\overline{Y}}} dn_{X\overline{Y}} = \overrightarrow{grad}(\widetilde{mgk}). \begin{pmatrix} dn \\ dn_X \\ dn_Y \\ dn_{X\overline{Y}} \end{pmatrix}$$

En fait, la differentielle de la fonction $\widetilde{mgk}$ apparait donc comme le produit scalaire de son gradient et des variations de $\widetilde{mgk}$ sur la surface représentant les variations de la fonction $\widetilde{mgk}(n, n_X, n_Y, n_{X\overline{Y}})$. Ainsi, le gradient de $\widetilde{mgk}$ représente ses propres variations en fonction de celles de ses composantes.

Si nous examinons le cas où seuls n_Y et $n_{X\overline{Y}}$ varient (n et n_X constants), on obtient alors :

$$\frac{\partial \widetilde{mgk}}{\partial n_Y} = n_{X\overline{Y}} \left(\frac{n_X n_Y}{n} \right) > 0$$

$$\frac{\partial \widetilde{mgk}}{\partial n_{X\overline{Y}}} = \frac{1}{\frac{n_X n_{\overline{Y}}}{n}} = \frac{1}{\frac{n_X(n-n_Y)}{n}} > 0$$

Ceci s'interprète comme si le nombre d'exemples n_Y et celui de contre-exemples $n_{X\overline{Y}}$ augmentent, alors la mesure statistique mgk diminue pour n et n_X constants.

Si nous examinons le cas où seul n_X varie, nous obtenons la dérivée partielle ci-après :

$$\frac{\partial \widetilde{mgk}}{\partial n_X} = -\frac{n_{X\overline{Y}}}{\frac{n_X^2 n_{\overline{Y}}}{n}} < 0$$

Ceci signifie que la fonction $\widetilde{mgk}$ est décroissante sur $[0, n_Y]$, et est minimum pour $n_X = n_Y$. Par suite, la mesure statistique mgk croît lorsque n_X croît.

Etude des variations en fonction des fréquences

Dans ce cadre, nous examinons les variations de $\widetilde{mgk}$ en fonction des fréquences relatives. Notons $f_X = \frac{n_X}{n}$ (resp. $f_Y = \frac{n_Y}{n}$ et $f_{X\overline{Y}} = \frac{n_{X\overline{Y}}}{n}$), la fréquence des variables n_X (resp. n_Y et $n_{X\overline{Y}}$) pour le cardinal n. Par suite, la fonction $\widetilde{mgk}$ s'écrit alors de la manière suivante :

$$\widetilde{mgk}(X, \overline{Y}) = \frac{f_{X\overline{Y}} - f_X f_{\overline{Y}}}{f_X f_{\overline{Y}}} = \frac{f_{X\overline{Y}}}{f_X f_{\overline{Y}}} - 1$$

Remarquons qu'en étant indépendant de n, il n'a pas un sens statistique aussi intéressant pour $\widetilde{mgk}$ en fonction des fréquences f_X, $f_{\overline{Y}}$ et $f_{X\overline{Y}}$. Sa différentielle s'exprime de la façon suivante :

$$d\widetilde{mgk} = \frac{\partial \widetilde{mgk}}{\partial f_X} df_X + \frac{\partial \widetilde{mgk}}{\partial f_{\overline{Y}}} df_{\overline{Y}} + \frac{\partial \widetilde{mgk}}{\partial f_{X\overline{Y}}} df_{X\overline{Y}} = \overrightarrow{grad}(\widetilde{mgk}). \begin{pmatrix} df_X \\ df_{\overline{Y}} \\ df_{X\overline{Y}} \end{pmatrix}$$

La sensibilité de $\widetilde{mgk}$ aux variations des fréquences f_X, $f_{\overline{Y}}$ et $f_{X\overline{Y}}$ se lit avec les dérivées partielles :

$$\frac{\partial \widetilde{mgk}}{\partial f_X} = -\frac{f_{X\overline{Y}}}{f_{\overline{Y}} f_X^2} < 0$$

$$\frac{\partial \widetilde{mgk}}{\partial f_{\overline{Y}}} = -\frac{f_{X\overline{Y}}}{f_X f_{\overline{Y}}^2} < 0$$

$$\frac{\partial \widetilde{mgk}}{\partial f_{X\overline{Y}}} = \frac{1}{f_X f_{\overline{Y}}} > 0$$

Les deux premiers résultats (2 premières relations) s'interprètent comme la fonction $\widetilde{mgk}$ décroît quand les fréqences f_X et $f_{\overline{Y}}$ croissent, à $f_{X\overline{Y}}$ constante. En conséquence, la mesure statistique mgk diminue lorsque la fréquence f_X (resp. $f_{\overline{Y}}$) croît (resp. décroît), mais la vitesse des variations reste constante, indépendante des variations de n. Puis, le dernier résultat quant à lui signifie que si la fréquence de contre-exemples $f_{X\overline{Y}}$ augmente (pour f_X et $f_{\overline{Y}}$ constantes), alors mgk diminue, mais la vitesse de variations reste encore constante et indépendante des variations de n.

Bibliographie

[AIS93] R. Agrawal, T. Imielinski, and A. N. Swami. Mining association rules between sets of items in large databases. In *Proceedings of the ACM SIGMOD Conference*, pages 207–216, Washington, DC, 1993.

[AM11] R. Aldecoa and I. Marín. Deciphering network community structure by surprise. In *PloSone*, 2011.

[AS94] R. Agrawal and R. Srikant. Fast Algorithms for Mining Association Rules. In *Proceedings of 20th VLDB Conference*, pages 487–499, Santiago, Chile, 1994.

[BB04] D. BARSKY and B. BENZAGHOU. Nombres de bell et somme de factorielles. pages 1–17, 2004.

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

[BCT21] P. Bemarisika, R. Couturier, and A. Totohasina. Une construction condensée et interactive du graphe implicatif d'une base de données. In *J-C. Régnier et al.(Eds), Analyse Statistique Implicative (ASI'2021)*, pages 199–224, 2021.

[Bem16] P. Bemarisika. *Extraction des règles d'association selon le couple support-M_{GK} : Graphes implicatifs, et Applications en didactique des mathématiques.* PhD thesis, Université d'Antananarivo, Madagascar, 2016.

[Ben18] A. Benyattou. *Congruences pour quelques suites de nombres combinatoires.* PhD thesis, Université des Sciences et de la Technologie Houari Boumediène, Algérie, 2018.

[BJT22a] P. Bemarisika, A. Jerson, and A. Totohasina. Une méthode de construction d'arbres hiérarchiques implicatifs. In *Société Francophone de Classification*, 2022.

[BJT22b] P. Bemarisika, A. Jerson, and A. Totohasina. Une méthode simultannée pour la compression, le partitionnement et la construction d'un graphe implicatif. In *Société Francophone de Classification*, 2022.

[Bor03] C. Borgelt. Efficient Implementations of Apriori and Eclat. In *FIMI'03 Workshop on Frequent Item Set Mining Implementations*, Aachen, Germany, CEUR Workshop Proceedings 90, November 2003.

[BRT18] P. Bemarisika, H. Ramanantsoa, and A. Totohasina. An Efficient Approach for Extraction Positive and Negative Association Rules from Big Data. In *International Cross-Domain for Machine Learning and Knowledge Extraction, CD-MAKE 2018*, pages 79–97. Springer, 2018.

[BRTR12] P. Bemarisika, H. Ramanantsoa, A. Totohasina, and L. Ramifidisoa. Enseignement et apprentissage de la résolution d'équations polynomiales par l'utilsation de TIC au niveau secondaire. In *Colloque international sur les TIC*, ENS, Antananarivo, 2012.

[BT14a] P. Bemarisika and A. Totohasina. Apport de règles négatives à l'extraction des règles d'association. In *Société Francophone de Classification (SFC)*, pages 99–104, 2014.

[BT14b] P. Bemarisika and A. Totohasina. Elaboration of implicative graph according to measure M_{GK}. *International Journal of Computer Science Issues*, Vol. 11, No 1, Issue 4 :52–59, 2014.

[BT14c] P. Bemarisika and A. Totohasina. A Novel Algorithm for Mining Negative and Positive Association Rules. *International Journal of Computer and Information Technology*, Volume 03-Issue 04 :792–798, 2014.

[BT16] P. Bemarisika and A. Totohasina. EOMF, Un algorithme d'extraction optimisée des motifs fréquents. In *Apprentissage Artificiel & Fouille de Données (AAFD), et Société Francophone de Classification (SFC)*, pages 198–203, 2016.

[BT17] P. Bemarisika and A. Totohasina. Optimisation de l'extraction des règles d'association positives et négatives. In *Société Francophone de Classification (SFC)*, pages 25–28, 2017.

[BT18] P. Bemarisika and A. Totohasina. ERAPN, an algorithm for extraction positive and negative association rules in big data. In *Big Data Analytics and Knowledge Discovery (DaWaK)*, pages 329–344. Springer, 2018.

[BT19a] P. Bemarisika and A. Totohasina. Elimination of redundant association rules. In *Information Systems Architecture and Technology (ISAT)*, pages 208–218. Springer, 2019.

[BT19b] P. Bemarisika and A. Totohasina. An informative base of positive and negative association rules on big data. In *International Conference on Big Data*, pages 2428–2437. Springer, 2019.

[BT19c] P. Bemarisika and A. Totohasina. Nouvelles bases des règles d'association non-redondantes. In *Société Francophone de Classification (SFC)*, pages 77–82, 2019.

[BT20a] P. Bemarisika and A. Totohasina. Concise representations for positive and negative association rules. In *Intnal Conf. on Fuzzy Systems and Knowledge Discovery (FSKD)*. Springer, 2020.

[BT20b] P. Bemarisika and A. Totohasina. Consise extraction for informative positive and negative association rules. In *ACM International Conference on Information Integration and Web-based Applications and Services (iiWAS'20)*, 2020.

[BT20c] P. Bemarisika and A. Totohasina. An efficient method for mining informative association rules in knowledge extraction. In *International Cross-Domain for MAchine Learning and Knowledge Extraction (CD-MAKE)*, pages 227–247. Springer, 2020.

[BT20d] P. Bemarisika and A. Totohasina. Nonred, An efficient algorithm for mining non-redundant rules in big data. In *International Conference on Big Data Analytics and Knowledge Discovery, DaWaK 2020*, 2020.

[BT20e] P. Bemarisika and A. Totohasina. Visualisation interactive des graphes d'une règle d'association informative. In *Workshop sur Visualisation d'informations, Interactions et Fouille de données (VIF)*, pages 5–6, 2020.

[BT20f] P. Bemarisika and A. Totohasina. Visualisation interactive des graphes implicatifs. In *REVUT Scientific Journal (RSJ),* DOI *: https ://doi.org/10.46857/rsj.2021.3*, 2020.

[BT21a] P. Bemarisika and A. Totohasina. CONCISE : An Algorithm for Mining Positive and Negative Non-Redundant Association Rules. In R. Pal and K.P. Shukla (eds), editors, *SCRS Conference Proceedings on Intelligent Systems*, pages 13–34, 2021.

[BT21b] P. Bemarisika and A. Totohasina. Generating a condensed representation for positive and negative association rules. In *Intnal Conf. on BIS*, pages 175–186. Springer, 2021.

[CG06] T. Calders and B. Goethals. Non-derivable itemset mining. In *Data Mining and Knowledge Discovery(DMKD)*, pages 1–35, 2006.

[CG15] R. Couturier and S. Ghanem. Ajout de la confiance au graphe implicatif. In *Analyse Statistique Implicative (ASI)*, pages 117–129, 2015.

[CP15] R. Couturier and R. Pazmiño. Statistical implicative analysis for educational data sets : Analysis with RCHIC. In *EDUTEC XVIII*, 2015.

[DLL17] T. Delacroix, P. Lenca, and S. Lallich. Du local au global : Un nouveau défi pour l'analyse statistique implicative. In *J-C. Régnier et al.(Eds), ASI*, pages 102–115, 2017.

[DQ13] N. Durand and M. Quafafou. Approximation de bordures de motifs fréquents par le calcul de traverses minimales approchées d'hypergraphes. In *Conférence Francophone sur l'Apprentissage Automatique (CAp'2013)*, 2013.

[DRT07] J. Diatta, H. Ralambondrainy, and A. Totohasina. Towards a unifying probabilistic implicative normalized quality measure for association rules. In *Quality Measures in Data Mining*, pages 237–250, 2007.

[FDT06] D.R. Feno, J. Diatta, and A. Totohasina. Une base pour les règles d'association d'un contexte binaire valides au sens de la mesure de qualité M_{GK}. In *Proc. Société Francophone de Classification*, pages 105–109, 2006.

[FDT07] D.R. Feno, J. Diatta, and A. Totohasina. Génération de bases pour les règles d'association M_{GK}-valides. In *Proc. Société Francophone de Classification*, pages 101–104, 2007.

[Fen07] D. R. Feno. *Mesures de qualité des règles d'association : normalisation et caractérisation de bases.* PhD thesis, Université de La Réunion, France, 2007.

[Fle96] L. Fleury. *Découverte de connaissances dans une base de données de gestion des ressources humaines.* PhD thesis, Université de Nantes, 1996.

[For10] S. Fortunato. Community detection in graphs. In *Physics Reports*, pages 75–174, 2010.

[GAB+96] R. Gras, S. Almouloud, M. Bailleul, A. Larher, M. Polo, H. Ratsimba-Rajohn, and A. Totohasina. *L'implication statistique, nouvelle méthode exploratoire des données.* La Pensée Sauvage, 1996.

[Gay09] D.J. Gay. *Calcul de motifs sous contraintes pour la classification supervisée.* PhD thesis, Université de la Nouvelle-Calédonie, 2009.

[GCG15] R. Gras, R. Couturier, and P. Gregori. Un mariage arrangé entre l'implication et la confiance. In *ASI, Analyse Statistique Implicative*, 2015.

[GDGB07] R. Gras, J. David, F. Guillet, and H. Briand. Stabilité en ASI de l'intensité d'implication et comparaisons avec d'autres indices de qualité de règles d'association. In *EGC*, 2007.

[GKB03] R. Gras, P. Kuntz, and H. Briand. Hiérarchie orientée et règles généralisées en analyse implicative. In M.S. Hacid, Y. Kodratoff, and D. Boulanger, editors, *EGC'2003*, pages 145–158, 2003.

[GL10] R. Ghosh and K. Lerman. Community detection using a measure of global influence. In *Advances in Social Network Mining and Analysis*, pages 20–35, 2010.

[GRMG13] R. Gras, J-C. Régnier, C. Marinica, and F. Guillet. L'analyse statistique implicative, méthode exploratoire et confirmatoire à la recherche de causalités. In *Cépaduès Editions*, pages 11–40, 2013.

[GS96] J. Galambos and I. Simonelli. *Bonferroni-type inequalities with applications.* Springer, 1996.

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

[GW99] B. Ganter and R. Wille. *Formal concept analysis : Mathematical foundations.* Springer Verlag, 1999.

[GYNS06] G. Gasmi, S.B. Yahia, E.M Nguifo, and Y. Slimani. IGB : Une nouvelle base générique informative des règles d'association. In *Revue I3 (Information-Interaction-Intelligence)*, pages 31–65, 2006.

[Hah17] M. Hahsler. arulesviz : Interactive visualization of association rules with r. In *R Journal*, pages 163–175, 2017.

[HYN11] T. Hamrouni, S.B. Yahia, and E.M Nguifo. Construction efficace du treillis des motifs fermés fréquents et extraction simultanée des bases génériques de règles. *Mathématiques et Sciences humaines*, pages 5–54, 2011.

[Ler81] I-C. Lerman. *Classification et analyse ordinale des donnés.* Dunod, 1981.

[Ler08] I.C. Lerman. Analyse de vraisemblance des liens relationnels : Une méthodologie d'analyse classificatoire des données. *IRISA*, 2008.

[LHH12] C. Latiri, H. Haddad, and T. Hamrouni. Towards an effective automatic query expansion process using an association rule mining approach. pages 209–247, 2012.

[LLWH14] G. Liu, J. Li, L. Wong, and W. Hsu. Positive borders or negative borders : How to make lossless generator based representations concise. In *SIAM*, pages 469–472, 2014.

[LMV+04] P. Lenca, P. Meyer, P. Vaillant, P. Picouet, and S. Lallich. Evaluation et analyse multicritères de qualité des règles d'association, mesures de qualité pour la fouille de données. In *RNTI-E*, pages 219–246, 2004.

[Maa17] M. Maamar. *Fouille de motifs basée sur la programmation par contraintes-Appliquée à la validation de logiciels.* PhD thesis, Université d'Oran 1 - Ahmed Ben Bella, 2017.

[MD09] A.D. Medus and C.O. Dorso. Alternative approach to community detection in networks. In *Physical Review*, 2009.

[MLLL16] M. Maamar, N. Lazaar, S. Loudni, and Y. Lebbah. F-CPminer : Une approche pour la localisation de fautes basée sur l'extraction de motifs ensemblistes sous contraintes. In *Actes JFPC*, pages 83–92, 2016.

[MT97] M. Mannila and H. Toivonen. Mining top-k non-redundant association rules. In *Data Mining Knowledge Discovery*, pages 241–258, 1997.

[New06] M. Newman. Finding community structure in networks using the eigenvectors of matrices. In *Physical Review E-Statistical, Nonlinear and Soft Matter Physics*, 2006.

[NG04] M.E.J. Newman and M. Girvan. Finding and evaluating community structure in networks. In *Physical Review*, 2004.

[Ngu12] T.K.N. Nguyen. *Generalizing Association Rules in N-ary Relations : Application to Dynamic Graph Analysis.* PhD thesis, Institut National des Sciences Appliquées de Lyon, France, 2012.

[OLL+16] A. Ouali, S. Loudni, Y. Lebbah, P. Boizumault, A. Zimmermann, and L. Loukil. Clustering conceptuel en PLNE. In *Actes JFPC*, pages 27–36, 2016.

[PTB+05] N. Pasquier, R. Taouil, Y. Bastide, G. Stumme, and L. Lakhal. Generating a condensed representation for association rules. In *J. of Intell. Info. Syst.*, pages 29–60, 2005.

[QUE13] F. QUEYROI. *Partitionnement de grands graphes : Mesures, Algorithmes et Visualisation.* PhD thesis, Université de Bordeaux I, 2013.

[Ram16] H. Ramanantsoa. *Contributions à l'amélioration de génération des bases des règles d'association M_{GK}-valides et applications en didactique des mathématiques.* PhD thesis, Université d'Antananarivo, 2016.

[RBT20a] T. Rabenantenaina, P. Bemarisika, and A. Totohasina. Une approche efficace pour estimer un modèle de séries temporelles. In *Journées de Recherche de ISTs et leurs partenaires internationaux*, 2020.

[RBT20b] T. Rabenantenaina, P. Bemarisika, and A. Totohasina. Une stratégie d'estimation d'un processus temporel. In *REVUT Scientific Journal (RSJ)*, 2020.

[RBTR12] H. Ramanantsoa, P. Bemarisika, A. Totohasina, and L. Ramifidisoa. Enseignement de limite d'une fonction et TIC au niveau secondaire. In *Colloque international sur les TIC*, ENS, Antananarivo, 2012.

[STB+02] G. Stumme, R. Taouil, Y. Bastide, N. Pasquier, and L. Lakhal. Computing iceberg concept lattices with titanic. In *Journal on Knowledge and Data Engineering (KDE)*, pages 189–222, 2002.

[TF08] A. Totohasina and D.R. Feno. De la qualité des règles d'association : Etude comparative des mesures M_{GK} et confiance. In *CARI'08*, pages 561–568, 2008.

[Tot08] A. Totohasina. *Contribution à l'étude des mesures de qualité des règles d'association : Normalisation sous cinq contraintes et cas de M_{GK} ; Propriété, base composite des règles d'association, et extension en vue d'applications en statistique et en sciences physiques.* PhD thesis, Université d'Antsiranana, Madagascar, 2008. HDR.

[TR05] A. Totohasina and H. Ralambondrainy. Ion : A pertinent new measure for mining information from many types of data. In *IEEE, SITIS'05*, pages 202–207, 2005.

[Vai06] B. Vaillant. *Mesurer la qualité des règles d'association : Etudes formelles et expérimentales.* PhD thesis, Université de Bretagne Sud, 2006.