Intégration de connaissances expertes dans le processus de fouille de données pour l'extraction d'informations pertinentes

Laurent Brisson

UNIVERSITÉ DE NICE - SOPHIA ANTIPOLIS - UFR SCIENCES

ÉCOLE DOCTORALE
SCIENCES ET TECHNOLOGIES DE L'INFORMATION ET DE LA COMMUNICATION

THÈSE

pour obtenir le titre de

DOCTEUR EN SCIENCES

de l'université de Nice - Sophia Antipolis

Spécialité : INFORMATIQUE

présentée par et soutenue par

Laurent BRISSON

Intégration de connaissances expertes dans le processus de fouille de données pour l'extraction d'informations pertinentes

Thèse dirigée par **Martine COLLARD**
soutenue le 13 décembre 2006 devant le jury :

Michel RIVEILL, Professeur, UNSA	Président
Oscar PASTOR, Professeur, Universitad Politécnica de Valencia	Rapporteur
Jean-Marie PINON, Professeur, INSA de Lyon	Rapporteur
Jérôme AZÉ, Maître de Conférences, Université Paris XI	Examinateur
Engelbert MEPHU NGUIFO, Maître de Conférences (HDR), Université d'Artois	Examinateur
Martine COLLARD, Maître de Conférences (HDR), UNSA	Directeur de thèse
Pierre BOURGEOT, Directeur du CERTIAM/CNEDI de Sophia Antipolis	Membre invité
Jacques FAVEEUW, Directeur du CNEDI de Lyon	Membre invité

Remerciements

Cette thèse a été financée par une Convention Industrielle de Formation par la Recherche (CIFRE) avec le CERTIAM/CNEDI de Sophia-Antipolis. Ce travail a été réalisé au sein de l'équipe EXeCO du laboratoire Informatique Signaux et Systèmes de Sophia Antipolis (I3S, UMR UNSA/CNRS) et du CERTIAM/CNEDI de Sophia Antipolis.

Je tiens en tout premier lieu à remercier ma directrice de thèse Mme Martine COLLARD, pour la confiance qu'elle m'a témoignée tout au long de ces années de travail, sa grande disponibilité au quotidien et ses précieux conseils.

Je remercie M. Pierre BOURGEOT et M. Jacques FAVEEUW pour m'avoir accueilli au CNEDI et rassemblé toutes les conditions nécessaires à la réussite de cette thèse aussi bien dans le cadre de l'entreprise que dans celui de la recherche.

Je remercie M. Cyrille BROILLIARD, M. Hugues SANIEL, M. Hussein CHAMI, Mme Marie-Hélène BARON-ROYER et les membres du BGPEO de Grenoble pour leur collaboration fructueuse.

Je remercie les professeurs Oscar PASTOR et Jean-Marie PINON pour l'honneur qu'ils m'ont fait en acceptant d'être rapporteurs et pour toutes les remarques constructives qu'ils ont formulées au sujet de mon manuscrit.

Je remercie le professeur Michel RIVEILL pour m'avoir fait l'honneur de présider ce jury.

Je remercie également M. Jérôme AZÉ M. Engelbert MEPHU NGUIFO qui m'ont fait le plaisir de participer à ce jury.

Je tiens tout particulièrement à remercier M. Nicolas PASQUIER, pour le temps qu'il a pu m'accorder tout au long de ces trois années et la patience avec laquelle il a répondu à mes nombreuses questions ; nos discussions m'ont beaucoup appris.

Je remercie également chaleuresement M. Jean-Louis CAVARERO, Mme Isabelle

MIRBEL ainsi que tous les autres membres de l'équipe EXeCO pour les discussions intéressantes que nous avons partagées lors de nos séminaires et le soutien qu'ils m'ont apporté tout au long de cette thèse.

Enfin une pensée particulière à mes amis, Sylvie, Florian et Nepher ainsi qu'à mes parents et mes frères pour le soutien et l'aide qu'ils m'ont apportés durant mes études et sans qui ce travail n'aurait pas pu avoir lieu. Un dernier clien d'oeil à Pistou et Pitchoun, qui n'ont jamais omis de me réveiller aux toutes premières lueurs de l'aube ...

Laurent BRISSON

Résumé

L'extraction automatique de connaissances à partir des données peut être considérée comme la découverte d'informations enfouies dans de très grands volumes de données. Les approches actuelles, pour évaluer la pertinence des informations extraites, se distinguent en deux catégories : les approches objectives qui mettent en œuvre des mesures d'intérêt afin d'évaluer les propriétés statistiques des modèles extraits et les approches subjectives qui confrontent les modèles extraits à des connaissances exprimées sur le domaine et nécessitent généralement l'interrogation d'experts. Toutefois, le choix de modèles pertinents en regard de la connaissance métier d'un expert reste un problème ouvert et l'absence de formalisme dans l'expression des connaissances nuit à la mise au point de techniques automatiques de confrontation des modèles permettant d'exploiter toute la richesse sémantique des connaissances expertes.

L'approche KEOPS que nous proposons dans ce mémoire, répond à cette problématique en proposant une méthodologie qui intègre les connaissances des experts d'un domaine tout au long du processus de fouille. Un système d'information dirigé par une ontologie (ODIS) joue un rôle central dans le système KEOPS en permettant d'organiser rationnellement non seulement la préparation des données mais aussi la sélection et l'interprétation des modèles générés. Une mesure d'intérêt est proposée afin de prendre en compte les centres d'intérêt et le niveau de connaissance des experts. Le choix des modèles les plus pertinents se base sur une évaluation à la fois objective pour évaluer la précision des motifs et subjective pour évaluer l'intérêt des modèles pour les experts du domaine. Enfin l'approche KEOPS facilite la définition de stratégies pour améliorer le processus de fouille de données dans le temps en fonction des résultats observés. Les différents apports de l'approche KEOPS favorisent l'automatisation du processus de fouille de données, et ainsi, une dynamique d'apprentissage peut être initiée pour obtenir un processus de fouille particulièrement bien adapté au domaine étudié. KEOPS a été mise en œuvre dans le cadre de l'étude de la gestion des relations avec les allocataires au sein des Caisses d'Allocations Familiales. L'objectif de cette étude a été d'analyser la relation de service rendu aux allocataires afin de fournir aux décideurs des connaissances précises, pertinentes et utiles pour l'amélioration de ce service.

Mots-Clefs : Fouille de données, mesures d'intérêt, ontologies, expression de connaissances

Table des matières

Liste des figures

Liste des tables

Liste des Algorithmes

Chapitre 1

Introduction

Sommaire

L'extraction automatique de connaissances à partir des données peut être considérée comme la découverte d'informations enfouies dans de très grands volumes de données. Ces informations, selon la tâche effectuée et les algorithmes mis en œuvre, sont exprimées sous la forme de modèles ou de motifs. Hand et al. définissent un *modèle* relativement à une structure globale qui couvre toutes les données alors qu'un *motif* correspond à une description d'une partie de l'espace de données [HSM01]. Parmi les modèles on peut considérer les arbres de décision résultant d'une classification supervisée ou les clusters en classification non supervisée tandis que les motifs sont souvent exprimés sous la forme de règles, par exemple dans le cas de la recherche d'associations. Bien que l'approche proposée dans cette thèse soit mise en œuvre pour la recherche de motifs exprimés sous forme de règles d'association, elle est également adaptée à la recherche d'autres formes de motifs et modèles. Une différence essentielle entre les techniques statistiques et les techniques de fouille de données réside dans le fait que ces dernières visent à construire les modèles de façon automatique, sans formuler d'hypothèses sur les relations liant les données et en évitant ainsi de recourir à un expert. Cependant nous pouvons constater que les approches actuelles ne permettent pas de s'abstraire de la présence d'un expert du domaine pour sélectionner des résultats réellement intéressants et utilisables. Les indices permettant de caractériser la pertinence des informations extraites sont traditionnellement séparés en deux classes : les indices objectifs d'une part, et les indices subjectifs d'autre part.

1.1 Problématique

Les approches objectives mettent en œuvre des mesures d'intérêt afin d'évaluer la précision des modèles extraits. Les nombreux travaux menés sur l'optimisation des algorithmes d'extraction ont abouti à des solutions performantes fournissant des modèles plus concis et plus fiables. Cependant, dans le cadre d'un domaine donné, il demeure toujours assez difficile de choisir des modèles pertinents en regard de la connaissance métier d'un expert. De nombreuses approches définissent des critères de qualité guidant le choix d'une mesure ou d'un ensemble de mesures ; toutefois, la communauté n'est pas parvenue actuellement à un consensus sur le choix d'un ensemble de mesures adéquates. Une méthode d'aide à la décision a été proposée afin d'assister les experts dans le choix d'une mesure adaptée à une situation donnée.

Les approches subjectives, quand à elles, se sont focalisées sur des mécanismes interactifs nécessitant l'interrogation d'experts du domaine avant de pouvoir évaluer les modèles extraits par les techniques de fouille. Elles se basent par exemple sur l'expression de patrons qui modélisent les attentes et les connaissances des experts et qui sont ensuite confrontés aux modèles extraits grâce à des mesures de similarité prenant parfois en compte des relations taxonomiques. Ces approches demeurent limitées car elles mélangent les connaissances explicites des experts (celles qui sont formulées et apparaissent sous forme de documents tangibles) et les connaissances tacites, qui relèvent plus du savoir-faire et qui demeurent parfois incertaines. D'autre part l'absence de formalisme au niveau de l'expression des connaissances empêche la mise au point de techniques automatiques de confrontation des modèles et des connaissances qui pourraient profiter de toute la richesse sémantique des connaissances des experts.

Avec le développement des techniques de gestion et de représentation des connaissances il apparaît plus aisé d'intégrer au processus de fouille de données des connaissances sémantiquement riches : les ontologies, par exemple, qui formalisent la connaissance et apportent un support exploitable de manière automatisée. Toutefois, à la différence des ontologies du domaine, les ontologies à définir pour la fouille de données doivent également contenir les connaissances nécessaires à la réalisation de la tâche de fouille de données : ce sont des ontologies d'application. L'utilisation de ce type d'ontologie tout au long du processus de fouille de données apparaît donc comme une perspective prometteuse afin de faire évoluer les approches subjectives.

Par ailleurs, un processus de fouille de données est habituellement considéré comme éminemment itératif. Il est généralement mené de manière expérimentale, par essais et erreurs. Bien que permettant de comparer deux modèles extraits, les environnements dédiés à la fouille de données n'offrent pas actuellement la possibilité d'organiser de manière très rationnelle de véritables stratégies de fouille. Il serait intéressant de disposer d'outils permettant de maîtriser la gestion des expériences et de comparer des modèles extraits en fonction des variations des paramètres caractérisant les expériences.

1.2 Contributions

L'approche KEOPS que nous proposons dans ce mémoire a pour objectif d'apporter de nouvelles solutions palliant en partie les limitations évoquées ci-dessus. Les contributions de ce travail sont à la fois d'ordre méthodologique et applicatif :

- Une méthodologie qui intègre les connaissances des experts d'un domaine tout au long du processus de fouille. Cette méthodologie concentre l'intervention de l'expert du domaine lors de la phase de pré-traitement des données afin que l'intégration de la sémantique du domaine puisse s'effectuer dès la préparation des données. Cette étape devient ainsi plus complexe mais le processus global de fouille de données profite largement de l'apport des connaissances expertes.
- Un système d'information dirigé par une ontologie (ODIS pour Ontology Driven Information System) qui joue un rôle central dans le système KEOPS. L'ODIS est constitué d'une ontologie d'application et d'une base de données relationnelle (MODB pour Mining Oriented DataBase) dont les valeurs et attributs sont associés à des concepts de l'ontologie. Ce système d'information permet d'organiser rationnellement la préparation des données mais aussi la sélection et l'interprétation des modèles générés.
- Une mesure pour évaluer l'intérêt des modèles générés prenant en compte les centres d'intérêt et le niveau de connaissance des experts. Cette mesure se base sur une approche à la fois objective pour évaluer la précision des motifs et subjective pour évaluer l'intérêt des modèles pour les experts du domaine. Son originalité réside dans le fait qu'elle repose sur l'ODIS qui formalise les relations entre concepts et fournit des données nettoyées et structurées au sein de la MODB.

- La possibilité de définir des stratégies permettant d'améliorer le processus de fouille de données dans le temps en fonction des résultats observés. Les différents apports de l'approche KEOPS favorisent l'automatisation du processus de fouille de données, et ainsi, une dynamique d'apprentissage peut être initiée pour obtenir un processus de fouille particulièrement bien adapté au domaine étudié.

La méthodologie proposée a été mise en œuvre dans le cadre de deux tâches de fouille sur des données réelles. La première est du type CRM et répond à une demande des CAF (Caisses d'Allocation Familiales) ; la seconde concerne des données médicales issues d'une étude sur l'athérosclérose.

1.3 Organisation du mémoire

Ce mémoire est organisé comme suit. La première partie, structurée en trois chapitres, est consacrée à un état de l'art des approches existantes. Le chapitre 2 présente les approches objectives pour l'extraction de connaissances. Le chapitre 3 se focalise sur les approches subjectives. Le chapitre 4 décrit les méthodes de représentation des connaissances ainsi que la manière dont ces dernières permettent d'intégrer la connaissance à un processus de fouille de données.

La seconde partie est consacrée à la présentation complète du système, elle est structurée en quatre chapitres. Le chapitre 5 présente les objectifs du système. Ces objectifs guident les choix méthodologiques et de conception ainsi que les méthodes d'évaluation et d'exploration des modèles générés par la fouille. Ce chapitre présente également une vue d'ensemble du système et justifie l'approche qui est adoptée dans la suite du mémoire. L'expression des connaissances fait l'objet du chapitre 6. Au chapitre 7 nous présentons notre méthode de confrontation des connaissances aux modèles générés par la fouille de données. Enfin, dans le chapitre 8, nous présentons les stratégies qui peuvent être mises en place afin d'organiser et gérer les différentes expériences menées dans un processus de fouille de données.

Le chapitre 9 présente le bilan des travaux effectués et présente nos projets de travaux futurs.

Première partie

État de l'art

Chapitre 2

Approches objectives pour l'extraction de connaissances

Sommaire

2.1 Introduction

Le volume important de motifs générés lors d'un processus de fouille de données a conduit à la mise au point de techniques pour sélectionner les meilleurs d'entre eux selon différents critères. De nombreuses mesures d'intérêt ont été mises au point dans le but d'évaluer la fiabilité et la qualité de ces motifs. Des études ont été menées pour énoncer des critères auxquels doivent répondre ces mesures (section 2.2). Une approche différente a été de clarifier la situation, soit par la création d'une mesure unique soit en aidant à choisir une mesure réalisant le meilleur compromis selon le contexte (section 2.3). Enfin, des algorithmes qui intègrent les mesures à leur mécanisme ont été mis au point pour éviter d'utiliser les mesures d'intérêt uniquement en tant que filtre appliqué aux modèles découverts (section 2.4).

Dans cette section la plupart des approches présentées s'appliquent à l'évaluation de motifs exprimés sous forme de règles d'association. Il est donc indispensable de présenter les notations utilisées par la suite.

Définition 2.1 (Table d'une base de données)
Une table d'une base de données est composée d'attributs définis sur un domaine de valeurs. Une table représente une relation de la base de données, c'est-à-dire un sous-ensemble du produit cartésien des domaines d'un ensemble d'attributs. Chaque ligne de la base de données est également appelée exemple, tuple ou objet.

Définition 2.2 (Item et itemset)
On définit un item par un triplet {A,op,V} où :

- *A est un attribut*
- *op est un opérateur parmi* $<$, $\leq$, $>$, $\geq$, $=$
- *V est une valeur*

Un itemset est constitué d'un ensemble d'items.

Remarque : Dans le cas de valeurs discrètes seul l'opérateur $=$ est utilisé.

Notation 2.1
Soit une règle R de la forme $A \rightarrow B$. *On appelle A l'antécédent de la règle et B son conséquent.*
Soit n le nombre d'exemples d'une table. On nomme $n(A)$ *et* $n(B)$ *les nombres*

d'exemples qui contiennent respectivement les items de A et de B et $n(AB)$ le nombre d'exemples qui contiennent à la fois A et B.

L'objectif de ce chapitre n'est pas d'effectuer un état de l'art des différentes mesures d'intérêt objectif. Une synthèse complète sur ce sujet été réalisée par Azé qui introduit de plus des critères de choix sur les mesures d'intérêt [Azé03], ou encore par McGarry qui établit un récapitulatif des différentes mesures d'intérêt objectif et subjectif [McG05]. Ce chapitre traite des différentes approches utilisées pour l'étude des mesures d'intérêt. La première partie examine les différents critères de qualité pour évaluer les mesures. La deuxième partie présente les travaux unifiant les mesures d'intérêt. La troisième partie présente différentes solutions algorithmiques.

2.2 Critères de qualité

Freitas propose dans [Fre99] cinq critères évaluer la qualité de mesures d'intérêt utilisées dans le contexte de la classification supervisée :

- Les événements rares, également appelés *pépites de connaissances* ou *small disjunct* en anglais
- Le coût des attributs
- L'asymétrie des règles
- Le déséquilibre de la distribution des classes
- Le coût des erreurs de classification

Toutefois, les trois premiers critères s'appliquent à des mesures qui évaluent de façon individuelle les règles qui décrivent les modèles et non pas à des mesures évaluant les modèles dans leur intégralité. Lallich et Teytaud ont, quand à eux, définis dix critères [LT04] qui ont été ensuite complétés par Guillaume [Gui00] et Azé [Azé03] :

- Compréhensibilité de la mesure
- Nature des règles visées
- Prise en compte des contre-exemples
- Sens de variation de la mesure
- Nature de la variation de la mesure
- Comportement par rapport à la couverture de l'antécédent et du conséquent

- Sensibilité à la taille des données
- Fixation d'un seuil
- Sensibilité au bruit

Dans cette section, un aspect essentiel des mesures est présenté : leur intelligibilité ou encore leur simplicité de compréhension. Ensuite les différents critères proposés, parfois similaires, sont classés en trois catégories : les critères liés au type de motif que l'on souhaite extraire, les critères liés aux propriétés intrinsèques des mesures et les critères liés au comportement des mesures en fonction des données.

2.2.1 Intelligibilité et simplicité

Une première évaluation lors de l'étude de motifs est de considérer les mesures de support et de confiance dont le sens concret est parfaitement assimilable pour l'utilisateur non spécialiste [LT04]. Nous rappelons les définitions ci-dessous.

Définition 2.3 (Support)
Le support d'une règle $A \rightarrow B$ sur une table T est la proportion d'exemples de T qui contiennent à la fois A et B. Il peut s'exprimer en terme de probabilité :

$$Supp(A \rightarrow B) = p(AB) = \frac{n(AB)}{n}$$

Définition 2.4 (Confiance)
La confiance d'une règle $A \rightarrow B$ sur une table T est la proportion d'exemples de T qui contiennent B parmi ceux qui contiennent A, c'est-à-dire la probabilité conditionnelle de B sachant A :

$$Conf(A \rightarrow B) = p(B\,|A) = \frac{p(AB)}{p(A)} = \frac{n(AB)}{n(A)}$$

Le principal intérêt des mesures de support et de confiance réside dans leur grande intelligibilité pour les utilisateurs non statisticiens. Toutefois les approches basées sur le support et la confiance souffrent de deux inconvénients majeurs. Dans les algorithmes de recherche d'associations, le seuil de support écarte les règles ayant un trop petit support, alors que parmi elles certaines peuvent avoir une bonne valeur de confiance et présenter un réel intérêt ! Une solution est alors de baisser la valeur du seuil de support, ce qui engendre de nouveaux problèmes : le nombre de motifs extraits devient très grand et les algorithmes sont asphyxiés. L'utilisation du support est donc une approche limitative pour l'extraction de règles rares.

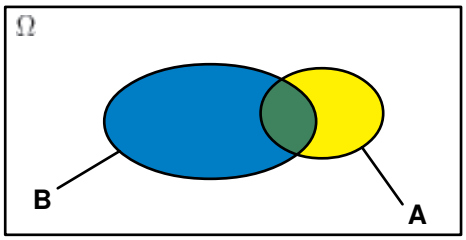


FIG. 2.1: Situation où la confiance est inférieure à 1

Le deuxième inconvénient est relatif à la définition de la confiance. En effet, la faiblesse de la confiance est de ne pas prendre en compte p(B), la proportion d'exemples qui contiennent les items de B.

Remarque 2.1 Dans les figures 2.1, 2.2 et 2.3 le rectangle représente l'univers Ω. L'ellipse bleue représente la couverture du conséquent B d'une règle, l'ellipse jaune la couverture de l'antécédent A d'une règle et l'ellipse verte l'intersection de A et B.

Comme on peut le voir sur la figure 2.1, la confiance n'est pas optimale dans cette situation. Il existe alors trois situations où la confiance est améliorée : lorsque la couverture de B inclut celle de A et met en évidence un événement rare (figure 2.2), lorsque la couverture de A est incluse dans celle de B (figure 2.3 a) ou lorsque la couverture de B couvre quasiment tout l'univers (figure 2.3 b). Dans ces trois derniers cas la confiance sera égale à 1, cependant si le cas représenté par la figure 2.2 est très intéressant, celui de la figure 2.3b n'apporte aucune information qui puisse être intéressante puisque la probabilité de B approche de près la probabilité conditionnelle de B sachant A.

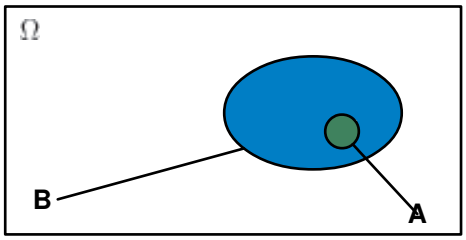
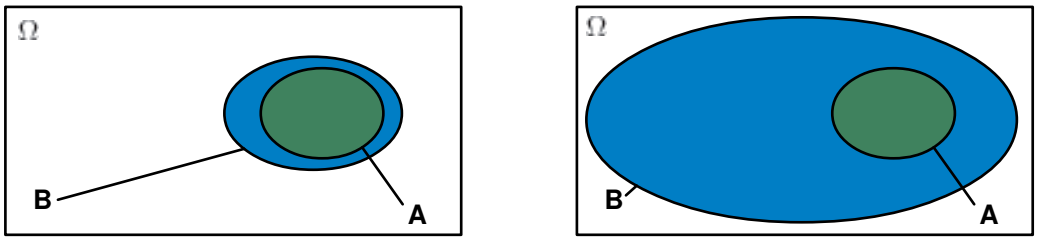


FIG. 2.2: Événement rare avec une confiance égale à 1

La mesure de *lift* représente un bon compromis satisfaisant le critère d'intelligibilité d'une part et permettant de résoudre le problème énoncé ci-dessus d'autre



a) Règle intéressante b) Règle inintéressante

FIG. 2.3: Deux situations différentes où la confiance est égale à 1

part. Le lift permet en effet de faire la distinction entre les corrélations positives et négatives entre antécédent et conséquent. Cependant c'est une mesure symétrique qui ne différencie donc pas la règle $A \rightarrow B$ de la règle $B \rightarrow A$.

Définition 2.5 (Lift)
Le lift d'une règle $A \rightarrow B$ représente sa distance à l'indépendance. Les attributs A et B sont indépendants lorsque : $p(AB) = p(A) \times p(B)$

$$Lift(A \rightarrow B) = \frac{p(AB)}{p(A)p(B)} = \frac{p(B\,|A)}{p(B)}$$

En conclusion, les approches utilisant des mesures simples et intelligibles ont en contrepartie quelques inconvénients. En complément de celles-ci, il peut alors être intéressant de faire appel à des mesures d'intérêt plus complexes. Ces dernières pouvant toutefois introduire un biais il est important de connaître leurs propriétés et leur comportement. Un certain nombre de travaux [Azé03][LT04][McG05] ont été consacrés récemment à l'analyse, selon des points de vue divers, des différentes mesures proposées. Certains auteurs [Fre99][LMV+04][TKS02] se sont attachés à définir des critères de satisfaction.

Dans la suite de ce chapitre nous présentons un état de l'art de ces travaux selon trois axes : les critères liés au type des motifs à extraire, les critères liés aux propriétés intrinsèques des mesures et enfin les critères liés aux propriétés des mesures vis à vis des données.

2.2.2 Critères liés au type des motifs à extraire

Cette section présente les critères permettant d'extraire un type particulier de motif. Nous nous intéressons au cas des événements rares et à celui des motifs surprenants.

2.2.2.1 Les événements rares

Le concept d'*événement rare* correspond à une règle dont la couverture de l'antécédent est faible. Selon Freitas, la prise en compte des événements rares est essentiel afin de mettre au point une mesure d'intérêt. Cependant cet aspect est souvent mis de côté lors de la fouille de données car, à première vue, ces règles sont potentiellement source d'erreurs. En effet, en les étudiant, on peut observer qu'elles peuvent aussi bien représenter une exception véritable qu'un simple bruit.

Les mesures permettant d'évaluer les événements rares sont évidemment très différentes de celles permettant d'évaluer les règles ayant une large couverture. On voit ainsi la nécessité d'adapter la mesure au type de l'information ou du motif ciblé par la tâche de fouille.

2.2.2.2 Les motifs surprenants

Les mesures basées sur la surprise relèvent le plus souvent des approches subjectives décrites dans la section 3. Cependant Freitas propose dans [Fre98] de s'intéresser à la manière d'évaluer le degré de surprise d'une règle de façon objective. Dans le cadre de ses travaux relatifs à des règles de classification il propose trois approches différentes :

- une première relative aux événements rares en utilisant la notion de généralisation minimale d'une règle,
- une deuxième qui prend en compte l'effet de surprise de chacun des items d'une règle,
- une troisième se basant sur la détection du paradoxe de Simpson.

Généralisations minimales d'une règle

Soit une règle de classification $R : A \rightarrow B$, A représentant une conjonction d'items[1] : $A_1 \wedge A_2 \wedge ... \wedge A_k$. On dit que la $i$ème généralisation minimale de R a pour antécédent A privé de l'item A_i et pour conséquent la classe majoritaire, sur le jeu de données, qui vérifie A. Une règle d'antécédent A de taille $|A| = k$ possède donc k généralisations minimales.

[1] voir définition 2.2

Exemple 2.1
Considérons une étude concernant l'analyse de la clientèle susceptible d'acheter un ordinateur. La règle 2.1 possède trois généralisations minimales : les règles 2.2, 2.3 et 2.4.

Règle 2.1
Sexe="Homme" $\wedge$ *Revenus="Faibles"* $\wedge$ *Age="Jeune"* $\rightarrow$ *AchetePC="Non"*

Règle 2.2
Sexe="Homme" $\rightarrow$ *AchetePC="Oui"*

Règle 2.3
Revenus="Faibles" $\rightarrow$ *AchetePC="Non"*

Règle 2.4
Age="Jeune" $\rightarrow$ *AchetePC="Oui"*

Selon cette approche, plus le conséquent de la règle 2.1 est contredit par les conséquents des généralisations minimales 2.2, 2.4 et 2.3 plus la règle sera considérée comme surprenante. Dans cet exemple, la règle 2.1 est donc surprenante car deux de ses trois généralisations minimales ne prédisent pas la même classe « Non ».

Effet de surprise de chacun des items de l'antécédent

Selon Freitas, la plupart des mesures évaluant la surprise prennent en compte l'antécédent d'une règle comme un tout, sans prendre en considération chacun de ses items. Ces mesures sont appelées mesure de surprise à *forte granularité*. Ainsi deux mesures de surprise à *forte granularité* pourraient évaluer la surprise d'une règle de la même façon alors que celles-ci auraient un intérêt différent pour l'utilisateur si l'on avait considéré chacun des items lors du calcul de l'intérêt.

Freitas propose donc une mesure de surprise à *faible granularité* se basant sur la théorie de l'information [CT91]. Pour cela, il se base sur la mesure de *gain d'information* appliquée à chacun des attributs composant l'antécédent d'une règle. Les attributs avec un haut niveau de gain d'information sont de bons prédicteurs de la classe. Dans cette approche, le postulat est que ces attributs sont très certainement déjà connus de l'utilisateur et ne lui sont d'aucun intérêt. C'est ainsi que les règles considérées comme les plus surprenantes sont celles qui réunissent un grand nombre

d'attributs mauvais prédicteurs de la classe. Chacun des attributs pris individuellement n'est pas intéressant, mais une règle qui en réunit un grand nombre dans son antécédent est potentiellement surprenante et donc intéressante pour l'utilisateur.

Le paradoxe de Simpson

Ce paradoxe repose sur une distribution particulière des classes. Considérons une population partitionnée en deux populations exclusives Pop_1 et Pop_2 selon un critère $Crit_1$. Soit G un attribut binaire définissant un objectif. G_1 et G_2 sont respectivement les valeurs de l'objectif pour les populations Pop_1 et Pop_2. On définit $p(G_1)$ et $p(G_2)$ les probabilités que l'objectif étudié soit vérifié sur les populations Pop_1 et Pop_2.

Considérons le cas où chacune des populations Pop_1 et Pop_2 est partitionnée selon un deuxième critère $Crit_2$ possédant n valeurs. On définit alors $p(G_{ij})$ la probabilité que l'objectif soit atteint pour chacune des sous-populations, avec $i = \{1, 2\}$ l'identifiant de $Crit_1$ et $j = \{1, ..., n\}$ l'identifiant de $Crit_2$.

Le paradoxe de Simpson apparaît lorsque :

$$\begin{cases} p(G_1) > p(G_2) \\ et \\ \forall j = 1, ..., n \qquad p(G_{1j}) \leq p(G_{2j}) \end{cases}$$

et dans la situation duale :

$$\begin{cases} p(G_1) < p(G_2) \\ et \\ \forall j = 1, ..., n \qquad p(G_{1j}) \geq p(G_{2j}) \end{cases}$$

Exemple 2.2 (Cursus de formation et concours)
Des étudiants, afin de préparer deux concours X et Y décident de suivre des cursus de formation afin de s'améliorer. Ils ont le choix entre suivre le cursus A ou suivre le cursus B. Cependant au moment de passer les concours, ils apprennent que les deux ont lieu en même temps ! Chacun fait donc le choix de passer le concours pour l'école qui l'intéresse le plus.

Le tableau 2.1 donne les résultats du concours (en effectifs et pourcentage), pour simplifier il y a 100 étudiants dans chacun des cursus A et B.

	Cursus A	Cursus B
Concours X	29/30 : 96,5%	1/1 : 100%
Concours Y	1/70 : 1,5%	19/99 : 19,2%
Total	30%	20%

TAB. 2.1: Illustration du paradoxe de Simpson : Taux de réussite des étudiants par groupe et par concours

Le paradoxe de Simpson s'illustre par le fait que globalement les étudiants ayant suivis le cursus A ont mieux réussi sur l'ensemble des concours alors que pour chacun des concours le pourcentage de réussite des étudiants du groupe B est meilleur. Ce paradoxe est dû au fait que, lors d'une évaluation globale, on ne prend pas en compte l'impact de la variable Concours et la répartition des effectifs dans les deux sous-populations.

Freitas propose un algorithme de détection du paradoxe de Simpson au sein des modèles générés afin d'extraire des *informations surprenantes*.

2.2.3 Critères liés aux propriétés intrinsèques des mesures

Les mesures d'intérêt mises au point peuvent avoir des comportements très différents lors de l'évaluation des règles. Nous présentons ici quelques critères de qualité liés aux propriétés des mesures ; ils prennent en compte le sens et la nature de la variation d'une mesure, sa compréhensibilité, son comportement dans différentes situations mais aussi la nature des règles visées.

2.2.3.1 Sens de variation de la mesure

Piatetsky-Shapiro [PSF91] a proposé trois critères pour construire une mesure m ayant la capacité d'évaluer la corrélation entre l'antécédent et le conséquent d'une règle :

- Si A et B sont statistiquement indépendant alors $m(R) = 0$
- $m(R)$ augmente de façon monotone avec n(AB) quand n(A) et n(B) restent constants
- $m(R)$ diminue de façon monotone avec n(AB) quand n(A) et n(B) restent constants

Major et Mangano ont introduit un quatrième critère [MM95] :

- $m(R)$ augmente avec n(A) avec une confiance fixée $conf(R) > p(B)$

2.2.3.2 Étude de situations caractéristiques

Azé [Azé03] et Guillaume [Gui00]s'intéressent au comportement des mesures dans trois cas spécifiques pour une règle $A \rightarrow B$, qui correspondent à des situations dans lesquelles l'intérêt de la règle est facilement évaluable :

1. La règle ne vérifie aucun exemple ; A et B sont totalement incompatibles : $p(AB) = 0$
2. A et B sont indépendants : $p(AB) = p(A) \times p(B)$
3. $A \rightarrow B$ est une règle exacte car sa confiance est égale à 1 : $p(AB) = p(A)$ et $A \subset B$

Dans le premier cas, la règle $A \rightarrow B$ n'a aucun lieu d'être et l'évaluation d'une mesure devrait être égale à une valeur nulle ou à la valeur minimale associée à la mesure. Le second cas correspond à une situation d'indépendance entre les attributs et devrait se caractériser par une évaluation égale à la valeur nulle. Enfin, le troisième cas détermine, selon Azé, une situation où la règle représente un événement rare.

L'étude effectuée compare, selon ces critères, différentes mesures d'intérêt. Ainsi, lorsque A et B sont indépendants un grand nombre de mesures prend une valeur constante et donc indépendante de la taille de A et B. De même, lorsque $A \rightarrow B$ est une règle logique, plusieurs mesures prennent des valeurs constantes et évaluent de manière identique les situations représentées par les figures 2.2 et 2.3.

Il est donc intéressant, lors du choix d'une mesure, de prendre en compte sa capacité à distinguer ces situations caractéristiques.

2.2.3.3 Nature des règles visées

Selon les travaux de Lallich et Teytaud une mesure d'intérêt devrait permettre de [LT04] :

- Faire la différence entre les règles $A \rightarrow B$ et $A \rightarrow \overline{B}$, les exemples de l'une étant les contre-exemples de l'autre.
- Prendre en compte l'orientation du lien entre A et B : les évaluations des règles $A \rightarrow B$ et $B \rightarrow A$ ne devraient pas être équivalentes. Si elles ont bien les mêmes

exemples, leurs contre-exemples sont différents. Ainsi une règle inversant la cause et la conséquence ne devrait absolument pas avoir le même intérêt !

- Faire la distinction entre $A \rightarrow B$ et $\overline{B} \rightarrow \overline{A}$: si les règles sont équivalentes au sens logique, elles ne le sont pas au sens des règles d'association car elles ne sont pas vérifiées par les mêmes exemples.

2.2.3.4 Nature de la variation de la mesure

Lallich et Teytaud s'intéressent également à la variation linéaire des mesures. Les mesures peuvent varier de façon linéaire ou non-linéaire en fonction du nombre d'exemples ou de contre-exemples. Toutes les mesures créées par une transformée affine de la confiance évoluent de façon linéaire, cependant les mesures non linéaires présentent l'avantage d'être moins sensibles au bruit. En effet, une mesure non-linéaire voit sa valeur changer lentement à l'apparition des premiers contre-exemples. Cela lui confère ainsi meilleure résistance au bruit que les mesures linéaires qui vont, dès les premiers contre-exemples, changer leur valeur de façon plus rapide.

2.2.3.5 Comportement par rapport à la couverture de l'antécédent et du conséquent

Il est important qu'une mesure prenne en considération la couverture de l'antécédent et du conséquent d'une règle. Lorsque ce n'est pas le cas la mesure peut avoir des valeurs égales pour des règles ayant un intérêt totalement différent : c'est le cas pour la mesure de confiance comme nous l'avons expliqué en section 2.4. Lallich suggère donc qu'une mesure soit une fonction croissante selon $1 - p(B)$, qui représente la rareté du conséquent.

2.2.3.6 Fixation d'un seuil

Les mesures doivent pouvoir être utilisées avec un seuil d'élagage afin de sélectionner les règles les plus intéressantes d'entre elles. Les mesures compréhensibles et les mesures normalisées permettent à l'utilisateur de fixer assez facilement un seuil. Cependant, lorsque le seuil est fixé par l'utilisateur, ce dernier prend rarement en compte la nature des données ce qui peut conduire à une mauvaise interprétation des résultats. Le calcul automatique du seuil est alors une approche intéressante ; cependant la complexité du calcul est un élément à considérer.

2.2.4 Critères liés au comportement des mesures en fonction des données

Une même mesure peut se comporter différemment selon le jeu de données sur lequel elle est employée. Nous évoquons ici les différents aspects d'un jeu de données à prendre en compte lors de l'élaboration d'une mesure d'intérêt.

2.2.4.1 Le déséquilibre de la distribution des classes

On parle de déséquilibre dans la distribution des classes lorsque les exemples contenant une classe sont beaucoup plus fréquents ou beaucoup plus rares que les exemples relatifs aux autres classes. Pour illustrer ce problème, prenons l'exemple de données relatives à 2 classes dont la distribution est déséquilibrée. Découvrir les règles concernant la classe majoritairement présente est très facile, mais il est beaucoup plus dur de découvrir les règles décrivant la classe minoritaire. Selon Freitas les règles prédisant une classe minoritaire, difficiles à découvrir, sont intéressantes. Inversement les règles décrivant la classe majoritaire sont plutôt inintéressantes.

Il est donc important qu'une mesure prenne en compte ce déséquilibre dans la distribution des classes (voir section 2.2.2.2).

2.2.4.2 Le coût des attributs

Certaines mesures prennent en compte l'antécédent d'une règle comme un tout. Freitas met en évidence le fait qu'il est intéressant de considérer chacun des attributs d'un antécédent comme une entité individuelle. Il présente ainsi la notion de coût : dans un cadre prédictif, deux règles fiables statistiquement n'auront pas forcément la même valeur si l'on prend en compte le coût associé à chacun des attributs. Une mesure d'intérêt devrait donc, à fiabilité et utilité identiques, trouver les règles peu coûteuses.

2.2.4.3 Le coût des erreurs de classification

Dans certains domaines, les erreurs de classification peuvent avoir des coûts différents. Par exemple, dans le domaine bancaire refuser un prêt à un bon client (c'est-à-dire un client qui va le rembourser) coûte moins cher que d'accepter le prêt d'un mauvais client.

La prise en compte du coût des erreurs de classification est donc, selon le domaine d'étude, un paramètre à prendre en compte avant de choisir une mesure.

2.2.4.4 Prise en compte des contre-exemples

Une règle peut être intéressante si son nombre d'exemples $n(AB)$ est élevé ou si son nombre de contre-exemples $n(A\overline{B})$ est faible. La plupart des mesures prenant en compte $n(AB)$ considèrent implicitement les contre-exemples ; cependant certains auteurs envisagent une mesure évaluant $n(A\overline{B})$, bien que la complexité du calcul soit non négligeable.

2.2.4.5 Sensibilité à la taille des données

Lallich fait la distinction entre les mesures descriptives et les mesures statistiques. Selon cette classification les mesures descriptives ne changent pas l'évaluation d'une règle en cas de dilatation des données, c'est-à-dire lorsque tous les effectifs sont multipliés par un même facteur, et les mesures statistiques sont sensibles à la taille de l'échantillon considéré.

A première vue, une mesure statistique semble plus intéressante. Si l'on fixe la taille de A et de B et que l'on dilate la taille des données, la règle $A \rightarrow B$ décrivant les données dilatées est plus intéressante car elle décrit une fraction des données ; il est donc possible qu'elle ne soit pas encore connue ! Toutefois, Lallich démontre que lorsque la taille des données devient très grande les mesures perdent leur pouvoir discriminant et il devient très difficile de comparer l'intérêt de deux règles [LVL05].

2.2.4.6 Sensibilité au bruit

Le bruit est présent dans toutes les données réelles : il est engendré par les erreurs de mesures, les erreurs de saisies, les données manquantes ou encore le prétraitement appliqué aux données. La sensibilité au bruit d'une mesure est donc un critère important. Cependant il demeure assez difficile de différencier le bruit des événements rares et intéressants (voir section 2.2.2.1). Une méthode est d'utiliser des mesures ne variant pas linéairement (voir section 2.2.3.4) mais il existe également d'autres approches basées sur les événements rares que nous présentons par la suite (voir section 2.4.2).

2.2.5 Formalisation matricielle de certains critères

Les mesures sont définies à partir des fréquences d'occurrence des items dans les données. Tan [TKS02] présente une approche dans laquelle, pour chaque règle d'association $A \rightarrow B$, ces fréquences peuvent être exprimées sous la forme d'une table de contingence 2x2 comme l'illustre le tableau 2.2.

	B	$\overline{B}$	
A	f_{11}	f_{10}	f_A
$\overline{A}$	f_{01}	f_{00}	$f_{\overline{A}}$
	f_B	$f_{\overline{B}}$	n

TAB. 2.2: Table de contingence 2x2 pour une règle $A \rightarrow B$ extraite de [TKS02] : chaque f_{ij} représente une fréquence d'occurence dans les données.

Tan définit les propriétés d'une mesure en utilisant une formulation matricielle dans laquelle chaque table de contingence est représentée par la matrice M donnée ci-dessous :

$$M = \begin{pmatrix} f_{11} & f_{10} \\ f_{01} & f_{00} \end{pmatrix}$$

Chaque mesure est représentée par un opérateur matriciel $\mathcal{O}$, qui transforme une matrice M en une valeur scalaire k tel quel $\mathcal{O}M = k$. On peut noter que A et B sont indépendants statistiquement lorsque le déterminant de la matrice est nul, c'est-à-dire lorsque : $Det(M) = f_{11}f_{00} - f_{01}f_{10} = 0$.

Tan formalise aussi 5 propriétés qui, pour certaines d'entre elles, correspondent aux critères définis de manière informelle dans la section précédente.

Symétrie de la mesure

Pour toute matrice de contingence M, $\mathcal{O}$ est dite symétrique si $\mathcal{O}(M^T) = \mathcal{O}(M)$; sinon elle est dite asymétrique. En effet, l'évaluation des règles $A \rightarrow B$ et $B \rightarrow A$ ne devrait pas donner le même résultat pour les deux règles. Si elles ont bien les mêmes exemples, leurs contre-exemples sont différents. Ainsi, une règle inversant la cause et la conséquence d'une règle R ne devrait absolument pas avoir le même intérêt que R ! (figure 2.4)

	B	$\overline{B}$
A	p	q
$\overline{A}$	r	s

	A	$\overline{A}$
B	p	r
$\overline{B}$	q	s

FIG. 2.4: Permutation des variables

Invariance lors du changement d'échelle des lignes et des colonnes

Soient $R = \begin{pmatrix} k_1 & 0 \\ 0 & k_2 \end{pmatrix}$ et $C = \begin{pmatrix} k_3 & 0 \\ 0 & k_4 \end{pmatrix}$ deux matrices carrées 2x2 où k_1, k_2, k_3 et k_4 sont des constantes positives. Le produit $R \times M$ correspond à un changement de l'échelle de la première ligne de la matrice par k_1 et de la deuxième ligne par k_2 tandis que le produit $M \times C$ correspond à un changement de l'échelle de la première colonne de la matrice par k_3 et de la deuxième colonne par k_4 (figure 2.5).

Pour toute matrice de contingence M, une mesure $\mathcal{O}$ est dite invariante lors d'un changement d'échelle des lignes et colonnes si $\mathcal{O}(RM) = \mathcal{O}(M)$ et si $\mathcal{O}(MC) = \mathcal{O}(M)$.

	B	$\overline{B}$
A	p	q
$\overline{A}$	r	s

	B	$\overline{B}$
A	k_3k_1p	k_4k_1q
$\overline{A}$	k_3k_2r	k_4k_2s

FIG. 2.5: Mise à l'échelle des lignes et des colonnes

Une mesure vérifiant cette propriété sera donc plus portée sur la comparaison de la proportion des effectifs et ne changera pas en fonction du changement de la taille de l'antécédent ou du conséquent d'une règle.

Antisymétrie lors d'une permutation de lignes et de colonnes

Soit $S = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ une matrice 2x2. Une mesure $\mathcal{O}$ normalisée (c'est-à-dire dont les valeurs sont comprises entre -1 et 1), est dite :

- antisymétrique lors d'une permutation de lignes si : $\mathcal{O}(SM) = -\mathcal{O}(M)$ pour toute matrice de contingence M (figure 2.6) ;
- antisymétrique lors d'une permutation de colonnes si : $\mathcal{O}(MS) = -\mathcal{O}(M)$ pour toute matrice de contingence M.

Une mesure antisymétrique lors d'une permutation de lignes et de colonnes ne permet pas de distinguer les corrélations positives des corrélations négatives entre

	B	$\overline{B}$
A	p	q
$\overline{A}$	r	s

	B	$\overline{B}$
A	r	s
$\overline{A}$	p	q

FIG. 2.6: Permutation des lignes

l'antécédent et le conséquent d'une règle. Il faut donc prendre ce critère en compte lors de l'interprétation des résultats !

Mesure invariante lors d'une inversion

Soit $S = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ une matrice de permutation 2x2. Pour toute matrice de contingence M, une mesure $\mathcal{O}$ est dite invariante lors d'une inversion si $\mathcal{O}(SMS) = -\mathcal{O}(M)$. Nous sommes dans un cas particulier de la propriété précédente où l'on effectue à la fois une permutation des lignes et des colonnes. Les mesures vérifiant cette propriété évaluent donc de la même façon les règles $A \rightarrow B$ et $\overline{A} \rightarrow \overline{B}$ (figure 2.7).

	B	$\overline{B}$
A	p	q
$\overline{A}$	r	s

	B	$\overline{B}$
A	s	r
$\overline{A}$	q	p

FIG. 2.7: Inversion

Mesure invariante lors de l'ajout d'exemples « vides »

Une mesure est invariante lors de l'ajout d'exemples « vides » si :
$\mathcal{O}(M + C) = \mathcal{O}(M)$, où $C = \begin{pmatrix} 0 & 0 \\ 0 & k \end{pmatrix}$ avec k une constante positive.

Cette opération consiste à ajouter des tuples qui ne contiennent aucun des items présents dans la règle considérée. Cette propriété est intéressante dans les domaines où la co-occurence des items est plus importante que leur absence simultanée (figure 2.8).

	B	$\overline{B}$
A	p	q
$\overline{A}$	r	s

	B	$\overline{B}$
A	p	q
$\overline{A}$	r	s+k

FIG. 2.8: Invariance lors de l'ajout d'exemples vides

L'approche de Tan définit donc des propriétés mathématiques permettant de vérifier la satisfaction d'un critère. L'antisymétrie et l'invariance au changement d'échelle, par exemple, correspondent à des critères importants.

2.2.6 Conclusion

Afin de répondre au mieux à la nécessité d'extraire des règles intéressantes, de plus en plus de mesures ont été créées sans toutefois qu'aucune ne fasse le consensus. Nous avons présenté différentes études qui, bien que possédant quelques points communs, constituent des approches différentes permettant de mettre en valeur une mesure plutôt qu'une autre en fonction de critères différents. Le choix d'une bonne mesure reste donc un problème largement ouvert qui, comme le note Lenca dans [LMV+04], résulte de plusieurs éléments :

- la difficulté à comparer des mesures caractérisées par des propriétés très différentes,
- la difficulté à exprimer les critères de l'utilisateur à l'aide de ces propriétés,
- la difficulté à intégrer des critères propres à l'utilisateur et à son domaine d'application, qui sont très subjectifs.

2.3 Travaux sur les mesures d'intérêt

Dans la section précédente, nous avons vu que les mesures sont souvent hétérogènes de par leur comportement et leurs propriétés. Ainsi se pose le problème du choix de la mesure adaptée aux besoins du processus de fouille et du domaine étudié. Pour cela, deux approches peuvent être envisagées : la première consisterait à faire le choix de la mesure la plus adaptée (sections 2.3.1 et 2.3.2), la deuxième approche, quand à elle, conduit à trouver un compromis entre plusieurs mesures (sections 2.3.3 et 2.3.4).

2.3.1 Agrégation de mesures

Barthelemy propose trois stratégies pour faire émerger un compromis entre mesures [BLLV06] :

- L'agrégation directe des mesures en une seule en utilisant une moyenne généralisée. La difficulté de cette approche étant de gérer les différences d'échelles entre mesures.
- L'agrégation des rangs induits par les différentes mesures en un rang unique. Cependant l'agrégation ordinale pose les mêmes problèmes que ceux présentés par Condorcet pour le choix d'un ordre social à partir d'ordres individuels.
- L'agrégation de relations valuées. Les classements sont des préordres totaux qui sont généralisés sous forme de relations valuées.

Barthelemy propose de mettre en œuvre la troisième stratégie. Les relations valuées permettent d'échapper aux effets d'échelle. Elles peuvent présenter de nombreuses propriétés comme la transitivité. L'un des avantages de la modélisation de relations valuées est la retranscription des évaluations numériques.

Étant donné, $\mathcal{R} = \{r_1, ..., r_k\}$ un ensemble de règles et, $\mu_1, ..., \mu_m$ les mesures sélectionnées, on considère que chaque mesure μ_j induit une relation valuée $\mathcal{R}_j$ sur $\mathcal{R}$.

L'idée générale est que la différence normalisée entre les valeurs prises par la mesure μ_j sur les règles r_i et $r_{i'}$ doit permettre de modéliser un système de préférences sur l'ensemble des règles.

La procédure d'agrégation produit une matrice carrée de valeurs entre 0 et 1, chaque valeur représentant la préférence agrégée d'une règle sur une autre. Les préférences sont alors représentées par des graphes où les arcs entre deux sommets représentent la préférence d'une règle sur l'autre, la flèche pointant sur la règle qui est préférée.

2.3.2 La nouveauté comme mesure unique

L'approche de Lavrač [LFZ99] a pour objectif de démontrer que quelques unes des mesures utilisées traditionnellement comme la confiance, la fiabilité négative, la sensibilité et la spécificité ne visent à exprimer qu'un seul critère : *la nouveauté*, qui selon elle joue un rôle central dans les mécanismes d'évaluation de règles.

Définition 2.6 (Nouveauté)
La nouveauté est une mesure relative dans le sens où elle compare le support d'une règle $A \rightarrow B$ à son support attendu sous l'hypothèse d'indépendance des attributs

A et B :

$$Nouveauté(A \rightarrow B) = p(AB) - p(A)p(B)$$

Lavrač propose alors d'exprimer les mesures traditionnelles relativement à un seuil exprimant la valeur attendue de la règle sous l'hypothèse d'indépendance (voir tableau 2.3).

Mesure \ Forme	Normale	Relative	Relative pondérée
Confiance	$p(B\|A)$	$p(B\|A) - p(B)$	$p(A) \times (p(B\|A) - p(B))$
Fiabilité négative	$p(\overline{B}\|\overline{A})$	$p(\overline{B}\|\overline{A}) - p(\overline{B})$	$p(\overline{A}) \times (p(\overline{B}\|\overline{A}) - p(\overline{B}))$
Sensibilité	$p(A\|B)$	$p(A\|B) - p(A)$	$p(B) \times (p(A\|B) - p(A))$
Spécificité	$p(\overline{A}\|\overline{B})$	$p(\overline{A}\|\overline{B}) - p(\overline{A})$	$p(\overline{B}) \times (p(\overline{A}\|\overline{B}) - p(\overline{A}))$

TAB. 2.3: Confiance, fiabilité négative, sensibilité et spécificité exprimée sous leur forme normale, relative et relative pondérée

La confiance relative, connue également sous le nom de confiance centrée, possède toutefois un inconvénient : il est facile d'obtenir des règles avec une confiance relative élevée et un support très faible. On introduit alors $p(A)$ dans la mesure afin de pondérer la confiance relative par le support de l'antécédent de la règle. De la même manière, $p(\overline{B})$ est introduit dans le calcul de la fiabilité négative car les règles générales ont une fiabilité négative élevée. L'utilisation de la pondération permet donc de lutter contre les solutions triviales.

Enfin, Lavrač démontre que ces quatre mesures relatives pondérées sont équivalentes à l'expression de *la nouveauté* qui est donc une mesure essentielle pour l'évaluation de l'intérêt des règles. Cependant Azé fait remarquer que cette mesure n'est pas adaptée à la recherche d'événements rares [Azé03].

2.3.3 Transformées affines de la confiance

Lallich et Teytaud montrent que de nombreuses mesures d'intérêt s'écrivent comme une normalisation de la confiance par le biais d'une transformation affine [LT04].

Si l'on considère une règle $A \rightarrow B$, on peut définir une transformation affine de la confiance comme une mesure m qui peut s'exprimer sous la forme $\theta_1(p(B|A) - \theta_0)$,

avec θ_0 et θ_1 ne dépendant que des marges relatives de la table qui croise A et B et éventuellement du nombre de tuples n présents dans la base de données [Azé03].

Par le biais de cette transformation, on corrige la principale critique faite à la confiance au sujet de la prise en compte de la taille de B. Cependant les mesures résultantes héritent de quelques unes des propriétés de la confiance :

- elles sont des fonctions affines du nombre d'exemples et de contre-exemples
- elles sont invariantes en cas de dilatation des données dans le cas où le facteur d'échelle ne dépend pas de n

2.3.4 Aide à la décision

Lenca et al. proposent d'utiliser un processus d'aide multicritères à la décision afin de prendre en compte les différents aspects des mesures sans négliger le rôle de l'utilisateur et ses besoins spécifiques [LMV+04]. Il présente quatre problèmes de référence relatifs à ce type d'approche :

1. Choisir un sous-ensemble de mesures en vue de faciliter un choix final.
2. Déterminer toutes les bonnes mesures en éclairant la décision par un tri résultant d'une affectation de chaque mesure à une catégorie.
3. Classer les mesures de la meilleure à la moins bonne.
4. Décrire les mesures et/ou leurs conséquences de façon formalisée.

Lenca définit cette approche dans le cadre des problèmes 1 et 3. Il propose de structurer le processus d'aide à la décision en quatre grandes étapes :

- Établir la liste $\mathcal{M}$ des mesures possibles.
- Établir la famille $\mathcal{F}$ des critères servant à évaluer les mesures.
- Évaluer les mesures de $\mathcal{M}$ sur les critères de $\mathcal{F}$ afin de bâtir des matrices de décision.
- Agréger les performances en fonction de la problématique choisie.

Les propriétés choisies pour évaluer les mesures sont traditionnelles et ont été présentées dans la section 2.2 :

1. traitement non symétrique de A et B,
2. décroissance avec n_B,

3. évaluation de l'indépendance,
4. évaluation des règles logiques,
5. non-linéarité de $p(A\overline{B})$ autour de 0^+,
6. prise en compte du nombre de cas n,
7. facilité à fixer un seuil,
8. intelligibilité de la règle.

Des valeurs et des évaluations ordinales sont proposées pour chacune de ces propriétés afin de les transformer en critères. Ensuite pour chacune des mesures, les critères sont évalués et le résultat est représenté au sein d'une matrice de décision. La méthode PROMETHEE-GAIA permet alors de déterminer des rangements et de les visualiser sur un plan [BM02].

2.3.5 Conclusion

Les approches présentées proposent des solutions très différentes cependant le sujet de recherche reste ouvert. D'autres méthodes tentent de parvenir à un compromis au moyen de solutions algorithmiques, comme les algorithmes génétiques que nous présentons dans la section suivante.

2.4 Mise en œuvre des mesures dans les algorithmes

Nous avons précédemment introduit les différents critères considérés pour construire une mesure d'intérêt ainsi que différentes approches afin de faire un choix parmi elles. La plupart de ces approches s'effectuent dans une phase de post-traitement lors de laquelle les règles précédemment générées sont filtrées selon une mesure particulière. Il est également possible d'intégrer un ensemble de mesures d'intérêt au sein d'un algorithme d'extraction. Nous nous intéressons ici aux algorithmes utilisés en classification ou encore dans la recherche d'associations [AIS93a, PBTL99a, PBTL99b, PBTL99c] pour l'extraction de règles. Dans cette section nous présentons trois approches algorithmiques qui mettent en œuvre plusieurs mesures d'intérêt. En premier lieu, la recherche de règles d'association, ensuite la recherche d'événements rares et enfin une approche évolutionnaire.

2.4.1 Algorithme d'extraction de règles d'association

Agrawal présente dans [AS94] une formalisation du problème d'extraction de règles d'association. Dans un premier temps nous introduisons ce formalisme afin d'évoquer ensuite les différentes approches envisagées pour la résolution de ce problème. Dans [Pas00] Pasquier présente une synthèse des algorithmes existants et en introduit de nouveaux (CLOSE, ACLOSE et CLOSE+) qui optimisent les performances et la taille des résultats. Il est important de noter que nos contributions présentées en deuxième partie se basent sur cette dernière approche.

Soit $\mathcal{I} = \{i_1, i_2, ..., i_m\}$ un ensemble de m littéraux, appelés **items**. Soit $\mathcal{B} = \{t_1, t_2, ..., t_n\}$ une base de données de n tuples, chaque **tuple** t_i étant constitué d'un sous-ensemble $I \subseteq \mathcal{I}$ d'items et identifié par un identifiant unique. Un sous-ensemble $I \subseteq \mathcal{I}$ de taille k est appelé un **k-itemset**. Un tuple t_i contient un itemset I si et seulement si $I \subseteq t_i$. Un itemset dont le support est supérieur ou égal au seuil minimal de support défini par l'utilisateur est appelé **itemset fréquent**.

Étant donnée une base de données de tuples $\mathcal{B}$, le problème de l'extraction des règles d'association dans $\mathcal{B}$ consiste à déterminer l'ensemble des règles d'association dont le support et la confiance sont supérieurs ou égaux à des seuils minimaux de support *minsupport* et de confiance *minconfiance* définis par l'utilisateur. Ce problème est en général divisé en deux sous-problèmes :

1. Déterminer l'ensemble des itemsets fréquents dans $\mathcal{B}$, c'est-à-dire les itemsets dont le support est supérieur ou égal à *minsupport*
2. Pour chaque itemset fréquent I_1, générer toutes les règles d'association de la forme $r : I_2 \rightarrow I_1 - I_2$ telles que $I_2 \subset I_1$ et dont la confiance est supérieure ou égale à *minconfiance*

Agrawal introduit le concept de contexte d'extraction de règles d'association. Un contexte d'extraction de règles d'association est un triplet $\mathcal{B} = (\mathcal{T}, \mathcal{I}, \mathcal{R})$ dans lequel $\mathcal{T}$ et $\mathcal{I}$ sont respectivement des ensembles finis de tuples et d'items et $\mathcal{R} \subseteq \mathcal{T} \times \mathcal{I}$ est une relation binaire entre les tuples et les items. Un couple $(t, i) \in \mathcal{R}$ dénote le fait que le tuple $t \in \mathcal{T}$ est en relation avec l'item $i \in \mathcal{I}$.

Étant donné un contexte d'extraction de règles d'association $\mathcal{B}$, la découverte des

OID	Items			
1	A	C	D	
2	B	C	E	
3	A	B	C	E
4	B	E		
5	A	B	C	E
6	B	C	E	

TAB. 2.4: Contexte d'extraction de règles d'association $\mathcal{B}$

itemsets fréquents est un problème non trivial car le nombre d'itemsets fréquents potentiels est exponentiel en fonction du nombre d'items du contexte $\mathcal{B}$.

Si l'on considère un contexte d'extraction $\mathcal{B} = (\mathcal{T}, \mathcal{I}, \mathcal{R})$ et un seuil minimal de support *minsupport*, l'ensemble F des itemsets fréquents dans $\mathcal{B}$ est :

$$F = \{l \subseteq \mathcal{I} \mid l \neq \emptyset \wedge support(l) \geq minsupport\}$$

Dans le cadre d'un ensemble d'items $\mathcal{I}$ de taille m, le nombre d'itemsets potentiellement fréquents est de 2^m. Ces itemsets forment le *treillis des parties* de $\mathcal{I}$, dont la hauteur est m+1.

Exemple 2.3
Le treillis des parties de l'ensemble d'items $\mathcal{I}$ du contexte $\mathcal{B}$ est représenté dans la figure 2.4. L'ensemble $\mathcal{I}$ contenant cinq items, ce treillis contient 32 itemsets et sa hauteur est égale à six.

La phase de découverte des itemsets fréquents constitue la phase la plus coûteuse en temps d'exécution de l'extraction de règles d'association du fait de cet espace de recherche de taille exponentielle dans le nombre d'items et de la nécessité de réaliser des balayages du contexte. Une approche simpliste consiste à tester le support de chacun des itemsets du treillis. Cette approche, bien que ne nécessitant qu'un seul balayage du contexte, est impraticable lorsque le nombre d'items est grand. Dans la section 2.4.1.1 nous présentons trois méthodes différentes permettant de réduire l'espace de recherche ainsi que le nombre de balayages du contexte.

Étant donné un ensemble F d'itemsets fréquents dans un contexte d'extraction $\mathcal{B}$ pour un seuil minimal de support *minsupport*, la génération des règles d'association pour un seuil minimal de confiance *minconfiance* est un problème exponentiel dans

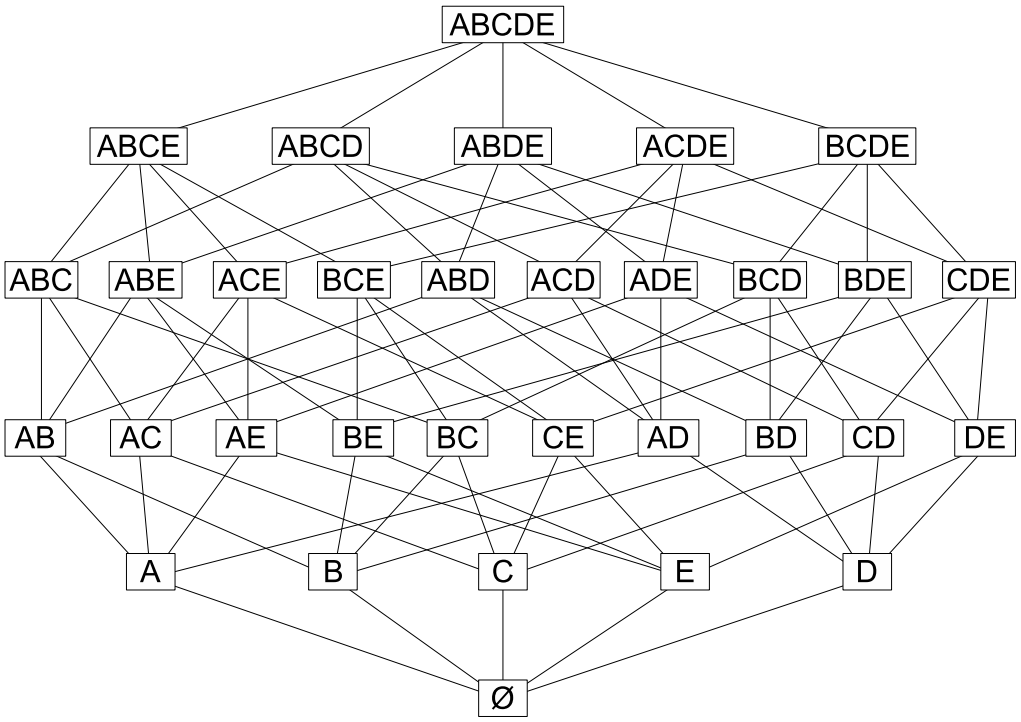


FIG. 2.9: Treillis des itemsets associé au contexte $\mathcal{D}$

la taille de F.

Soit un ensemble F d'itemsets fréquents dans un contexte d'extraction $\mathcal{B}$ pour un seuil minimal de support *minsupport*. Étant donné un seuil minimal de confiance *minconfiance*, l'ensemble $\mathcal{AR}$ des règles d'association valides dans $\mathcal{B}$ est :

$$\mathcal{AR} = \{r : l_2 \rightarrow (l_1 - l_2) | l_1, l_2 \in F \wedge l_2 \subset l_1 \wedge support(l_1)/support(l_2) \geq minconfiance\}$$

En pratique, la génération des règles d'association est réalisée de manière directe sans accéder au contexte d'extraction, et le coût de cette phase en temps d'exécution est donc faible comparé au coût de l'extraction des itemsets fréquents. Pour chaque itemset fréquent l_1 dans F, tous les sous-ensembles l_2 de l_1 sont déterminés et la valeur du rapport $support(l_1)/support(l_2)$ est calculée. Si cette valeur est supérieure ou égale au seuil de confiance *minconfiance* alors la règle d'association $l_2 \rightarrow (l_1 - l_2)$ est générée.

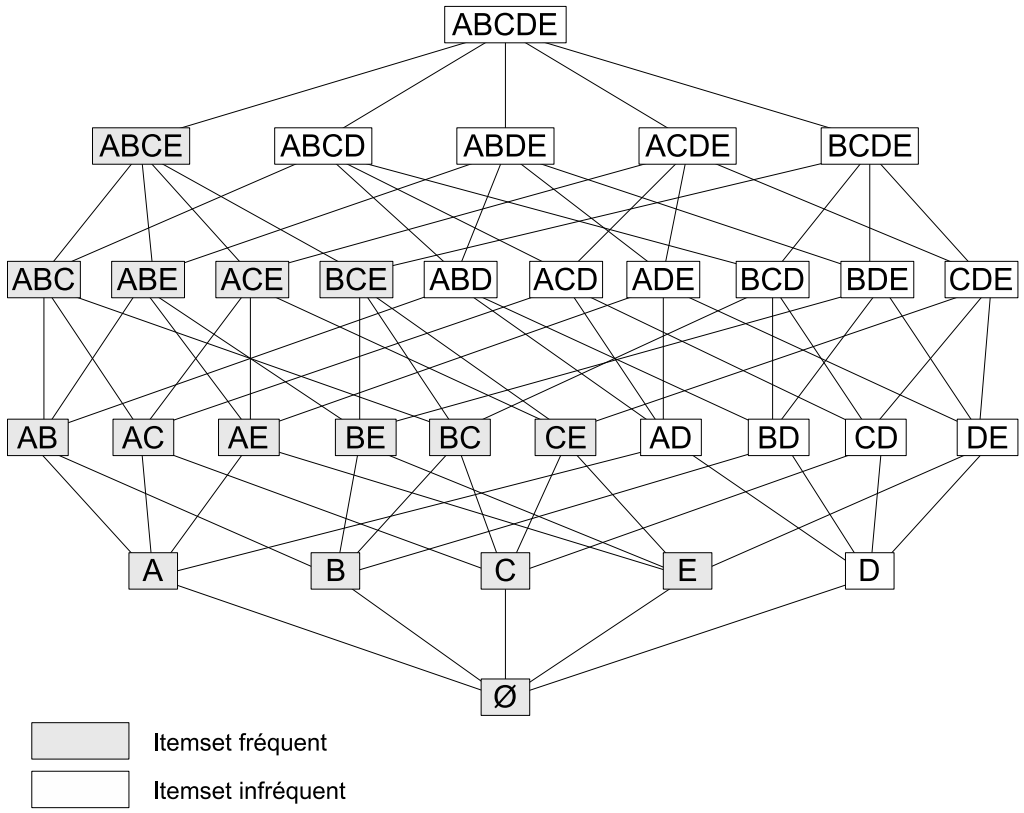


FIG. 2.10: Itemsets fréquents dans le treillis des itemsets associé au contexte $\mathcal{D}$ pour $minsupport = 2/6$

2.4.1.1 Extraction des itemsets fréquents

Les algorithmes d'extraction des itemsets fréquents procèdent de manière itérative, en parcourant le treillis des itemsets fréquents « par niveaux » : ils réalisent un parcours en largeur du treillis des itemsets fréquents, du bas vers le haut, en déterminant lors de chaque itération tous les itemsets fréquents d'un niveau, c'est-à-dire d'une taille donnée. Pour chaque itération k, un ensemble de k-itemsets candidats (k-itemsets fréquents potentiels) est généré et les supports de ces candidats sont calculés lors d'un seul et même balayage, ce qui permet de limiter le nombre total de balayages réalisés. Les premiers algorithmes d'extraction des itemsets fréquents « par niveaux » sont proposés en 1993 [AIS93b]. Ensuite, plusieurs autres algorithmes permettant de réduire les temps d'extraction des itemsets fréquents ont été proposés, parmi eux Apriori [AS94] et l'algorithme OCD [MTV94].

2.4.1.2 Extraction des itemsets fermés fréquents

Pasquier propose une nouvelle approche basée sur la connexion de Galois afin de résoudre le problème de l'extraction de règles d'association [PBTL98, PTB+05]. L'utilisation des opérateurs de fermeture de la connexion de Galois permet de définir les *itemsets fermés*, qui forment le *treillis des itemsets fermés*, et les *itemsets fermés fréquents* [YN04, YHN06]. Il est démontré que l'ensemble des itemsets fermés fréquents constitue un *ensemble générateur*, également appelé *base*, pour l'ensemble des itemsets fréquents. Cela signifie que les itemsets fréquents et leurs supports peuvent être générés à partir des itemsets fermés fréquents et leurs supports sans accéder à la base de données. Le problème de l'extraction des itemsets fréquents dans le contexte $\mathcal{B}$ est divisé en deux sous-problèmes :

1. Déterminer l'ensemble des itemsets fermés fréquents dans $\mathcal{B}$, c'est-à-dire des itemsets fermés dont le support est supérieur ou égal à minsupport.
2. Dériver l'ensemble des itemsets fréquents à partir de l'ensemble des itemsets fermés fréquents extraits durant la phase précédente. Cette phase consiste à générer tous les sous-ensembles des itemsets fermés fréquents maximaux et dériver leurs supports à partir des supports des itemsets fermés fréquents.

Cette nouvelle approche, qui a conduit au développement des algorithmes CLOSE, ACLOSE et CLOSE+, permet d'améliorer les temps de réponse car dans la plupart des cas le nombre d'itemsets fermés fréquents est bien inférieur au nombre d'itemsets fréquents. L'utilisation des itemsets fermés fréquents permet de définir des bases pour les règles d'association. Ces bases, qui sont des sous-ensembles de l'ensemble des règles d'association valides, permettent d'améliorer la pertinence et l'utilité de l'ensemble de règles extraites.

2.4.2 Algorithme de recherche d'événements rares

Selon Weiss [Wei95], bien que les événements rares ne couvrent individuellement qu'une faible fraction de tuples, ils peuvent en couvrir collectivement un pourcentage significatif. Le problème avec les événements rares est qu'ils ont un taux d'erreur plus important que les règles avec support élevé mais qu'ils ne peuvent pas être éliminés sans réduire de façon importante la précision d'un modèle. Cette difficulté réside dans le fait que les événements rares sont très sensibles aux données bruitées.

Apprendre à partir des données bruitées est très difficile car il n'est pas évident de faire la différence entre le bruit et une règle exceptionnellement vraie.

Un certain nombre de travaux ont été consacrés à ce sujet comme nous l'avons vu en section 2.2. Freitas traite ce problème en proposant une mesure de détection de la surprise basée sur ces événements rares [Fre99]. Cependant la mise au point de mesures d'intérêt n'est pas la seule façon de prendre en compte les événements rares, des algorithmes dédiés à leur détection sont également mis au point.

Azé propose dans sa thèse [Azé03] une solution innovante. La définition d'un support minimal, couramment utilisé afin de parcourir l'espace de recherche, est très contraignante pour l'expert qui, sans connaissance a priori sur les règles recherchées, est souvent incapable de définir cette valeur minimale. De plus, fixer une valeur faible pour ce seuil induit une asphyxie des algorithmes qui ne sont pas conçus pour cela. Azé suggère alors de ne pas définir ce support minimal et propose une approche permettant d'extraire un ensemble de règles intéressantes ayant la propriété d'être peu contredites par les données. Pour cela, il utilise la mesure de moindre contradiction qui possède deux atouts : elle est simple à comprendre par un utilisateur non expert et présente d'excellentes performances en terme de résistance au bruit, propriété nécessaire pour l'extraction d'événements rares.

La moindre contradiction mesure la différence entre le nombre d'exemples et de contre-exemples de la règle, cette différence est normalisée par le nombre d'exemples vérifiant la conclusion de $A \rightarrow B$:

$$contramin(A \rightarrow B) = \frac{p(AB) - p(A\overline{B})}{p(B)}$$

Cependant, cette mesure étant une transformation affine de la confiance, elle ne possède pas de propriété de monotonie ou d'anti-monotonie. Ainsi, il est difficile de rechercher de manière exhaustive l'intégralité des règles les moins contradictoires. Azé propose alors de restreindre la recherche aux règles d'association présentant les propriétés suivantes :

- se distinguer le plus des autres règles,
- être telles que les antécédents des règles ne contiennent pas plus de k_{max} attributs et telles que la conclusion de celles-ci soit réduite à un seul attribut.

Cette approche ne permet donc pas d'extraire toutes les règles d'association vérifiant les contraintes de l'utilisateur, mais seulement les plus significatives d'entre elles.

2.4.3 Approches évolutionnaires

Un algorithme génétique travaille sur une population de solutions potentielles, ces solutions étant également appelées *individus* ou *chromosomes* (par analogie à la biologie). Le processus conduit à l'élimination des individus les plus faibles pour favoriser la survie des individus les plus performants, c'est-à-dire les mieux adaptés.

Les algorithmes génétiques renvoient aux principaux mécanismes de la sélection naturelle, c'est-à-dire essentiellement la sélection, le croisement et la mutation. Un algorithme génétique décrit l'évolution, au cours de générations successives, d'une population d'individus en réponse à leur environnement. Il sélectionne les individus au moyen d'une *fonction de fitness*, en accord avec le principe de la survie du plus adapté. Comme leurs équivalents biologiques, les individus sont constitués d'un ensemble de gènes qui ont chacun un rôle propre. Dans une simulation génétique, les individus les plus adaptés ont une probabilité plus élevée d'être sélectionnés et reproduits, donc d'être présents à la génération suivante. L'opérateur de croisement permet de parcourir l'espace de recherche, quant à l'opérateur de mutation d'un gène il permet de maintenir une certaine diversité au sein de la population [Fra04].

La tâche d'apprentissage consistant à apprendre les meilleurs motifs ou modèles à partir des données peut être reformulée sous la forme d'un problème d'optimisation, où l'on cherche à optimiser une fonction qui est une mesure de la discrimination des exemples positifs ou négatifs, pondérée par des indices que l'on se donne. Ce point de vue est adopté dans un certain nombre de travaux en fouille de données [Wei99, RFP02, Bra02, FBC03a, FBC03b, Fra04] qui s'intéressent à l'apport d'algorithmes d'optimisation stochastiques, comme les algorithmes génétiques, pour l'apprentissage de descriptions de concepts en premier ordre à partir de données. De par leur fonctionnement, ces algorithmes se révèlent être lents et ne garantissent pas la découverte de la meilleure solution. Néanmoins, ils sont intéressants et connaissent un large succès car ils effectuent une recherche globale évitant ainsi les inconvénients d'une stratégie locale. Notons que leurs performances peuvent être améliorées par la parallélisation.

Weiss a développé un système d'apprentissage basé sur un algorithme génétique nommé Timeweaver afin de résoudre le problème d'identification des événements rares [Wei99]. Il applique son système à la prédiction de pannes dans un réseau. Les jeux de données du domaine étant très volumineux, l'algorithme génétique doit donc minimiser le nombre de motifs à évaluer. Pour cela une sélection *steady state* est utilisée c'est-à-dire que seuls les plus mauvais individus sont remplacés. Ainsi il n'est pas nécessaire à chaque génération de réévaluer toute la population. Ensuite, dans le contexte du domaine étudié, la précision d'une prédiction n'est pas une mesure adéquate pour la fonction de fitness car la stratégie de ne jamais prédire un événement est la plus précise. Weiss propose donc d'utiliser les mesures de support et de confiance. L'objectif étant d'extraire également des événements rares intéressants, il est nécessaire de prendre en compte le problème du bruit. Pour cela un langage flexible et tolérant au bruit est proposé afin d'exprimer les règles à l'origine des différents événements.

L'approche de Carvalho diffère de celle de Weiss bien que la finalité soit la même : la prédiction des événements rares [CF00, CF02]. Carvalho utilise dans un premier temps, l'algorithme C4.5 afin de produire un arbre de décision. Cet arbre de décision est ensuite transformé en un ensemble de règles dont certaines sont considérées comme des événements rares. L'algorithme génétique mis au point permet alors de rechercher les individus qui prédisent au mieux ces événements rares.

2.5 Conclusion

Dans ce chapitre nous avons présenté un panorama des différentes études concernant les mesures d'intérêt objectif. De multiples critères de qualité ont été énoncés pour faire un choix parmi les mesures ; cependant aucun consensus n'a pu être trouvé sur la mesure la plus appropriée dans un contexte donné. Des méthodes d'aide à la décision ont été proposées afin d'aider un utilisateur à effectuer un choix : cela met en évidence le rôle primordial de l'utilisateur afin d'exprimer la notion d'intérêt. Dans le chapitre suivant nous présentons les approches subjectives pour l'évaluation des modèles extraits.

Chapitre 3

Approches subjectives pour l'extraction de connaissances

Sommaire

3.1 Intérêt subjectif

Les mesures d'intérêt subjectif ont pour vocation de quantifier à quel point un motif peut intéresser un utilisateur. La difficulté dans la mise au point de ce type de mesure est que la notion d'intérêt dépend du temps, du domaine étudié, de l'utilisateur et de ses objectifs. Parfois, les utilisateurs ne savent pas eux-mêmes ce qui peut les intéresser et n'ont donc aucun a priori sur les résultats [Bri04]. Il est cependant possible de discerner deux grandes catégories de motifs intéressants : les motifs qui vont surprendre les utilisateurs, et les motifs utilisables par les utilisateurs [ST96].

Une première façon d'aborder le problème est de considérer une règle intéressante si elle est inattendue ou inconnue de l'utilisateur. Ce concept d'inattendu se base sur le fait que si une règle surprend l'utilisateur elle sera nécessairement intéressante. Ces règles inattendues sont donc intéressantes dans la mesure où elles contredisent nos « convictions » [PT99, PT00, LHM01]. Une règle est également intéressante si l'utilisateur peut grâce à elle agir et en tirer un avantage. L'utilisabilité est une mesure d'intérêt subjective importante car beaucoup d'utilisateurs sont souvent intéressés par des connaissances leur permettant de mettre en œuvre des actions appropriées à l'accomplissement de leurs objectifs [AT97, PSM94, ST95, ST96].

Selon Silberschatz [ST95] tous les motifs utilisables sont inattendus, par conséquent l'étude de l'inattendu suffirait à extraire des motifs intéressants. Il n'est toutefois pas possible de généraliser cette affirmation à tous les domaines : certains motifs attendus et utilisables sont intéressants car les utilisateurs peuvent ne pas y avoir porté attention. L'objectif n'est donc pas seulement de montrer des choses intéressantes et surprenantes mais aussi de mettre en évidence les motifs intéressants *évidents et utilisables* qui sont négligés par les utilisateurs.

Liu propose une technique permettant d'éliminer les règles non-utilisables qui ont été considérées comme « inintéressantes » par les différentes méthodes d'élagage [LHM01].

Exemple 3.1 (Prédiction des risques cardio-vasculaires)
Prenons pour exemple une base de données de 1000 tuples. Nous avons un attribut cible « Maladie » pouvant prendre deux valeurs, « Oui » et « Non », exprimant si

un patient est atteint d'une maladie. Parmi les 1000 patients, 500 ont la maladie et les 500 autres sont sains. Après une fouille de données effectuée sur ces données médicales relatives aux risques cardio-vasculaires l'algorithme d'extraction produit les trois règles suivantes :

Règle R1

Tension Artérielle="Elevée" → Maladie="Oui"

- support = 6%
- confiance = 60%

Règle R2

Tension Artérielle="Elevée" ∧ Sexe="Masculin" → Maladie="Oui"

- support = 3,6%
- confiance = 90%

Règle R3

Tension Artérielle="Elevée" ∧ Niveau Glucose="Anormal" → Maladie="Oui"

- support = 3%
- confiance = 100%

Nous rapellons la définition de *couverture* d'une règle adoptée par l'auteur qui correspond au nombre de tuples vérifiants l'antécédent de la règle. Si l'on considère une règle $R : A \rightarrow B$ on a : $couverture(R) = support(R)/confiance(R)$.

Dans cet exemple, la règle R1 a un support de 6%, soit 60 tuples, et une couverture de 100 (60 * 100 / 60). La règle R2 a un support de 3,6%, soit 36 tuples, et une couverture de 40 (36 * 100 / 90). La règle R3 a un support de 3%, soit 30 tuples, et une couverture de 30 (30 * 100 / 100). Une interprétation graphique des règles permet de dessiner la figure 3.1 extraite de [LHM01]. Sur cette figure les hachûres diagonales représentent les tuples couverts par la règle R1, tandis que les traits horizontaux représentent les tuples pour lesquels la valeur de « Maladie » est « Oui ».

Supposons que le nombre de tuples couverts par R2 ou R3 avec « Maladie=Oui » soient au nombre de 58. La couverture de R2 ou R3 comprenant tous les tuples (in-

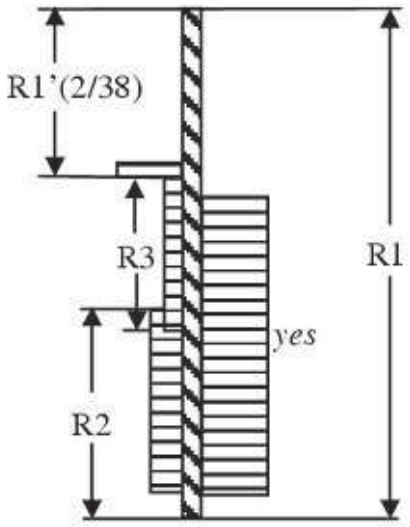


FIG. 3.1: Représentation des règles R1, R2, R3 et R1' = R1 - (R2 ∪ R3)

cluant « Maladie=Oui » et « Maladie=non ») est donc de 62 (car R2 couvre 4 tuples avec « Maladie=non »). Etant donné que R2 et R3 ont une meilleure confiance que R1, ces deux règles seront utilisées en premier dans une application. Cela implique que les tuples restant couverts par R1 - (R2 ∪ R3) sont : 2 tuples avec « Maladie=oui » (60 tuples de R1 avec « Maladie=oui » moins les 58 de R2 ∪ R3) et 36 tuples avec « Maladie=non » (40 tuples de R1 avec « Maladie=non » moins les 4 de R2 ∪ R3). Sur la figure 3.1 nous avons R1' = R1 - (R2 ∪ R3). La règle R1' n'est pas significative car sa confiance est faible (2 / (2+36) = 5,3%) comparée à la confiance que toute la population soit malade une fois que les règles R2 et R3 aient été appliquées : (500-58)/(1000-63) = 47%. La règle R1' est donc pire qu'un choix aléatoire et par conséquent elle n'est pas utilisable. La règle R1 correspondant à la règle R1' une fois les règles R2 et R3 appliquées n'est donc pas, à son tour, utilisable.

Il ne s'agit cependant pas d'éliminer toutes les règles ayant une faible confiance (des règles générales le plus souvent) mais d'éliminer les règles générales non-intéressantes à la lumière de règles plus précises !

Laurent BRISSON

3.2 Les patrons

Klemettinen propose que les informations intéressantes et inintéressantes soient spécifiées au moyen de patrons, appelés également *templates* en anglais [KMR+94]. Les patrons décrivent un ensemble de règles spécifiant les attributs apparaissant dans l'antécédent et un attribut prédictif dans le conséquent. Chacun des attributs appartient également à une hiérarchie de concepts.

Exemple 3.2 (Cours d'informatique à l'université)
Considérons que les cours se divisent en trois niveaux : débutant, avancé et confirmé. On peut considérer la hiérarchie suivante :

- {Programmation en Pascal, Systèmes d'information} $\subset$ Niveau débutant $\subset$ Tous les cours
- {Programmation en C, Bases de données} $\subset$ Niveau avancé $\subset$ Tous les cours
- {Algorithmique, Réseaux de neurones, Intelligence Artificielle} $\subset$ Niveau confirmé $\subset$ Tous les cours

Klemettinen définit un patron comme une expression $A_1 \wedge ... \wedge A_k \rightarrow A_{k+1}$ où chaque A_i est soit une valeur, soit une expression $C+$ ou $C*$ où $C+$ signifie au moins une instance d'une classe C et $C*$ aucune ou plusieurs instances de C. Une règle de la forme $B_1 \wedge ... \wedge B_k \rightarrow B_{k+1}$ vérifie un patron si elle peut être considérée comme une instance du patron.

Patron : Niveau confirmé $\wedge$ Tous les cours* $\rightarrow$ Algorithmique

Un patron est dit *inclusif* s'il correspond à une notion intéressante pour l'utilisateur et *exclusif* dans le cas contraire. Les règles A et B suivantes vérifient le patron précédent :

Règle A : Réseaux de neurones $\rightarrow$ Algorithmique
- Support : 2%
- Confiance : 48%

Règle B : Intelligence Artificielle $\rightarrow$ Algorithmique
- Support : 2%
- Confiance : 48%

Avec les patrons, un utilisateur peut spécifier explicitement à la fois ce qui l'intéresse ou pas. Afin d'être intéressante une règle doit vérifier un patron inclusif. Si elle vérifie un patron exclusif elle est considérée comme inintéressante et ne sera pas présentée à l'utilisateur. La limitation de cette approche réside dans la définition des patrons. Il arrive souvent que les utilisateurs ne sachent pas ce qui les intéresse. De plus avec le système de « matching » envisagé de nombreuses règles générées par la fouille de données sont écartées.

3.3 Les convictions

La notion d'inattendu est très liée aux convictions d'un utilisateur. Ainsi il est important de mettre au point un système de convictions cohérent avec la façon de penser des utilisateurs.

3.3.1 La création d'un ensemble de convictions

L'approche de Silberschatz [ST95, ST96] Les convictions sont définies par Silberschatz comme des expressions logiques du premier ordre auxquelles on associe un « degré de conviction ». On peut classer les convictions en deux catégories :

- Les convictions « fortes » : quelles que soient les nouvelles connaissances acquises ou les règles générées les convictions fortes demeurent exactes.
- Les convictions « faibles » : ce sont des convictions que l'utilisateur aimerait confirmer ou voir évoluer selon les connaissances nouvellement acquises.

Silberschatz propose différentes méthodes pour évaluer le degré de conviction :

- Approche Bayésienne : le degré de conviction est défini comme une probabilité conditionnelle que la conviction existe étant données certaines conditions supportant cette conviction.
- Approche de Dempster-Shafer : le degré de conviction est la somme de toutes les probabilités des événements B qui impliquent A.
- Approche fréquentielle : pour une conviction $A \rightarrow B$ le degré de conviction est le rapport du nombre d'exemples satisfaisant A et B avec le nombre d'exemples ne satisfaisant que A.

L'approche de Padmanabhan[PT99, PT00] Les convictions s'expriment pour Padmanabhan sous la même forme que les règles générées. Padmanabhan ajoute une propriété de monotonie qui doit être vérifiée par les convictions : si une conviction est supposée être vérifiée sur un jeu de données $\mathcal{D}$ avec un support s alors cette conviction doit être également vérifiée sur un sous-ensemble de $\mathcal{D}$ assez grand statistiquement avec un support $s' \geq 0,5$. Par exemple la règle 1 n'est pas monotone à cause de la règle 2 et devrait être plutôt remplacée par la règle 3.

Règle 1 oiseau(X) $\rightarrow$ vole(X)

Règle 2 oiseau(X) $\wedge$ manchot(X) $\rightarrow \neg$ vole(X)

Règle 3 oiseau(X) $\wedge\neg$ manchot(X) $\rightarrow$ vole(X)

3.3.2 Le processus de raffinement

La découverte de règles inattendues et le raffinement de la connaissance sont deux aspects importants d'un processus plus global [ST95, PT00]. Les connaissances a priori d'un domaine sont basées sur l'expérience passée et, dans certains cas, les conditions ont pu changer et les connaissances doivent être remises en question ! Dans le cadre du système proposé par Silberschatz si une règle contredit :

- une conviction forte : cela met évidence une règle fausse ou un problème dans le recueil des données ;
- une conviction faible : s'il n'y a pas d'erreurs un véritable cas inattendu a peut-être été découvert et dans ce cas il est important de mettre à jour la conviction !

Le principal problème réside au niveau du classement des deux types de convictions.

3.4 Les attentes de l'utilisateur

Cette méthode demande une participation active de l'utilisateur qui doit exprimer de façon basique une partie de ses connaissances. La méthode de Liu consiste

à extraire les motifs qui vont correspondre aux attentes de l'utilisateur. Dans cette première phase aucun aspect inattendu ou utilisable est introduit [LHML99].

L'utilisateur doit fournir un ensemble de règles avec la même syntaxe que les règles générées. Les règles sont considérées comme des règles floues qui sont définies par des variables floues. Une variable linguistique floue est définie par le quintuplet : $(x, T(x), U, G, M)$ avec :

- x : le nom de la variable
- T(x) : l'ensemble des termes définissant les valeurs de x
- U : l'ensemble de définition de la variable
- G : règle syntaxique générant le nom X des valeurs de x
- M : règle sémantique associant à X un sens

Exemple 3.3
Considérons la *vitesse* comme une variable linguistique floue avec $U = [1, 140]$.
Les termes *T(vitesse)* sont représentés par l'ensemble *{lente, modérée, rapide}*.
$M(X)$ donne une signification à chacun des termes. Par exemple, $M(lente)$ peut être définie par :

$$M(lente) = \{(u, \mu_{lente}(u)) \mid u \in [1, 140]\}$$

où :

- $\mu_{lente}(u) = 1$, si $u \in [1, 30[$
- $\mu_{lente}(u) = -\frac{1}{30}u + 2$, si $u \in [30, 60[$
- $\mu_{lente}(u) = 0$, si $u \in [60, 140[$

$\mu_{lente}(u)$ exprime le degré d'appartenance de u au terme *lente*.

Un système de logique floue extrait alors les règles qui correspondent aux attentes de l'utilisateur. Afin d'extraire les règles utilisables, l'utilisateur spécifie dans un premier temps toutes les actions qu'il est en mesure d'effectuer. Par exemple dans le cadre de la sécurité routière un responsable pourrait envisager les actions suivantes :

Action 1 : Inciter la population à rester prudente en conduisant, même dans les zones où la visibilité est bonne : Zone="Bonne visibilité" → Blessure="Légère"

Action 2 : Placer des radars dans les zones à faible visibilité Zone="Mauvaise visibilité" ∧ Vitesse="Rapide" → Blessure="Mortelle"

Pour chaque action l'utilisateur spécifie les situations pour lesquelles il pourra effectuer l'action. Les situations sont représentées par un ensemble de règles floues dans lesquelles les valeurs des variables sont représentées par une classe, par exemple : "Mauvaise visibilité", "Rapide", "Lent". Il est important de remarquer que l'utilisateur ne doit pas fournir des situations qu'il pense possibles mais toutes les situations pour lesquelles une action est possible ! Un système de logique floue extrait alors les règles qui correspondent aux situations envisagées par l'utilisateur :

Règle 1 : Age>50 $\wedge$ Zone="Ligne droite" $\rightarrow$ Blessure="Légère"

Règle 2 : Age>65 $\wedge$ Zone="Virage" $\wedge$ Vitesse>90 $\rightarrow$ Blessure="Mortelle"

Règle 3 : Age>50 $\wedge$ Zone="Croisement" $\rightarrow$ Blessure="Légère"

Ces règles sont ensuite classés par niveau d'intérêt pour chacune des actions :

- Action 1 :
  - Rang 1 : Règle 1 : Age>50 $\wedge$ Zone="Ligne droite" $\rightarrow$ Blessure="Légère"
  - Rang 2 : Règle 3 : Age>50 $\wedge$ Zone="Croisement" $\rightarrow$ Blessure="Légère"
  - Rang 3 : Règle 2 : Age>65 $\wedge$ Zone="Virage" $\wedge$ Vitesse>90 $\rightarrow$ Blessure="Mortelle"
- Action 2 :
  - Rang 1 : Règle 2 : Age>65 $\wedge$ Zone="Virage" $\wedge$ Vitesse>90 $\rightarrow$ Blessure="Mortelle"
  - Rang 2 : Règle 3 : Age>50 $\wedge$ Zone="Croisement" $\rightarrow$ Blessure="Légère"
  - Rang 3 : Règle 1 : Age>50 $\wedge$ Zone="Ligne droite" $\rightarrow$ Blessure="Légère"

Avec les résultats précédents, l'utilisateur peut alors décider de mieux inciter la population âgée d'être plus prudente au volant et/ou d'installer des radars dans les virages.

Cette technique a l'avantage de permettre de trouver les règles utilisables mais aussi de déterminer l'action à entreprendre, l'inconvénient est qu'il est rare qu'un utilisateur puisse appréhender l'ensemble des situations possibles.

Laurent BRISSON

3.4.1 L'extraction de règles inattendues

L'extraction de règles inattendues est similaire à l'extraction de règles utilisables. La nuance se situe au niveau de la fonction de « matching » qui est remplacée par un moyen d'évaluer le degré de surprise de la règle générée. Lors de la comparaison, Liu fait la différence entre conséquence inattendue et cause inattendue et évalue différemment le degré de surprise selon qu'il y ait une contradiction entre les attributs ou juste des attributs identiques aux valeurs différentes.

3.5 Les impressions générales

Selon Liu [LHC97] il existe deux principaux types de concepts :

- Les impressions générales : l'utilisateur n'a pas de concept détaillé d'un domaine mais il a quelques vagues sentiments.
- Les connaissances relativement précises : l'utilisateur a une idée très précise des choses.

Exemple 3.4
Dans le domaine de l'accord de crédit les définitions précédentes pourraient s'illustrer de la façon suivante :

- Une impression générale : « Plus les revenus sont importants plus les chances d'accorder un crédit sont élevées ».
- Une connaissance précise : « Si les revenus mensuels sont supérieurs à 5000€ alors le crédit est toujours accordé ».

3.5.1 Représentation des impressions générales

Liu n'a étudié les impressions générales que dans le cadre particulier des règles de classement. Par rapport à la méthode basée sur les *attentes de l'utilisateur* celle-ci met à disposition une plus grande variété d'opérateurs pour nuancer les différentes expressions :

- a $< \rightarrow$ C : plus a est petit plus il y a de chances que l'on ait C
- a $> \rightarrow$ C : plus a est grand plus il y a de chances que l'on ait C
- a « $\rightarrow$ C : si a est dans la moyenne il y a des chances que l'on ait C
- a $| \rightarrow$ C : s'il existe une relation avec a il y a des chances que l'on ait C

- a [S] → C : si a appartient à S il y a des chances que l'on ait C

3.5.2 Évaluation des règles

La méthode permettant de repérer les règles inattendues est semblable à celle utilisée pour les « attentes de l'utilisateur » à quelques différences près dues à la structure des connaissances. Cette méthode, proposée par Liu, se divise en deux étapes :

- L'utilisateur définit toutes les impressions générales qu'il a sur le domaine grâce au langage de spécification présenté précédemment.
- Le système analyse les règles découvertes et les compare de différentes façons pour découvrir des règles intéressantes et les évaluer.

La comparaison des règles aux impressions permet de mettre en évidence les règles qui confirment une impression, les règles dont le conséquent est inattendu et les règles dont l'antécédent est inattendu.

3.6 Les prédictions floues

Romão [RFP02] propose une approche utilisant des techniques évolutionnaires afin de découvrir des règles de prédiction floues. La logique floue est selon les auteurs particulièrement bien adaptée à la description de situations réelles car elle permet d'exprimer de façon flexible les incertitudes auxquelles on peut être confrontées. Romão conçoit un algorithme génétique de manière à extraire les règles ayant une bonne précision mais qui présentent aussi de l'intérêt pour l'utilisateur. Dans ces travaux l'intérêt des utilisateurs est assimilé à la notion de surprise, et pour mettre en œuvre cette approche les impressions générales définies par Liu (voir section 3.5) sont utilisées. Toutefois à la différence de Liu l'approche évolutionnaire n'utilise pas les impressions générales comme moyen de filtrage des règles générées mais les incorpore au cœur de l'algorithme afin de ne pas générer des règles inintéressantes.

Chaque individu de l'algorithme représente une règle de prédiction, toutefois seul l'antécédent est codé dans le génome, le conséquent étant fixé pour une exécution de l'algorithme génétique. Le génome est de taille fixe et chaque gène représente un attribut possible associé à une valeur booléenne indiquant si cet attribut est

utilisé ou non dans le génome de l'individu. La fonction de fitness utilisée est le produit d'un indice de précision et d'un indice d'intérêt. L'indice d'intérêt est évalué en comparant la règle (caractérisée par le génome) aux impressions générales, et permet de caractériser la notion de surprise. Plus il y a de similarités entre les antécédents de la règle et des impressions générales et plus il y a de contradictions entre leurs conséquents, plus le degré de surprise est élevé.

3.7 Conclusion

Les approches subjectives complètent les mesures d'intérêt objectif qui sont basées sur des critères statistiques. Si la plupart des approches se positionnent dans une phase de post-traitement en filtrant les règles intéressantes à partir d'un ensemble de règles générées (et donc déjà filtrées par les mesures objectives), certaines approches permettent d'intégrer les deux types de mesures au sein d'un système dans le but de ne générer que des règles intéressantes en une seule phase [Cou05].

L'intérêt de l'utilisateur est très souvent assimilé à la notion de surprise et à la notion d'utilisable. Cet intérêt est généralement défini au sein de règles plus ou moins complexes et parfois associé à divers indicateurs exprimant leur degré de certitude. Au final, même si ils présentent quelques différences, les patrons, les convictions et les impressions générales sont des systèmes très similaires.

Les limitations des approches présentées dans ce chapitre sont principalement dues à certains choix conceptuels et à la manière de représenter et d'utiliser la connaissance dans les mécanismes de comparaison. Le choix conceptuel qui est susceptible de poser problème réside dans la nécessité pour l'utilisateur expert du domaine d'exprimer a priori son intérêt pour telle ou telle information. Les experts d'un domaine ont généralement beaucoup de difficultés à exprimer leur connaissance de manière formelle ; aussi les contraindre à créer des filtres pour les règles à extraire s'avère être une tâche difficile. Dans ce sens les règles floues apportent une solution, cependant la représentation des connaissances utilisée n'est pas très riche sémantiquement et mériterait de profiter des dernières avancées dans le domaine de l'ingénierie des connaissances. L'utilisation d'ontologies pourrait permettre d'exprimer un grand nombre de relations entre concepts (et pas uniquement des relations

de généralisation) ce qui permettrait leur utilisation au sein des mécanismes de comparaison.

Une évolution possible pour le développement de mesures d'intérêt subjectif serait donc de les appliquer dans le cadre d'algorithmes utilisant une représentation évoluée des connaissances du domaine et capables de cerner l'intérêt évoluant dans le temps de différents utilisateurs.

Chapitre 4

Utilisation des connaissances en fouille de données

Sommaire

4.1 Expression formelle des connaissances

4.1.1 Les langages de représentation des connaissances

L'élément central de tout système basé sur des connaissances est une *base de connaissances*. Les connaissances sont exprimées dans la base au moyen d'un langage de représentation. De nouvelles connaissances peuvent être ajoutées à la base afin que le système soit plus précis et plus complet, et des requêtes permettent d'interagir avec la base de connaissances pour en extraire les informations. Le choix du langage de représentation des connaissances va déterminer la représentation syntaxique et sémantique des entités, des événements, des actions et des processus au sein de la base. Toutefois, tous les langages de représentation des connaissances ne permettent pas de mettre en œuvre les différentes techniques de représentation des connaissances comme les règles, les frames ou les réseaux sémantiques. Les langages de représentation des connaissances se caractérisent par leur *expressivité*, c'est-à-dire leur capacité à permettre d'exprimer facilement les connaissances, et leur *compréhensibilité* par les humains.

4.1.1.1 Les langages basés sur les frames

Les langages à base de frames ou encore langages de frames fournissent une structure de données pour représenter des concepts ou des classes, appelées *frame*, et leurs instances (des objets ou individus). Une frame possède un ensemble de *slots* ou *propriétés* qui sont en fait des structures complexes auxquelles sont attachées des *facettes*. Les facettes sont des procédures activées contextuellement et permettant de calculer la valeur du slot. Les langages de frames s'inspirent du paradigme orienté objet tout en fournissant un support pour la description de hiérarchies et pour permettre le raisonnement sur les connaissances. Grâce à leurs propriétés les langages de frames sont donc très intéressants à mettre en œuvre. L'exemple 4.1 illustre une frame au moyen du langage KM [CP04].

Exemple 4.1
Un achat est caractérisé par :

- Un vendeur et un client (tous les deux du type *agent*)
- Un produit acheté
- Une somme d'argent équivalent au prix de l'objet

- Deux événements lors desquels :
  1. Le client donne de l'argent au vendeur
  2. Le vendeur donne le produit au client

```
(Buy has (superclasses (Event)))      ; Properties of the class
                                      ; (‘owns’ properties)
(chaque Achat a                        ; Properties of its members
  (client ((un Agent)))                 ’ (‘template’ properties)
  (produit ((une Chose)))
  (vendeur ((un Agent)))
  (argent ((le prix de (le produit de Achat))))
  (sous-événement1 ((un Echange avec
                (agent ((le client de Achat))
                (objet ((l’argent de Achat)))
                (destinataire ((le vendeur de Achat))))))
  (sous-événement2 ((un Echange avec
                (agent ((le vendeur de Achat)))
                (objet ((le produit de Achat)))
                (destinataire ((le client de Achat)))))))
```

Les langages de frames présentent toutefois quelques limitations. Tout d'abord les nombreux langages existants, même s'ils ont l'air semblables à première vue, peuvent structurer la connaissance différemment ce qui rend difficile l'interopérabilité des systèmes utilisant différents langages de frames. De plus s'ils sont bien adaptés à la représentation des connaissances, les langages de frames ont toutefois des difficultés à modéliser des changements non-monotone dans les connaissances. De façon plus générale, un problème crucial de définition d'un système formel de raisonnement non-monotone est de définir syntaxiquement ce qui reste vrai et ce qui devient faux lorsqu'une des formules des prémisses est modifiée, validée ou invalidée [Fab96]. Enfin, au sein des facets, les procédures sont codées en dur dans d'autres langages et ainsi certaines connaissances ne sont pas exprimées dans le même formalisme. Ainsi les systèmes basés sur les frames peuvent raisonner avec mais non pas à propos des connaissances.

4.1.1.2 Les langages basés sur la logique

Les langages basés sur la logique satisfont plusieurs des critères qui caractérisent un bon langage de représentation des connaissances. Tout d'abord leur sémantique est déclarative, c'est-à-dire que l'expression des connaissances dans la base est totalement indépendante des applications qui vont opérer sur cette base. Ensuite ils fournissent un support pour des mécanismes de récupération des connaissances dans la base. Il est possible de retrouver, par inférence, des connaissances qui ne sont exprimées qu'implicitement et cela uniquement grâce à la sémantique des informations stockées.

Le principal reproche fait aux langages basés sur la logique est leur incapacité à gérer les connaissances incomplètes ou contradictoires ainsi que les connaissances subjectives ou dépendantes du temps. Toutefois, Baadar [Baa99] souligne que ces reproches ne sont valables que dans le cadre de la logique des prédicats et que les approches basées sur les logiques modales et non-monotones apportent une solution à ces problèmes.

Parmi les langages basés sur la logique nous pouvons citer :

- COL, un langage basé sur la logique pour les objets complexes [AG90],
- SAFIN, un langage mis au point pour développer des logiciels basés sur des réseaux d'agents [XZF98]
- Delegation Logic, un langage utilisé pour résoudre des problèmes de délégation d'autorité dans des systèmes distribués [LFG99].

4.1.1.3 Les langages basés sur les règles

Les langages de représentation à base de règles sont très populaires. Ils ont le mérite d'être très faciles à comprendre par des utilisateurs non experts et il existe de nombreux outils pour construire des systèmes de règles. Tous les langages à base de règles permettent de représenter une structure de causalité de type « Si ... alors ... ». Cependant, entre les différents langages à base de règles il peut exister de très nombreuses différences syntaxiques.

Les langages à base de règles peuvent avoir des niveaux d'expressivité différents, par exemple ils peuvent être déclaratifs ou conçus pour des systèmes de production. Parfois les nouveaux faits ne peuvent être ajoutés que dans la partie « alors » des règles alors que certains langages permettent l'ajout de code procédural pouvant

être déclenché de façon contextuelle [DDV06].

Les langages à base de règles possèdent deux principales limitations [Wel96] :

- Dans le cas de problèmes complexes où la base de règles est très importante, il n'y a pas de méthode afin d'ordonner les règles.
- Les inférences ne peuvent être limitées aux règles manipulant seulement les objets qui sont intéressants dans un contexte donné.

Les langages à base de règles sont très utilisés dans le domaine de l'ingénierie web, toutefois le manque d'interopérabilité pose problème. Afin d'y remedier le standard RuleML homogénéise la syntaxe des règles. RuleML permet d'encoder différentes formes de règles en XML et RDF afin de réaliser des tâches de transformations et de déductions. Les catégories de règles supportées par RuleML sont les suivantes :

1. Production : règles de la forme « Si ... alors ».
2. Réaction : règles vérifiant des conditions avant de déclencher des actions.
3. Transformation : règles de réécriture utilisées lorsque certaines conditions sont vérifiées.
4. Dérivation : règles de déduction utilisées lorsque certaines conditions sont vérifiées.
5. Intégrité : contraintes devant être vérifiées dans tous les états d'un système.

Exemple 4.2

Voici comment s'exprime en RuleML la connaissance suivante : « Un client a le statut premium si ses dépenses l'an passé ont été d'au moins 5000 euros ».

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE rulebase SYSTEM
  "http://www.ruleml.org/0.8/dtd/ruleml-datalog.dtd">
<rulebase>
  <imp>
    <head>
      <atom>
        <rel>premium</rel>
        <var>client</var>
      </atom>
```

```
    </head>
    <body>
      <atom>
        <rel>dépenses</rel>
        <var>client</var>
        <ind>min 5000 euro</ind>
        <ind>année précédente</ind>
      </atom>
    </body>
  </imp>
</rulebase>
```

La DTD de RuleML est disponible en ligne, toutefois voici briévement le sens des différentes balises :

- rulebase : décrit une base de règles
- imp : décrit une implication entre la condition d'une règle (head) et sa conséquence (body)
- atom : permet de constituer une formule atomique
- rel : décrit une relation entre une variable (var) et une ou plusieurs constantes (ind)

4.1.2 Ontologies

4.1.2.1 Définitions

De manière informelle, l'ontologie d'un domaine d'intérêt est constituée de l'ensemble de sa terminologie, de tous les concepts essentiels du domaine (ainsi que de leur classification, de leur taxonomie, de toutes les relations existantes entre eux et des contraintes associées) et des axiomes du domaine. Une ontologie représente donc dans un langage $\mathcal{L}$ toutes les choses existantes dans un domaine d'intérêt $\mathcal{D}$. C'est un fragment essentiel des connaissances permettant de décrire un domaine.

Parmi les différentes définitions qui ont été données aux ontologies, la plus fréquemment citée est celle de Gruber [Gru93] qui énonce : « Une ontologie est la spécification explicite d'une conceptualisation partagée ». *Conceptualisation* signifie une vision abstraite et simplifiée du monde. Tout système désirant représenter un

fragment de notre monde doit obligatoirement créer une conceptualisation explicite ou implicite. Cette conceptualisation est composée des concepts et des objets supposés exister dans un certain domaine d'intérêt ainsi que des relations existantes parmi eux. Une *spécification* est une représentation formelle et déclarative : dans une ontologie les structures de données, les concepts et leurs types ainsi que les différentes relations sont exprimés de manière explicite en utilisant un langage formel qui peut être analysé de façon automatique.

Guarino étend cette définition en introduisant la notion d'*ontologie formelle* qui établit la distinction entre les objets du monde réel et le moyen de les représenter conceptuellement dans l'ontologie. Une ontologie formelle est alors une théorie des distinctions formelles entre les éléments d'un domaine, indépendamment de leur réalité [Gua97].

4.1.2.2 Caractéristiques et objectifs

Les ontologies sont très intéressantes pour la représentation et le traitement automatisé des connaissances. Elles décrivent un domaine en fournissant :

- un vocabulaire contrôlé : un ensemble de termes décrits par des énoncés logiques et dont les relations sont spécifiées au moyen de règles ;
- une hiérarchie de concepts afin de classifier les entités d'un domaine ;
- une théorie du domaine qui permet de vérifier la consistance du contenu de l'ontologie ;
- un moyen de partager et de réutiliser la connaissance, ainsi les applications reposant sur des ontologies peuvent être interopérables.

Fikes présente une liste de cinq domaines clefs dans lesquels les ontologies peuvent jouer un rôle important [Fik98] :

- La collaboration : différentes personnes pouvant avoir une vue différente du même domaine, les ontologies sont un moyen de fournir une représentation des connaissances unifiée et partageable par tous. Cet aspect est très important dans les domaines interdisciplinaires.
- L'interopérabilité : les ontologies permettent l'intégration de l'information à partir de sources différentes et disparates. La conversion automatique des différents formats est plus simple à effectuer et l'utilisateur final n'a alors accès qu'à une seule source d'information homogène.

- L'éducation : les ontologies regroupent des informations consensuelles sur un domaine et permettent à un utilisateur d'acquérir des informations objectives, s'il désire approfondir ses connaissances sur ce domaine.
- La modélisation : dans les systèmes basés sur la connaissance, les ontologies structurent les informations en ensembles facilement réutilisables.

Jusqu'à présent, nous avons présenté les ontologies comme un moyen de modéliser les connaissances d'un domaine. Cependant, cette notion de domaine d'intérêt demeure assez vague et nécessite d'être précisée. Guarino présente différents types d'ontologies [Gua98] :

- **Les ontologies de haut-niveau** décrivent des concepts très généraux comme le temps, l'espace, la matière, les objets, les événements, les actions, etc., qui sont indépendants d'un domaine ou d'un problème particulier. Selon Guarino les ontologies permettent d'unifier à haut niveau les différents domaines.
- **Les ontologies du domaine et les ontologies dédiées à une tâche** décrivent respectivement le vocabulaire lié à un domaine générique ou à une activité générique en spécialisant les termes des ontologies de haut-niveau.
- **Les ontologies d'application** dépendent à la fois des ontologies d'un domaine particulier et d'une tâche particulière : elles sont une spécialisation de ces deux types d'ontologies. Les concepts représentent alors le rôle joué par les entités du domaine lors d'une certaine activité.

Guarino établit également une distinction entre ontologie d'application et base de connaissances. Une ontologie d'application est une base de connaissances *particulière* décrivant des faits considérés comme vrais par une communauté qui est parvenue à un consensus sur les termes du domaine à utiliser. Une base de connaissances générique peut, quand à elle, décrire à la fois les faits spécifiques à un domaine mais aussi des assertions sur le domaine. Ainsi dans une base de connaissances générique on peut distinguer deux principaux composants :

- l'ontologie contenant des informations indépendantes de tout état ;
- le cœur de la base de connaissance contenant les informations dépendant d'un état, pouvant donc évoluer au cours du temps.

4.1.2.3 Les langages de représentation

XML, RDF et OWL sont les trois pièces maîtresses autour desquelles s'articule parfaitement le web sémantique. XML apporte syntaxe, règles et documents structurés. RDF apporte une infrastructure de données pour le Web. OWL fournit des ontologies opérationnelles pour le Web. Ces standards fournissent un cadre de représentation pour la gestion des ressources, l'intégration, le partage et la réutilisation des données. Ces formats de partage de données se rapportent aux applications, à la vie des entreprises et à celle d'autres communautés - tous ces différents types « d'utilisateurs » peuvent partager les mêmes informations, même s'ils ne partagent pas les mêmes logiciels.

XML et XML Schema XML fournit une syntaxe pour les documents structurés mais n'impose aucune contrainte sémantique. XML Schema est un langage pour définir la structure, le contenu et la sémantique de documents XML en étendant le langage XML avec de nouveaux types de données.

Exemple 4.3
Un Schema pour représenter un pays :

```
<xs:schema
 xmlns:xs="http://www.w3.org/2001/XMLSchema">
 <xs:element name="country">
  <xs:complexType>
   <xs:sequence>
    <xs:element name="name" type="xs:string"/>
    <xs:element name="population" type="xs:decimal"/>
   </xs:sequence>
  </xs:complexType>
 </xs:element>
</xs:schema>
```

L'utilisation de ce schema dans un document XML :

```
<country
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:noNamespaceSchemaLocation="country.xsd">
```

```
 <name>France</name>
 <population>59.7</population>
</country>
```

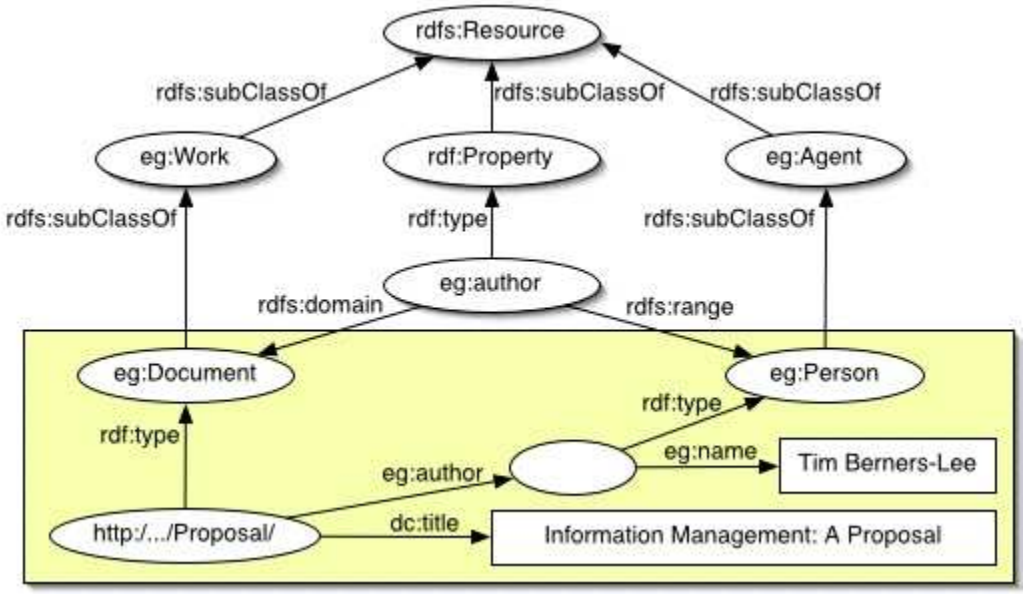


FIG. 4.1: Exemple d'utilisation d'un RDF Schema pour la description de classez et de propriétés

RDF et RDF Schema RDF est un modèle de données pour décrire des ressources et les relations entre elles. Ce modèle de données utilise la syntaxe XML. RDF Schema représente un vocabulaire pour décrire les propriétés et classes des ressources RDF. Il contient également une sémantique pour gérer la généralisation au sein des classes et propriétés. La figure 4.1, issue d'un document de travail du W3C [W3C02], illustre comment RDF peut être utilisé pour décrire des ressources (personnes ou documents représentés respectivement par les classes *eg :Person* et *eg :Document*) et les propriétés utilisées pour relier les membres de ces classes (*eg :author*). En utilisant le RDF Schema on peut décrire les relations entre les propriétés RDF et les classes de ressources. Dans cet exemple le RDF Schema est utilisé pour dire que la propriété *eg :author* relie un document à une personne. Cet exemple montre également que tous les documents sont considérés comme des travaux et toutes les personnes comme des agents.

OWL OWL, langage d'ontologies, permet de définir des ontologies garantissant une intégration plus riche et l'interopérabilité des données au travers des frontières applicatives. Les premiers langages utilisés pour le développement d'outils et d'ontologies pour des communautés d'utilisateurs spécifiques n'ont pas été définis pour être compatibles avec l'architecture du Web en général, et du Web sémantique en particulier. OWL répare ce manque en utilisant à la fois les URIs pour nommer, et la fonctionnalité fournie par RDF pour créer des liens.

OWL s'appuie sur un modèle et un schéma RDF pour ajouter plus de vocabulaire dans la description de propriétés et de classes, comme par exemple : les relations entre classes, la cardinalité, l'égalité, une typographie plus riche des propriétés, des caractéristiques de propriétés, et des classes énumératives. OWL fournit trois sous langages conçus pour différentes communautés d'utilisateurs et de développeurs :

- OWL Lite est dédié aux utilisateurs qui nécessitent simplement une taxonomie et quelques contraintes simples. Par exemple, bien que supportant les cardinalités seules les valeurs 0 ou 1 sont admises. OWL Lite ayant une complexité formelle moins importante que OWL DL, il est plus simple de fournir un outil supportant OWL Lite.
- OWL DL (pour OWL Description Logic) est dédié aux utilisateurs désirant un maximum d'expressivité tout en conservant complétude (toutes les conclusions sont calculables) et décidabilité (tous les calculs peuvent se faire en un temps fini). OWL DL inclut toutes les constructions du langage OWL toutefois elles ne peuvent être utilisées que sous certaines conditions.
- OWL Full est dédié aux utilisateurs désirant un maximum d'expressivité et la liberté d'expression de RDF sans garantie de calculabilité. OWL Full permet à une ontologie d'enrichir la signification du vocabulaire prédéfini de RDF ou de OWL.

Cyc et CyCL Le serveur de connaissances Cyc est constitué d'une base de connaissances et d'un moteur d'inférence développé par Cycorp [LPS86]. Le but de Cyc est de pouvoir rassembler un grand nombre de connaissances du « sens commun » afin d'aider les applications d'intelligence artificielle et de traitement du langage naturel. Le moteur Cyc se base sur le langage CycL qui utilise la logique des prédicats. La technologie Cyc fournit un système pour développer une ontologie mais aussi un moteur pour effectuer des inférences sur les connaissances. CycL est fondé sur trois

types de notions : les collections, les individus et les instances. Toute collection ou tout individu peut être instance d'une collection. Les individus, quand à eux, ne peuvent avoir d'instance : ils sont uniques. Une particularité de Cyc est la notion de microthéorie. Toutes les relations et assertions sont définies dans une microthéorie, qui est une sorte de domaine. Ainsi il est possible de gérer simultanément des ontologies portant sur des points de vue différents. En définissant correctement les connaissances contradictoires au sein de différentes microthéories on peut alors effectuer des inférences sur des connaissances contradictoires qui peuvent correspondre à la notion d'intérêt de différents utilisateurs.

4.2 Les ontologies en fouille de données

4.2.1 Les ontologies dans le cycle de vie CRISP-DM

Le modèle CRISP-DM [Wir00] distingue six phases principales dans le processus d'extraction des connaissances : la compréhension du domaine, la compréhension des données, la préparation des données, la modélisation, l'évaluation et le déploiement. Dans [CRS+04], Češpivová évoque le rôle que peuvent avoir les ontologies dans chacune de ces phases.

Compréhension du domaine Le rôle des ontologies dans la compréhension du domaine n'est pas particulier à l'extraction de connaissances. Les ontologies du domaine sont importantes pour explorer un domaine et effectuer une tâche spécifique. Les ontologies permettent de se familiariser avec le domaine et de mettre en évidence des connaissances incompatibles qui ne l'étaient pas à première vue.

Compréhension des données Afin d'améliorer la compréhension des données les concepts de l'ontologie doivent être associés aux éléments du schéma des données. Cet effort met en évidence les attributs manquants ou redondants dans la base de données.

Préparation des données La phase de préparation des données est toujours intimement liée à la phase de modélisation, car l'utilisation d'une ontologie peut influencer le choix des algorithmes qui seront mis en jeu. Lors de cette phase l'ontologie

permet d'identifier des groupes d'attributs ou de valeurs ayant la même sémantique.

Modélisation Lors de la phase de fouille de données les ontologies peuvent être utilisées par les algorithmes afin de générer des modèles prenant en compte la sémantique des données. Dans cette optique Han propose un algorithme d'extraction de règles d'association multi-niveaux reposant sur un encodage spécifique des valeurs de la base de données qui permet à une seule valeur de représenter toute une branche d'une taxonomie [HF95]. Liu propose quand à lui un algorithme de clustering guidé par les informations contenues dans une ontologie [LWY04].

Évaluation Lors de la phase d'évaluation les modèles extraits peuvent être interprétés en utilisant les termes de l'ontologie et confrontés aux connaissances a priori. Les méthodes de confrontation de règles aux connaissances sont basées sur des mesures de similarité. Bisson présente dans [Bis94] un panorama de ces différentes mesures et souligne une distinction importante : les mesures de similarités *non-informées* des mesures de similarité *informées*. Dans le premier cas, le calcul se fait sur une base purement locale en ne prenant en compte que les informations qui sont explicitement présentes dans les exemples alors que dans le second cas on utilise en outre des informations d'ordre statistique ou encore symbolique portant sur l'ensemble des exemples de l'univers sur lequel on travaille.

Selon Bisson, la similarité totale entre deux exemples se ramène toujours plus ou moins à faire la somme des similarités partielles sur les attributs communs. Ce qui pose un problème crucial : celui de la pondération. Si l'échelle des pondérations entre les attributs est mal établie, le résultat final risque fort d'être non pertinent. Étant donné que les critères sur lesquels les êtres humains jugent la similarité ne sont pas stables et qu'ils varient dynamiquement (en fonction du contexte dans lequel on se trouve et des connaissances a priori que l'on possède sur le domaine), il peut être judicieux d'intégrer les informations sur la similarité au sein de la base de connaissances.

Déploiement Dans la phase de déploiement la connaissance extraite permet d'enrichir les connaissances existantes sur le domaine : la structure fournie par une ontologie permet de guider ce processus d'intégration de nouvelles connaissances.

4.2.2 Un environnement de post-traitement intégrant la connaissance

Très peu d'approches, dans le domaine de la fouille de données, se sont intéressées à l'intégration cohérente de connaissances dans plusieurs des étapes du processus de fouille de données. Carsten Pohle propose toutefois KAIMAN (Knowledge-based Assistance for Intelligent Mined patterns Analysis), un environnement de post-traitement supportant l'utilisation d'ontologies [Poh03]. Cette approche propose des solutions aux trois problèmes suivants :

- L'utilisation de tout type d'algorithme de fouille de données.
- L'acquisition des connaissances du domaine.
- La confrontation des connaissances avec les modèles extraits.

Afin de pouvoir traiter les résultats de tous les types d'algorithme de fouille de données, le standard PMML (Predictive Model Language) est utilisé pour exprimer les modèles extraits. L'acquisition des connaissances repose, quand à elle, sur l'environnement Protégé2000 permettant la manipulation d'ontologies. Enfin, la confrontation des connaissances est gérée par une technique similaire à celle présentée par Liu (voir section 3.5) qui utilise des ensembles flous pour prendre en compte l'incertitude dans les connaissances exprimées par les utilisateurs experts. Un système gérant toutes les phases de l'approche KAIMAN a été mis en œuvre dans le cadre de la fouille de données web.

4.3 Conclusion

La méthode KAIMAN, bien que se limitant à la phase de post-traitement demeure assez ambitieuse en voulant supporter tous les types d'algorithmes de fouille de données. L'intégration de connaissance en fouille de données ne se limite pas à l'apport d'une ontologie du domaine mais plutôt à l'adaptation des connaissances à une tâche de fouille spécifique. Le choix d'une ontologie d'application adaptée à la tâche à effectuer semble donc plus judicieux, toutefois les connaissances du domaine ne seraient alors pas totalement réutilisables.

Il est intéressant de solliciter au minimum les experts du domaine afin d'initialiser un système de fouille de données, comme le propose cette approche. Toutefois, cet objectif ne doit pas être une fin en soi mais un compromis à atteindre. Le fait de

ne pas impliquer les ontologies au cœur du processus de pré-traitement n'allège pas forcément la mise en œuvre du système qui pourrait y gagner en cohérence dans les étapes suivantes.

Enrichir les connaissances a priori avec de nouvelles informations est très intéressant. Cependant, le problème de la confrontation des connaissances aux modèles extraits se pose, et l'approche proposée n'y répond que partiellement en utilisant un mécanisme à base d'ensembles flous. En effet, l'utilisation de relations taxonomiques (reliant des valeurs réelles à des valeurs plus subjectives définies par les utilisateurs) ne représente qu'un aspect du problème. Les ontologies contiennent un nombre important de relations de natures différentes et différentes connaissances peuvent ressembler partiellement à un modèle extrait. La confrontation des connaissances aux modèles extraits demeure donc un problème ouvert.

Deuxième partie

L'approche KEOPS pour l'intégration des connaissances au processus de fouille de données

Chapitre 5

Présentation générale de l'approche KEOPS

Sommaire

5.1 Introduction

Ce chapitre présente une vue d'ensemble du système KEOPS. Il présente en particulier la méthodologie choisie pour intégrer les connaissances expertes tout au long du processus de fouille de données. Cette méthodologie, qui enrichit l'approche CRISP-DM [Wir00], repose sur trois composants principaux : un système d'information dirigé par une ontologie (ODIS) pour la préparation des données et l'expression des connaissances expertes, des algorithmes de fouilles de données et un mécanisme novateur d'évaluation de l'intérêt des modèles générés en fonction des connaissances de l'expert. Nous présentons ensuite le déploiement du système KEOPS ainsi qu'une application sur le domaine de la gestion de la relation allocataire au sein des Caisses d'Allocations Familliales.

5.2 Rappel des objectifs

Les motivations générales pour la réalisation d'un système permettant de guider le processus de fouille de données au moyen de connaissances expertes ont été posées en introduction. Il s'agit ici de rappeler les besoins qui ont guidé les choix effectués dans la construction du système KEOPS :

- Définir une méthodologie afin de formaliser les différentes étapes du processus de fouille de données : préparation des données, création des jeux de données, choix de l'algorithme pour la fouille et visualisation des résultats.
- Gérer les interactions entre la connaissance des experts et les données tout au long du processus de fouille de données.
- Évaluer les modèles générés par la fouille relativement aux connaissances exprimées par les experts du domaine et fournir une navigation aisée au sein des résultats.
- Proposer des stratégies pour la gestion du processus de fouille de données, afin qu'il soit pérenne mais aussi qu'il puisse s'adapter en fonction du résultat des fouilles antérieures.

5.3 Présentation de la méthodologie

KEOPS intègre la connaissance experte tout au long du processus d'extraction des connaissances à partir des données. La première étape de la méthodologie structure et organise la connaissance au sein d'un système d'information conceptuel tandis que les étapes suivantes l'utilisent et l'enrichissent. La méthodologie proposée est composée de 10 phases :

1. Sélection des attributs
2. Construction de l'ontologie
3. Construction de la MODB (Mining Oriented Database)
4. Construction de la base de connaissances
5. Génération des jeux de données
6. Fouille de données
7. Simplification et combinaison des motifs
8. Sélection des motifs les plus pertinents en fonction des connaissances
9. Exploration des résultats
10. Mise à jour de la base de connaissances

Chacune de ces phases met en jeu des acteurs et des systèmes qui sont représentés sur la figure 5.1.

1. Sélection des attributs : il s'agit de définir la portée de l'étude effectuée par la fouille de données en choisissant parmi les données disponibles celles qui seront réellement utiles. Cette étape nécessite souvent la participation des experts du domaine.

2. Construction de l'ontologie : à partir des données existantes et des connaissances de l'expert une ontologie doit être créée afin de déterminer les concepts essentiels du domaine (ainsi que leurs relations) qui seront utiles lors de la fouille de données.

3. Construction de la MODB : une base de données relationnelle dont les attributs et valeurs sont des concepts de l'ontologie est construite lors de cette étape.

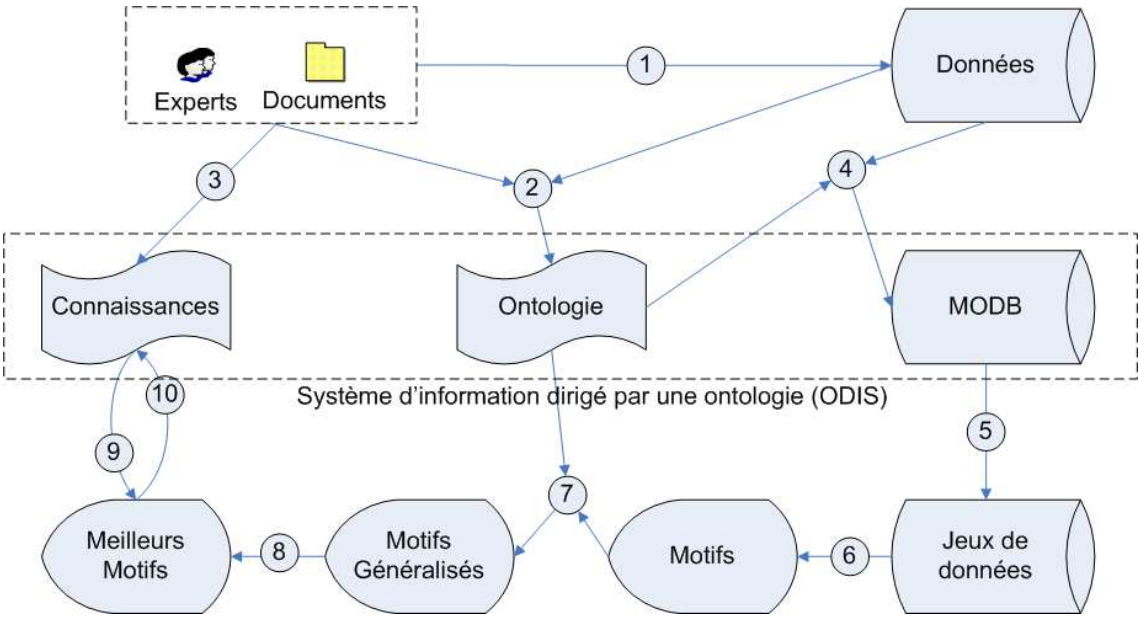


FIG. 5.1: Approche KEOPS

4. Construction de la base de connaissances : les connaissances des experts servent de support pour déterminer leurs centres d'intérêt.

5. Génération des jeux de données : la MODB permet de générer facilement de nouveaux jeux de données ayant un niveau de généralisation différent.

6. Fouille de données : l'algorithme CLOSE[1] est mis en œuvre afin d'extraire des motifs fermés fréquents générant des règles non redondantes.

7. Simplification et combinaison des motifs : Le but de la simplification est de conserver un maximum d'information au sein de règles de taille minimale. Il s'agit de combiner un ensemble de règles proches en une nouvelle qui permet de toutes les résumer.

8. Sélection des motifs les plus pertinents en fonction des connaissances : une mesure d'intérêt novatrice permet de prendre en compte des critères statistiques

[1] L'algorithme CLOSE, développé par Pasquier est présenté en section 2.4.1.2.

pour déterminer les modèles les plus fiables ainsi que les connaissances des experts du domaine afin de sélectionner parmi eux les plus intéressants et les plus utiles.

9. Exploration des résultats : elle doit permettre à l'expert de naviguer simplement au sein des résultats en fonction de ses centres d'intérêts. Les structures définies par l'ontologie et le classement généré par notre mesure d'intérêt favorisent une exploration efficace des résultats.

10. Mise à jour de la base de connaissances : la méthodologie présentée a pour but d'aider à la gestion du processus de fouille de données au cours du temps, il est donc nécessaire de maintenir une base de connaissances à jour.

Dans ce chapitre et les suivants nous utilisons parfois le terme « modèle » pour définir les résultats d'une fouille de données. Toutefois ayant fait le choix d'utiliser CLOSE, un algorithme de recherche de règles d'association, nos modèles se présentent sous la forme de règle « si ... alors ... ».

La base de données orientée fouille de données (MODB) est générique car elle est utilisée en tant qu'entrepôt de données à partir duquel des jeux de données spécifiques à une tâche peuvent être générés. La construction d'un système d'information dirigé par une ontologie (ODIS) permet d'obtenir une structure qui fournit plus de flexibilité non seulement lors de la phase de préparation des données mais aussi pour filtrer et interpréter les motifs découverts lors la fouille de données. Les relations au sein de l'ontologie jouent un rôle central pendant ces étapes.

5.4 Déploiement

La plate-forme KEOPS est formée de plusieurs composants communiquant entre eux. Le choix a été fait de ne pas concevoir une application monolithique car, si certaines tâches de visualisation et de gestion du processus peuvent être effectuées à partir d'un système portable, d'autres comme le stockage, la fouille des données ou le traitement des résultats nécessitent un serveur dédié afin de ne pas monopoliser les ressources de l'utilisateur.

Les informations manipulées dans l'approche étant de nature différente, différents systèmes de stockage sont déployés :

- Une base de données orientée fouille de données (MODB) qui contient les données pré-traitées et permet de générer des jeux de données à la volée.
- Un système de gestion des connaissances contenant l'ontologie applicative de l'ODIS ainsi que les connaissances de l'utilisateur exprimées sous forme de règles.
- Une base de données contenant les résultats de la fouille et de l'évaluation des modèles.

Les composants de la plate-forme KEOPS effectuent les tâches suivantes :

- Préparation des données permettant de générer différents jeux de données en tenant compte des relations exprimées dans l'ontologie.
- Fouille de données basée sur l'algorithme CLOSE qui effectue une recherche de règles d'association.
- Confrontation des modèles extraits aux connaissances afin d'évaluer leur intérêt.
- Visualisation des résultats en utilisant les relations définies dans l'ontologie afin de permettre à l'utilisateur d'explorer les résultats selon différents points de vue.
- Gestion du processus d'extraction des connaissances à partir des données.

5.5 Mise en œuvre expérimentale

L'approche KEOPS a été mise en œuvre dans le cadre de l'étude de la relation allocataire au sein des Caisses d'Allocations Familiales. L'objectif de cette étude était l'analyse de la relation avec les allocataires afin d'améliorer de fait la qualité de service grâce à une meilleure connaissance de cette relation. Des études ont été précédemment effectuées au sein des CAF sur la modélisation et la structuration sémantique de la réglementation [JACP03]. Toutefois, dans notre mise en œuvre nous nous focaliserons plutôt sur la confrontation des connaissances aux résultats de la fouille de données. Pour cela, les Caisses d'Allocations Familiales ont transmis les données suivantes qui concernent les allocataires de la CAF de Grenoble en 2004 :

- Bases mensuelles concernant les informations sur les allocataires
- Base concernant la politique de contrôle
- Base concernant le RMI
- Base concernant les faits générateurs

- Base des courriers
- Base des contacts physiques et téléphoniques
- Base des droits allocataires
- Base des consultations des bornes interactives
- Bases CRISTAL concernant différentes prestations
- Base des créances

Au total 73 tables ont été fournies ce qui constitue plus d'une centaine d'attributs et plusieurs millions d'enregistrements.

Chapitre 6

Expression des connaissances

Sommaire

6.1 Motivations

Pour pouvoir présenter à l'utilisateur des résultats aussi intelligibles que possible il est souhaitable de les exprimer en utilisant son langage. Par exemple, il est plus compréhensible de dire qu'un allocataire de la CAF a consulté une borne interactive pour imprimer un formulaire concernant les prestations logement que de dire qu'il a consulté la rubrique « simulation »[1]. Il est donc nécessaire de préparer les données brutes afin d'en extraire toute la richesse sémantique nécessaire à une fouille de données performante et à une interprétation des résultats plus aisée.

La méthodologie KEOPS a pour vocation d'aider à l'intégration des connaissances de l'expert dans tout le processus d'extraction de connaissances. L'expert intervient lors de la création de l'ontologie (étape n°2 fig 5.1), qui va permettre de définir les concepts du domaine utiles à la fouille de donnée, et lors de la création de la base de connaissances (étape n°3 fig 5.1), dont le rôle est d'aider à sélectionner les modèles les plus intéressants (étape n°8 fig 5.1).

Nous présentons également dans ce chapitre, les principes de la création de la base de données orientée fouille de données (MODB), dont la particularité est d'être formée uniquement d'attributs et de valeurs correspondant à des concepts définis dans l'ontologie. L'association entre l'ontologie et la base de données spécialement conçue pour la fouille de données constitue l'ODIS (Ontology Driven Information System) [Bri06b][2]. Son rôle, comme nous allons le voir, est de faciliter la création de jeux de données (étape n°5 fig 5.1) et d'obtenir des modèles facilement comparables aux connaissances exprimées par les experts.

Le chapitre est organisé de la façon suivante : dans la première partie nous présentons les choix fixés dans l'approche KEOPS pour la structure de l'ontologie, la deuxième partie est consacrée à l'ODIS et la dernière partie présente la base de connaissance.

[1] La rubrique simulation correspond au fait d'effectuer une simulation sur les prestations logements qu'un allocataire pourrait toucher selon sa situation.

[2] Dans l'article qui est cité, le terme Conceptuel Information System est employé : il correspond à la notion d'ODIS présenté dans ce chapitre. Le nom a été modifié suite à des suggestions, car il pouvait porter à confusion dans certains domaines.

6.2 Choix ontologiques

6.2.1 Préambule

L'ontologie KEOPS s'inscrit dans le cadre d'un système d'information dirigé par une ontologie, qui est étroitement associé à la base de données relationnelle MODB. Par conséquent les choix ontologiques qui ont été faits nécessitent de revenir sur la définition du modèle relationnel. Ce modèle a, pour la première fois, été présenté par Codd en 1970 [Cod70], à partir des notions fondamentales de domaine, d'attribut, de relation et de schéma.

Rappels 6.1 Un *domaine* est un ensemble de valeurs atomiques.
Le rôle que joue un domaine dans une relation est identifié par un *attribut*. On parle de « domaine de l'attribut ».
Une *relation* R($A_1, ..., A_n$) (où $A_1, ..., A_n$ sont des attributs) est un sous-ensemble fini du produit cartésien des domaines de $A_1, ..., A_n$. Le produit cartésien d'un ensemble de domaines $D_1, D_2, ..., D_n$, noté $D_1 \times D_2 \times ... \times D_n$, est l'ensemble des *tuples* $< v_1, v_2, ..., v_n >$ tels que $v_i \in D_i$.
Le *schéma d'une relation* R est un nom suivi de la liste des attributs, chaque attribut étant associé à son domaine : $R(A_1 : D_1, A_2 : D_2, ...A_n : D_n)$.

Une relation peut être vue comme une table dont chaque colonne représente un attribut. Dans une colonne on trouve les valeurs du domaine associé à l'attribut. Un ensemble de schémas de relations participe à la définition du *schéma relationnel* d'une base de données.

Remarque 6.1 La notion de domaine dans KEOPS étend la définition de domaine au sens relationnel de Codd. Le domaine *Prestation*, par exemple, contient toutes les informations sur les prestations existantes. En suivant la définition de Codd ce domaine serait défini par l'ensemble de valeurs suivant : {APL, ALS, ARS, AF} (voir figure 6.1 pour la signification des acronymes).

6.2.2 Structure de l'ontologie

La notion d'ontologie utilisée dans la méthodologie KEOPS correspond à celle présentée par Gruber dans [Gru95] c'est-à-dire « La spécification formelle et explicite

d'une conceptualisation partagée ». Nous définissons la structure générale d'une ontologie KEOPS dans la définition 6.1.

Définition 6.1
Une ***structure d'ontologie*** *est un triplet* $O := \{C, R, A\}$

- C *: ensemble des concepts de l'ontologie*
- R *: ensemble des relations de l'ontologie entre deux ou plusieurs concepts*
- A *: ensemble d'axiomes exprimés dans un langage logique permettant de définir des propriétés sur les concepts et les relations.*

Tous les concepts d'une ontologie KEOPS possèdent une propriété (attribut ou valeur) définissant leur rôle au sein de la MODB :

Définition 6.2 (Concept-attribut et concept-valeur)
On distingue deux différentes classes de concepts, les concepts-attributs et les concepts-valeurs :

- *Un* ***domaine*** *de la MODB est associé à un unique* ***concept-attribut*** *et réciproquement. On l'appelle* le domaine associé au concept-attribut.
- *Une valeur d'un domaine de la MODB est associé à un unique* ***concept-valeur****. Cette relation n'est cependant pas réciproque : un concept-valeur peut ne pas être associé à une valeur de la MODB.*

Un domaine défini au sein de la MODB est donc représenté dans l'ontologie par :

- un concept-attribut CA, qui peut être associé à plusieurs attributs de la MODB : chaque attribut décrit alors un rôle différent du domaine ;
- des concepts-valeurs organisés en une taxonomie et en relation « valeurDe » avec le concept-attribut CA ? Par extension on dit que ces concepts-valeurs appartiennent au domaine associé à CA.

Cette définition étend celle de Codd (voir définition 6.1) dans la mesure où un domaine peut contenir une hiérarchie de valeurs.

Exemple 6.1
La définition de domaine donnée par l'approche KEOPS permet d'inclure des valeurs non-atomiques généralisant d'autres valeurs du domaine. Par exemple, les concepts *APL* et *ALS* sont généralisés par le concept *Prestation Logement* et les concepts *ARS* et *AF* sont généralisés par le concept *Prestation entretien*. Le domaine *Prestation*,

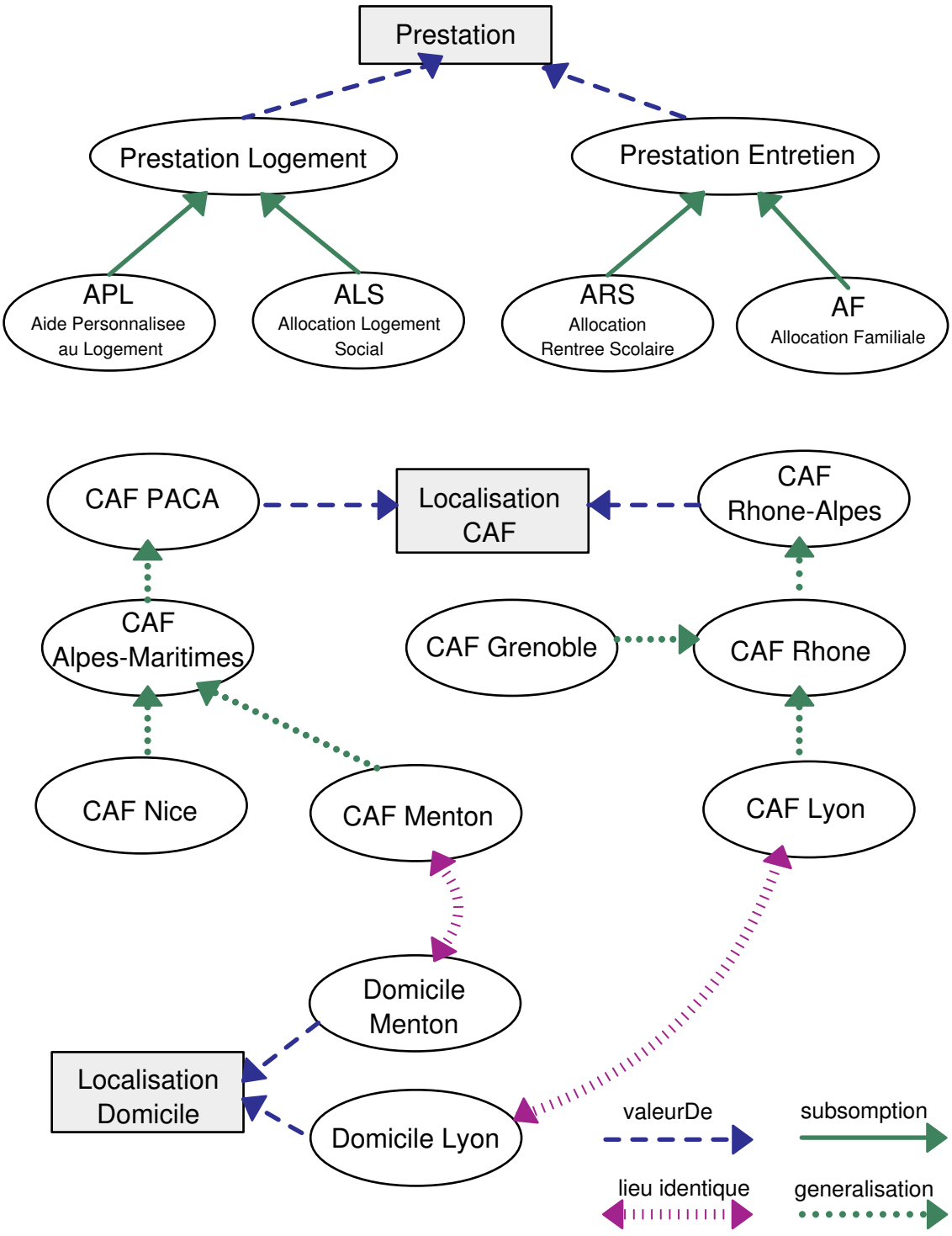


FIG. 6.1: Extrait d'une ontologie sur la gestion du contact allocataire dans les CAF

représenté par un concept-attribut, est alors défini par l'ensemble des concepts-valeurs suivants : {APL, ALS, Prestation Logement, ARS, AF, Prestation entretien} comme l'illustre la figure 6.1.

Définition 6.3 (Sous-domaine)
Soit CA un concept-attribut et un ensemble de concepts-valeurs en relation « valeurDe » avec CA définissant un domaine $\mathcal{D}$. Un concept-valeur $CV \in \mathcal{D}$ est appelé un sous-domaine de $\mathcal{D}$ s'il existe un ensemble de concepts-valeurs $\{CV_1, \ldots, CV_n\}$ de $\mathcal{D}$ en relation de généralisation avec CV. On dit que $\{CV_1, \ldots, CV_n\}$ définit le sous-domaine CV.

En considérant la classification proposée par Guarino [Gua98] l'ontologie KEOPS est une *ontologie d'application*, c'est-à-dire que les concepts définis dépendent à la fois du domaine étudié mais aussi de la tâche à effectuer, en l'occurrence la fouille de données. Au niveau de la structure de l'ontologie cela se traduit par le fait que les concepts possèdent une propriété indiquant qu'ils peuvent être utilisés lors de la fouille soit en tant qu'attribut soit en tant que valeur. Au niveau du choix des concepts cela se traduit par le fait que l'ontologie KEOPS décrit les hiérarchies de concepts utiles lors des différentes phases de la fouille de données.

L'ontologie ainsi définie a pour vocation de faciliter la tâche de construction de la base de données dédiée à la fouille de données (MODB), de fournir un support formel pour la simplification et la combinaison des modèles générés par la fouille de données et enfin de permettre la création d'une base de connaissances formalisée.

6.2.3 Relations entre concepts

Les relations de l'ontologie sont des relations binaires définies sur l'ensemble des concepts. Selon la définition 6.1 ces relations font partie de l'ensemble R des relations. La nature de la relation est très importante dans la mesure où elle permet de déterminer à quel moment celle-ci est utilisée dans le processus d'extraction des données.

Les relations entre deux concepts peuvent être :
- La relation « valeurDe » entre un concept-attribut et un concept-valeur.
- Une relation de généralisation entre deux concepts-valeurs.
- Une relation sémantique propre au domaine entre deux concepts-valeur.

6.2.3.1 La relation « valeurDe »

Définition 6.4

Soit C l'ensemble des concepts de l'ontologie.
Soit $A \subset C$ l'ensemble des concepts-attributs de l'ontologie.
Soit $V \subset C$ l'ensemble des concepts-valeurs de l'ontologie.
La relation valeurDe est une relation fonctionnelle de V dans A : $valeurDe \subseteq V \times A$.
Il existe une relation valeurDe entre un concept-valeur CV et un concept-attribut CA si et seulement si il existe dans la MODB un domaine associé à CA qui contient la valeur associée à CV.

Au sein de l'ontologie, l'ensemble de concepts-valeurs en relation « valeurDe » avec un concept-attribut CA définit l'ensemble des valeurs du domaine associé à CA. Cela permet de décrire un ensemble de concepts ayant une sémantique proche.

Etant donné qu'un concept-attribut décrit un domaine, un concept-valeur ne peut être lié par la relation « valeurDe » qu'à un unique concept-attribut, ce qui n'est toutefois pas réciproque, un même concept-attribut pouvant être relié à plusieurs concepts-valeurs.

Exemple 6.2 (Relation valeurDe)
Les figures 6.1 et 6.2 illustrent par des flèches discontinues bleues différentes relations du type « valeurDe ». Pour ne pas surcharger la figure 6.1 toutes les flèches bleues n'apparaissent pas notamment pour les sous-concepts d'un concept déjà en relation « valeurDe » avec un concept-attribut. Ainsi *APL*, *ALS*, *ARS* et *AF* sont bien des valeurs de *Prestation* et tous les concepts du type *CAF x* sont des valeurs de *Localisation CAF*.

6.2.3.2 Les relations de généralisation

Une *relation de généralisation* est une relation d'ordre strict partiel. C'est une relation d'ordre partiel car tous les concepts de C ne sont pas liés par cette relation. De façon générale les relations de généralisation entre deux concepts-valeurs peuvent être utilisées tout au long du processus de fouille de données, lors des étapes suivantes :

- la préparation des jeux de données (voir section 6.3.4),
- la factorisation des modèles générés (voir section 7.2.2),

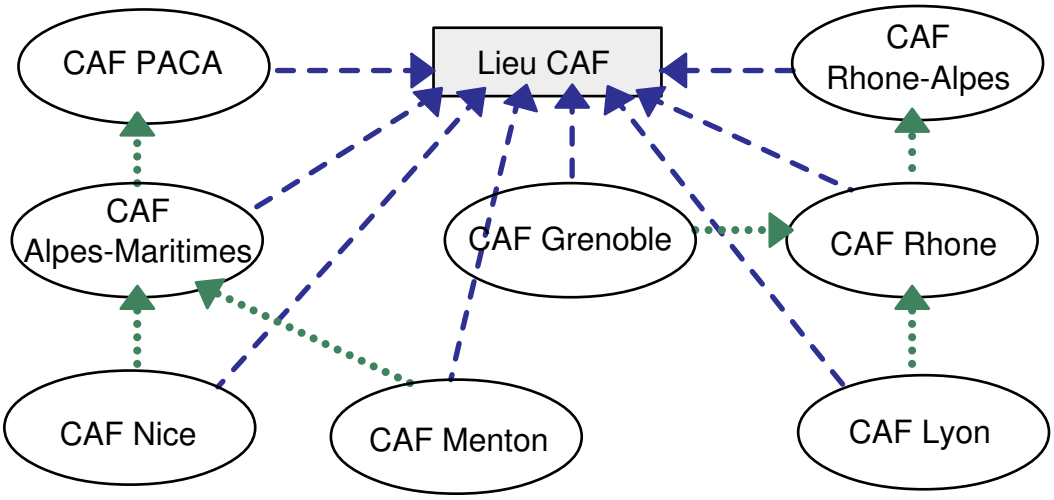


FIG. 6.2: Relations valeurDe (en bleu) au sein de l'ontologie

- la comparaison des modèles avec les connaissances (voir section 7.3.2),
- la visualisation des résultats par l'utilisateur.

Les notations et termes utilisés pour décrire les relations de généralisations sont parfois déroutants car, selon le domaine ou les habitudes, leur signification diffère. Pour définir la notion de subsomption nous nous basons sur la définition de Brachman [Bra83] : « A subsume B si et seulement si, dans toutes les interprétations possibles, chaque instance de B est nécessairement une instance de A ». Une autre confusion peut venir des termes dénommant la relation de subsomption et la relation entre un concept et ses instances. Considérons deux concepts A et B tel que A subsume B, et une instance de A dénommée α. Différents termes sont utilisés :

- Selon Fuchs, B « est une sorte de » A, et α « est un » A [Fuc97].
- Selon Guarino, B « est un » A, et α « est une instance de » A [Gua99].

Dans l'approche KEOPS nous n'utilisons pas d'instances de concepts, mais nous adoptons le terme « est une instance de » afin d'éviter les confusions. De plus, nous employons le terme « est une sorte de » pour décrire la relation de subsomption au sens défini par Brachman. Nous introduisons ensuite un autre type de relation de généralisation qui diffère de la relation de subsomption habituellement employée.

a) La relation de subsomption sur les concepts-valeurs

La relation de subsomption permet d'organiser les concepts-valeurs en une hiérarchie. Une relation de subsomption est définie, dans l'ontologie KEOPS, comme une relation de généralisation entre deux concepts-valeurs appartenant au même domaine, c'est-à-dire en relation « valeurDe » avec le même concept-attribut. Elle traduit le lien sémantique « est une sorte de ».

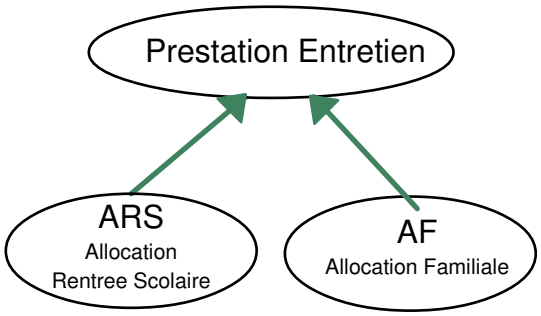


FIG. 6.3: Relations de subsomption (en vert) au sein de l'ontologie

Exemple 6.3 (Relation de subsomption)
Les figures 6.1 et 6.3 illustrent les relations de subsomption par des flèches vertes continues. Ainsi *ARS* est en relation avec *Prestation entretien* car l'*Allocation de rentrée scolaire (ARS)* **est** une *Prestation entretien*.

b) Les autres relations de généralisation sur les concepts-valeurs

Lors de la phase de préparation des jeux de données, une relation de généralisation peut être utilisée afin de substituer une valeur du domaine à une autre. Cette fonctionnalité est utile afin de discrétiser les valeurs de différentes façons. Par conséquent, ces relations de généralisation sont définies entre concepts-valeurs appartenant au même domaine (donc tous reliés par la relation « valeurDe » au même concept-attribut).

Exemple 6.4 (Relation de généralisation)
Les figures 6.1 et 6.4 illustrent les relations de généralisation (autres que la subsomption) par des flèches vertes en pointillés. Ainsi *CAF Menton* se généralise par *CAF Alpes-Maritimes* : un concept est plus général que l'autre cependant ils ne sont pas de même nature (l'un symbolise la ville d'une CAF, l'autre le département d'une

CAF), ce qui explique que la relation de subsomption n'a pas été choisie dans ce cas là.

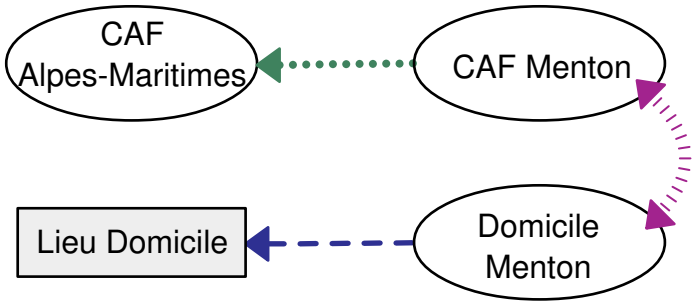


FIG. 6.4: Relations de généralisation (en vert pointillés) et relation d'équivalence (en violet) au sein de l'ontologie

6.2.3.3 Les relations sémantiques

Les relations sémantiques sont très particulières au domaine étudié : ces relations peuvent être par exemple des relations d'ordre, de composition, d'exclusion ou d'équivalence. Dans l'approche KEOPS il est nécessaire d'identifier formellement les relations sémantiques afin de pouvoir déterminer leur rôle dans les différentes étapes de la fouille de données. De façon générale les relations sémantiques n'interviennent que lors de deux étapes :

- la comparaison des modèles avec les connaissances,
- la visualisation des résultats par l'utilisateur.

La relation d'équivalence est une relation sémantique qui permet de décrire simplement certaines informations sur un domaine. Celle-ci se définie sur un sous-ensemble des concepts de l'ontologie afin de déterminer une classe d'équivalence.

Exemple 6.5 (Les classes d'équivalences sur la localisation)
Les figures 6.1 et 6.4 illustrent les relations sémantiques par des flèches discontinues violettes. Dans cet exemple la relation sémantique correspond au lien « être localisés dans le même lieu ». Ainsi on peut exprimer que deux concepts de domaines différents, « CAF Nice » et « Domicile Nice » représentent des entités équivalentes (situées dans la même ville). Lors de la phase d'exploration, ce type de relation permet de présenter les résultats selon différents points de vue.

6.3 Le système d'information dirigé par une ontologie (ODIS)

Un système d'information dirigé par une ontologie (ODIS : Ontology Driven Information System) est constitué d'une ontologie d'application, d'une base de données relationnelle dédiée à la fouille de données (MODB pour Mining Oriented DataBase) et d'une base de connaissances.

La MODB est une base de données relationnelle constituée de plusieurs tables dont les attributs décrivent le rôle d'un domaine qui est exprimé dans l'ontologie au moyen d'un concept-attribut. Les valeurs de ce domaine sont décrites dans l'ontologie à travers une hiérarchie de concepts-valeurs. La MODB ne contient toutefois que les valeurs de granularité la plus fine possible c'est-à-dire celles dont les concepts-valeurs sont à la base de la hiérarchie.

Les choix qui guident la construction de l'ODIS sont effectués dans le but de faciliter le processus de fouille de données. L'objectif est de structurer à la fois les connaissances et les données afin de réaliser une fouille de données efficace de manière à capitaliser le temps investi dans la préparation des données et à réutiliser les données préparées dans d'autres processus de fouilles sur des jeux de données créés à partir de la MODB.

L'ODIS est construit à partir de données existantes présentes dans une base de données qui est généralement dédiée à la gestion de tâches opérationnelles ou qui, dans certaines situations, fait partie d'un système décisionnel. Le plus souvent la base de données existante n'est pas adaptée à la fouille de données. Une première étape consiste à extraire de cette base de données les concepts de l'ontologie.

Dans la suite de cette section nous présentons une synthèse de travaux portant sur l'extraction d'ontologies à partir de bases de données ; puis nous détaillons les principes de construction de l'ontologie dans KEOPS.

6.3.1 Extraction d'ontologie à partir des données

Ces dernières années un certain nombre de travaux ont eu pour objectif d'automatiser l'extraction d'ontologies. Shamsfard présente dans [SB03] une synthèse des travaux les plus récents et introduit un environnement permettant de comparer les systèmes d'extraction d'ontologies. Cet environnement repose sur les critères

suivants :

- les caractéristiques des informations fournies en entrée de l'algorithme d'extraction,
- les méthodes d'apprentissage et d'acquisition de la connaissance,
- les éléments appris,
- l'ontologie résultante,
- l'évaluation du processus.

Si l'on s'intéresse plus particulièrement aux caractéristiques des informations fournies en entrée on remarque qu'elles peuvent être exprimées en langage naturel ou sous forme de données. Dans le cas où l'ontologie doit être extraite à partir de données il y a également trois situations possibles :

- les données sont structurées : cela peut-être au sein d'une base de données, de fichiers XML ou d'une base de connaissances ;
- les données sont semi-structurées : ce cas est le plus courant en web-sémantique ;
- les données sont non-structurés : ce cas est le plus difficile à traiter et nécessite des outils de fouille de texte par exemple.

Dans le contexte de la fouille de données, nous sommes donc en présence de données structurées au sein d'une base de données, c'est pourquoi nous allons plus particulièrement nous intéresser à deux approches spécifiques : celle de Kashyap se basant sur des techniques de rétro-conception et celle de Sampson basée sur la méthode NaLER permettant d'interpréter en langage naturel un modèle de données relationnel.

6.3.1.1 Rétro-conception du schéma relationnel

Le processus de création d'une ontologie à partir d'une base de données est très itératif. La méthode proposée par Kashyap [Kas99] est toutefois basée sur trois étapes principales :

- Rétro-conception du schéma de la base.
- Raffinement de l'ontologie grâce à des requêtes utilisateurs.
- Enrichissement de l'ontologie du domaine.

Rétro-conception du schéma de la base Ce processus implique l'analyse détaillée du schéma de la base de données afin de déterminer les clefs primaires, les clefs étrangères et les différentes dépendances. Cette analyse aide à déterminer les

principaux concepts et à regrouper les informations dispersées dans les différentes tables.

Raffinement de l'ontologie grâce à des requêtes utilisateurs Une ontologie construite à partir de l'analyse des schémas d'une base de données peut être améliorée afin d'être plus en adéquation avec les besoins des utilisateurs. Pour cela on peut utiliser un ensemble de requêtes utilisées par les utilisateurs pour obtenir les informations qu'ils désirent. Ce processus pourrait suggérer l'ajout à l'ontologie de concepts absents et potentiellement utiles pour les utilisateurs. Inversement, certains concepts n'intéressant pas les utilisateurs pourraient voir leur présence dans l'ontologie réévaluée. Pour finir ces requêtes pourraient suggérer la création de nouvelles relations permettant de structurer l'ontologie.

Enrichissement de l'ontologie du domaine Un des problèmes rencontrés dans les méthodes nécessitant beaucoup d'interactions avec l'utilisateur est le vocabulaire partagé. L'approche de Kashyap comprend donc une dernière phase permettant d'associer les concepts à différents termes extraits de thésaurus afin que les utilisateurs puissent exprimer les requêtes dans un langage qui leur est familier.

Il est important de souligner que la méthode proposée est essentiellement interactive et itérative, les informations données par les experts du domaine permettant d'évaluer la pertinence des concepts choisis et d'évaluer leur importance relative.

6.3.1.2 Expression du schéma relationnel en langage naturel

L'approche proposée par Sampson dans [Sam05] est d'utiliser une méthode d'interprétation d'un schéma relationnel en langage naturel et d'utiliser les résultats pour créer une ontologie. La méthode NaLER (Natural Language for Entity Relationship) a été développée dans le but de fournir un moyen d'exprimer de façon compréhensible un schéma relationnel [AP98]. Elle se base sur l'analyse du modèle, des clefs primaires et étrangères et de façon générale de toutes les relations présente dans la base de données afin de générer des descriptions en langage naturel. Ces descriptions sont de différents types : informations sur des classes disjointes, identification de propriétés mono ou multi-valuées, relations taxonomiques ou restrictions sur des propriétés.

Pour chacune de ces descriptions Sampson propose une traduction en terme de structure ontologique, ainsi une relation taxonomique permettra de définir des types et des sous-types tandis qu'une propriété multi-valuée permettra de définir différentes relations.

Pour conclure ce bref état de l'art, on peut rappeler que les techniques d'extraction d'ontologie à partir des données sont nombreuses mais aucune d'entre elles n'est réellement générique. Il est ainsi nécessaire d'adapter les méthodes existantes à la tâche à effectuer. Dans le cas de l'approche KEOPS nous avons essayé de proposer quelques principes directeurs avant de mettre au point notre propre solution.

6.3.2 Principes fondateurs pour la création de l'ODIS

L'objectif de l'approche KEOPS est d'extraire des modèles qui s'avèrent intéressants après confrontation aux connaissances de l'utilisateur. Lors de la construction de l'ontologie il est alors nécessaire d'être particulièrement attentif sur quelques points essentiels qui sont toutefois habituels dans le domaine de la fouille de données :

- Le niveau d'agrégation ou de discrétisation des concepts-valeurs qui peut rendre inefficace une fouille de données.
- La corrélation des données qui risque de produire des motifs triviaux et peu intéressants.
- Le nombre et la structure des tables de la base de données.

Le niveau d'agrégation des données

Lors de la création de l'ontologie, un soin particulier doit être apporté à la création de concepts. Ceux-ci doivent définir des domaines sémantiques différents afin que les domaines de leurs valeurs soient disjoints.

Exemple 6.6 (La rubrique *Simulation*)
Dans le domaine des CAF l'attribut *« Rubrique »* est utilisé pour caractériser les contacts qui ont lieu au moyen d'une borne interactive. Cette approche très opérationnelle (on enregistre le nom du menu consulté) pose toutefois des problèmes lorsque l'on désire effectuer une fouille de données. En effet, les intitulés des rubriques ne donnent pas une unique information. Par exemple, la rubrique *« Simulation »*

permet à l'allocataire de faire une simulation du montant des prestations logements qui lui seront versées. Cette valeur ne peut donc être transformée directement en un seul concept car elle donne plusieurs informations :

- La nature de la prestation concernée : une prestation logement.
- Le motif du contact : une demande d'information.
- La modalité du contact : l'utilisation d'une borne interactive.
- Le résultat du contact : l'allocataire a reçu l'information.

La corrélation des données

Si deux attributs de la base de données d'origine sont très corrélés il est plus pertinent de ne créer qu'un seul domaine de valeurs dans la MODB et d'enrichir l'ontologie des relations entre ces valeurs. On évite ainsi la génération de relations triviales et la perte d'informations.

Exemple 6.7 (Ville, département et région)

Si l'on a dans une table trois attributs différents pour décrire la ville, le département et la région où se situe une CAF cela pourrait générer lors de la fouille de données un certain nombre de motifs triviaux si les trois attributs sont conservés. Dans l'approche KEOPS un seul domaine est créé et celui-ci est associé à un ensemble de valeurs parmi lesquelles on définit des relations de généralisation (voir l'exemple de *Localisation CAF* sur la figure 6.1).

Le nombre et la structure des tables de la MODB Chacune des tables de la MODB doit représenter une thématique intéressante pour effectuer une fouille. Souvent les bases de données métiers possèdent plusieurs tables décrivant des informations très proches. Il est souhaitable, dans la mesure du possible, de structurer la MODB afin que ces tables soient intégrées en une seule. Toutefois le but n'est pas d'avoir un nombre minimal de tables, il est important de respecter autant que possible les formes normales définies pour les bases de données relationnelles. Ainsi lors de la génération des jeux de données il est alors aisé d'effectuer des jointures afin de réaliser une fouille de données selon des objectifs précis ; les formes normales n'ayant pas à être respectées dans les jeux de données prêt à être fouillés.

Exemple 6.8

Cet exemple concerne les différentes modalités de contacts : à l'accueil d'une agence, par téléphone et par borne interactive. Les informations sont stockées dans trois

tables, une table « gca2004 » pour les contacts à l'accueil et par téléphone et deux tables « bip » et « biw » pour les informations concernant deux différents types de bornes interactives. Le tableau 6.1 contient la liste des attributs présents dans « gca2004 », « bip » et « biw ». Ces trois tables serviront d'exemple dans la section suivante sur la méthode de construction de l'ODIS.

Table	Attribut
bip	matricule
	numéro caf
	numéro borne
	date connexion borne
	site implantation borne
	rubrique sélectionnée
biw	matricule
	numéro caf
	numéro borne
	date contact borne
	rubrique contact borne
	sous-rubrique contact borne
	site accueil borne
gca	matricule
	date contact
	rang contact
	heure arrivée contact
	nature contact
	motif contact
	heure début contact
	durée contact
	nature pf contact
	résultat contact
	type personne contact
	site accueil contact
	guichet contact

TAB. 6.1: Attributs des tables d'origine

6.3.3 Méthode de construction

La méthodologie KEOPS introduit une variante à CRISP-DM lors de la phase de préparation des données. Dans cette phase CRISP-DM décrit les cinq tâches suivantes : sélection, nettoyage, construction, intégration et formatage des données. Dans la méthodologie KEOPS les phases de nettoyage, construction et intégration sont fusionnées dans le but d'éliciter les concepts de l'ontologie et de permettre la construction de la base de données orientée fouille (MODB) [BCP05, BCP06].

Étape 1 : Définition de la portée et sélection des attributs d'origine

Les premières étapes de la méthode KEOPS correspondent aux étapes de compréhension du domaine et de compréhension des données de la méthode CRISP-DM. Ces étapes nécessitent une interaction importante avec les experts du domaine afin de :

- Déterminer les objectifs de la fouille de données
- Définir des thèmes : les données sont regroupées en ensembles sémantiques que l'on appelle thèmes.
- Associer à chaque thème un ensemble d'attributs avec l'aide des experts

Exemple 6.9

Dans le cadre de la CAF l'objectif de la fouille était de mieux appréhender la problématique de la relation allocataire. Cette problématique couvre trois thèmes : les profils d'allocataires, les contacts (par téléphone, courrier, courriel, à l'accueil, ...) et les événéments pouvant influencer le comportement de l'allocataire (vacances, rentrée scolaire, naissance, mariage, ...). Enfin pour chaque thème il a été nécessaire de déterminer un ensemble d'attributs pertinents. Le tableau 6.2 montre, pour le thème du contact, les attributs qui n'ont pas été conservés.

Étape 2 : Analyse des données et création des concepts-attributs

Soit C l'ensemble des concepts de l'ontologie. Soit z un attribut de la base de données d'origine. On appelle C_z l'ensemble des concepts de l'ontologie associés à l'attribut z.

- Pour chaque attribut z sélectionné dans l'ensemble des attributs définissant un des thèmes :

Table	Attribut	Raison de l'élimination
bip et biw	numéro caf	Les données récoltées ne concernaient qu'une seule CAF
	numéro borne	Le numéro de série de la borne utilisée n'a pas été jugé pertinent par les experts
gca	rang contact	Redondant avec heure d'arrivé contact
	heure début contact	A remplacer par la durée d'attente
	type personne contact	Attribut non jugé pertinent car une valeur est sur-représentée
	guichet contact	La fouille ne doit pas porter sur le comportement des employés

TAB. 6.2: Attributs non considérés pour l'étude du contact allocataire

  - Considérer son nom et sa description pour :
    - Associer n concepts à l'attribut z.
    - Nettoyer C des homonymes (concepts différents avec le même nom), des synonymes (même concepts mais des noms différents comme l'âge et la date de naissance par exemple), et des attributs inutiles selon nos objectifs
  - Étudier les valeurs (distribution, valeurs manquantes, doublons, ...) afin de :
    - Raffiner C_z (ajout ou suppression de concepts) selon les informations obtenues lors de l'étude
    - Nettoyer de nouveau les homonymes, synonymes et attributs inutiles
- Pour chaque concept associé à z, créer la procédure qui générera les concepts-valeurs

L'ensemble des concepts-attributs C de l'ontologie doit rester cohérent, c'est-à-dire sans homonyme ou synonyme. Par exemple l'attribut *« nature pf contact »* de la table *« gca »* serait susceptible de conduire à la création d'un concept synonyme à *« Prestation »*. L'analyse des valeurs des attributs est également importante pour obtenir une ontologie cohérente. Par exemple, nous pourrions avoir deux concepts *« Allocation »* dont l'un serait le montant de l'allocation et l'autre le type d'allocation versé, c'est-à-dire une *« Prestation »*.

Laurent BRISSON

Exemple 6.10
Lors de cette phase un attribut comme *« rubrique sélectionnée »* de la table *« bip »* est associé à quatre concepts-attributs différents, qui seront représentés dans la MODB par quatre attributs :

- Modalité Contact : pour exprimer la nature du contact.
- Prestation : pour exprimer la prestation concernée par le contact.
- Motif Contact : pour exprimer la nature de l'objet du contact.
- Résultat Contact : pour avoir une information sur la réponse donnée à l'allocataire.

Afin de pouvoir générer la MODB il est nécessaire lors de cette étape de conserver certaines informations. Comme l'illustre le tableau 6.3 il faut, pour chaque concept-attribut créé, enregistrer quatre informations :

- les attributs d'origine associés au concept-attribut,
- le nom des tables contenant chaque attribut de la base de données d'origine,
- le domaine de valeur de l'attribut d'origine pour lequel la procédure est définie (le symbole « * » signifiant toutes les valeurs du domaine),
- une référence à une procédure permettant de générer la hiérarchie des concepts-valeurs constituant le domaine.

Exemple 6.11
Parfois, les concepts-valeurs associés à un concept-attribut ne dépendent pas des valeurs d'un attribut de la base d'origine. Comme on peut l'observer dans le tableau 6.3 c'est par exemple le cas pour *« Modalité Contact »* qui prendra la valeur *« Borne Interactive »* au moyen de la procédure *« FModaliteborne »* quelle que soient les données de la table biw2004. Dans ce cas l'information sur l'attribut d'origine est remplacée par le symbole « ? ».

Les procédures évoquées précédemment peuvent être de différentes natures. Dans le cadre de la construction de l'ODIS pour les CAF nous avons deux types de procédures :

- Celles qui utilisent une fonction externe pour générer les concepts valeurs (leur nom commence par F dans le tableau 6.3). Dans notre implantation du système KEOPS nous utilisons plus particulièrement des plugins JAVA.
- Celles qui utilisent une table de correspondance entre valeurs d'origine et concepts valeurs (leur nom commence par T dans le tableau 6.3). Un exemple

Concept-attribut	Attribut d'origine	Table d'origine	Domaine	Procédure
Jour	date connexion borne	biw2004	*	Fjour
Mois	date connexion borne	biw2004	*	Fmois
Semaine	date connexion borne	biw2004	*	Fsemaine
Modalité Contact	?	biw2004	*	FModaliteborne
Prestation	rubrique sélectionnée	biw2004	*	Tnatpfbiw
Motif Contact	rubrique sélectionnée	biw2004	*	Tmotctabiw
Résultat Contact	rubrique sélectionnée	biw2004	*	Tresctabiw
Lieu Contact	site implantation borne	biw2004	*	Tsiteimpl
Type Structure	?	biw2004	*	Ttypstrbiw

TAB. 6.3: Informations nécessaires à la création des concepts-attributs concernant les données des bornes interactives (biw2004)

de table de correspondance est représentée par le tableau 6.4. Parfois un attribut recouvre plusieurs domaines mais dans certaine situation il n'apporte aucune information concernant un domaine, la valeur est alors considérée comme manquante et est représentée par le symbole « ? ».

Étape 3 : Création des concepts-valeurs

Lors de cette étape, il s'agit de définir tous les concepts valeurs :

Valeur d'origine	Concept-valeur	Concept-attribut associé
Accès Rubrique Simulation Borne	Prestation Logement	Nature Prestation
Accès Rubrique Simulation Borne	Demande Renseignement	Motif Contact
Accès Rubrique Simulation Borne	Réponse Finalisee	Résultat Contact
Changement de situation (EDFBOR[3])	?	Nature Prestation
Changement de situation (EDFBOR)	Déclaration Événement	Motif Contact
Changement de situation (EDFBOR)	Retrait Document	Résultat Contact

TAB. 6.4: Une valeur de la base de données d'origine peut être associée à plusieurs domaines (Exemple concernant les bornes interactives)

- Donner un nom à chaque concept-valeur.
- Nettoyer homonymes et synonymes parmi les concepts-valeurs.

Exemple 6.12
Dans le cadre de la CAF nous avons souvent des montants en euros, des âges ou encore des durées. Ainsi le concept *« 1-5 »* pourrait tout aussi bien déterminer le nombre d'enfants à charge d'un foyer qu'une durée d'attente. Il est donc particulièrement recommandé de prêter attention à de pareilles situations, où il faudrait créer deux concepts différents : *« 1-5enf_ acharge »* (pour les enfants à charge) et *«1-5min_ da »* (pour la durée d'attente) et en conserver également une description en langage naturel.

Étape 4 : Construction de l'ontologie

- Identifier les relations de généralisation parmi les concepts-valeurs (voir flèches vertes figure 6.1).

[3]EDFBOR est un acronyme pour la rubrique Édition De Formulaire

- Créer les relations sémantiques entre concepts-valeurs de hiérarchies différentes (voir flèches violettes figure 6.1).

Étape 5 : Construction de la base de données orientée pour la fouille (MODB)

- Générer la base de données en utilisant les procédures définies lors de la deuxième étape.

Lors de cette dernière étape un programme lit les informations conservées pour chaque concept-attribut et exécute chacune des procédures afin de générer la MODB.

Exemple 6.13
Dans le contexte des CAF nous avons créé au sein de la MODB une table *contacts* dont le but était de fournir une structure de données générique pour tous les types de contacts qui étaient auparavant dispersés dans plusieurs tables (bip,biw et gca). Les attributs suivants ont été choisis pour créer la table *contacts* :

- Distance
- Durée Contact
- Heure Arrivée Contact
- Jour
- Mois
- Semaine
- Modalité Contact
- Motif Contact
- Prestation
- Résultat Contact
- Temps Attente
- Type Structure

Á cela on ajoute l'attribut « id », la clef primaire de la table, l'attribut « matricule » la clef étrangère de la table concernant les profils allocataire et « date », la clef étrangère de la table concernant les événements. Les clefs ne correspondent cependant à aucun des concepts-attributs définis dans l'ontologie.

6.3.4 Génération de jeux de données

En général, un processus de fouille de données nécessite d'itérer les tests et conduit à reconsidérer les choix effectués lors de la construction du jeu de données. Bien que des algorithmes existent pour choisir les attributs les plus pertinents parfois les résultats ne correspondent pas aux attentes de l'utilisateur et le jeu de données à fouiller doit être recréé. Dans notre approche, l'ontologie permet de décrire des domaines de valeurs mais aussi les relations existant entre ces valeurs. Par conséquent de nombreux jeux de données peuvent être générés en fonction des choix que l'on effectue.
L'ODIS permet de factoriser toutes les opérations de pré-traitements sur les données, afin de générer à partir d'une même base de données des jeux de données partageant les mêmes valeurs. La construction d'un jeu de données s'effectue en combinant les trois opérations suivantes :

- La projection et la sélection (au sens de l'algèbre relationnelle) qui permettent de ne conserver respectivement que certains attributs d'une relation et que certains tuples d'une table
- Le choix de la granularité des données : cet opérateur permet de choisir au sein de l'ontologie le niveau de généralisation des concepts présents dans le jeu de données

Le tableau 6.5 montre un extrait du contenu de la MODB pour quelques attributs.

id	Heure Arrivée Contact	Modalité Contact	Prestation	Résultat Contact
1	?	Appel Entrant	API	?
2	8h30-11h29	Appel Entrant	AFEAMA	Réponse Finalisée
3	11h30-13h29	Contact Borne	BAFA	Réponse Finalisée
4	?	Contact Borne	AGED	Réponse Finalisée

TAB. 6.5: Extrait de valeurs de la MODB pour quelques attributs

Si l'on s'intéresse aux différentes catégories de prestation selon la modalité du contact il serait par exemple possible de générer un premier jeu de données (voir tableau 6.6) où l'on aurait les concepts les plus généraux pour l'attribut Prestation.

Cependant si l'on s'intéresse aux moments de la journée (matin, midi et soir) lors desquels ont lieu des contacts par borne interactive le jeu de données contiendra des

id	Heure Arrivée Contact	Modalité Contact	Prestation	Résultat Contact
1	?	Appel Entrant	Prestation Minima Sociaux	?
2	8h30-11h29	Appel Entrant	Prestation Jeune Enfant	Réponse Finalisée
3	11h30-13h29	Contact Borne	Prestation Action Sociale	Réponse Finalisée
4	?	Contact Borne	Prestation Jeune Enfant	Réponse Finalisée

TAB. 6.6: Exemple de jeu de données avec des concepts plus généraux pour les Prestations

concepts plus généraux pour l'heure d'arrivée mais la colonne *« Résultat Contact »* sera supprimée car la valeur *« Réponse Finalisée »* est extrêmement fréquente pour les bornes interactives et risquerait de fausser les résultats de la fouille de données (voir tableau 6.7).

id	Heure Arrivée Contact	Modalité Contact	Prestation
3	Midi	Contact Borne	BAFA
4	?	Contact Borne	AGED

TAB. 6.7: Exemple de jeu de données avec des concepts plus généraux pour l'heure d'arrivée, la sélection des contacts par borne interactive et la suppression de la colonne Résultat

6.4 La base de connaissances

Lorini, dans ses travaux sur les agents cognitifs, présente les connaissances comme un état mental décrivant une mémoire mais aussi des perceptions, des convictions,

des prédictions et des objectifs [LF05]. L'ensemble de ces informations permet alors de modéliser les attentes d'un utilisateur en conservant les propriétés épistémiques (le savoir) et les propriétés cognitives (la capacité de déduction). Lorini propose de décrire un état mental en définissant ses deux principaux aspects :

- les convictions auxquelles on associe un degré de certitude,
- les objectifs auxquels on associe une valeur permettant de déterminer leur importance aux yeux d'un utilisateur.

Dans cette section nous nous intéressons à la description du premier aspect, c'est-à-dire la modélisation des convictions de l'utilisateur. La prise en compte d'objectifs sera présentée dans le chapitre 8 concernant les stratégies à mettre en œuvre dans un processus de fouille de données intégrant les connaissances de l'utilisateur.

6.4.1 Les propriétés des connaissances

Les connaissances s'expriment sous la forme de règles du type « si ... alors ... ». Ce choix est justifié par le fait qu'il est très intuitif pour un expert du domaine d'exprimer ses connaissances sous cette forme. Toutefois l'information contenue dans une règle n'est pas suffisante. Il est nécessaire que l'expert renseigne également les propriétés suivantes :

- Indice de confiance : plage de valeurs décrivant la confiance de la règle estimée par l'expert.
- Degré de certitude :
  - Trivialité : connaissance dont l'expert est certain de l'exactitude,
  - Standard : connaissance du domaine,
  - Hypothèse : connaissance à vérifier.

L'indice de confiance tel que nous le définissons est conforme à la définition de la confiance utilisée en fouille de données, c'est-à-dire :

Rappels 6.2 La confiance d'une règle $R : A \rightarrow B$, notée $conf(A \rightarrow B)$, est la probabilité conditionnelle que B soit vérifié sachant que A est vérifié.

Exemple 6.14 (Rôle de l'indice de confiance)
La connaissance suivante signifie qu'on émet l'hypothèse que dans 60% à 80% des cas les allocataires viennent à l'accueil, s'ils habitent près de la CAF et qu'ils désirent une information sur le paiement de leur prestation logement :

Connaissance 6.1
Motif Contact="Paiement" ∧ Prestation="Prestation Logement" ∧ Distance="Proche"
→ Modalité Contact="Accueil"

- *Indice de confiance : 60-80%*
- *Degré de certitude : Hypothèse*

On pourrait par contre avoir envie de préciser que quelle que soit la prestation logement concernée, si les allocataires habitent loin de la CAF il ne viendront pas à l'accueil pour obtenir une information. Ne pouvant utiliser de négation dans nos connaissances, nous allons exprimer cette connaissance en influant sur l'indice de confiance :

Connaissance 6.2
Motif Contact="Paiement" ∧ Prestation="Prestation Logement" ∧ Distance="Loin"
→ Modalité Contact="Accueil"

- *Indice de confiance : **0-20%***
- *Degré de certitude : Standard*

Exemple 6.15 (Les trivialités)
Les connaissances triviales sont principalement de deux natures différentes :

- une connaissance évidente pour un expert du domaine,
- un artefact gênant dans les données qui pourrait générer des modèles inintéressants.

Prenons l'exemple d'une CAF qui vient de placer des bornes interactives dans les locaux de son siège afin de les tester. Aucune autre agence n'en possède encore dans le département. Il est alors utile de créer la connaissance suivante :

Connaissance 6.3
Modalité Contact="Borne Interactive" → TypeStructure="Siège"

- *Indice de confiance : 80-100%*
- *Degré de certitude : **Trivialité***

Dans le cas présent l'indice de confiance est à 80-100% car toutes les bornes ont été installées au siège ! Mais certaines trivialités n'ont pas nécessairement un indice de confiance élevé : il pourrait par exemple être trivial qu'un événement ait lieu une fois sur deux.

Les connaissances triviales sont également utiles pour exprimer des connaissances sur les données. Nous avons précédemment évoqué le concept *« Résultat Contact »* servant à décrire l'action qui a été effectuée : impression de document, renseignement transmis, modification du dossier. Les bornes interactives dans 80% des cas ne servent qu'à informer les allocataires et parfois (20% des cas) à imprimer un document. Ainsi si l'on désire ne pas retrouver ces informations triviales dans les résultats de la fouille de données on peut exprimer les connaissances suivantes :

Connaissance 6.4
Modalité Contact="Borne Interactive" → *Résultat Contact="Renseignement"*

- *Indice de confiance : 80-100%*
- *Degré de certitude :* ***Trivialité***

Connaissance 6.5
Modalité Contact="Borne Interactive" → *Résultat Contact="Retrait Document"*

- *Indice de confiance : 0-20%*
- *Degré de certitude :* ***Trivialité***

6.4.2 Structure de la base de connaissances

La connaissance, au même titre que la notion d'intérêt, est quelque chose qui diffère selon les personnes, évolue au cours du temps et est très dépendante de l'activité en cours. C'est pourquoi le choix a été fait de structurer les connaissances de la base de connaissances en différents ensembles. Il existe un ensemble général afin de définir toutes les connaissances consensuelles sur un domaine ; ensuite chaque utilisateur du système peut définir l'ensemble des connaissances qui lui sont propres et qui peuvent être contradictoires avec les connaissances des autres utilisateurs. L'objectif d'une telle structure est de permettre que les modèles présentés à l'utilisateur soient choisis en fonction des connaissances actuelles d'un utilisateur.

Exemple 6.16 (Connaissances personnalisées)
Si l'on considère par exemple une connaissance concernant les paiements, la suivante pourrait être consensuelle entre différentes CAF :

Connaissance 6.6
Semaine="1ère Semaine" ∧ *Modalité Contact="Accueil"* → *Motif Contact="Paiement"*

- *Indice de confiance : 60-80%*
- *Degré de certitude : Standard*

Cependant à Nice, la population étant plus âgée, l'expert voudrait préciser cette connaissance de la façon suivante :

Connaissance 6.7
Semaine="1ère Semaine" $\wedge$ *Modalité Contact="Accueil"* $\rightarrow$ *Motif Contact="Paiement"* $\wedge$ *Age Allocataire="**Plus de 40ans**"*

- *Indice de confiance : 60-80%*
- *Degré de certitude : Standard*

Tandis qu'à Grenoble, où la population est beaucoup plus jeune, l'expert habitué choisirait plutôt d'exprimer :

Connaissance 6.8
Semaine="1ère Semaine" $\wedge$ *Modalité Contact="Accueil"* $\rightarrow$ *Motif Contact="Paiement"* $\wedge$ *Age Allocataire="**Moins de 25ans**"*

- *Indice de confiance : 60-80%*
- *Degré de certitude : Standard*

Et si un expert niçois est un jour affecté à la CAF de Grenoble il saurait qu'il doit changer sa façon de penser, mais pas exactement dans quelle mesure. Il pourrait alors exprimer cette hypothèse :

Connaissance 6.9
Semaine="1ère Semaine" $\wedge$ *Modalité Contact="Accueil"* $\rightarrow$ *Motif Contact="Paiement"* $\wedge$ *Age Allocataire="**Moins de 30ans**"*

- *Indice de confiance : 60-80%*
- *Degré de certitude :* ***Hypothèse***

6.5 Conclusion

Dans les chapitres précédents le constat a été fait qu'il était nécessaire d'utiliser le formalisme des techniques de représentation des connaissances pour concevoir des mesures d'intérêt subjectif. Toutefois se limiter à la phase de post-traitement était limitatif et nous avons donc proposé d'utiliser cette formalisation tout au long du

processus d'extraction des connaissances à partir des données. L'originalité de l'approche KEOPS est de modéliser les connaissances lors de la phase de pré-traitement des données : non seulement les données peuvent alors être préparées en relation avec les connaissances du domaine mais les connaissances de l'expert du domaine peuvent être utilisées dans l'algorithme de fouille de données et lors de la phase de post-traitement. La confrontation des modèles extraits aux connaissances est d'autant plus facilitée que leur expression est très similaire car formée à partir des mêmes concepts.

Les connaissances modélisées dans le cadre de l'approche KEOPS sont stockées dans une base de connaissances sous forme de règles de causalité. Nous évoquons le fait que chaque utilisateur peut avoir des connaissances propres mais aussi partager les connaissances consensuelles avec d'autres utilisateurs. Dans l'état actuel du développement de l'approche KEOPS nous ne proposons pas de système de gestion des connaissances, cependant si un tel système n'est pas une nécessité pour le fonctionnement de KEOPS il peut être très intéressant afin de gérer les connaissances contradictoires et d'effectuer des vérifications préalables avant leur utilisation dans KEOPS. Nous avons donc le projet d'interfacer KEOPS avec un système de gestion des connaissances.

Chapitre 7

Confrontation des modèles aux connaissances

Sommaire

7.1 Motivations

Les algorithmes de fouille de données peuvent générer dans certains cas un nombre important de modèles selon les données fournies et les paramètres définis pour l'extraction. Une étape essentielle de la fouille de données est donc l'étape de post-traitement lors de laquelle les modèles sont réduits pour ne conserver que les connaissances les plus intéressantes. Des approches de différentes natures ont été proposées : il peut s'agir d'éliminer les redondances au sein des modèles, de filtrer les meilleurs modèles selon un certain seuil pour une mesure de qualité donnée ou selon un des critères subjectifs présentés dans le chapitre 3.

Dans l'approche KEOPS, nous caractérisons l'*intéressabilité* d'un modèle selon quatre critères :

1. L'intelligibilité,
2. La taille des modèles afin de ne pas présenter simultanément un trop grand nombre de motifs à l'utilisateur,
3. La facilité de confrontation avec la connaissance a priori afin d'éviter à l'utilisateur l'obligation d'effectuer des transformations logiques trop complexes ou trop nombreuses pour confirmer ses intuitions,
4. La personnalisation afin de ne fournir à un utilisateur que les règles intéressantes en fonction de ses connaissances et de ses centres d'intérêts.

Le critère d'intelligibilité est une constante dans la plupart des travaux sur ce sujet ce qui justifie le choix de modèles sous la forme de règles de causalité « si ... alors ... » du même type que les règles d'association. Les règles d'association ainsi que les mesures statistiques nécessaires afin de les interpréter correctement (support, confiance et lift) ont, en effet, l'avantage d'être facilement compréhensibles par un utilisateur non expert. Ainsi nous supposons donc être en mesure d'exprimer les connaissances du domaine dans ce format.

Pour satisfaire le second critère, nous utilisons en premier lieu l'algorithme CLOSE qui assure qu'aucune redondance syntaxique ne soit générée. Ensuite, nous étendons le concept de maximisation du niveau d'information en prenant en compte toutes les relations de généralisation définies dans l'ontologie.

La structure du système d'information conceptuel et l'utilisation de la base de connaissances permettent de respecter le troisième et le quatrième critère. Nous proposons en particulier une mesure d'intérêt intégrant les intérêts objectifs et subjectifs

afin d'évaluer l'intérêt des règles en fonction des connaissances définies dans la base de connaissances par un utilisateur.

Dans le cadre de ce chapitre où nous nous intéressons à des motifs sous forme de règles d'association, nous employons par abus de langage le terme d'*attribut* pour désigner un concept-attribut et de *valeur* pour désigner un concept-valeur au sein d'une règle. Cet abus de langage se justifie par le fait qu'un unique concept-attribut (respectivement concept-valeur) est associé à un attribut (respectivement une valeur) de la MODB.

Ce chapitre est organisé de la manière suivante : dans la première partie nous nous intéressons aux techniques pour éliminer les règles d'association redondantes, la deuxième partie décrit notre approche pour l'évaluation relative du niveau d'information de deux règles et la troisième partie présente la mesure d'intérêt que nous avons mise au point afin d'évaluer la pertinence d'une règle en fonction des connaissances.

7.2 Réduction du nombre de règles d'association

Il existe différentes approches afin de sélectionner les modèles les plus intéressants à présenter à l'utilisateur. Les approches les plus simples sélectionnent les n meilleures règles selon une mesure d'intérêt ou encore toutes les règles dépassant un seuil fixé. Dans le chapitre 2 nous avons évoqué différents critères de qualité pour la création de mesures d'évaluation des règles extraites. Ces mesures permettent de sélectionner les règles les plus intéressantes en fixant un seuil minimal d'intérêt. D'autres approches ont pour objectif de filtrer les règles redondantes comme par exemple la méthode d'extraction basée sur les itemsets fermés fréquents présentée en section 2.4.1.2. Dans l'approche KEOPS le choix a été fait de se baser sur l'algorithme CLOSE pour l'extraction de motifs qui sont ensuite traités afin d'éliminer les redondances d'origine sémantique qui ne peuvent être identifiées par l'algorithme.

7.2.1 Maximisation du niveau d'information

L'algorithme CLOSE [Pas00] permet d'extraire des règles d'association non redondantes minimales. Dans ce contexte, une règle est dite redondante si elle convoie la même information ou une information moins générale que l'information convoyée

par une autre règle de même utilité et de même pertinence.

7.2.2 Factorisation de règles d'association

L'approche KEOPS se fonde sur la définition de *règle d'association généralisée* présentée par Srikant [SA95] dans laquelle une règle d'association généralisée est composée d'items organisés en une taxonomie $\mathcal{T}$. Avant d'introduire notre définition il est nécessaire de présenter certaines notions.

Définition 7.1 (Chemin entre deux concepts)
On appelle chemin entre les concepts C_1 et C_n une suite de concepts $C_1, C_2, ..., C_n$ dans laquelle deux concepts successifs quelconques C_i et C_{i+1} sont reliés par une relation orientée de C_i vers C_{i+1}

Dans le cadre de la factorisation des données on s'intéresse plus particulièrement aux chemins créés à partir d'une même relation de généralisation reliant un concept à son ancêtre, on les appelle : *chemin de généralisation.*

Exemple 7.1
La figure 6.1 illustre un chemin composé d'une relation sémantique et de relations de généralisation entre « Domicile Lyon » et « CAF Rhône-Alpes ». Sur la figure 7.1 on peut observer un chemin de généralisation entre les concepts-valeurs « CV_{111} » et « CV_1 ».

On rappelle qu'un item est définit par un triplet {A,op,V} où :
- A est un attribut du jeu de données,
- op est un opérateur parmi $<$, $\leq$, $>$, $\geq$, $=$,
- V est une valeur du domaine de l'attribut.

Définition 7.2 (Généralisation d'un item)
Soit un item I_A défini par un couple (Att, Val_A) et un item I_B défini par le couple (Att, Val_B).
On dit qu'un item I_A généralise un item I_B si $I_A = I_B$ ou bien s'il existe un chemin de généralisation de Val_B vers Val_A.

Exemple 7.2
La figure 7.1 illustre que l'item « $CA = CV_1$ » généralise l'item « $CA = CV_{111}$ ».

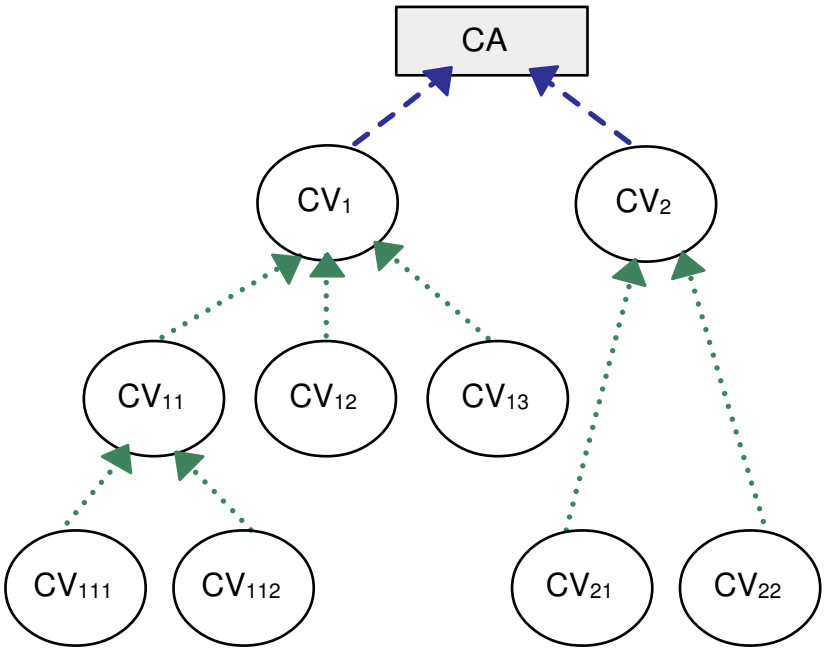


FIG. 7.1: Un domaine et ses sous-domaines

La notion de généralisation d'un item permet de définir les règles d'association généralisées. Ces règles, formées uniquement à partir des concepts d'une ontologie KEOPS, sont « minimales » dans le sens où elles excluent toute relation non pertinente entre les concepts-valeurs de leurs items :

Définition 7.3 (Règle d'association généralisée)
Soit CA et CV respectivement l'ensemble des concepts-attributs et des concepts-valeurs d'une ontologie KEOPS.
Soit $\mathcal{I}$ l'ensemble des items de la MODB créés à partir de CA et CV.
Une règle $R : A \rightarrow C$ est une règle d'association généralisée si et seulement si :

- $A \subset \mathcal{I}$
- $C \subset \mathcal{I}$
- *Aucun item de A ne généralise un autre item de A*
- *Aucun item de C ne généralise un item de A ou un autre item de C*

Les seules relations autorisées entre les items d'une règle d'association généralisée sont les relations sémantiques et les relations de généralisation d'un item de C vers un item de A. Afin de factoriser plusieurs règles d'association généralisées il est nécessaire de définir entre elles une relation d'ordre :

L'étape de factorisation des règles consiste à combiner un ensemble de règles proches en une nouvelle qui permet de toutes les résumer. Avant de la présenter il est nécessaire de rappeler la notion de « sous-domaine ». Si l'on considère CA un concept-attribut et un ensemble de concepts-valeurs en relation « valeurDe » avec CA définissant un domaine $\mathcal{D}$. Un concept-valeur CV de $\mathcal{D}$ est appelé un sous-domaine de $\mathcal{D}$ s'il existe un ensemble de concepts-valeurs $\{CV_1, \ldots, CV_n\}$ de $\mathcal{D}$ en relation de généralisation avec CV. On dit que $\{CV_1, \ldots, CV_n\}$ définit le sous-domaine CV.

Sur la figure 7.1 le domaine $\mathcal{D}$ de CA est représenté par $\{CV_1, CV_2\}$ où CV_1 et CV_2 sont deux sous-domaines de $\mathcal{D}$. Le sous-domaine CV_1 est défini quant à lui par l'ensemble $\{CV_{11}, CV_{12}, CV_{13}\}$ ou encore par $\{CV_{111}, CV_{112}, CV_{12}, CV_{13}\}$. La figure 7.2 représente le sous-domaine « Prestation Action Sociale » tandis que la figure 6.1 du chapitre 6 représente les sous-domaines « Prestation Logement » et « Prestation Entretien ».

Définition 7.4 (Factorisation de règles)
Soit un ensemble de règles $E = \{R_1, \ldots, R_n\}$ *tel que :*

$$\forall i \in \{1, n\}\ R_i : att_1 = val_{(i,1)}, \ldots, att_p = val_{(i,p)} \rightarrow att_{p+1} = val_{(i,p+1)}, \ldots, att_q = val_{(i,q)}$$

On dit que l'ensemble E *se factorise en une seule règle* R *de la forme :*

$$R : att_1 = val_1, \ldots, att_p = val_p \rightarrow att_{p+1} = val_{p+1}, \ldots, att_q = val_q$$

si : $\forall j \in \{1, q\}\ \{val_{1,j}, \ldots, val_{n,j}\}$ *définit le sous-domaine* val_j
ou bien si : $\forall r \in \{1, q\}\ val_{(r,j)} = val_j$.

Exemple 7.3
Considérons l'ontologie simplifiée illustrée par la figure 7.2. Les règles 7.2, 7.3, 7.4 ne peuvent se factoriser en la règle 7.1 car les concepts-valeurs « BAFA », « PAH » et « PEL » ne définissent pas complètement le sous-domaine « Prestation Action Sociale ». Toutefois, dès lors que la règle 7.5 apparait il devient possible de factoriser l'ensemble des règles. Bien entendu dans notre exemple Lyon et Grenoble définissent le sous-domaine « Rhône-Alpes » ce qui ne reflète pas la situation réelle.

Règle 7.1
Localisation CAF="Rhône-Alpes" $\wedge$ *Prestation="Prestation Action Sociale"* $\rightarrow$ *Heure Arrivée="Matin"*

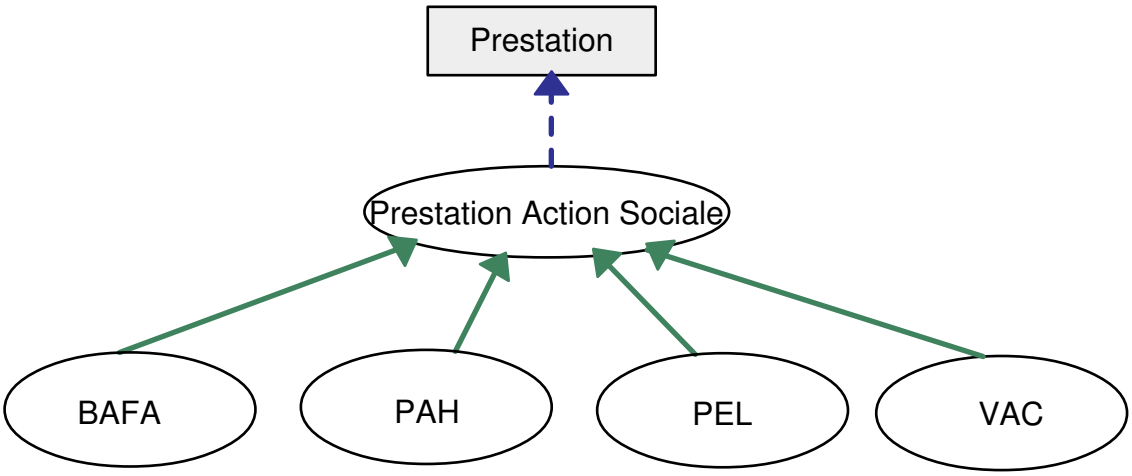


FIG. 7.2: Le sous-domaine « Prestation Action Sociale » dans une ontologie dédiée au contact allocataire

Règle 7.2
Localisation CAF="Grenoble" $\wedge$ *Prestation="BAFA"* $\rightarrow$ *Heure Arrivée="Matin"*

Règle 7.3
Localisation CAF="Grenoble" $\wedge$ *Prestation="PAH"* $\rightarrow$ *Heure Arrivée="Matin"*

Règle 7.4
Localisation CAF="Grenoble" $\wedge$ *Prestation="PEL"* $\rightarrow$ *Heure Arrivée="Matin"*

Règle 7.5
Localisation CAF="Lyon" $\wedge$ *Prestation="VAC"* $\rightarrow$ *Heure Arrivée="Matin"*

7.3 Évaluation relative des niveaux d'informations

Un des objectifs de l'approche KEOPS est de confronter les motifs extraits par les techniques de fouilles de données aux connaissances déjà acquises sur le domaine. La confrontation nécessite de disposer de critères de comparaison : un de ces critères est de comparer le niveau d'information de chaque règle aux connaissances. Étant donné qu'une ontologie KEOPS peut définir de nombreuses relations pouvant relier les différents items d'une règle de manières différente, une comparaison syntaxique aboutirait à la nécessité d'associer à chacun des concepts un poids afin de pouvoir comparer deux règles. L'approche que nous présentons permet de comparer le niveau

d'information de deux règles en deux étapes : la première consiste à comparer les couvertures des antécédents et des conséquents des deux règles, et la deuxième consiste à évaluer le niveau d'information en se conformant au principe de maximisation du niveau d'information. Nous définissons également la notion de « comparabilité » de deux règles qui est basée sur l'existence d'un lien quelconque dans l'ontologie entre leurs items respectifs. Dans cette section nous définissons cette notion, nous abordons le mécanisme de comparaison de la couverture de deux itemsets et nous présentons le mécanisme d'évaluation du niveau d'intérêt. Pour finir, nous donnons un récapitulatif de la méthode ainsi que l'algorithme qui lui est associé.

Définition 7.5 (Items comparables)
On dit que deux items sont comparables s'il existe une relation dans l'ontologie entre leurs valeurs.

Définition 7.6 (Itemsets comparables)
On dit qu'un itemset I_1 est comparable à un itemset I_2 s'il existe au moins un item de I_1 comparable à un item de I_2.

Définition 7.7 (Règles comparables)
Soit R_1 une règle du type : $A_1 \rightarrow C_1$.
Soit R_2 une règle du type : $A_2 \rightarrow C_2$.
On dit que deux règles R1 et R2 sont comparables si :

- *A_1 est comparable à A_2* ***et***
- *C_1 est comparable à C_2*

Dans la section 2.4.1 nous avons présenté le formalisme concernant les règles d'association. Cela nous permet d'introduire la définition de la couverture d'un itemset.

Définition 7.8 (Couverture d'un itemset)
Soit un contexte d'extraction de règles d'association $\mathcal{B} = (\mathcal{T}, \mathcal{I}, \mathcal{R})$. La couverture d'un itemset $I \subseteq \mathcal{I}$, notée $f(I)$, est l'ensemble des tuples $\mathcal{T} \subseteq \mathcal{B}$ contenant I. On dit d'un tuple T de $f(I)$ qu'il est couvert *par I.*

7.3.1 Comparaison de la couverture des itemsets

Notation 7.1
Nous appelons couverture d'un itemset I, et on note $f(I)$, l'ensemble des tuples de la base de données contenant I.

Notation 7.2
Nous notons $E \setminus F$ la différence ensembliste entre deux ensembles E et F et $|E|$ le cardinal de l'ensemble E.

Définition 7.9 (Couvertures similaires)
Deux itemsets I_1 et I_2 ont une couverture similaire si :

- $|f(I_1) \setminus f(I_2)| < \delta\,|f(I_1)|$
- $|f(I_2) \setminus f(I_1)| < \delta\,|f(I_2)|$

où $\delta \in$ [0,1] est un coefficient permettant de définir un seuil au-delà duquel le nombre d'exemples de $f(I_2)$ n'appartenant pas à $f(I_1)$ est trop grand. On note $f(I_1) \sim f(I_2)$.

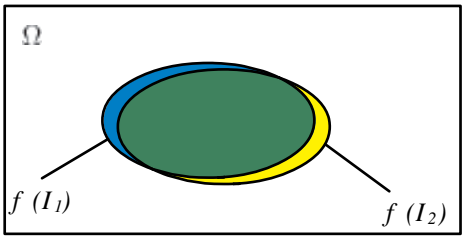


FIG. 7.3: Couverture des itemsets lorsque $f(I_1) \sim f(I_2)$

Comme on peut le voir sur la figure 7.3, I_1 a une couverture similaire à I_2 si :

- Le nombre d'exemples couverts uniquement par I_1 est négligeable par rapport au nombre total d'exemples couverts par I_1,
- Le nombre d'exemples couverts uniquement par I_2 est négligeable par rapport au nombre total d'exemples couverts par I_2.

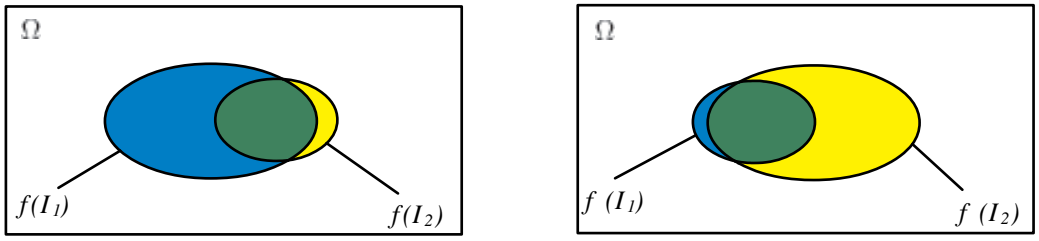
Enfin, la notion de « négligeable » est définie par le coefficient δ paramétrable par l'utilisateur en fonction des données.

Définition 7.10 (Couverture plus étendue)
Un itemset I_1 a une couverture plus étendue qu'un itemset I_2 si :

1. $|f(I_1) \setminus f(I_2)| \geq \delta\,|f(I_1)|$,
2. $|f(I_2) \setminus f(I_1)| < \delta\,|f(I_2)|$.

où $\delta \in$ [0,1] est un coefficient permettant de définir un seuil au-delà duquel le nombre d'exemples de $f(I_2)$ n'appartenant pas à $f(I_1)$ est trop grand. On note $f(I_1) \rhd f(I_2)$.

La couverture de $I1$ est donc plus étendue que celle de I_2 si :



a) $f(I_1) \rhd f(I_2)$ b) $f(I_1) \lhd f(I_2)$

FIG. 7.4: Comparaison de la couverture des itemsets I_1 et I_2

- Le nombre d'exemples couverts uniquement par I_1 est important par rapport au nombre total d'exemples couverts par I_1
- Le nombre d'exemples couverts uniquement par I_2 est négligeable par rapport au nombre total d'exemples couverts par I_2

On peut souligner que cette définition introduit une notion d'égalité *approximative* entre couvertures, moins rigide que l'égalité stricte.

Remarque 7.1 Si l'inégalité $f(I_1) \rhd f(I_2)$ est fausse cela n'implique pas que $f(I_1) \trianglelefteq f(I_2)$ soit vraie car les couvertures des itemsets peuvent être incomparables.

Définition 7.11 (Couvertures incomparables)
On dit que deux itemsets I_1 et I_2 ont une couverture incomparable si :

1. $|f(I_1) \setminus f(I_2)| \geq \delta\,|f(I_1)|$
2. $|f(I_2) \setminus f(I_1)| \geq \delta\,|f(I_2)|$

où $\delta \in [0,1]$ est un coefficient permettant de définir un seuil au-delà duquel le nombre d'exemples de $f(I_2)$ n'appartenant pas à $f(I_1)$ est trop grand. On note $f(I_1) \nsim f(I_2)$.

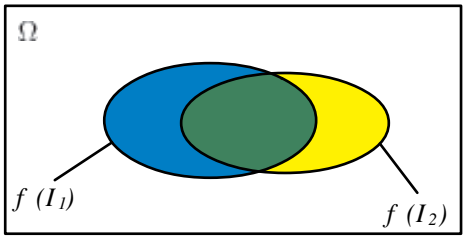


FIG. 7.5: Couverture des itemsets lorsque $f(I_1) \nsim f(I_2)$

La figure 7.5 illustre le cas où les couvertures de I_1 et I_2 sont incomparables : le nombre d'exemples couverts par l'intersection de leurs couvertures $f(I_1) \cap f(I_2)$ est

faible par rapport à leurs tailles respectives.

7.3.2 Comparaison des règles

L'objectif de l'approche KEOPS est de confronter les règles aux connaissances afin d'évaluer leur intérêt. Cependant dans la phase de comparaison des règles il n'est pas encore nécessaire de distinguer la nature des règles. Nous avons introduit précédemment des critères de comparaison de la couverture de deux itemsets. Ceux-ci peuvent être appliqués aux antécédents et aux conséquents de deux règles et permettent d'évaluer leur niveau de généralisation respectif.

Dans le but de comparer deux règles nous reprenons le paradigme qui a conduit au développement de l'algorithme CLOSE, c'est-à-dire la maximisation du niveau d'information. Dans cette section nous présentons une méthode afin de comparer les niveaux d'information de deux règles, en tenant compte des aspects sémantiques liant les items des règles tandis que CLOSE a permis lors de la fouille de données de prendre en compte les aspects syntaxiques.

Nous nous basons sur l'axiome suivant : *Plus la condition d'une règle est restrictive et plus la prédiction est large plus le niveau d'information de la règle augmente.* Dans notre contexte cet axiome se traduit par le fait que la couverture de l'antécédent d'une règle doit être minimal tandis que la couverture de son conséquent doit être maximal.

Définition 7.12 (Niveau d'information supérieur / inférieur)
Pour deux règles comparables R_1 et R_2, R_1 possède un niveau d'information supérieur à R_2 dans les situations suivantes :

- $f(A_1) \trianglelefteq f(A_2)$ *et* $f(C_1) \trianglerighteq f(C_2)$
- $f(A_1) \triangleleft f(A_2)$ *et* $f(C_1) \sim f(C_2)$

Définition 7.13 (Niveaux d'information similaires)
On dit que les niveaux d'information de deux règles comparables R_1 et R_2 sont similaires lorsque les couvertures respectives de leurs antécédents et conséquents sont similaires.

Définition 7.14 (Niveaux d'information non comparables)
Si deux règles $R_1 : A_1 \rightarrow C_1$ et $R_2 : A_2 \rightarrow C_2$ sont non comparables ou si les couvertures de leurs antécédents et conséquents sont non comparables on dit que $R_1 : A_1 \rightarrow C_1$ et $R_2 : A_2 \rightarrow C_2$ ont des niveaux d'information non comparables.

Exemple 7.4
Considérons la connaissance C et la règle extraite R, données ci-dessous. Si l'on compare leur syntaxe on remarque que leur conséquent est identique et que leur antécédent, bien que constitué des mêmes concepts-attributs, est différent. Étant donné que les concepts-valeurs de chacune des règles sont en relation de généralisation, les deux règles sont bien comparables. Cependant le sens de la relation de généralisation n'est pas le même entre antécédents et conséquents, c'est-à-dire que le concept-valeur « Rhône-Alpes » de la connaissance C est plus général que le concept-valeur « Grenoble » de la règle R tandis que le concept-valeur « Prestation Logement » de la règle R est plus général que le concept-valeur « APL » de la connaissance C. Ainsi cet exemple illustre bien le fait que, selon la distribution des valeurs dans les données, l'une ou l'autre des règles sera plus informative au sens où nous l'avons défini.

Connaissance C

Localisation CAF = "Rhône-Alpes" $\wedge$ Prestation = "APL" $\rightarrow$ Motif Contact = "Paiement"

Règle R

Localisation CAF = "Grenoble" $\wedge$ Prestation = "Prestation Logement" $\rightarrow$ Motif Contact = "Paiement"

Remarque 7.2 Dans le cadre de cet exemple, {Grenoble,Hors-Grenoble} est le sous-domaine de « Rhône-Alpes » et {APL,ALS} le sous-domaine de « Prestation Logement ».

Nous étudions par la suite les conclusions diverses qui peuvent être obtenues sur le niveau d'information respectif des règles R et C selon la situation relative de leur couverture.

Situation où la règle extraite est plus informative

Le tableau 7.1 illustre un exemple virtuel de répartition des allocataires en fonction du lieu de contact et du type de prestation logement qu'ils reçoivent. La figure 7.6 illustre la même situation graphiquement.

	Grenoble	Hors-Grenoble
APL	95	200
ALS	5	

TAB. 7.1: Répartition des effectifs allocataires en fonction du lieu de contact et du type de prestation

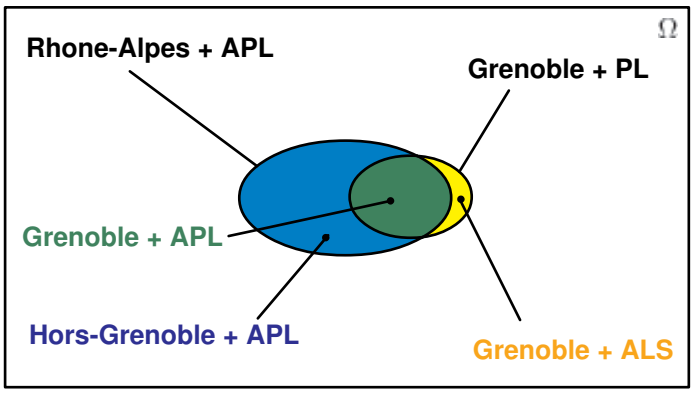


FIG. 7.6: Comparaison des couvertures lorsque $f(C) \rhd f(R)$

La comparaison des deux règles nous montre que l'antécédent de la connaissance a une couverture plus étendue que celle de l'antécédent de la règle extraite. De plus nous savons que leurs conséquents sont identiques. Ainsi la connaissance C de l'utilisateur consiste à déduire le même prédicat que la règle R à partir de conditions plus *couvrantes*. On en déduit que la connaissance est moins informative que la règle extraite (voir définition 7.12) qui dans le cas présent donne une information plus précise.

Situation où la connaissance est plus informative

Le tableau 7.2 illustre un exemple virtuel de répartition différente de la même population selon les mêmes critères que le tableau 7.1. La figure 7.7 illustre cette deuxième situation graphiquement.

La comparaison des deux règles nous montre que l'antécédent de la connaissance a une couverture plus restreinte que celle de l'antécédent de la règle extraite. De plus nous savons que leurs conséquents sont identiques. Ainsi la connaissance C de l'utilisateur consiste à déduire le même prédicat que la règle R à partir de conditions moins *couvrantes*. On en déduit que la connaissance est plus informative que la règle

	Grenoble	Hors-Grenoble
APL	95	5
ALS	200	

TAB. 7.2: Répartition des effectifs allocataires en fonction du lieu de contact et du type de prestation

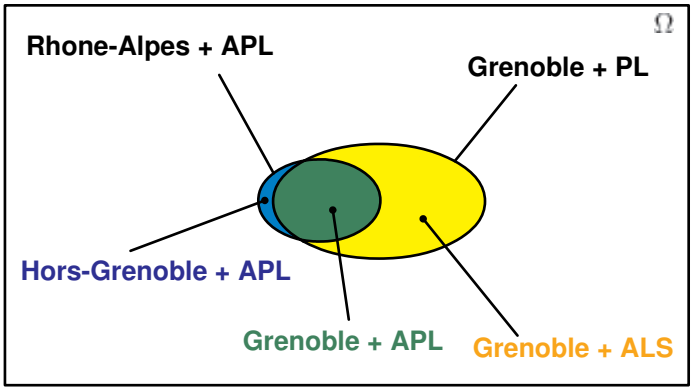


FIG. 7.7: Comparaison des couvertures lorsque $f(C) \lhd f(R)$

extraite (voir définition 7.12).

Situation où connaissance et règle extraite ont le même niveau d'information

Le tableau 7.3 illustre un exemple virtuel de répartition différente de la même population selon les mêmes critères que les tableaux 7.1 et 7.2. La figure 7.8 illustre la même situation graphiquement.

	Grenoble	Hors-Grenoble
APL	95	5
ALS	5	

TAB. 7.3: Répartition des effectifs allocataires en fonction des différents critères dans la situation où connaissance et règle extraite ont le même niveau d'information

La comparaison des deux règles nous montre que l'antécédent de la connaissance a une couverture similaire à l'antécédent de la règle extraite. De plus nous savons que leurs conséquents sont identiques, ainsi le niveau d'information des deux règles

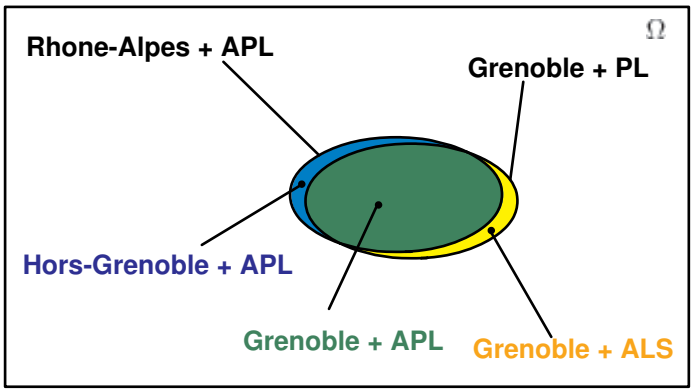


FIG. 7.8: Comparaison des couvertures dans la situation où connaissance et règle extraite ont le même niveau d'information

est similaire (voir définition 7.12) !

7.3.3 Émergence de règles inattendues

Dans les travaux sur les mesures d'intérêt subjectif la notion d'inattendu a souvent été introduite en comparant des motifs extraits à des motifs attendus exprimés par les utilisateurs. Nous reprenons, dans KEOPS, la distinction entre les notions *cause* et *conséquence* inattendues introduites par Liu pour déterminer des motifs inattendus. Une règle peut être intéressante si sa cause est inattendue par rapport à une connaissance présente dans la base de connaissances :

Définition 7.15 (Cause inattendue)
On dit qu'une règle R_1 représente une cause inattendue s'il existe une règle R_2 pour laquelle :

- *A_1 n'est pas comparable à A_2 (voir définition section 7.7).*
- *La couverture de C_1 est soit :*
  - *plus étendue que la couverture de C_2 ;*
  - *moins étendue que la couverture de C_2 ;*
  - *similaire à la couverture de C_2.*

Cependant si la cause de la règle est plus ou moins connue mais que sa conséquence est inattendue, cette règle demeure intéressante.

Définition 7.16 (Conséquence inattendue)
Une règle R_1 représente une conséquence inattendue s'il existe une règle R_2 de la base pour laquelle :

- *C_1 n'est pas comparable à C_2.*
- *La couverture de A_1est soit :*
  - *plus étendue que la couverture de A_2 ;*
  - *moins étendue que la couverture de A_2 ;*
  - *similaire à la couverture de A_2.*

Remarque 7.3 Une règle est inattendue par rapport à l'ensemble de la base de connaissances si elle est inattendue avec au moins une des connaissances et qu'elle est inattendue ou incomparable avec les autres connaissances.

7.3.4 Récapitulatif

Lors de la comparaison d'une règle à une connaissance nous avons vu que nous pouvions nous retrouver dans l'une des six situations suivantes :

- La connaissance a un niveau d'information supérieur
- La règle extraite a un niveau d'information supérieur
- La connaissance et la règle extraite ont des niveaux d'information similaires
- La règle extraite exprime une cause inattendue
- La règle extraite exprime une conséquence inattendue
- La connaissance et la règle extraite ont des niveaux d'information non comparables

Afin d'évaluer le niveau d'information d'une règle en fonction d'une connaissance, nous avons présenté des critères de comparaison des couvertures des antécédents et conséquents des règles :

1. $|f(A_1) \setminus f(A_2)| < \delta\,|f(A_1)|$
2. $|f(A_2) \setminus f(A_1)| < \delta\,|f(A_2)|$
3. $|f(C_1) \setminus f(C_2)| < \delta\,|f(C_1)|$
4. $|f(C_2) \setminus f(C_1)| < \delta\,|f(C_2)|$

Enfin, selon les critères vérifiés, on en déduit la règle qui a le niveau d'information le plus élevé conformément à l'axiome de la section 7.3.2. Le tableau 7.4 résume toutes les situations possibles.

Critères vérifiés				
1	2	3	4	Niveau d'information
Vrai	Vrai	Vrai	Vrai	Similaire
Vrai	Faux	Faux	Vrai	Supérieur
Vrai	Faux	Vrai	Vrai	
Vrai	Vrai	Faux	Vrai	
Faux	Vrai	Vrai	Faux	Inférieur
Vrai	Vrai	Vrai	Faux	
Faux	Vrai	Vrai	Vrai	
Faux	Faux	Vrai	Vrai	Cause inattendue
Faux	Faux	Vrai	Faux	
Faux	Faux	Faux	Vrai	
Vrai	Vrai	Faux	Faux	Conséquence inattendue
Vrai	Faux	Faux	Faux	
Faux	Vrai	Faux	Faux	
Vrai	Faux	Vrai	Faux	Indéterminé
Faux	Vrai	Faux	Vrai	
Faux	Faux	Faux	Faux	Non comparable

TAB. 7.4: Niveau d'information de R_1 par rapport à R_2

7.3.5 Algorithmes

Afin d'évaluer le niveau d'information des règles par rapport aux connaissances notre approche implique de calculer la couverture de nombreux itemsets ce qui peut être couteux en temps de calcul. L'algorithme 7.2 permet de créer un arbre de hachage contenant l'ensemble minimal des itemsets dont la couverture est à calculer.

```
// Fonction indiquant si les itemsets passés en paramètres sont comparables
Fonction estComparable( I1 : Itemset, I2 Itemset ) : Booléen ;

// Fonction retournant l'antécédent d'une règle
Fonction ant( R : Regle) : Itemset ;

// Fonction retournant le conséquent d'une règle
Fonction cons( R : Regle) : Itemset ;

// Union de deux itemsets en tenant compte des relations de généralisation
Opérateur ∪*( I1 : Itemset, I2 Itemset ) : Itemset ;
```

Algorithme 7.1: Fonctions utilisées dans les algorithmes 7.2, 7.3 et 7.4

L'algorithme 7.3 utilise l'opérateur $\cup^*$ qui effectue l'union de deux itemsets en tenant compte des relations de généralisation entre items. L'arbre de hachage créé par l'algorithme 7.4 contient dans ses nœuds feuilles un ensemble d'itemsets associés à un compteur. Cet arbre est passé en paramètre de l'algorithme 7.3 qui incrémente les différents compteurs et calcule le support de chacun des itemsets.

```
Initialisation :
A : ArbreHachage ;
C : Connaissances, R : Règles
T : Ensemble des tuples d'une base de données ;
A ← MinEnsemble(C, R) ;
Pour chaque t ∈ T faire
    ParcoursArbre(racine(A), t, 1, taille(t)) ;
Fin Pour
```

Algorithme 7.2: Programme principal

```
Fonction MinEnsemble( C : Connaissances, R : Règles) : ArbreHachage
    A : ArbreHachage ;
    A ← ∅ ;
    Pour chaque r ∈ R faire
        Pour chaque c ∈ C faire
            boolANT, boolCONS : booleen ;
            boolANT ← estComparable(ant(r), ant(c)) ;
            boolCONS ← estComparable(cons(r), cons(c)) ;
            Si (boolANT OU boolCONS) Alors
                antRC ← ant(r) ∪* ant(c) ;
                consRC ← cons(r) ∪* cons(c) ;
                insérer(A, {ant(r), ant(c), cons(c), cons(c), antRC,
                consRC}) ;
            Fin Si
        Fin Pour
    Fin Pour
    Retourner A ;
Fin
```

Algorithme 7.3: Calcul d'une ensemble minimal d'itemsets

```
Procédure ParcoursArbre( n : Nœud, t : Tuple, niveau, tailleMax : Entier)
    Si (estUneFeuille(n)) Alors
        Incrémenter le compteur de tous les itemsets présents dans le nœud ;
    Sinon
        Pour i de niveau à tailleMax faire
            ParcoursArbre(fctHachage(n, t[i]), t, niveau+1, tailleMax) ;
        Fin Pour
    Fin Si
Fin
```

Algorithme 7.4: Calcul de la couverture de tous les itemsets

7.4 Évaluation de l'intérêt d'une règle en fonction des connaissances

7.4.1 Méthode

Dans l'approche KEOPS nous définissons une mesure d'intérêt à mi-chemin entre les mesures d'intérêt objectif et les mesures d'intérêt subjectif [Bri06a]. Cette mesure prend en compte différents critères statistiques pour comparer la précision des règles générées par rapport à celle des connaissances et compare les différents niveaux d'information pour déterminer les règles proches des centres d'intérêts de l'utilisateur. Enfin la nature des connaissances avec lesquelles les règles sont comparées permet d'évaluer le niveau d'intérêt que représente la règle pour l'utilisateur. Ainsi trois critères sont pris en compte afin d'évaluer l'intérêt d'une règle vis à vis d'une connaissance :

- Le type de la connaissance,
- La différence entre les niveaux de confiance de la règle et de la connaissance,
- La différence entre les niveaux d'information de la règle et de la connaissance.

Il est important de souligner qu'afin d'éviter le problème engendré par la mesure de confiance (voir section 2.2.1) nous ne nous intéressons qu'aux règles dont le lift est supérieur à 1. L'intérêt résultant de cette évaluation est estimé par un indice

qualitatif (ordinal) ayant 5 niveaux d'intérêt : **aucun**, **faible**, **moyen**, **fort** et **très fort**.

La valeur « **faible** » signifie que la règle est intéressante dans la mesure où elle confirme une connaissance. Enfin les trois derniers indices caractérisent des règles intéressantes apportant des informations nouvelles par rapport aux connaissances ; plus la règle extraite apporte de nouvelles informations plus l'indice est élevé. Le niveau de référence est établi lorsque la règle extraite est comparée à une connaissance standard. Une comparaison avec une hypothèse est jugée plus intéressante (+1 niveau par rapport à celui de référence), une comparaison avec une trivialité est jugée moins intéressante (-1 niveau par rapport à celui de référence).

La modification apportée sur l'indice par le niveau d'information de la règle dépend de la comparaison des niveaux de confiance. Nous présentons le résultat des différents cas dans les tableaux 7.5, 7.6 et 7.7.

Connaissance C2 :

Heure Arrivée Contact = "Matin" $\wedge$ Modalité Contact = "Contact Telephonique" $\rightarrow$ Durée Contact = "Moins de 15 minutes"

- Indice de confiance : 60-80%
- Degré de certitude : Hypothèse

Cette connaissance traduit l'hypothèse, formulée par un agent de la CAF, selon laquelle 60 à 80% des contacts téléphoniques ayant lieu le matin durent moins de quinze minutes.

7.4.2 La règle générée et la connaissance ont des indices de confiance similaires

Dans le cas d'une comparaison avec une connaissance standard, une règle avec un niveau d'information similaire confirme la connaissance mais n'apporte pas de nouvelles informations : son intérêt est **faible**. Il en est de même pour une règle avec un niveau d'information inférieur. Dans le cas d'une règle ayant un niveau d'information supérieur l'intérêt sera **moyen**.

Règle R1 :

Heure Arrivée Contact = "Matin" ∧ Modalité Contact = "Contact Téléphonique" → Durée Contact = "Moins de 15 minutes"

- Confiance : 75%
- Niveau d'information calculé lors de la confrontation avec la connaissance C : similaire

La règle R1 et la connaissance C2 étant identiques, leur niveau d'information est similaire. Le niveau de confiance de la règle et de la connaissance étant également similaires l'intérêt de la règle sera **moyen** car elle est confrontée à une hypothèse.

Type de connaissance	Niveau d'information de la règle par rapport à la connaissance		
	Supérieur	Similaire	Inférieur
Trivialité	**faible**	**aucun**	**aucun**
Connaissance standard	**moyen**	**faible**	**faible**
Hypothèse	**fort**	**moyen**	**moyen**

TAB. 7.5: Indice d'intérêt dans le cas où règle générée et connaissance ont des indices de confiance similaires

7.4.3 La règle générée a un indice de confiance plus élevé que la connaissance

Dans le cas d'une comparaison avec une connaissance standard, une règle avec un niveau d'information similaire apporte tout de même une nouvelle information : un taux de confiance plus élevé, son intérêt est **moyen**. Une règle avec un taux d'information supérieur aura un intérêt encore supérieur car elle apporte potentiellement de nouvelles informations.

Règle R2 :

Heure Arrivée Contact = "8h30-11h29" ∧ Modalité Contact = "Contact Téléphonique" → Durée Contact = "Moins de 15 minutes"

- Confiance : 95%
- Niveau d'information calculé lors de la confrontation avec la connaissance C : supérieur

La règle R2 a un niveau d'information supérieur à la connaissance C2, son antécédent ayant une couverture plus restreinte et son conséquent ayant une couverture identique. Son niveau de confiance étant également plus élevé la règle aura un intérêt **très fort** car elle est comparée à une hypothèse.

Type de connaissance	Niveau d'information de la règle par rapport à la connaissance		
	Supérieur	Similaire	Inférieur
Trivialité	**moyen**	**faible**	**faible**
Connaissance standard	**fort**	**moyen**	**moyen**
Hypothèse	**très fort**	**fort**	**fort**

TAB. 7.6: Indice d'intérêt dans le cas où la règle générée a un indice de confiance plus élevé que la connaissance

7.4.4 La connaissance a un indice de confiance plus élevé que la règle générée

Dans le cas d'une comparaison avec une connaissance standard, une règle avec un niveau d'information similaire apporte tout de même une information sur un changement de confiance. Ce changement étant peu pertinent l'intérêt de la règle sera **faible**. Les règles avec un niveau d'information supérieur n'apportent toutefois aucune information intéressante car leur fiabilité est inférieure aux connaissances, leur intérêt sera **aucun** sauf dans le cadre de la comparaison avec une hypothèse où il est intéressant de signaler à l'utilisateur qu'une règle, même peu fiable, a des points communs avec l'hypothèse.

Règle R3 :

Heure Arrivée Contact = "Matin" $\wedge$ Modalité Contact = "Contact Téléphonique" $\rightarrow$ Durée Contact = "Moins de 5 minutes"

- Confiance : 45%
- Niveau d'information calculé lors de la confrontation avec la connaissance C : inférieur

La règle R3 a, à la fois, un niveau de confiance et un niveau d'information inférieurs à la connaissance C2. Cette dernière étant une hypothèse l'intérêt de la règle sera

tout de même **faible** car elle permet à l'utilisateur de vérifier que son hypothèse n'est pas totalement invalide.

Type de connaissance	Niveau d'information de la règle par rapport à la connaissance		
	Supérieur	Similaire	Inférieur
Trivialité	**aucun**	**aucun**	**aucun**
Connaissance standard	**faible**	**faible**	**aucun**
Hypothèse	**moyen**	**moyen**	**faible**

TAB. 7.7: Indice d'intérêt dans le cas où la connaissance a un indice de confiance plus élevé que la règle générée

Cette situation décrit un cas particulier, l'implication énoncée par la connaissance se réalise moins souvent que ne le pensent les experts.

7.5 Conclusion

Un des principaux objectifs de l'approche KEOPS était d'extraire des informations intéressantes du point de vue d'un utilisateur, sans qu'auparavant celui-ci ait à dire ce qui l'intéresse. Notre méthode se base donc sur la confrontation des connaissances actuelles de l'utilisateur aux modèles extraits afin de sélectionner ceux qui vont apporter une information nouvelle. Cette information nouvelle peut confirmer une connaissance existante, être surprenante mais aussi infirmer une idée fausse. Cela fait l'originalité de l'approche KEOPS qui est capable de prendre en compte les *mauvais résultats intéressants*.

Une autre originalité de l'approche est de proposer une mesure d'intérêt reposant sur l'ontologie présente dans l'ODIS. Cette ontologie d'application a été conçue pour modéliser les connaissances du domaine pour une tâche spécifique : la fouille de données. Ainsi chacune des relations présentes dans l'ontologie peut être utilisée dans le processus d'évaluation de l'intérêt des règles, notamment lors du calcul de la couverture des itemsets qui prend en compte les différents niveaux de généralisation spécifiés dans l'ontologie. Enfin, lors de la présentation des résultats à l'utilisateur

l'ontologie est à nouveau utilisée afin de lui permettre de visualiser selon différents points de vues un nombre réduit de modèles intéressants.

Chapitre 8

Stratégies et résultats

Sommaire

8.1 Stratégies en fouille de données

8.1.1 Gestion de l'expérience

La gestion de l'expérience est un domaine émergeant et prometteur. Dans des domaines aussi différents que le commerce ou les sciences le besoin se fait sentir d'expliciter la connaissance sous-jacente aux différents processus, produits et technologies. Généralement cette connaissance est basée sur les informations implicites acquises par un apprentissage individuel. Toutefois la connaissance individuelle est difficilement accessible à l'organisation car elle n'est pas souvent transmise et reste éphémère.

Un des objectifs de la gestion de l'expérience est donc d'améliorer l'apport pour l'organisation de l'expertise de chaque acteur. L'organisation peut ainsi profiter de l'expérience passée [BCBB04]. Decker présente dans [JAD+01] une approche pour la gestion de l'expérience (et des expériences) reposant sur six étapes. Durant la planification d'un projet, on caractérise le *contexte*, on définit les *objectifs* et on choisit le *processus* à effectuer. Durant la phase d'*exécution*, les résultats sont analysés afin d'aider à la gestion de projet. À la fin du projet, les résultats sont *analysés* rétrospectivement. Enfin les résultats sont ajoutés à la *mémoire d'expérience*, ce qui consiste en la validation de l'expérience acquise au cours du projet.

L'extraction de connaissance à partir des données est le résultat d'un processus exploratoire impliquant l'utilisation de divers algorithmes pour préparer les données, construire les modèles et les évaluer. Le processus de découverte de la connaissance est complexe et met en œuvre de multiples composants interagissant de manière non triviale. Les experts en fouille de données eux-mêmes ne sont pas familiers avec l'intégralité du domaine et peuvent négliger ainsi l'utilisation de certains processus. L'extraction des connaissances à partir des données est donc également un des domaines pouvant profiter avantageusement de la gestion de l'expérience.

Dans ce contexte, Bernstein propose l'utilisation d'assistants (IDAs : Intelligent Discovery Assistants) afin d'aider les experts en fouille de données à explorer l'espace des processus de fouille valides [BPH05] ; un processus valide étant un processus ne violant aucune des contraintes imposées par les techniques qui le constituent. Les IDAs utilisent une ontologie sur la fouille de données qui définit les différentes techniques et leurs propriétés. Grâce à l'utilisation de cette ontologie, les IDAs sont

capables de rechercher des processus valides. Selon Bernstein l'intérêt des IDAs en fouille de données est triple car ils permettent :

- l'énumération de tous les processus de fouille de données valides afin de ne pas en oublier,
- l'évaluation des différents processus afin d'aider les experts à choisir entre les différentes options,
- la création d'une infrastructure afin de partager la connaissance au sujet des processus de fouille de données.

Actuellement, dans les logiciels de fouille de données présents sur le marché, il existe la possibilité de définir des processus sous forme de flux permettant de tester différentes combinaisons de techniques et d'en comparer les résultats. Toutefois aucun système actuel ne propose l'intégration des connaissances de l'utilisateur, ne guide les experts vers le meilleur processus ni ne propose de nouveaux choix afin d'améliorer les résultats d'une fouille de données. Un des objectifs dans la définition de l'approche KEOPS a été de favoriser une gestion rationnelle des stratégies de fouille sur un même ensemble de données.

8.1.2 Dimension temporelle en fouille de données

La mise en place de stratégies pour la fouille de données nécessite la prise en compte d'un important paramètre : le temps. La dimension temporelle peut intervenir à différents niveaux dans le processus d'extraction de connaissances à partir des données. Tout d'abord, la fouille de données peut s'effectuer sur des séries temporelles dans lesquelles la notion d'événement est présente. L'analyse de la chronologie de ces événements et de la durée qui les sépare peut être effectuée au moyen de *règles d'association séquentielles*. En second lieu, la fouille de données ne s'effectue pas obligatoirement sur des données statiques. Le développement rapide des systèmes d'information a accéléré le développement de *la fouille de flux de données* et de *la fouille de données ubiquitaire* qui s'effectue sur des systèmes embarqués, dans des environnements sans fil, communicants. Enfin, l'utilisation des connaissances peut être également améliorée si l'on considère la dimension temporelle. Dans le cas de la fouille de flux de données, si l'on est capable de définir des objectifs en plus des connaissances du domaine il est alors possible de faire émerger une nouvelle notion d'intérêt prenant en compte le temps. On se rapproche ainsi de plus en plus d'une

personnalisation des résultats pour un utilisateur donné à un moment donné.

8.1.2.1 Règles d'association séquentielles

Un certain nombre de travaux se sont intéressés à l'extraction efficace de règles d'association séquentielles [SA96, TS98, CCH02]. La plupart d'entre eux se sont basés sur des variantes de l'algorithme fondateur APRIORI [AIS93b] mis au point pour extraire des règles d'association. Nous avons également abordé cette problématique en développant l'approche HASAR [BPHC04], approche hybride combinant une technique de recherche d'itemsets fermés fréquents utilisant l'algorithme CLOSE [PBTL99c] avec une heuristique basée sur un algorithme évolutionnaire afin d'extraire des règles d'association séquentielles. Nous définissons une règle d'association séquentielle comme une règle d'association possédant trois paramètres :

- Une fenêtre temporelle A qui est associée à l'antécédent de la règle exprimant des actions effectuées.
- Une fenêtre temporelle L associée à un temps de latence entre les actions et les observations.
- Une fenêtre temporelle O qui est associée au conséquent de la règle décrivant les observations effectuées après la période L.

La stratégie mise en œuvre consiste en trois étapes :

1. Préparation des données afin de pouvoir effectuer une fouille de données prenant en compte des fenêtres temporelles.
2. Recherche de règles d'association en utilisant l'algorithme CLOSE.
3. Exécution d'un algorithme évolutionnaire dont la population de départ est initialisée en utilisant les résultats de la recherche d'associations.

Un des problèmes majeurs des algorithmes évolutionnaires est leur difficulté à converger rapidement vers de bons résultats avec une population initiale aléatoire. L'approche HASAR utilise donc l'algorithme CLOSE afin de fournir à l'algorithme évolutionnaire une population initiale de bonne qualité.

L'algorithme évolutionnaire offre un moyen d'optimiser une population constituée d'individus ou chromosomes représentant chacun une règle d'association séquentielle différente. On définit une *fonction de fitness* qui permet l'évaluation d'un chromosome. Notre choix a été de définir cette fonction comme le produit du support, du lift et de la confiance. La fitness d'un gène (c'est-à-dire l'évaluation numérique

de la fonction de fitness sur un gène), permet de sélectionner les plus intéressants d'entre eux afin de procéder aux étapes suivantes de l'algorithme : croisement et mutation (voir section 2.4.3). Lorsque l'algorithme se termine les résultats permettent de déterminer de nouvelles règles dont la fitness est supérieure ou égale à celles de la population initialie.

L'approche HASAR a été mise en œuvre dans le cadre de l'analyse de facteurs de risque pour l'athérosclérose. Un des principaux objectifs de cette étude a été d'évaluer l'impact du changement de comportement sur les facteurs de risques pour le développement de maladies cardio-vasculaires. Le jeu de données utilisé a été constitué dans le cadre d'une importante étude évaluant l'impact de prescriptions non pharmacologiques (c'est-à-dire uniquement des recommandations sur le régime alimentaire, l'activité physique, la consommation d'alcool et de tabac etc.) sur l'évolution des facteurs de risques pour l'athérosclérose. Ce jeu contient des données collectées entre 1975 et 2001 sur une population de 1417 hommes nés entre 1926 et 1937 en Tchécoslovaquie.

Tout d'abord un examen médical a été effectué afin de rassembler des données concernant le régime alimentaire, la consommation d'alcool et de tabac, l'activité physique ainsi que des résultats d'analyses. Les patients ont été classés en différents groupes en prenant en compte plusieurs facteurs de risque :

- les informations sur les patients avant l'étude (niveau d'éducation, âge, consommation d'alcool, facteur de risque initial),
- les informations sur le comportement des patients durant l'étude (consommation de tabac, activité physique, travail, régime, prise de médicaments pour le cholestérol ou la tension),
- les facteurs de risques (niveau de cholesterol, de triglycérides et de glycémie, poids, tension).

Au début de l'étude les groupes suivants ont été créés :

- les patients sains (NG : Normal group),
- les patients malades (PG : Pathological group),
- les patients à risque (RG : Risk group).

Durant les vingts années de l'étude les patients ont été examinés afin d'observer le changement de leur comportement et l'évolution de leur santé. En fin d'étude les patients ont pu être affectés à une des deux classes suivantes :

- les patients ayant contractés une maladie cardio-vasculaire durant l'étude (CVD),

- les patients n'ayant pas contractés une maladie cardio-vasculaire durant l'étude (NCVD).

Lors d'une expérience l'algorithme HASAR a été mis en œuvre sur la classe PG afin d'obtenir des règles d'association séquentielles concernants les patients malades. Ces règles ont ensuite été évaluées sur les classes constituées des patients sains (NG) et à risque (RG) afin de voir si les comportements des patients malades avaient les mêmes conséquences pour les autres classes. La figure 8.1 illustre cette expérience : en abscisse nous avons une dizaine de règles dont la fitness, exprimée en ordonnée, a été évaluée sur les différents groupes PG, RG et NG.

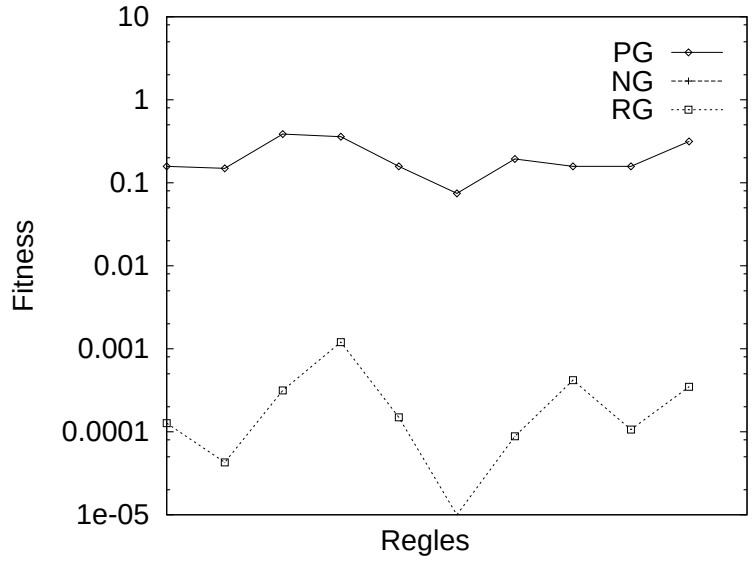


FIG. 8.1: Meilleures règles apprises sur le groupe PG et évaluées sur les groupes RG et NG

Sur la figure 8.1 seules deux courbes sont visibles car celles concernant les groupes NG et RG sont confondues. De plus nous observons que la fitness des règles évaluées sur le groupe PG est de 100 à 1000 fois supérieure à la fitness des mêmes règles évaluées sur le groupe PG. On en conclut que les règles apprises sur le groupe des patients malades (PG) ne sont pas applicables pour modéliser le comportement (et les conséquences de ce comportement) des patients sains (NG) et à risque (RG).

Lors d'une seconde expérience l'algorithme HASAR a été mis en œuvre sur la classe CVD représentant les patients ayant contractés une maladie cardio-vasculaire lors de l'étude. Les règles obtenues ont ensuite été évaluées sur l'ensemble des pa-

tients n'ayant pas contracté de maladie (NCVD). La figure 8.2 illustre cette expérience : en abscisse nous avons une dizaine de règles dont la fitness pour chacun des groupes est exprimée en ordonnée.

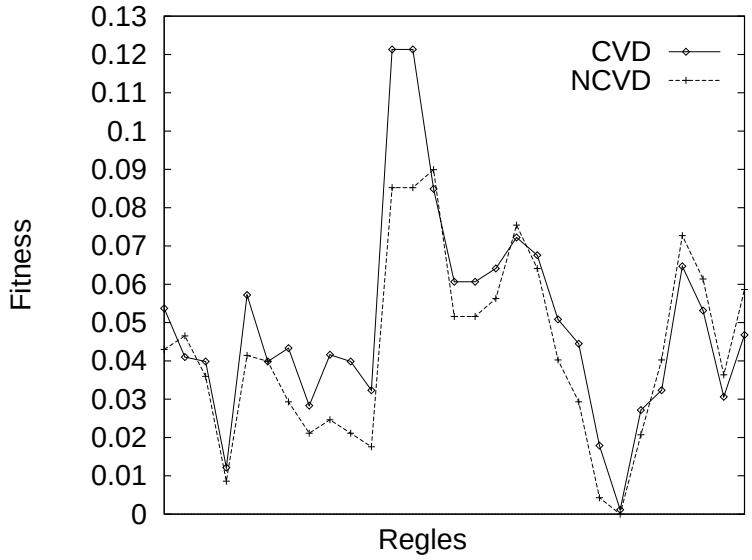


FIG. 8.2: Meilleures règles apprises sur le groupe CVD et évaluées sur le groupe NCVD

Sur la figure 8.2 nous observons que la fitness des règles évaluées sur le groupe CVD est assez similaire à la fitness des mêmes règles évaluées sur le groupe NCVD. On en conclut que les règles apprises sur le groupe CVD permettent également de modéliser le comportement des patients qui ne sont pas tombés malades.

La méthode HASAR permet donc, en combinant une technique de recherche d'itemsets fermés fréquents avec un algorithme évolutionnaire, d'extraire des régles d'association séquentielles pertinentes permettant d'étudier l'impact du comportement d'invidus sur leur santé.

8.1.2.2 La fouille de flux de données

La multiplication des flux de données, des réseaux sans fil et des interfaces mobiles justifie le besoin d'outils d'analyse de données efficaces capables de traiter des flux continus de données. On appelle fouille de données ubiquitaire (UDM : Ubiquitous Data Mining) un processus d'analyse de données provenant de sources hétérogènes et

distribuées qui utilise des interfaces mobiles ou intégrées à des réseaux de capteurs. Les applications en sont diverses :

- Contrôle des flux financiers pour la gestion d'un portefeuille d'actions
- Analyse d'informations pour la détection d'intrusions
- Analyse des données issues de capteurs pour la détection d'incidents de la route
- Contrôle du flux de contacts allocataires dans les CAF pour la détection de nouveaux comportements sur différentes échelles de temps

Différentes stratégies ont été proposées pour répondre à ces besoins [GZK04] :

1. Modifier la vitesse du flux de données entrant : cette approche a pour objectif d'*échantilloner* les données, de *filtrer* les données selon leur importance, d'*agréger* les données selon certains critères et de *délester* le flux de données, c'est-à-dire supprimer des blocs entiers de données.

2. Abstraire le niveau des connaissances : cette approche a pour but de créer des catégories de haut niveau afin de ne pas traiter chacune des données de façon individuelle.

3. Mettre au point des algorithmes approximatifs : cette approche consiste à effectuer une fouille de données en une seule passe avec une marge d'erreur acceptable.

Les perspectives dans le domaine de la fouille de flux de données se situent au niveau du développement de nouvelles techniques de pré-traitement, de la création de plate-formes de fouille de données en temps réel et de l'adaptation des algorithmes, qui vont mettre en œuvre une fouille de données ubiquitaire, au contexte réel des applications.

8.1.2.3 Glissement de concepts et définition d'objectifs

De nombreux systèmes d'apprentissage et de fouille de données font l'hypothèse que les données d'apprentissage sont un échantillon aléatoire d'une distribution stationnaire. Malheureusement, la plupart des bases et des flux de données ne respectent pas ce principe. Pour la plupart ils existent depuis plusieurs mois ou années et les processus qui ont guidé leur création ont changé, parfois de façon radicale, durant ce temps : c'est ce que l'on peut appeler un *glissement de concepts* [HSD01].

Au cours du temps nous avons donc à la fois une évolution du flux de données et des concepts du domaine ; la comparaison des connaissances aux modèles générés se doit alors de prendre en compte l'aspect temporel afin de demeurer efficace. En définissant les objectifs de l'utilisateur il est possible d'introduire de nouvelles modélisations de l'intérêt de l'utilisateur qui dépendent du temps : la déception et le soulagement, deux variantes de la surprise. Cette proposition s'inspire et repose sur le formalisme définit par Lorini pour la modélisation d'agents intelligents [LF05].

Dans un premier temps, Lorini revient sur la notion de certitude en faisant la distinction entre une prévision et une hypothèse. Une *prévision* implique que le seuil de probabilité que l'événement se produise a été dépassé alors que ce n'est pas le cas pour une *hypothèse*. On peut également définir la notion de *prédiction* pour laquelle la probabilité que l'événement se produise est proche de 1. Il introduit ensuite la notion de *souhait*, c'est-à-dire le désir qu'un événement ait lieu et la notion de *crainte*, le désir que l'événement n'ait pas lieu [1]. Dans un second temps, Lorini définit la notion de *déception* comme le fait qu'un désir fort ne se soit pas réalisé et la notion de *soulagement* comme le fait qu'une crainte forte ne se soit pas réalisée.

Ces propositions de modélisation de concepts cognitifs peuvent être facilement utilisées dans l'approche KEOPS en y ajoutant simplement la gestion des flux de données et des objectifs de l'utilisateur.

8.2 Expérimentations avec l'approche KEOPS

8.2.1 Caractéristiques du processus de fouille de données

L'approche KEOPS repose sur une plate-forme d'outils permettant de gérer le processus de la préparation des données à la visualisation des modèles en passant par la fouille elle-même. Le principal intérêt de cette plate-forme est qu'elle est spécifiquement conçue afin d'intégrer la connaissance des experts du domaine tout au long du processus d'extraction de connaissances à partir des données. De plus, en concentrant l'intervention des experts en fouille de données dans la conception de l'ODIS (cf. section 6.3), l'approche KEOPS permet également de réduire le nombre

[1] Pour ces concepts, différentes nuances sont introduites par Lorini dans son article. Le but n'étant pas de les présenter, nous vous suggérons de vous y référer afin de découvrir le formalisme proposé pour modéliser ces notions cognitives [LF05].

d'opérations de traitement complexes à effectuer lors du processus d'extraction de connaissances. De plus, afin de pouvoir optimiser le processus il est possible d'influer sur certains paramètres :

- l'expression des connaissances de l'utilisateur sur le domaine : exactitude, cohérence, richesse ;
- la création de jeux de données plus appropriés : sélection des attributs, niveau d'abstraction des valeurs, discrétisation ;
- le choix des algorithmes : le type d'algorithme[2], les seuils à fixer (de support, confiance ou lift par exemple) ;
- la réduction du nombre de modèles : seuil d'élagage, regroupement des règles (choix des relations de l'ontologie à prendre en compte).

Dans la suite de ce chapitre, nous montrons l'influence de ces différents paramètres sur les résultats de nos expérimentations.

8.2.2 Sélection de règles intéressantes

L'approche KEOPS a été mise en œuvre sur des données fournies par la Caisse d'Allocations Familiales de Grenoble concernant les contacts allocataires ayant eu lieu en 2004. Le jeu de données contenait 15 attributs et 443716 contacts et l'exécution de l'algorithme CLOSE a permis de générer 4404 règles d'association. La confrontation de ces 4404 règles avec les 70 connaissances définies par les experts a permis de sélectionner 1609 règles dont :

- 426 jugées très intéressantes,
- 97 jugées intéressantes,
- 88 jugées moyennement intéressantes,
- 317 jugées peu intéressantes,
- 603 ayant une cause inattendue,
- 78 ayant une conséquence inattendue.

Pour chaque connaissance nous avons évalué les critères de confiance, support et lift sur toutes les règles ayant été jugées *intéressantes* lors de la confrontation. Afin de visualiser les résultats sur les figures 8.3, 8.4 et 8.5 nous définissons la *confiance relative*, le *support relatif* et le *lift relatif*.

[2] Actuellement la méthode KEOPS ne supporte que les algorithmes de recherche d'associations.

Remarque 8.1 Étant données une règle R et une connaissance C, nous disons que R est une règle associée à C si R et C sont compatibles (voir définition 7.7) et si R a été jugée intéressante lors de sa confrontation avec C (voir section 7.4).

Définition 8.1 (Confiance relative)
Étant donnée une règle R et une connaissance C on appelle confiance relative la différence des confiances de R et de C :

$$ConfianceRelative(R, C) = confiance(R) - confiance(C)$$

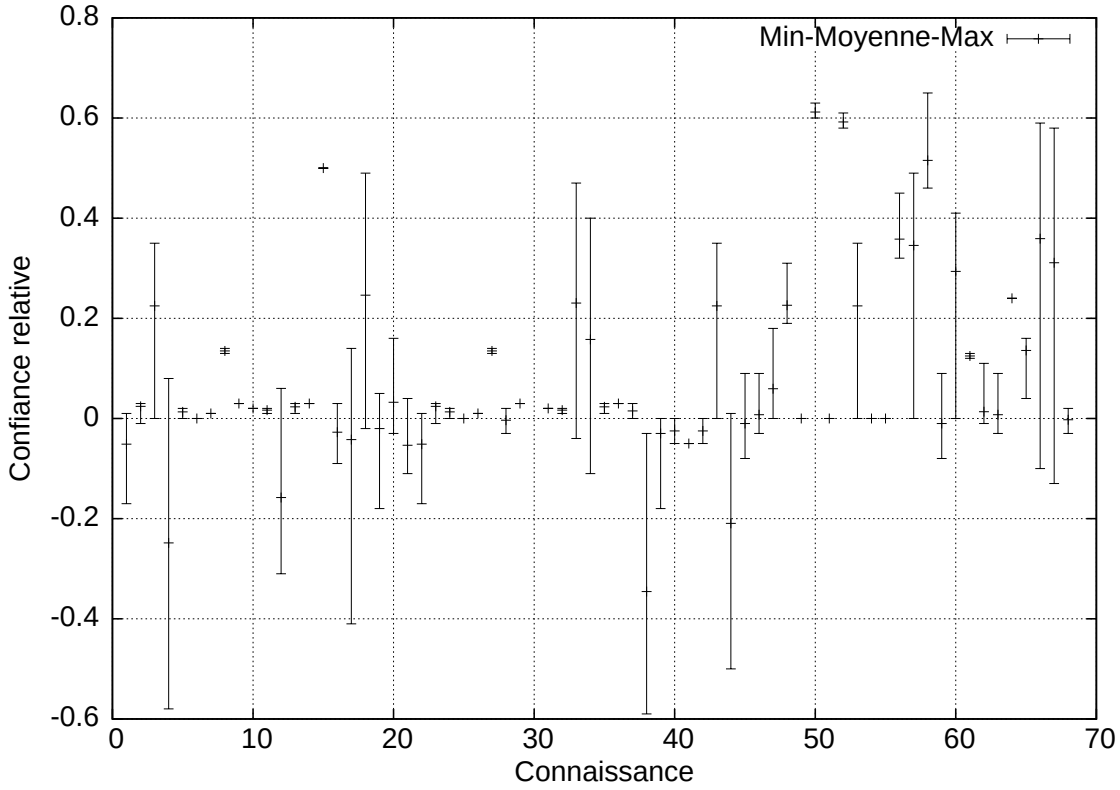


FIG. 8.3: Niveaux de confiance relative entre les connaissances et leurs règles associées

La figure 8.3 permet d'observer les valeurs de confiance relative des règles extraites jugées intéressantes lors de la confrontation aux connaissances. En abscisse nous avons le numéro des différentes connaissances exprimées par les experts. En ordonnée, nous avons pour chaque connaissance une barre verticale dont :

- le point supérieur représente la valeur maximale de la confiance relative atteinte par les règles confrontées à la connaissance,
- le point intermédiaire représente la moyenne de la confiance relative des règles confrontées à la connaissance,
- le point inférieur représente la valeur minimale de la confiance relative atteinte par les règles confrontées à la connaissance.

On observe que pour la quasi-totalité des connaissances il existe une règle jugée intéressante par notre approche dont la confiance est supérieure à celle de la connaissance (c'est-à-dire dont la confiance relative est positive). Dans la majorité des cas les règles jugées intéressantes ont en moyenne également une meilleure confiance.

Définition 8.2 (Support relatif)
Étant donnée une règle R et une connaissance C on appelle support relatif la différence des supports de R et de C :

$$SupportRelatif(R, C) = support(R) - support(C)$$

La figure 8.4 permet d'observer les valeurs de support relatif des règles extraites jugées intéressantes lors de la confrontation aux connaissances. En abscisse nous avons le numéro des différentes connaissances exprimées par les experts. En ordonnée, nous avons pour chaque connaissance une barre verticale dont :

- le point supérieur représente la valeur maximale de support relatif atteinte par les règles associées à la connaissance,
- le point intermédiaire représente la moyenne du support relatif des règles associées à la connaissance,
- le point inférieur représente la valeur minimale de support relatif atteinte par les règles associées à la connaissance.

On observe cette fois que pour la majorité des connaissances la valeur maximale du support relatif est inférieure à 0. On en déduit que dans la majorité des situations le support de la règle est inférieur au support de la connaissance à laquelle elle est confrontée. Les règles sélectionnées représentent donc des cas plus particuliers, voir des événements rares lorsque les valeurs de support sont les plus faibles.

Définition 8.3 (Lift relatif)
Étant donnée une règle R et une connaissance C on appelle lift relatif la différence des lifts de R et de C :

$$LiftRelatif(R, C) = lift(R) - lift(C)$$

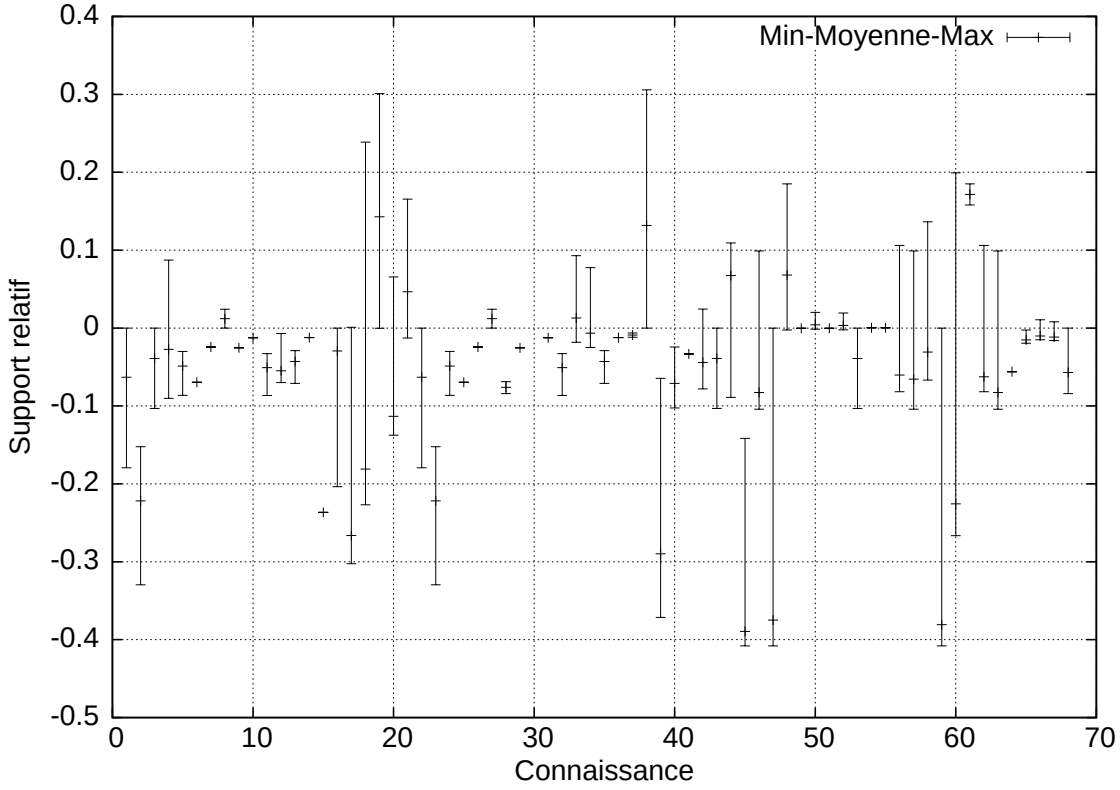


FIG. 8.4: Niveaux de support relatif entre les connaissances et leurs règles associées

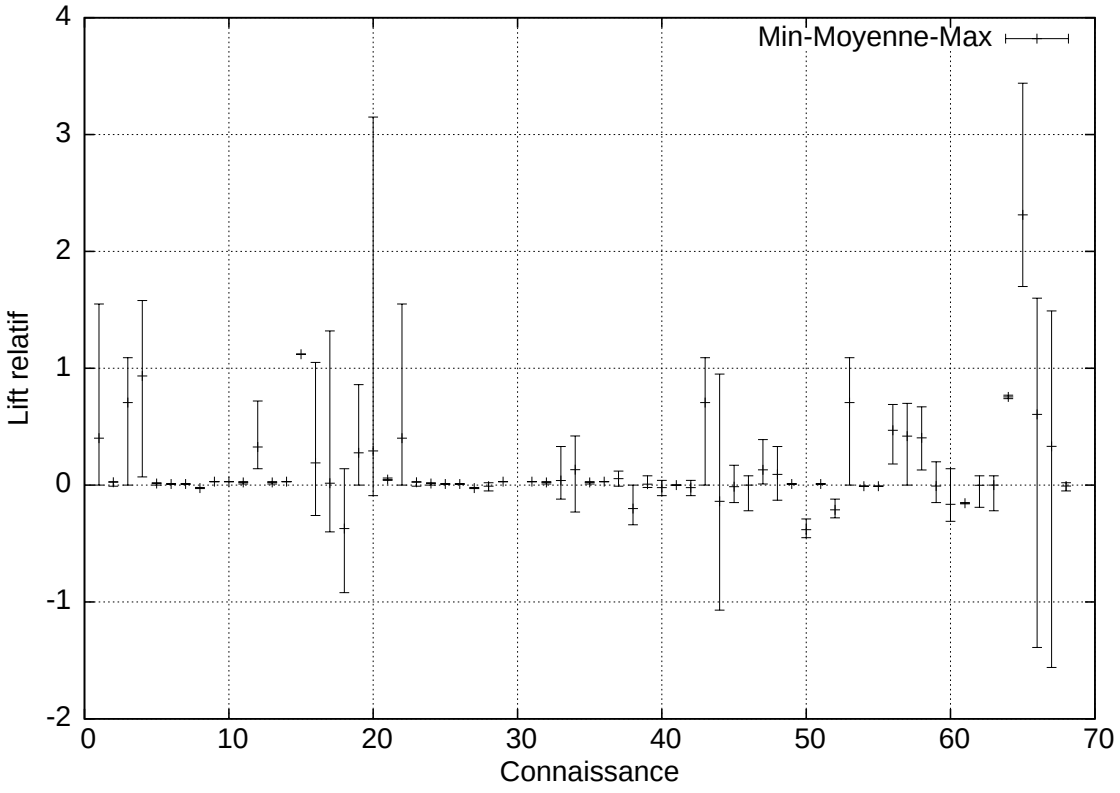


FIG. 8.5: Niveaux de lift relatif entre les connaissances et leurs règles associées

La figure 8.5 permet d'observer les valeurs de lift relatif des règles extraites jugées intéressantes lors de la confrontation aux connaissances. En abscisse nous avons le numéro des différentes connaissances exprimées par les experts. En ordonnée, nous avons pour chaque connaissance une barre verticale dont :

- le point supérieur représente la valeur maximale de lift relatif atteinte par les règles associées à la connaissance,
- le point intermédiaire représente la moyenne du lift relatif des règles associées à la connaissance,
- le point inférieur représente la valeur minimale de lift relatif atteinte par les règles associées à la connaissance.

On remarque que le lift des règles intéressantes est, dans la majorité des cas, similaire au lift de la connaissance. Nous rappelons que la méthode KEOPS supprime les règles ayant un lift inférieur ou égal à 1.

La méthode proposée par KEOPS permet la sélection de règles jugées intéressantes en fonction des connaissances. Le faible support d'un grand nombre des règles sélectionnées montre que KEOPS permet de sélectionner des événements rares ayant une bonne confiance et un lift supérieur à 1. Notre approche permet donc d'extraire des règles qui optimisent à la fois quelques critères statistiques standard (confiance, support et lift), et apportent des informations nouvelles aux connaissances exprimées par les experts.

8.2.3 Mise en œuvre de stratégies

Dans le cadre de la fouille des données de la CAF évoquée précédemment l'algorithme CLOSE permet l'extraction de plus de 4000 règles d'association. Celles-ci sont représentées sur la figure 8.6 qui montre en abscisse leur confiance et en ordonnée leur lift. Cette figure va nous permettre de comparer les résultats obtenus lors de la mise en œuvre des deux stratégies que nous allons décrire.

Après la phase de fouille des données, les règles ayant un lift inférieur à 1 sont supprimées et les règles restantes sont confrontées aux connaissances. Avant d'examiner plus en détail les résultats de la confrontation, il est intéressant d'évoquer la manière dont les connaissances ont été générées. L'expérience a pu montrer qu'il était très difficile, si l'on est peu familier avec les techniques de représentation des connaissances, d'exprimer de façon formelle des connaissances sur un domaine. En effet, comme nous l'avons évoqué dans les chapitres précédents, la connaissance métier est très souvent implicite. Afin d'amorcer la mise en œuvre de notre approche nous avons proposé la stratégie suivante :

- Dans un premier temps le processus d'ECD est mis en pause après la phase de fouille de données. Avec plusieurs milliers de règles à leur disposition les experts ont pu choisir quelques dizaines de connaissances au moyen d'un moteur de recherche de règles d'association que nous leur avions fourni. La base ainsi initialisée, la phase de confrontation des règles aux connaissances sélectionnées a pu être initiée.
- Dans un second temps, les experts, ayant plus de recul sur l'approche et possédant les résultats de la première expérience, ont exprimé quelques dizaines de nouvelles connaissances, en langage naturel cette fois-ci. Ces règles ont alors été traduites sous forme de règles de causalité qui ont été ensuite confrontées

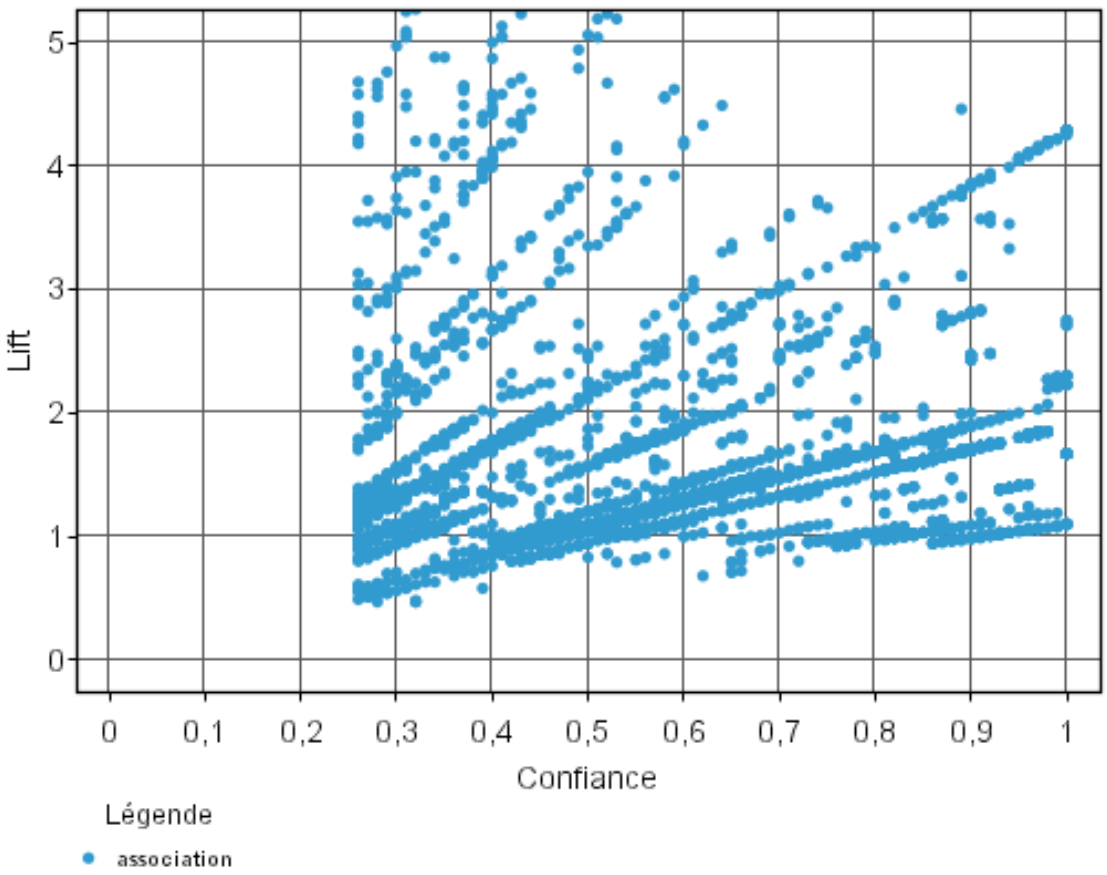


FIG. 8.6: Confiance et lift des règles extraites

aux règles extraites.

Ces deux stratégies sont présentées dans les sections suivantes.

8.2.3.1 Sélection de connaissances parmi les règles extraites

Dans une première approche, les experts ont choisi parmi les règles extraites celles qu'ils jugeaient éligibles au statut de connaissance. La figure 8.7, qui est à la même échelle que la figure 8.6, représente en rouge les connaissances choisies par les experts et en bleu les règles jugées intéressantes (c'est-à-dire dont l'intérêt est supérieur à 0) lors de la confrontation aux connaissances. Cette figure montre que KEOPS permet de ne sélectionner qu'un nombre réduit de règles. Si celles-ci, comme nous l'avons vu au chapitre précédent, apportent des informations nouvelles par rapport aux connaissances nous pouvons également constater que certaines d'entre elles optimisent les valeurs de lift et de confiance.

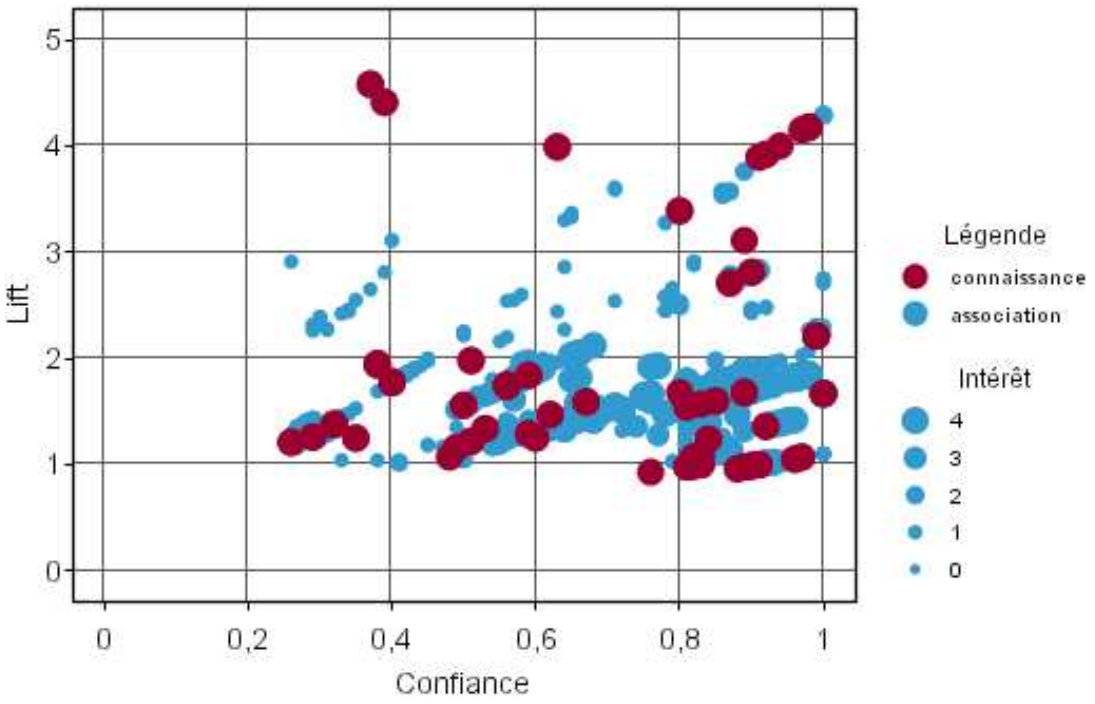


FIG. 8.7: Confiance et lift des connaissances sélectionnées par les experts parmi les règles (en rouge) et des règles qui leurs sont associées (en bleu)

Afin d'illustrer cette approche prenons l'exemple de la règle : « Dans 67% des cas un allocataire téléphone afin de demander un renseignement au siège de la CAF ». L'expert, n'étant pas complètement certain de la véracité de la règle, désire la rajouter à la base de connaissances en tant qu'hypothèse avec un indice de confiance de 60-80%. Il exprime alors la connaissance 8.1 suivante :

Connaissance 8.1
ModaliteContact="Appel Entrant" $\rightarrow$ *"MotifContact="DemandeRenseignement"* $\wedge$ *TypeStructure="Siege"*

- *Indice de confiance : 60-80%*
- *Degré de certitude : Hypothèse*

La confrontation de cette connaissance avec les règles extraites est illustrée par la figure 8.8. Le choix de règles intéressantes a été restreint par rapport à la figure 8.7. L'utilisateur dispose alors d'un ensemble de règles qui apportent des informations nouvelles par rapport à la connaissance et qui optimisent de différentes façons les critères de lift et de confiance. Afin de faire son choix, l'expert utilise le module de visualisation intégré à KEOPS qui permet d'effectuer une recherche dans les résultats selon différents critères.

Voici une liste de quelques unes des règles proposées à l'utilisateur lors de cette approche :

Règle 8.1
HeureArriveeContact="8h30-11h29" $\wedge$ *ModaliteContact="Appel Entrant"* $\wedge$ *Prestation="PAJE"* $\rightarrow$ *"MotifContact= "DemandeRenseignement"*
(support : 0,02 - confiance : 0,81)

Règle 8.2
DureeContact="0-4min" $\wedge$ *ModaliteContact="Appel Entrant"* $\wedge$ *Prestation= "PAJE"* $\rightarrow$ *"MotifContact= "DemandeRenseignement"*
(support : 0,02 - confiance : 0,81)

Règle 8.3
DureeContact="0-4min" $\wedge$ *ModaliteContact="Appel Entrant"* $\wedge$ *Prestation= "AFEAMA"* $\rightarrow$ *"MotifContact= "DemandeRenseignement"*
(support : 0,02 - confiance : 0,76)

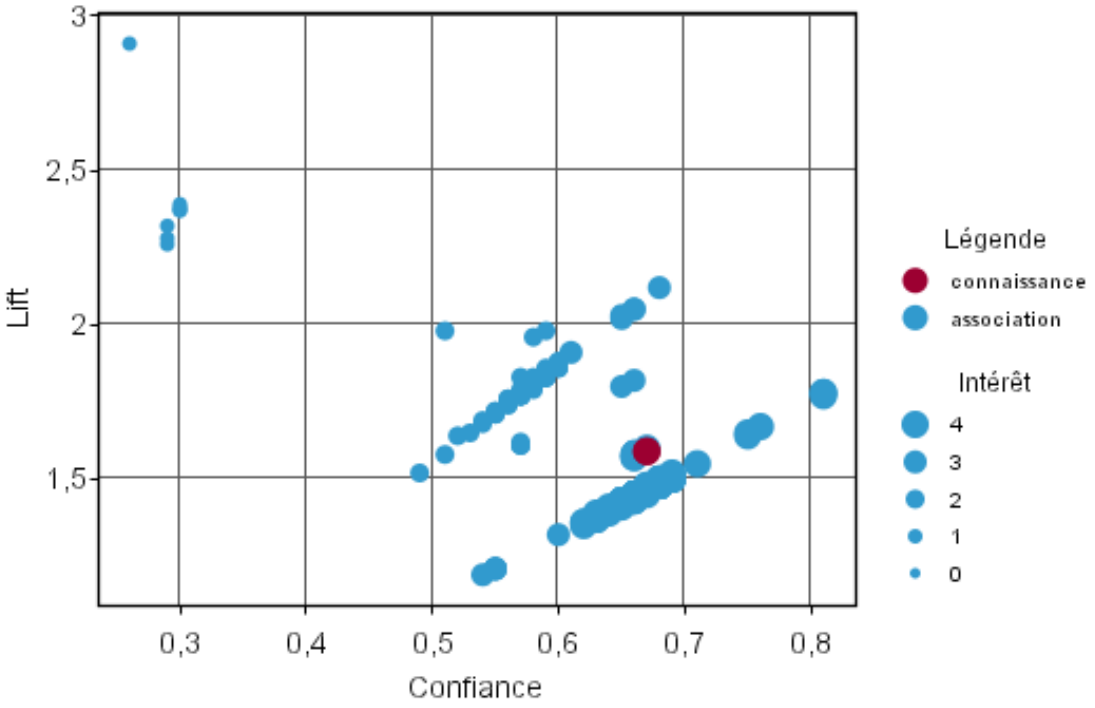


FIG. 8.8: Confiance et lift d'une connaissance (en rouge) extraite de la figure 8.7 et des règles qui lui sont associées (en bleu)

De façon générale, en observant les règles intéressantes sélectionnées par notre approche, nous pouvons constater que dans plus de 76% des cas les personnes appelant le matin pour une durée de moins de 5 minutes au sujet de la prestation PAJE ou AFEAMA désirent obtenir un renseignement. Si l'on observe les valeurs de support de ces règles on constate qu'elles sont très faibles : ce sont des événements rares.

Afin de parvenir aux résultats présentées sur les figures 8.7 et 8.8, il a été nécessaire d'effectuer différentes expériences. Ces experiences font varier plusieurs paramètres comme la nature du jeu de données ou les seuils minimaux de support et de confiance :

- Expérience n°1 (en bleu) : jeu n°1, minsupport : 0,1 et minconfiance : 0,25
- Expérience n°2 (en rouge) : jeu n°2, minsupport : 0,1 et minconfiance : 0,25
- Expérience n°3 (en rose) : jeu n°3, minsupport : 0,1 et minconfiance : 0,25
- Expérience n°4 (en vert) : jeu n°3, minsupport : 0,01 et minconfiance : 0,25

Les jeux de données présentent, quand à eux, des différences au niveau des attributs qui les composent et du degré de généralisation des valeurs :

- Jeu n°1 : jeu de données de référence contenant toutes les valeurs au plus faible niveau de généralisation
- Jeu n°2 : jeu de données basé sur le jeu n°1 mais contenant des valeurs au plus haut niveau de généralisation
- Jeu n°3 : jeu de données basé sur le jeu n°1 mais avec une sélection de certains attributs afin de supprimer ceux jugés non pertinents

La figure 8.9 illustre les différentes expériences qui ont été menées afin de sélectionner des règles intéressantes par rapport aux connaissances exprimées par les experts. La couleur des points permet ici de distinguer les différentes expériences.

La stratégie employée dans cet exemple a été, dans un premier temps, d'effectuer l'expérience n°1 avec un jeu de données contenant les valeurs les moins générales possibles. L'expérience n°2 est différente de la première sur un point : les valeurs étaient, cette fois-ci, les plus générales possibles. Les règles obtenues ont été peu nombreuses et certaines valeurs, plus fréquentes que les autres, étaient omniprésentes. Nous avons donc mené l'expérience n°3 en utilisant le jeu de données n°1 auquel nous avons supprimé des attributs dont les valeurs, trop fréquentes, pouvaient masquer certains résultats intéressants. Au vu du faible nombre de résultats, nous avons alors

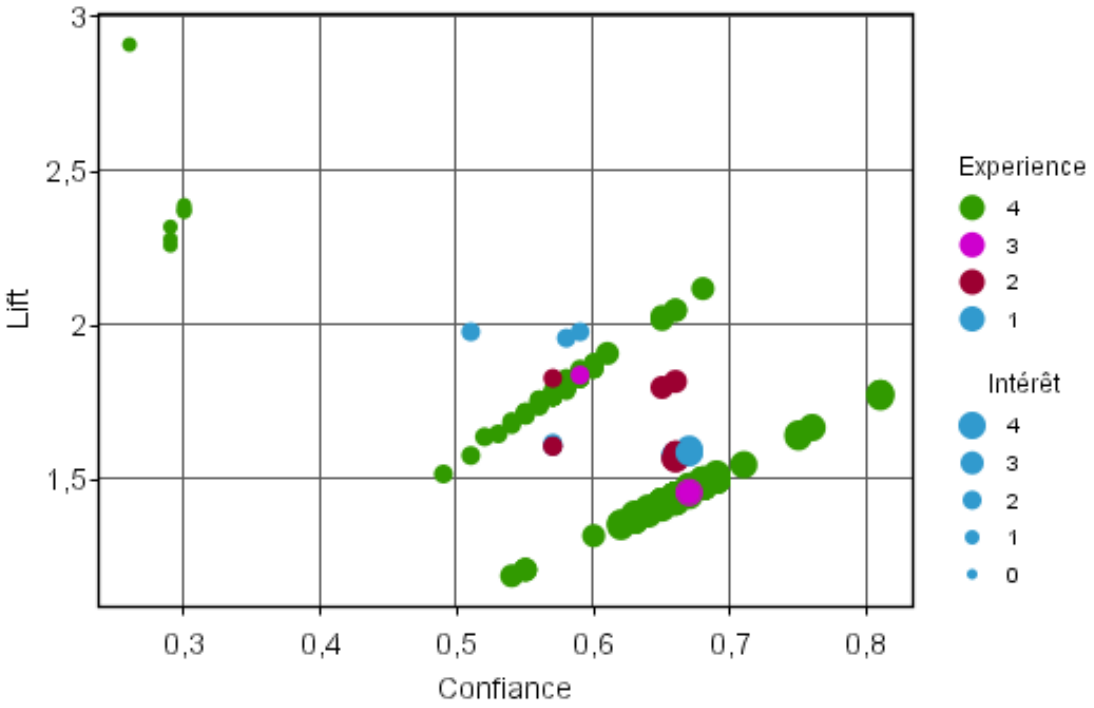


FIG. 8.9: Confiance et lift des règles extraites lors de différentes expériences

diminué le seuil minimal de support dans l'expérience n°4 ce qui nous a permis d'obtenir, comme nous pouvons l'observer sur la figure 8.9, des règles *fiables* du point de vue du lift et de la confiance et *intéressantes* par rapport aux connaissances des experts. Comme le seuil minimal de support a été beaucoup diminué, les règles de l'expérience n°4 représentent des événements intéressants, fiables et peu fréquents : c'est-à-dire des événéments rares.

8.2.3.2 Expression des connaissances en langage naturel

La deuxième stratégie mise en œuvre a consisté à laisser les experts exprimer des connaissances en langage naturel.

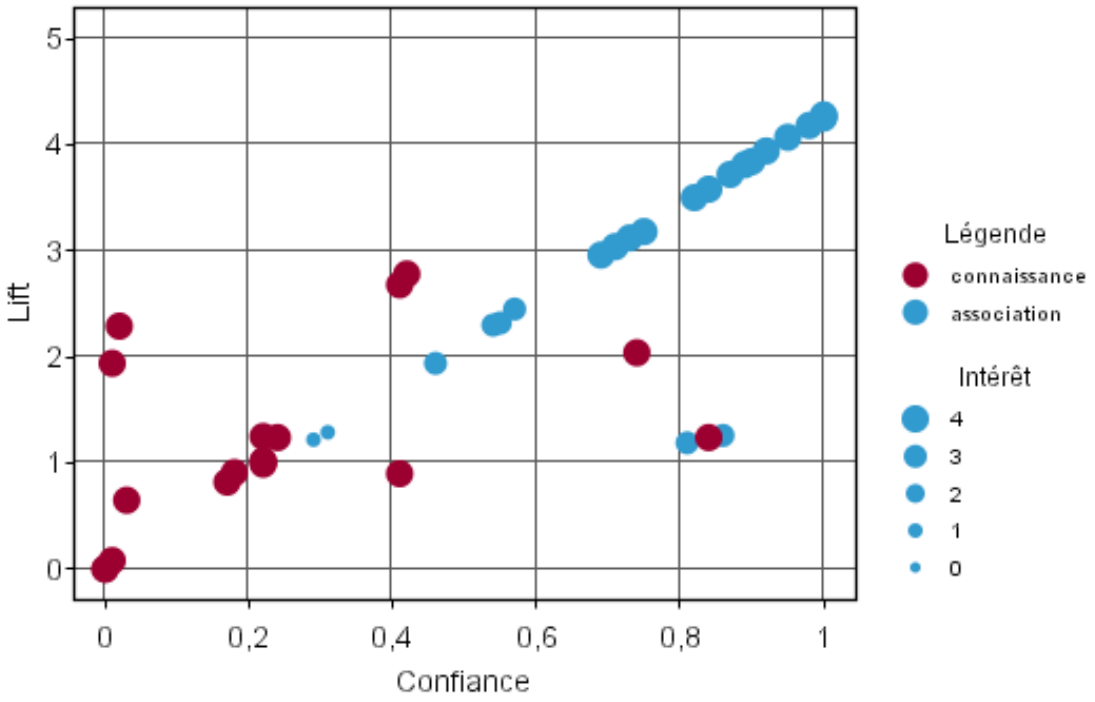


FIG. 8.10: Confiance et lift des connaissances exprimées par les experts en langage naturel (en rouge) et des règles qui leurs sont associées (en bleu)

Le premier écueil a été que, bien souvent, les connaissances exprimaient de nouveaux concepts absents de l'ontologie. En ne considérant que les connaissances au sujet des contacts allocataires nous avons pu toutefois en conserver une trentaine. La

figure 8.10 permet de visualiser ces connaissances ainsi que les règles extraites présentant un intérêt selon l'évaluation effectuée par KEOPS. La figure 8.11 illustre les mêmes informations que la figure précédente mais en ne considérant que la connaissance suivante :

***Allocataires Multi-Visitants - Lieu d'accueil** : Grenoble (Siège) - Par rapport aux autres visiteurs les multi-visitants sont plus souvent des bénéficiaires du RMI.*

Comme toutes les autres connaissances exprimées en langage naturel, cette connaissance a été formalisée sous forme de règle de causalité à laquelle on a associé un indice de confiance et un degré de certitude. Tous les concepts exprimés en langage naturel n'étant pas disponibles dans l'ontologie, certaines informations ont été extraites de ces connaissances. Nous obtenons ainsi la connaissance 8.2, qui exprime que 40 à 60% des contacts à l'accueil du siège sont initiés par des allocataires percevant le RMI. Cette connaissance, inspirée de celle en langage naturel, est toutefois sensiblement différente. Ainsi on lui associe le statut d'*hypothèse* à vérifier.

Connaissance 8.2
Prestation="RMI" → ModaliteContact="Accueil" ∧ TypeStructure="Siege"

- *Indice de confiance : 40-60%*
- *Degré de certitude : Hypothèse*

Voici quelques-unes des règles intéressantes sélectionnées par KEOPS qui donnent plus d'informations sur les contacts avec les bénéficiaires du RMI :

Règle 8.4
Prestation="RMI" ∧ TempsAttente="5-14min" → ModaliteContact="Accueil"
(support : 0,01 et confiance : 1)

Règle 8.5
Prestation="RMI" ∧ DureeContact="0-4min" → ModaliteContact="Accueil"
(support : 0,02 et confiance : 0,89)

Règle 8.6
Prestation="RMI" ∧ Distance="Proche" → ModaliteContact="Accueil"
(support : 0,01 et confiance : 0,84)

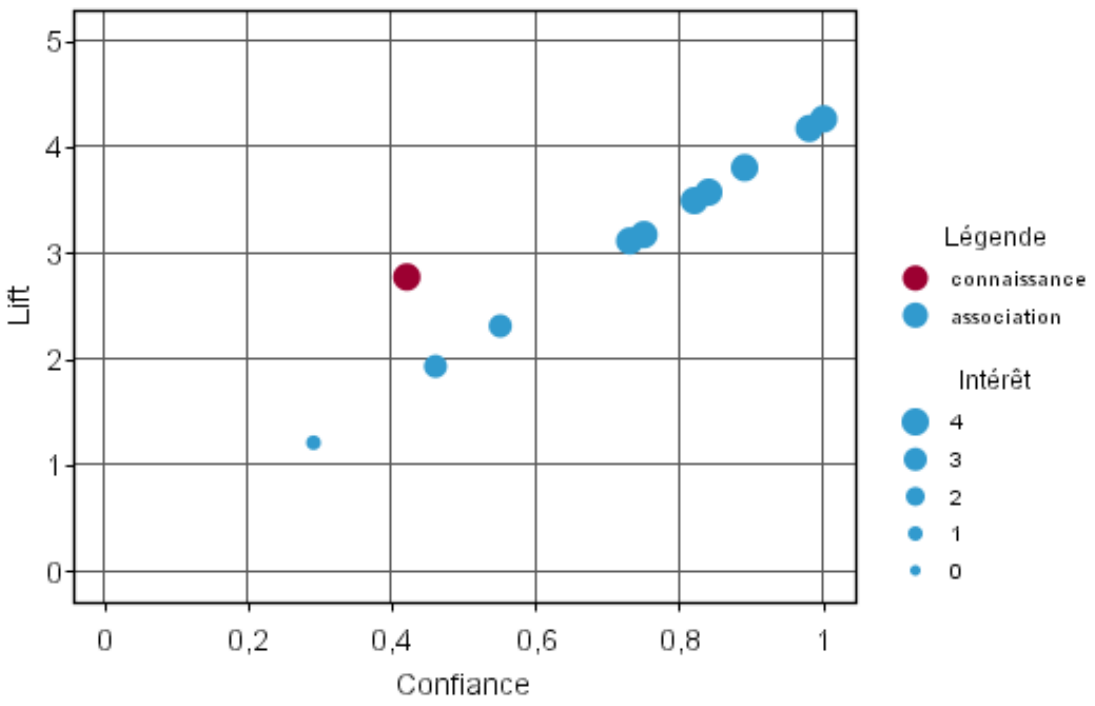


FIG. 8.11: Confiance et lift d'une connaissance (en rouge) extraite de la figure 8.10 et des règles qui lui sont associées (en bleu)

Dans le cas de cet exemple, aucune règle permettant de confirmer ou infirmer la connaissance n'a été trouvé. Ainsi KEOPS essaye de présenter des règles proches du thème de la connaissance : c'est-à-dire des informations sur des contacts à l'accueil effectués par des bénéficiaires du RMI. Nous apprenons ainsi que les bénéficiaires du RMI se déplacent à l'accueil de la CAF lorsqu'ils sont proches (règle 8.6) et que le temps d'attente est faible (règle 8.4). La règle 8.5 confirme le fait que les bénéficiaires du RMI viennent à l'accueil uniquement pour un contact bref : c'est-à-dire pour avoir une information sur la date de versement de leur RMI.

Dans le cadre de cette deuxième stratégie, les mêmes expériences ont été menées mais seule l'expérience n°4 a pu fournir des règles intéressantes. C'est pourquoi nous ne présentons pas de graphiques illustrant les résultats des différentes expériences.

Laurent BRISSON

8.3 Conclusion

L'approche KEOPS, en automatisant la plupart des tâches effectuées parfois manuellement par les experts en fouille de données, permet de mettre en œuvre facilement et rapidement différentes stratégies. Nous avons montré l'intérêt de déployer cette approche dans le cadre de stratégies qui vont permettre d'optimiser les résultats présentés à l'utilisateur.

Un aspect important à commenter est l'impact de l'approche KEOPS sur la charge de travail de l'expert en fouille de données et de l'expert du domaine. Bien que cela soit très difficile à quantifier nous avons pu constater l'intérêt de KEOPS car nous avons comparé le travail fourni avant la mise en place de KEOPS dans le cadre de l'étude des maladies cardio-vasculaires à celui fourni après la mise en place de KEOPS dans le cadre de l'étude de la relation allocataire.

Lors de la fouille sur les données cardio-vasculaires nous avons procédé par essais et erreurs, et chaque nouveau choix nécessitait un retour en phase de pré-traitement des données afin de générer de nouveaux jeux de données. L'absence de formalisation des connaissances du domaine a également engendré beaucoup de malentendus entre experts et fait perdre beaucoup de temps car la complexité du domaine médical n'était absolument pas à la portée d'un expert en fouille de données.

Au contraire, lors de la mise en place de KEOPS dans le contexte de l'analyse des données sur le contact allocataire, un important et coûteux effort a été fait sur la mise au point de l'ODIS. Par conséquent toutes les étapes suivantes ont été simplifiées, les concepts correctement définis ont permis aux différents experts de mieux communiquer et les différentes expériences variant les paramètres présentés dans ce chapitre ont pu être effectuées rapidement, sans risque d'erreur et sans devoir remettre en cause certains choix essentiels.

Pour finir, un point fort de l'approche KEOPS est qu'elle est particulièrement adaptée à un déploiement en entreprise. En effet, bien que la phase de pré-traitement soit légèrement plus complexe, dans la gestion d'un projet il sera préférable d'affecter des ressources durant une plus longue période une seule fois afin de les affecter ensuite à un autre projet, plutôt que de monopoliser cette même ressource plusieurs fois pour effectuer une nouvelle variante de la même tâche. L'approche KEOPS en permettant la définition de *jalons* ou *milestones* facilite la gestion des ressources, simplifie l'analyse des risques et donne un meilleur contrôle sur le projet.

Chapitre 9

Conclusion et Perspectives

9.1 Bilan

Une nouvelle approche a été définie pour intégrer des connaissances dans le processus de fouilles de données afin d'extraire des informations intéressantes pour l'utilisateur. Comme nous l'avons présenté dans les chapitres 2 et 3 il existe différentes études concernant l'évaluation de l'intérêt des connaissances extraites et proposant des mesures d'intérêt diverses. De multiples critères de qualité ont été énoncés pour faire un choix parmi les mesures ; cependant aucun consensus n'a pu être trouvé sur la mesure la plus appropriée dans un contexte donné.

Les approches subjectives complètent les mesures d'intérêt objectif qui sont basées sur des critères statistiques. Si la plupart des approches se positionnent dans une phase de post-traitement en filtrant les règles intéressantes à partir d'un ensemble de règles générées (et donc déjà filtrées par les mesures objectives), certaines approches permettent d'intégrer les deux types de sélection au sein d'un système dans le but de ne générer que des règles intéressantes en une seule phase. Les limitations de ces approches sont principalement dûes à certains choix conceptuels et à la manière de représenter et d'utiliser la connaissance dans les mécanismes de comparaison.

Nous avons fait le constat qu'il était nécessaire d'utiliser le formalisme des techniques de représentation des connaissances pour concevoir des mesures d'intérêt subjectives. Aussi, pour palier ces inconvénients, nous proposons d'utiliser cette formalisation tout au long du processus d'extraction pour évaluer l'intérêt subjectif.

L'originalité de l'approche KEOPS réside en partie dans la modélisation des connaissances lors de la phase de pré-traitement des données : non seulement les données peuvent alors être préparées en relation avec les connaissances du domaine mais les connaissances de l'expert du domaine peuvent être utilisées dans l'algo-

rithme de fouille de données et lors de la phase de post-traitement. La confrontation des modèles extraits aux connaissances est d'autant plus facilitée que leur expression est très similaire car formée à partir des mêmes concepts.

Un des objectifs de l'approche KEOPS est d'extraire des informations intéressantes du point de vue d'un utilisateur, sans qu'auparavant celui-ci n'ait à exprimer ce qui l'intéresse. Notre méthode se base donc sur la confrontation des connaissances actuelles de l'utilisateur aux modèles extraits afin de sélectionner ceux qui vont apporter une information nouvelle. Cette information nouvelle peut confirmer une connaissance existante, être surprenante mais aussi infirmer une idée préconçue. Cela fait l'originalité de l'approche KEOPS qui est capable de prendre en compte les mauvais résultats, intéressants dans la mesure où ils remettent en cause certaines connaissances.

Une autre originalité de l'approche est de proposer une mesure d'intérêt reposant sur l'ontologie présente dans l'ODIS. Cette ontologie d'application doit être conçue pour modéliser les connaissances du domaine pour une tâche spécifique : la fouille de données. Ainsi chacune des relations présentes dans l'ontologie peut être utilisée dans le processus d'évaluation de l'intérêt des règles, notamment lors du calcul de la couverture des itemsets qui prend en compte les différents niveaux de généralisation spécifiés dans l'ontologie. Enfin, lors de la présentation des résultats à l'utilisateur l'ontologie est à nouveau utilisée afin de lui permettre de visualiser, selon différents points de vue, un nombre réduit de modèles intéressants.

Nous avons montré l'intérêt de déployer l'approche KEOPS dans le cadre de stratégies qui vont permettre d'optimiser les résultats présentés à l'utilisateur. De plus, l'automatisation de la plupart des tâches effectuées parfois manuellement par les experts en fouille de données, permet de tester facilement et rapidement différentes stratégies. Un point fort de l'approche est qu'elle focalise l'intervention des experts lors de la phase de pré-traitement des données et de conception de l'ODIS. Ainsi toutes les stratégies sont développées sur une base commune et consistante ce qui permet de faciliter la gestion des ressources à déployer pour la tâche de fouille de données, de simplifier l'analyse des risques et de mieux contrôler le processus d'extraction de connaissances.

Laurent BRISSON

9.2 Perspectives

Les travaux présentés dans ce mémoire ouvrent plusieurs voies de recherches futures.

D'une part, les connaissances modélisées dans le cadre de l'approche KEOPS sont stockées dans une base de connaissances sous forme de règles de causalité. L'approche actuelle n'utilisant que des algorithmes d'extraction de règles d'association une première évolution de l'approche serait de l'étendre à d'autres tâches de fouille de données comme le clustering ou la classification. Nous avons évoqué le fait que chaque utilisateur puisse avoir des connaissances propres mais aussi partager des connaissances consensuelles avec d'autres utilisateurs. Dans l'état actuel du développement de l'approche KEOPS nous ne proposons pas de système de gestion des connaissances, cependant si un tel système n'est pas une nécessité pour son fonctionnement il peut être très intéressant afin de gérer les connaissances contradictoires et effectuer des vérifications préalables avant leur utilisation dans KEOPS. Nous avons donc le projet d'interfacer KEOPS avec un système de gestion des connaissances.

D'autre part, les perspectives dans le domaine de la fouille de flux de données ouvrent la voie au développement de nouvelles techniques de pré-traitement, à la création de plate-formes de fouille de données en temps réel et à l'adaptation des algorithmes au contexte réel des applications qui vont mettre en œuvre une fouille de données ubiquitaire. L'approche KEOPS pourrait bénéficier des recherches dans ce domaine notamment en adaptant la phase de pré-traitement pour qu'elle puisse s'occuper dynamiquement d'un flux de données et en modélisant les objectifs des utilisateurs afin de prendre en considération la dimension temporelle dans toutes les étapes du processus.

Enfin une perspective désormais rendue possible par KEOPS est d'acquérir une expertise à partir des échecs et des succès afin d'optimiser le processus de fouille de données et d'adapter les stratégies à un domaine particulier. L'approche KEOPS permettant l'automatisation des processus autorisera l'étude des paramètres qui ont conduit à des résultats intéressants ou moins intéressants. Une stratégie de fouille pourrait être ainsi dynamiquement modifiée en agissant sur les différents paramètres

afin de fournir à l'utilisateur des résultats toujours plus adaptés à ses besoins.

Bibliographie

[AG90] Serge Abiteboul and Stéphane Grumbach. Col : a logic-based language for complex objects. pages 347–374, 1990.

[AIS93a] Rakesh Agrawal, Tomasz Imielinski, and Arun N. Swami. Database mining : A performance perspective. *IEEE Trans. Knowl. Data Eng.*, 5(6) :914–925, 1993.

[AIS93b] Rakesh Agrawal, Tomasz Imielinski, and Arun N. Swami. Mining association rules between sets of items in large databases. In Peter Buneman and Sushil Jajodia, editors, *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, Washington, D.C., May 26-28, 1993*, pages 207–216. ACM Press, 1993.

[AP98] C. Atkins and J. Patrick. Naler : A natural language method for interpreting e-r models. In *SEEP '98 : Proceedings of the 1998 International Conference on Software Engineering : Education & Practice*, page 2, Washington, DC, USA, 1998. IEEE Computer Society.

[AS94] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules. In Jorge B. Bocca, Matthias Jarke, and Carlo Zaniolo, editors, *Proc. 20th Int. Conf. Very Large Data Bases, VLDB*, pages 487–499. Morgan Kaufmann, 12–15 1994.

[AT97] Gediminas Adomavicius and Alexander Tuzhilin. Discovery of actionable patterns in databases : The action hierarchy approach. In *KDD*, pages 111–114, 1997.

[Azé03] Jérôme Azé. *Extraction de connaissances à partir de données numériques et textuelles*. PhD thesis, Université Paris-Sud, 2003.

[Baa99] F. Baader. Logic-based knowledge representation. In M. J. Wooldridge and M. Veloso, editors, *Artificial Intelligence Today, Recent Trends and Developments*, number 1600, pages 13–41. Springer Verlag, 1999.

[BCBB04] Laurent Brisson, Martine Collard, Kevin Le Brigant, and Pascal Barbry. Kta : A framework for integrating expert knowledge and expe-

riment memory in transcriptome analysis. In *OntoKDD workshop in ECML/PKDD Conference*, Pisa, September 2004.

[BCP05] Laurent Brisson, Martine Collard, and Nicolas Pasquier. Improving the knowledge discovery process using ontologies. In *Mining Complex Data workshop in ICDM Conference*, Houston, 2005.

[BCP06] Laurent Brisson, Martine Collard, and Nicolas Pasquier. Ontologie et base de connaissances pour le pré-traitement et le post-traitement en fouille de données. In *Fouille de Données Complexes workshop in EGC Conference*, Lille, 2006.

[Bis94] Gilles Bisson. Une approche symbolique/numérique de la notion de similarité. In Yves Kodratoff Edwin Diday, editor, *4èmes journées sur l'induction symbolique/numérique*, pages 93–96, Orsay (FR), 1994.

[BLLV06] Jean-Pierre Barthélemy, Angélique Legrain, Philippe Lenca, and Benoît Vaillant. Agrégation de mesures d'intérêt de règles d'association. In *Atelier Qualité des Données et des Connaissances (associé à la conférence Extraction et Gestion des Connaissances)*, 2006.

[BM02] J.P. Brans and B. Mareschal. *PROMETHEE-GAIA. Une Méthodologie d'Aide à la Décision en Présence de Critères Multiples.* Ellipses, Paris, France, 2002.

[BPH05] Abraham Bernstein, Foster Provost, and Shawndra Hill. Toward intelligent assistance for a data mining process : An ontology-based approach for cost-sensitive classification. *IEEE Transactions on Knowledge and Data Engineering*, 17(4) :503–518, 2005.

[BPHC04] Laurent Brisson, Nicolas Pasquier, Céline Hebert, and Martine Collard. Hasar : Mining sequential association rules for atherosclerosis risk factor analysis. In *Proceedings of the PKDD'2004 conference - Discovery Challenge*, Pisa, September 2004.

[Bra83] Ronald J. Brachman. What is-a is and isn't : An analysis of taxonomic links in semantic networks. *IEEE Computer*, 16(10) :30–36, 1983.

[Bra02] Agnès Braud. *Fouille de données par algorithmes génétiques.* PhD thesis, Université d'Orléans, 2002.

[Bri04] Laurent Brisson. Mesures d'intérêt subjectif et représentation des connaissances. Technical Report I3S/RR-2004-35-FR, Laboratoire I3S - Université de Nice Sophia-Antipolis, October 2004.

[Bri06a] Laurent Brisson. Interesting patterns extraction using prior knowledge. In *Discovery Science*, pages 296–300, 2006.

[Bri06b] Laurent Brisson. Knowledge extraction using a conceptual information system (excis). In *ODBIS Workshop in VLDB Conference*, Seoul, 2006.

[CCH02] Yen-Liang Chen, Shih-Sheng Chen, and Ping-Yu Hsu. Mining hybrid sequential patterns and sequential rules. *Inf. Syst.*, 27(5) :345–362, 2002.

[CF00] Deborah R. Carvalho and Alex Alves Freitas. A genetic algorithm-based solution for the problem of small disjuncts. In *Principles of Data Mining and Knowledge Discovery*, pages 345–352, 2000.

[CF02] Deborah R. Carvalho and Alex Alves Freitas. A genetic algorithm with sequential niching for discovering small-disjunct rules. In *GECCO*, pages 1035–1042, 2002.

[Cod70] E. F. Codd. A relational model of data for large shared data banks. *Commun. ACM*, 13(6) :377–387, 1970.

[Cou05] Olivier Couturier. *Contribution à la fouille de données : règles d'association et interactivité au sein d'un processus déxtraction de connaissances dans les données.* PhD thesis, CRIL,Université d'Artois, 2005.

[CP04] P. Clark and B. Porter. *The KM Machine : Users manual*, 2004.

[CRS+04] Hanna Cespivova, Jan Rauch, Vojtech Svatek, Martin Kejkula, and Marie Tomeckova. Roles of Medical Ontology in Association Mining CRISP-DM Cycle. In *Workshop on Knowledge Discovery and Ontologies at ECML/PKDD*, 2004.

[CT91] Thomas M. Cover and Joy A. Thomas. *Elements of information theory.* Wiley-Interscience, New York, NY, USA, 1991.

[DDV06] Gašević Dragan, Djuric Dragan, and Devedžić Vladan. *Model Driven Architecture and Ontology Development.* Springer-Verlag, 2006.

[Fab96] Patrice Fabiani. *Représentation dynamique de l'incertain et stratégie de prise d'information pour un système autonome en environnement*

évolutif. PhD thesis, L'école nationale supérieure de l'aéronautique et de l'espace, 1996.

[FBC03a] Dominique Francisci, Laurent Brisson, and Martine Collard. Extraction de règles selon des critères multiples : l'art du compromis. Technical Report I3S/RR-2003-11-FR, Laboratoire I3S - Université de Nice Sophia-Antipolis, May 2003.

[FBC03b] Dominique Francisci, Laurent Brisson, and Martine Collard. A scalar evolutionnary approach to rule extraction. Technical Report I3S/RR-2003-12-FR, Laboratoire I3S - Université de Nice Sophia-Antipolis, May 2003.

[Fik98] R. Fikes. Multi-use ontologie, 1998.

[Fra04] Dominique Francisci. *Techniques d'optimisation pour la fouille de données*. PhD thesis, Université de Nice-Sophia Antipolis, 2004.

[Fre98] Alex Alves Freitas. On objective measures of rule surprisingness. In *Principles of Data Mining and Knowledge Discovery*, pages 1–9, 1998.

[Fre99] A. Freitas. On rule interestingness measures. *Knowledge-Based Systems journal*, 1999.

[Fuc97] Béatrice Fuchs. *Représentation de connaissances pour le raisonnement à partir de cas*. PhD thesis, Université Jean Monnet de Saint-Etienne, 1997.

[Gru93] Thomas R. Gruber. A translation approach to portable ontology specifications. *Knowl. Acquis.*, 5(2) :199–220, 1993.

[Gru95] Thomas R. Gruber. Toward principles for the design of ontologies used for knowledge sharing. *Int. J. Hum.-Comput. Stud.*, 43(5-6) :907–928, 1995.

[Gua97] Nicola Guarino. Some organizing principles for a unified top-level ontology. In *In Proceedings of AAAI Spring Symposium on Ontological Engineering*, Stanford, 1997. AAAI Press.

[Gua98] Nicola Guarino. *Formal Ontology in Information Systems : Proceedings of the 1st International Conference June 6-8, 1998, Trento, Italy*. IOS Press, Amsterdam, The Netherlands, The Netherlands, 1998.

[Gua99] Nicola Guarino. Avoiding is-a overloading : The role of identity conditions in ontology design. In *Intelligent Information Integration*, 1999.

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

[GZK04] Mohamed Medhat Gaber, Arkady B. Zaslavsky, and Shonali Krishnaswamy. Towards an adaptive approach for mining data streams in resource constrained environments. In *DaWaK*, pages 189–198, 2004.

[HF95] Jiawei Han and Yongjian Fu. Discovery of multiple-level association rules from large databases. In *VLDB '95 : Proceedings of the 21th International Conference on Very Large Data Bases*, pages 420–431, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc.

[HSD01] Geoff Hulten, Laurie Spencer, and Pedro Domingos. Mining time-changing data streams. In *KDD '01 : Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 97–106, New York, NY, USA, 2001. ACM Press.

[HSM01] David J. Hand, Padhraic Smyth, and Heikki Mannila. *Principles of data mining.* MIT Press, Cambridge, MA, USA, 2001.

[JACP03] David Jouve, Youssef Amghar, Bertrand Chabbat, and Jean-Marie Pinon. Conceptual framework for document semantic modelling : an application to document and knowledge management in the legal domain. *Data Knowl. Eng.*, 46(3) :345–375, 2003.

[JAD+01] Andreas Jedlitschka, Klaus-Dieter Althoff, Björn Decker, Susanne Hartkopf, and Markus Nick. Corporate information network (coin) : The fraunhofer iese experience factory. In *GI Jahrestagung (1)*, pages 54–60, 2001.

[Kas99] Vipul Kashyap. Design and creation of ontologies for environmental information retrieval. In *Proceedings of the 12th International Conference on Knowledge Acquisition, Modeling and Management*, Banff, Canada, 1999.

[KMR+94] Mika Klemettinen, Heikki Mannila, Pirjo Ronkainen, Hannu Toivonen, and A. Inkeri Verkamo. Finding interesting rules from large sets of

discovered association rules. In *CIKM '94 : Proceedings of the third international conference on Information and knowledge management*, pages 401–407, New York, NY, USA, 1994. ACM Press.

[LF05] Emiliano Lorini and Rino Falcone. Modeling expectations in cognitive agents. In *AAAI 2005 Fall Symposium-From Reactive to Anticipatory Cognitive Embodied Systems*, Arlington, Virginia, 2005.

[LFG99] Ninghui Li, Joan Feigenbaum, and Benjamin N. Grosof. A logic-based knowledge representation for authorization with delegation (extended abstract). In *Proceedings of the 1999 IEEE Computer Security Foundations Workshop*, pages 162–174. IEEE Computer Society Press, June 1999.

[LFZ99] Nada Lavrač, Peter A. Flach, and Blaz Zupan. Rule evaluation measures : A unifying view. In *International Workshop on Inductive Logic Programming*, pages 174–185, 1999.

[LHC97] Bing Liu, Wynne Hsu, and Shu Chen. Using general impressions to analyze discovered classification rules. In *Knowledge Discovery and Data Mining*, pages 31–36, 1997.

[LHM01] Bing Liu, Wynne Hsu, and Yiming Ma. Identifying non-actionable association rules. In *KDD '01 : Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 329–334, New York, NY, USA, 2001. ACM Press.

[LHML99] Bing Liu, Wynne Hsu, Lai-Fun Mun, and Hing-Yan Lee. Finding interesting patterns using user expectations. *IEEE Transactions on Knowledge and Data Engineering*, 11(6) :817–832, 1999.

[LMV+04] Philippe Lenca, Patrick Meyer, Benoît Vaillant, Philippe Picouet, and Stéphane Lallich. Evaluation et analyse multi-critères des mesures de qualité des règles d'association. *Revue des Nouvelles Technologies de l'Information, Mesures de Qualité pour la Fouille de Données, RNTI-E-1*, pages 219–246, 2004.

[LPS86] Douglas B. Lenat, M. Prakash, and M. Shepherd. Cyc : Using commonsense knowledge to overcome brittleness and knowledge acquisition bottlenecks. *AI Magazine*, 6(4) :65–85, 1986.

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

[LVL05] Stéphane Lallich, Benoît Vaillant, and Philippe Lenca. Parametrised measures for the evaluation of association rule interestingness. In *International Symposium on Applied Stochastic Models and Data Analysis (ASMDA 2005), Brest, France*, pages 220–229, 2005.

[LWY04] Jinze Liu, Wei Wang, and Jiong Yang. A framework for ontology-driven subspace clustering. In *KDD '04 : Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 623–628, New York, NY, USA, 2004. ACM Press.

[McG05] Ken McGarry. A survey of interestingness measures for knowledge discovery. *Knowl. Eng. Rev.*, 20(1) :39–61, 2005.

[MM95] J. Major and J. Mangano. Selecting among rules induced from a hurricane database. *J. Intelligent Information Systems*, pages 39–52, 1995.

[MTV94] Heikki Mannila, Hannu Toivonen, and A. Inkeri Verkamo. Efficient algorithms for discovering association rules. In Usama M. Fayyad and Ramasamy Uthurusamy, editors, *AAAI Workshop on Knowledge Discovery in Databases (KDD-94)*, pages 181–192, Seattle, Washington, 1994. AAAI Press.

[Pas00] Nicolas Pasquier. *Data mining : algorithmes d'extraction et de réduction des règles d'association dans les bases de données*. PhD thesis, Université de Clermont-Ferrand II, 2000.

[PBTL98] N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal. Pruning closed itemset lattices for association rules. In *Actes des 14èmes journées Bases de Données Avancées (BDA'98)*, pages 177–196, Octobre 1998.

[PBTL99a] N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal. Closed set based discovery of small covers for association rules. In *Actes des 15èmes journées Bases de Données Avancées (BDA'99)*, pages 361–381, Octobre 1999.

[PBTL99b] N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal. Discovering frequent closed itemsets for association rules. In *Proceedings of the 7th biennial International Conference on Database Theory (ICDT'99)*, Lecture

Notes in Computer Science, Vol. 1540, pages 398–416. Springer-Verlag, January 1999.

[PBTL99c] N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal. Efficient mining of association rules using closed itemset lattices. *Information Systems*, 24(1) :25–46, 1999.

[Poh03] Carsten Pohle. Integrating and updating domain knowledge with knowledge discovery. In *Doctoral Consortium held in conjunction with the 6th International Conference for Business Informatics 2003 (WI-2003)*, Dresden, Germany, 2003.

[PSF91] Gregory Piatetsky-Shapiro and William J. Frawley, editors. *Knowledge Discovery in Databases*. AAAI/MIT Press, 1991.

[PSM94] Gregory Piatetsky-Shapiro and Christopher J. Matheus. The interestingness of deviations. In *KDD-94*, pages 25–36, 1994.

[PT99] Balaji Padmanabhan and Alexander Tuzhilin. Unexpectedness as a measure of interestingness in knowledge discovery. *Decis. Support Syst.*, 27(3) :303–318, 1999.

[PT00] Balaji Padmanabhan and Alexander Tuzhilin. Small is beautiful : discovering the minimal set of unexpected patterns. In *KDD '00 : Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 54–63, New York, NY, USA, 2000. ACM Press.

[PTB+05] Nicolas Pasquier, Rafik Taouil, Yves Bastide, Gerd Stumme, and Lotfi Lakhal. Generating a condensed representation for association rules. *J. Intell. Inf. Syst.*, 24(1) :29–60, 2005.

[RFP02] W. Romão, Alex A. Freitas, and PCS Pacheco. A genetic algorithm for discovering interesting fuzzy prediction rules : applications to science and technology data. In WB Langdon and E Cantu-Paz et al, editors, *Proceedings of Genetic and Evolutionary Computation Conference (GECCO-2002)*, pages 1188–1195, San Francisco, CA, USA, July 2002. Morgan Kaufmann.

[SA95] Ramakrishnan Srikant and Rakesh Agrawal. Mining generalized association rules. In *VLDB '95 : Proceedings of the 21th International*

Conference on Very Large Data Bases, pages 407–419, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc.

[SA96] Ramakrishnan Srikant and Rakesh Agrawal. Mining sequential patterns : Generalizations and performance improvements. In Peter M. G. Apers, Mokrane Bouzeghoub, and Georges Gardarin, editors, *Proc. 5th Int. Conf. Extending Database Technology, EDBT*, volume 1057, pages 3–17. Springer-Verlag, 25–29 1996.

[Sam05] Jennifer Sampson. Converting data to knowledge : Applying a natural language technique. In *ODBIS 2005, VLDB Workshop on Ontologies-based techniques for Databases and Information Systems*, Trondheim, Norway, 2005.

[SB03] Mehrnoush Shamsfard and Ahmad Abdollahzadeh Barforoush. The state of the art in ontology learning : a framework for comparison. *Knowl. Eng. Rev.*, 18(4) :293–316, 2003.

[ST95] Abraham Silberschatz and Alexander Tuzhilin. On subjective measures of interestingness in knowledge discovery. In *Knowledge Discovery and Data Mining*, pages 275–281, 1995.

[ST96] A. Silberschatz and A. Tuzhilin. What makes patterns interesting in knowledge discovery systems. *Ieee Trans. On Knowledge And Data Engineering*, 8 :970–974, 1996.

[TKS02] P. Tan, V. Kumar, and J. Srivastava. Selecting the right interestingness measure for association patterns. In *Eight A CM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2002.

[TS98] Shiby Thomas and Sunita Sarawagi. Mining generalized association rules and sequential patterns using SQL queries. In *Knowledge Discovery and Data Mining*, pages 344–348, 1998.

[W3C02] W3C. Rdf vocabulary description language 1.0 : Rdf schema, 2002.

[Wei95] Gary M. Weiss. Learning with rare cases and small disjuncts. In *International Conference on Machine Learning*, pages 558–565, 1995.

[Wei99] Gary M. Weiss. Timeweaver : a genetic algorithm for identifying predictive patterns in sequences of events. In Wolfgang Banzhaf, Jason Daida, Agoston E. Eiben, Max H. Garzon, Vasant Honavar, Mark Jakiela, and

Robert E. Smith, editors, *Proceedings of the Genetic and Evolutionary Computation Conference*, volume 1, pages 718–725, Orlando, Florida, USA, 13-17 1999. Morgan Kaufmann.

[Wel96] Christopher A. Welty. *An integrated representation for software development and discovery.* PhD thesis, Troy, NY, USA, 1996.

[Wir00] Rüdiger Wirth. Crisp-dm : Towards a standard process model for data mining. In *Fourth International Conference on the Practical Applications of Knowledge Discovery and Data Mining*, pages 29–39, 2000.

[XZF98] D. Xu, G. Zheng, and X. Fan. A logic based language for networked agents. 40(8) :435–442, 1998.

[YHN06] S. Ben Yahia, T. Hamrouni, and E. Mephu Nguifo. Frequent closed itemset based algorithms : a thorough structural and analytical survey. *SIGKDD Explor. Newsl.*, 8(1) :93–104, 2006.

[YN04] S. Ben Yahia and E. Mephu Nguifo. Approches d'extraction de règles d'association basées sur la correspondance de galois. *Revue Ingénierie des Systèmes d'Information (ISI)*, 9(3/4) :109–132, septembre 2004.