HAL
archives-ouvertes.fr

Hiérarchisation des règles d'association en fouille de textes

Rokia Bendaoud, Yannick Toussaint, Amedeo Napoli

▶ **To cite this version:**

Rokia Bendaoud, Yannick Toussaint, Amedeo Napoli. Hiérarchisation des règles d'association en fouille de textes. Revue des Sciences et Technologies de l'Information - Série ISI : Ingénierie des Systèmes d'Information, Lavoisier, 2005, Extraction et gestion des connaissances (EGC'2005), 1, pp.263-274. inria-00000436

HAL Id: inria-00000436
https://hal.inria.fr/inria-00000436

Submitted on 14 Oct 2005

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

inria-00000436, version 1 - 14 Oct 2005

Hiérarchisation des règles d'association en fouille de textes

Rokia BENDAOUD*, Yannick TOUSSAINT*
Amedeo NAPOLI*

*LORIA Campus Scientifique - BP 239
54506 VANDOEUVRE-lès-NANCY CEDEX
{bendaoud,toussaint,napoli}@loria.fr,

Résumé. L'extraction de règles d'association est souvent exploitée comme méthode de fouille de données. Cependant, une des limites de cette approche vient du très grand nombre de règles extraites et de la difficulté pour l'analyste à appréhender la totalité de ces règles. Nous proposons donc de pallier ce problème en structurant l'ensemble des règles d'association en hiérarchies. La structuration des règles se fait à deux niveaux. Un niveau global qui a pour objectif de construire une hiérarchie structurant les règles extraites des données. Nous définissons donc un premier type de subsomption entre règles issue de la subsomption dans les treillis de Galois. Le second niveau correspond à une analyse locale des règles et génère pour une règle donnée une hiérarchie de généralisation de cette règle qui repose sur des connaissances complémentaires exprimées dans un modèle terminologique. Ce niveau fait appel à un second type de subsomption inspiré de la subsomption en programmation logique inductive.

Nous définissons ces deux types de subsomptions, développons un exemple montrant l'intérêt de l'approche pour l'analyste et étudions les propriétés formelles des hiérarchies ainsi proposées.

1 Introduction

L'extraction des règles d'association appliquée à des textes est une méthode de fouille de données qui permet de mettre en valeur des liens entre les termes des textes. Ces liens peuvent alors être interprétés par des experts en vue, par exemple, de la construction d'une ontologie.

Que ce soit à partir de textes où à partir de base de données, le nombre de règles extraites est souvent très grand et difficile à appréhender par un expert humain. De nombreux travaux se sont intéressés à élaguer l'ensemble des règles et à les classer soit par rapport à des critères statistiques, soit par rapport à une base de connaissances [Janetzko D. Cherfi H., 2004]. Nous proposons dans cet article une approche visant à structurer les règles sous forme hiérarchique afin de permettre à l'expert une approche descendante de la lecture de l'ensemble des règles. En réalité, nous proposons à l'expert deux approches d'analyse, un niveau global et un niveau local, tous deux reposant sur une structuration hiérarchique des règles. Ces deux types de structuration hiérarchique

nous ont conduit à définir deux types de subsomption qui, au final, peuvent être combinés.

L'approche globale a pour objectif de permettre à l'expert d'appréhender l'ensemble des règles extraites à partir d'un ensemble de textes. L'enjeu est donc de lui proposer une vision structurée et synthétique de cet ensemble de règles. Nous considérons alors les termes comme étant non hiérarchisés. La hiérarchie des règles ainsi construite repose sur un premier type de subsomption pour laquelle aucune propriété n'est plus générale (ni plus spécifique) qu'une autre. Par exemple des propriétés comme `vole`, `respire`, `pond` peuvent être considérées comme indépendantes les unes des autres.

Au niveau local, l'expert peut disposé de propriétés structurées. Nous construisons une hiérarchie locale autour d'une règle ou d'un sous-ensemble de règles en prenant en compte des propriétés hiérarchisées au sein d'un modèle terminologique, cette hiérarchie repose sur un second type de subsomption des règles.

Cet article est divisé en six parties. Dans la section 2, nous définissons les règles d'association, puis les treillis de Galois que nous utilisons comme support à l'extraction des règles. La section 3 précise notre méthodologie d'extraction de règles. La section 4 présente la structuration globale des règles et définit le premier type de subsomption qui repose sur la subsomption dans les treillis. La section suivante décrit la structuration locales des règles et définit le second type de subsomption permettant de généraliser les règles, en se reposant sur la subsomption en programmation logique inductive. Bien que ces travaux soient inspirés d'une problématique de fouille de texte, les deux types de subsomption ont été testées sur une base de données réduite pour faciliter l'interprétation des résultats.

2 Le contexte mathématique

2.1 Les règles d'association

Pour faire l'analogie avec le vocabulaire généralement utilisé en fouille de données, nous allons considérer un texte comme un individu et désignerons par I l'ensemble des individus. Les termes sont considérés comme des propriétés et l'ensemble des propriétés est noté P. Nous considérons la relation binaire $\mathcal{R}$ tel que $\mathcal{R} \subset \mathrm{I} \times \mathrm{P}$ et $\mathcal{R}$ (`i`,`p`) si l'individu `i` contient la propriété `p`.

Définition 1 (Les règles d'association)
Une règle d'association est une implication pondérée de la forme $A \Rightarrow B$, où A est la prémisse, et B est la conclusion, avec $A \subseteq P$, $B \subseteq P$ et $A \cap B = \emptyset$.

Les règles d'association [Agrawal et al., 1993] permettent de mettre en évidence les dépendances entre les propriétés. Par exemple, la règle : `vole` $\Rightarrow$ `pond` $\sqcap$ `respire` ($\sqcap$ désigne la conjonction des propriétés) peut s'interpréter comme le fait que si un individu `vole`, il est probable qu'il ait les propriétés `pond` et `respire`.

Définition 2 (Motif et image d'un motif)
*Un **motif** est un sous-ensemble de P. On dit qu'un individu i contient le motif M, si M et i sont en relation : $\forall\ p \in M : \mathcal{R}(i,p)$.*

*L'**image d'un motif** M est l'ensemble des individus qui contiennent le motif M.*

Le processus d'extraction des règles d'association est un processus exponentiel, en fonction du nombre d'individus et du nombre de propriétés. Il existe plusieurs méthodes pour réduire la complexité de ce processus, l'une d'elle est l'utilisation des indices statistique [Cherfi and Toussaint, 2002]. Les deux indices statistiques les plus couramment utilisés sont le support et la confiance, qui servent à réduire le nombre de règles extraites.

Rappels : support d'un motif, d'une règle, confiance et motif fréquent

Le support d'un motif représente le nombre d'individus qui possèdent le motif sur la cardinalité de l'ensemble des individus.

$$support(M_1) = \frac{Image(M_1)}{card(I)}$$

Le support d'une règle représente le nombre d'individus qui vérifient la règle, c'est-à-dire, qui possèdent le motif $\mathtt{A} \sqcap \mathtt{B}$

$$support(A \Rightarrow B) = support(A \sqcap B)$$

La confiance d'une règle $\mathtt{A} \Rightarrow \mathtt{B}$ est définie par le fait qu'un individu possède les propriétés B sachant qu'il possède celles de A :

$$confiance(A \Rightarrow B) = \frac{Support(A \Rightarrow B)}{Support(A)}$$

Motif fréquent Un motif est dit fréquent si et seulement si son support est supérieur à un seuil *minsupp*.

Définition 3 (Règle valide, règle totale, règle partielle et règle informative) *Soit* $\mathtt{R} : \mathtt{A} \Rightarrow \mathtt{B}$ *une règle :*

La règle R est ***valide*** *ssi support(*$\mathtt{R}$*)*$\geq$ *minsupp et si sa confiance est supérieure à un seuil* $\mathtt{minconf}$*.*

La règle R est ***totale*** *ssi confiance(*$\mathtt{R}$*) = 1, ce qui signifie qu'à chaque fois qu'un individu i possède A, i possède également B. Les règles totales ne possèdent donc pas de contre-exemple.*

La règle R est ***partielle*** *ssi confiance(*$\mathtt{R}$*)<1. Ce sont des règles qui possèdent des contre-exemples, c'est-à-dire des individus qui possèdent la partie gauche de la règle mais pas la partie droite.*

La règle $\mathtt{R}$ *est dite* ***informative*** *ssi elle est valide et* $A \cap B = \emptyset$*.*

Propriétés des règles Ces deux propriétés sont utilisées dans la section 5.1.

Prop1. transitivité : si $\mathtt{A} \Rightarrow \mathtt{B}$ et $\mathtt{B} \Rightarrow \mathtt{C}$ et que l'une des règles est valide et l'autre totales alors $\mathtt{A} \Rightarrow \mathtt{C}$ est valide.

Prop2. si $\mathtt{A} \Rightarrow \mathtt{B}$ et que Image(B) $\subseteq$ Image(B') alors $\mathtt{A} \Rightarrow \mathtt{B}$'.

Il existe différentes approches pour l'extraction des règles d'association. La première issue des travaux en bases de données, est l'extraction de règles à partir des algorithmes de motifs fréquents. La seconde est l'extraction des règles à partir d'un treillis de Galois. C'est ce deuxième type d'extraction de règles que nous allons utiliser.

2.2 Les treillis de Galois

Rappelons qu'un treillis de Galois ou treillis de concepts s'appuie sur une connexion de Galois et organise un ensemble de concepts formels –les fermés de la connexion– en un treillis ([Barbut and Monjardet, 1970], [Guénoche, 1990], [Ganter and Wille, 1999]). Les concepts se notent ci-dessous $C_k = (P_k,I_k)$ où P_k désigne les propriétés du concept C_k (l'intension du concept) et I_k les individus recouverts par le concept (l'extension du concept). La relation d'ordre partiel dans un treillis vérifie : $C_k \sqsubseteq C_{k'}$ ssi $I_k \subseteq I_{k'}$ (et de façon duale $P_{k'} \subseteq P_k$).

3 Extraction des règles d'association à partir d'un treillis de Galois

La formalisation mathématique de l'extraction de règles d'association à partir d'un treillis de Galois est présentée dans ([Guigues J.L, 1986], [Godin et al., 1995]) et fait appel à la notion de propriétés propres et de propriétés héritées pour un concept. De façon analogue, l'extraction de règles que nous proposons se fait en parcourant les concepts du treillis et en considérant l'intension du concept comme le motif commun à toutes les règles extraites à partir de ce concept. Le processus se déroule de la façon suivante :

- Soit $C_s = (P_s,I_s)$ sommet du treillis.
- si le support du motif $P_s \geq$ *minsupp*
  - alors extraire l'ensemble R_s des règles associées au motif P_s de la forme $P_i \Rightarrow P_s \backslash P_i$, tel que $P_i \subset P_s$.
  - calculer la confiance, supprimer les règles donc la confiance $<$ *minconf.*
  - appeler récursivement l'algorithme pour tous les concepts subsumés par C_s dans le treillis.
- sinon passer à une autre branche du treillis.

Soit la règle R_1 : `A` $\Rightarrow$ `B` extraite du concept $C_1 = (P_1,I_1)$, nous pouvons calculer support(R_1) et confiance(R_1) directement du treillis de Galois

$$support(R_1) = support(P_1) = \frac{card(I_1)}{card(I)} \;\; et \;\; confiance(R_1) = \frac{support(R_1)}{support(A)}$$

Pour trouver le support de `A` qui n'est peut-être pas un fermé, nous devons chercher le concept dont l'intension est le fermé minimal contenant le motif `A`. Pour cela, on part du sommet du treillis, cherchant le premier concept qui possède dans son intension le motif `A`, soit $C_j = (P_j\ ,I_j)$ ce concept, alors support(`A`) = support(I_j).

Seules les règles issues d'un motif fermé par rapport à la connexion de Galois sont extraites. L'algorithme est donc plus restrictif que Apriori (Agrawal et al. 1994) où la notion de fermé n'est pas utilisée. En revanche, cette méthode ne se limite pas à l'extraction de règles de type `clé` $\Rightarrow$ `fermé`$\backslash$`clé` comme c'est le cas pour *Close* [Bastide et al., 2002]. On obtient donc un sous-ensemble de règles par rapport à *Close*. [Bastide et al., 2002] montre que cet ensemble de règles constitue une base (non minimale). De même, notre ensemble de règles extraites constitue une base non minimale.

4 Classification de règles pour des propriétés non hiérarchisées

La subsomption de règles lorsque les propriétés sont non hiérarchisées repose directement sur la structure du treillis de Galois du contexte (I,P,$\mathcal{R}$) introduit en section 2. Elle est définie à partir de la subsomption sur l'intension des concepts que nous considérons comme des motifs. Nous appelons cette subsomption basée sur les motifs M-subsomption et nous la notons $\sqsubseteq_M$.

4.1 Subsomption des règles non hiérarchisées

Définition 4 (M-Subsomption des règles)

Soient C_1 = (P_1,I_1) et C_2 = (P_2,I_2) deux concepts du treillis de Galois. Soient R_1 une règle issue du motif P_1 et R_2 une règle issue du motif P_2.

R_2 M-subsome R_1 noté $R_2 \sqsubseteq_M R_1$ ssi $C_2 \sqsubseteq C_1$ dans le treillis du contexte (I,P,$\mathcal{R}$).

4.2 Les R-ensembles

Définition 5 (R-ensemble)

*Soit M un motif de longueur $\geq$ 1. Un **R-ensemble** engendré pour M, noté **R**(M), est défini comme l'ensemble des règles valides qu'il est possible d'extraire de M.*

Le fait que deux règles soient du même R-ensemble, signifie qu'elles ont été extraites du même concept dans le treillis. De ce fait nous allons les placer dans le même noeud de la hiérarchie des règles. Les règles d'un même R-ensemble ont le même support (en extension) mais pas forcément la même confiance.

Exemple : Soient le motif $\mathtt{P_1}$=

{`respire`, `pond`, `vole`}. Nous pouvons extraire 7 règles, supposons que 3 seulement sont valides :

R_1 : `respire`⇒`pond`, `vole`, R_2 : `pond`⇒`respire`, `vole`, R_3 : `vole`⇒`respire`, `pond`.

Ces règles font toutes partie du même R-ensemble noté **R**(`respire`, `pond`, `vole`).

4.3 Propriétés de la M-subsomption et du R-ensemble

Soient R_1, R_2 et R_3 trois règles extraites respectivement des concepts C_1, C_2 et C_3.

1. **transitivité** si $R_1 \sqsubseteq_M R_2$ et $R_2 \sqsubseteq_M R_3$ alors $R_1 \sqsubseteq_M R_3$. En effet, puisque si $C_1 \sqsubseteq C_2$ et $C_2 \sqsubseteq C_3$ alors $C_1 \sqsubseteq C_3$ car la subsomption entre concepts est transitive.
2. **réflexivité** $R_1 \sqsubseteq_M R_1$ car $C_1 \sqsubseteq C_1$.
3. **anti-symétrie** si $R_1 \sqsubseteq_M R_2$ et $R_2 \sqsubseteq_M R_1$ alors R_1 et R_2 sont du même R-ensemble, car si $C_1 \sqsubseteq C_2$ et $C_2 \sqsubseteq C_1$ cela implique que $C_1 = C_2$ et donc R_1 et R_2 sont extraites du même concept dans le treillis.

Les deux propriétés 1 et 2 définissent un pré-ordre sur l'ensemble des règles et les trois propriétés 1, 2 et 3 définissent un ordre partiel sur les R-ensembles.

4.4 Expérimentation sur la base du "zoo"

Dans cette section, nous présentons une expérimentation illustrant la M-subsomption sur une base de données réduite "Zoo" [Forsyth, 1991] où les individus dénotent des animaux (`antilope`, `ours`, `sanglier`, ..) et les propriétés (`pond`, `respire`, `vole`, ...) sont non hiérarchisées. Cette base de données compte 40 individus et 19 propriétés binaires. Nous avons construit le treillis de Galois et extrait les règles d'association à l'aide du logiciel Galicia [Valtchev et al., 2003]. Les règles d'associations ont était extraites avec $minsupp = 0.3$ et $minconf = 0.5$. Nous avons obtenu 38 règles partielles et 7 règles totales. Les règles extraites ont été hiérarchisées selon la M-subsomption.

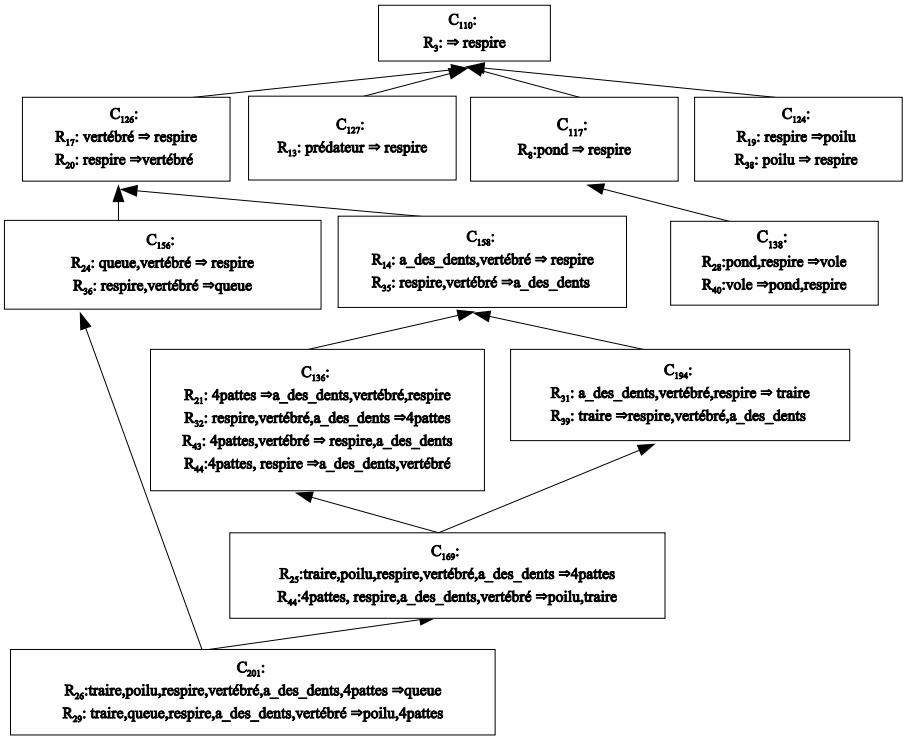

FIG. 1 – **Une partie de la hiérarchie des règles**

Une partie de la hiérarchie des règles est données par la figure 1. Chaque concept du treillis C_i (par exemple C_{117}, C_{127}...) définit un R-ensemble et les liens entre les R-ensembles sont les relations de subsomption existant entre les règles des différents R-ensembles. Supposons que l'expert prenne comme point de départ la règle R_3 : $\Rightarrow$ `respire`, qui veut dire l'ensemble des individus possèdent la propriété `respire`. Si cette règle lui semble intéressante, mais trop générale, il peut rechercher des règles plus spécifiques qui portent sur une population réduite. Les concepts C_{126}, C_{117}, C_{127} et C_{124} sont subsumés par le concept C_{110}, donc toutes les règles extraites de ces concepts sont M-subsumées par la règle R_3. Si l'expert veut prendre en compte la

propriété `vertébré`, il peut considèrer le concept C_{126}, et toutes les règles valides issues de ce concept lui sont proposées. Il peut réduire encore sa population ou augmenter l'information contenue dans les règles en travaillant sur un motif plus grand en ajoutant d'autres propriétés, comme : `a-des-dents` (concept C_{158}) ou `queue` (concept C_{156}). En choisissant la propriété `a-des-dents`, il accède à un R-ensemble de deux règles R_{14} : `a-des-dents`, `vertébré` ⇒ `respire` et R_{35} : `respire`,`vertébré`⇒ `a-des-dents`. S'il descend encore dans la hiérarchie vers les concepts C_{136} et C_{194} et qu'il trouve que sa population a été trop réduite (le support a trop diminué), il peut s'arrêter à ce niveau de la hiérarchie et ne pas consulter les règles plus spécifiques.

Maintenant, si l'analyste étudie la règle R_{13} : `prédateur` ⇒ `respire`, il se rendra compte qu'il n'existe aucune propriété pouvant être ajoutée au motif "`prédateur`, `respire`" pour avoir une règle valide, car il n'existe aucun descendant du R-ensemble **règles**(`prédateur`, `respire`) dans la hiérarchie des règles.

Cette méthode de classification des règles est simple et ne demande pas de calcul supplémentaire, la subsomption de règle étant directement issue du treillis. Elle offre à l'analyste une hiérarchie globale pour l'analyse des règles.

5 Subsomption des règles avec un modèle terminologique

La M-subsomption permie à l'expert d'avoir une vision globale et structurée de l'ensemble des règles. Supposons à présent que l'expert soit plus particulièrement intéressé par une règle et qu'il dispose d'un modèle terminologique qui structure en une hiérarchie l'ensemble des propriétés P. Nous définissons un second type de subsomption qui permet de générer de nouvelles règles généralisant la règle étudiée. Cette subsomption crée donc par rapport au treillis global une structure hiérarchique locale dont nous présentons les propriétés formelles.

En premier lieu, précisons ce que nous appelons modèle terminologique. De façon analogue au modèle de connaissances introduit dans la construction d'un treillis (Godin et al. 1995) notre modèle terminologique est une hiérarchie de propriétés $\mathcal{T}$ construite selon la relation *Est-un*, définie sur $\mathcal{T} \times \mathcal{T}$. L'interprétation de A *Est-un* B signifie que si un individu possède A alors il possède B qu'on lui rajoute car B n'est pas dans la base de données. La relation *Est-un* est réflexive, transitive et anti-symétrique, c'est donc un ordre partiel. Les propriétés de l'ensemble P (du contexte (I,P,$\mathcal{R}$) sont des feuilles pour la relation *Est-un*.

5.1 La subsomption en programmation logique inductive

La programmation logique inductive [Cornuéjols and Miclet, 2001] réalise l'apprentissage de formules de la logique des prédicats à partir d'exemples et de contre-exemples. L'enjeu est de construire des expressions logiques comportant des variables liées les unes aux autres.

L'objectif de la PLI est la construction de formules logiques incluant le plus d'exemples, et le moins de contre-exemples possibles. Notre objectif pour les règles est comparable. Nous souhaitons engendrer une règle qui généralise une ou plusieurs règles sans pour autant sur-généraliser et englober des contre-exemples.

Il existe plusieurs types de formules en logique des prédicats et celles qui nous intéressent sont les clauses, qui montrent une certaine similitude avec règles d'association. Nous allons rappeler la définition des clauses et nous inspirer de la définition de la subsomption entre clauses pour calquer la subsomption entre règles d'association.

Définition 6 (Clause, théorie et subsomption relative à une théorie)
*Une **clause** est une formule de la logique des prédicats, qui se compose d'une disjonction finie de littéraux dont toutes les variables sont quantifiées universellement. Une clause s'écrit :* $\neg B_1 \vee \neg B_2 \vee ... \vee \neg B_n \vee A_1 \vee A_2 .... \vee A_m$ *ou encore en abrégé :* $B_1, B_2, ..., B_n \rightarrow A_1, A_2, ..., A_m$.

*Une **théorie** est un ensemble de clauses.*

La clause C_1 ***subsume*** *la clause* C_2 *relativement à la théorie* T *si : de* $T \wedge C_1$ *nous pouvons déduire* C_2*, ce que nous notons :* $T \wedge C_1 \models C_2$ *ou* $C_1 \models_T C_2$.

De ces définitions, nous dérivons la subsomption entre règles d'association que nous nommons la H-subsomption, notée $\sqsubseteq_H$. En premier lieu, nous introduisons la notion d'ancêtre d'une propriété. Soient A, B deux propriétés du modèle terminologique. Si A $(Est\text{-}un)^*$ B (la relation *Est-un* peut-etre appliquée plusieurs fois), alors il existe un chemin dans la hiérarchie du modèle terminologique entre A et B. Tout ancêtre de A est noté Â.

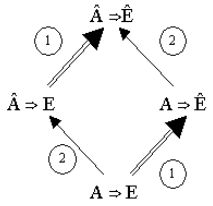

FIG. 2 – **Généralisation d'une règle**

Définition 7 (H-Subsomption des règles) *Soient deux règles* R_1 *:* $\mathtt{A} \Rightarrow \mathtt{B}$ *et* R_2 *:* $\mathtt{C} \Rightarrow \mathtt{D}$*,* $R_1 \sqsubseteq_H R_2$ *ssi C est un ancêtre de A et D est un ancêtre de B.*

Nous nous refusons à garder dans la hiérarchie des règles, des règles du type $\mathtt{A} \Rightarrow \hat{\mathtt{A}}$ ou $\hat{\mathtt{A}} \Rightarrow \mathtt{A}$. En effet, ces règles ne sont pas informatives ($A \cap \hat{A} \neq \emptyset$)

L'idée de la H-subsomption a été partiellement reprise de ([Agrawal and Srikant, 1995], [Maedche and Staab, 2000]), sur la généralisation de règles d'association en s'appuyant sur une hiérarchie des propriétés. Cependant au lieu de supprimer une règle du type A ⇒ B lorsqu'existe la règle Â ⇒ B avec Â l'ancêtre de A, nous avons défini la H-subsomption entre ces deux règles et nous les avons gardées toutes les deux, car nous pensons que la suppression de A ⇒ B va entraîner une perte d'information.

La figure 2 montre la hiérarchie des règles qui généralisent la règle $\mathtt{A} \Rightarrow \mathtt{E}$ telle que A et E sont des ensembles de propriétés. Nous allons décrire comment cette généralisation a été faite, sachant que pour un ensemble de propriétés $P_1 = p_1 p_2 ... p_n$ en remplaçant p_i tel que $1 \leq i \leq n$ par $\widehat{\mathtt{p_i}}$ dans P_1 on obtient un ancêtre de P_1 noté $\widehat{\mathtt{P_1}}$. Il y a deux types de généralisation différentes notées dans la figure 2 par ① et ② :

① Généralisation de la partie droite de la règle : Si $\mathtt{A} \Rightarrow \mathtt{E}$ alors $\mathtt{A} \Rightarrow \widehat{\mathtt{E}}$.
La démonstration est immédiate en appliquant la propriété Prop2 de la section 2.1. En effet Image(E)$\subseteq$ Image($\hat{\text{E}}$),puisque $\hat{\text{E}}$ est plus général que E et donc support($\widehat{E} \sqcap$ A) $\geq$ support(E $\sqcap$ A). La règle $\mathtt{A} \Rightarrow \widehat{\mathtt{E}}$ est valide car :

$$confiance(\mathtt{A} \Rightarrow \widehat{\mathtt{E}}) = \frac{support(\widehat{E} \sqcap A)}{support(A)} \geq confiance(\mathtt{A} \Rightarrow \mathtt{E}) = \frac{support(E \sqcap A)}{support(A)}$$

② Généralisation de la partie gauche de la règle : Si $\mathtt{A} \Rightarrow \mathtt{E}$ alors $\widehat{\mathtt{A}} \Rightarrow \mathtt{E}$ sous condition.
Ce type de généralisation est de nature inductive. Nous considérons la règle $\mathtt{A} \Rightarrow \widehat{\mathtt{A}}$ comme étant une théorie. De $\mathtt{A} \Rightarrow \widehat{\mathtt{A}}$ et de la règle $\widehat{\mathtt{A}} \Rightarrow \mathtt{E}$, nous pouvons déduire par transitivité des règles vu à la propriété Prop1 dans 2.1, la règle $\mathtt{A} \Rightarrow \mathtt{E}$, et d'après la définition de la subsomption en PLI, nous pouvons déduire que la règle $\widehat{\mathtt{A}} \Rightarrow \mathtt{E}$ subsume la règle $\mathtt{A} \Rightarrow \mathtt{E}$, par rapport à la théorie $\mathtt{A} \Rightarrow \widehat{\mathtt{A}}$. Pour ce type de généralisation, le support($\widehat{\mathtt{A}} \Rightarrow \mathtt{E}$) $\geq$ support($\widehat{\mathtt{A}} \Rightarrow \mathtt{E}$) $\geq$ *minsupp*, mais il faut vérifier la confiance($\widehat{\mathtt{A}} \Rightarrow \mathtt{E}$) pour que la nouvelle règle reste valide car le support de la partie gauche de la règle a augmenté ce qui peut entraîner une sur-généralisation et donc le fait d'englober trop de contre-exemples.

5.2 Propriétés de la H-subsomption

Soient $\mathtt{R}_1 : \mathtt{A} \Rightarrow \mathtt{B}$, $\mathtt{R}_2 : \mathtt{C} \Rightarrow \mathtt{D}$ et $\mathtt{R}_3 : \mathtt{E} \Rightarrow \mathtt{F}$

1. transitivité : si $\mathtt{R}_1 \sqsubseteq_H \mathtt{R}_2$ et $\mathtt{R}_2 \sqsubseteq_H \mathtt{R}_3$ alors : $\mathtt{R}_1 \sqsubseteq_H \mathtt{R}_3$, en effet car si C = $\hat{\text{A}}$ et E = $\widehat{C}$ alors E = $\hat{\text{A}}$ et D = $\widehat{B}$ et F = $\hat{\text{E}}$ alors F = $\widehat{B}$.
2. réflexivité : comme nous considérons que chaque propriété est son propres ancêtre alors : $\mathtt{R}_1 \sqsubseteq_H \mathtt{R}_1$.
3. anti-symétrie : si $\mathtt{R}_1 \sqsubseteq_H \mathtt{R}_2$ et $\mathtt{R}_2 \sqsubseteq_H \mathtt{R}_1$ alors $\mathtt{R}_1 = \mathtt{R}_2$, car si A = $\widehat{C}$ et C = $\hat{\text{A}}$ alors A = C et B = $\widehat{D}$ et D = $\widehat{B}$ alors D = B.

Ces trois propriétés définissent un ordre partiel.

La hiérarchie des règles peut ne pas être un treillis complet car la borne supérieure peut ne pas exister. Ceci est du à l'exclusion des règles du type $\mathtt{A} \Rightarrow \widehat{\mathtt{A}}$ et $\widehat{\mathtt{A}} \Rightarrow \mathtt{A}$ et par le fait que pour certaines généralisations nous devons contrôler la confiance.

5.3 Expérimentation sur des règles avec modèle terminologique

Nous avons expérimenté la H-subsomption sur une base de données de 6 individus et de 6 propriétés. On suppose que cette base a été créée par un professeur qui voudrait savoir quelles sont les grandes tendances dans le choix des modules. Le tableau de cette base est présenté dans la table 1.

Nous avons fixé $minsupp = 0.5$ et $minconf = 0.5$. Nous avons obtenu en appliquant un algorithme d'extraction de règles à partir de motifs fréquents 9 règles partielles P_i et 1 règle totale T_0, qui sont présentées dans le tableau 2.

Puis nous avons généralisé les règles pour lesquelles la partie droite ou gauche est composée d'une propriété ayant un ancêtre dans le modèle terminologique 3. Pour les

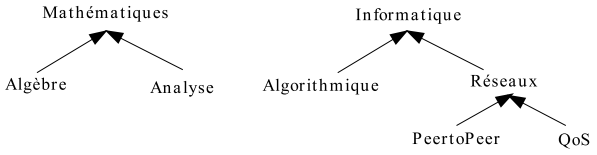

FIG. 3 – **Hiérarchie des propriétés**

R	Algèbre	Algorithmique	Probabilité	QoS	PeertoPeer	Biologie
I_1	1	1	1	1	1	0
I_2	1	1	0	0	0	1
I_3	0	0	1	0	0	0
I_4	0	1	1	0	1	0
I_5	1	1	1	1	0	0
I_6	0	0	0	0	1	1

TAB. 1 – **Représentation en tableau de la relation R**

	Règles	Sup.	Conf.		Règles	Sup.	Conf.
P_0	⇒ PeertoPeer	0.5	0.5	P_5	Algorithmique ⇒ Probabilité	0.5	0.75
P_1	⇒ Probabilité	0.66	0.66	P_6	⇒ Probabilité,Algorithmique	0.5	0.5
P_2	⇒ Algorithmique	0.66	0.66	P_7	Algorithmique ⇒ Algèbre	0.5	0.75
P_3	⇒ Algèbre	0.5	0.5	P_8	⇒ Algorithmique,Algèbre	0.5	0.5
P_4	Proba ⇒ Algorithmique	0.5	0.75	T_0	Algèbre ⇒ Algorithmique	0.5	1

TAB. 2 – **Les règles extraites du tableau de 1**

généralisations de type ②, nous avons contrôlé la confiance. Nous illustrons par la figure 4 la hiérarchie de règles construite à partir des deux règles : P7 : `Algorithmique` ⇒ `Algèbre` et P4 : `Algorithmique` ⇒ `Probabilité`, la hiérarchie résultante est présentée dans la figure 4

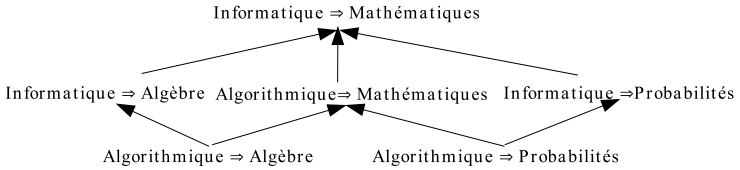

FIG. 4 – **Hiérarchisation des règles P5 et P7**

Supposons que le professeur cherche à caractériser le parcours des étudiants et qu'il considère la règle `informatique` ⇒ `Algèbre`. Il peut chercher des règles plus précises pour savoir s'il y a des sous-domaines de l'information qui sont plus particulièrement concernés. Ainsi, il accède à la règle `Algorithme` ⇒ `Algèbre`. À l'inverse, s'il veut une vision plus générale de la règle `informatique` ⇒ `Algèbre`, alors il accède à la règle `informatique` ⇒ `Mathématique` à travers un processus d'induction validé par le contrôle de confiance. De cet exemple nous pouvons déduire que la hiérarchie de

généralisation peut être partagée par plusieurs règles.

Ainsi, la subsomption entre règles avec des propriétés hiérarchisées permet d'exploiter un modèle terminologique et d'utiliser cette hiérarchie des propriétés pour généraliser les règles d'association.

6 Conclusion

Lorsque le nombre de règles d'association est important, l'analyste cherche à trouver des liens entre ces règles pour pouvoir en déduire les connaissances connues dans son domaine, et éventuellement de nouvelles connaissances. Nous avons ajouté la classification des règles d'association dans l'étape de fouille de données pour faciliter le travail de l'analyste lors de l'évaluation et de l'interprétation des règles extraites. Nous lui fournissons une hiérarchie de règles d'après les propriétés qui les composent, qui lui permet de faire ressortir les liens dont il a besoin.

Les règles d'association ayant des propriétés non hiérarchisées sont classifiées dès qu'elles sont extraites du treillis de Galois. De ce fait cette classification ne demande pas une étape supplémentaire et offre plusieurs avantages à l'analyste tels que le fait de redéfinir un support minimal s'il trouve que sa population a trop été réduite dans le bas de la hiérarchie, et de voir toutes les règles valides qui sont extraites du même concept (les règles du même R-ensemble).

La structuration des règles d'association avec des propriétés hiérarchisées permet de tenir compte des liens entre les différentes propriétés et de pouvoir généraliser l'une des deux parties de la règle.

La classification des règles d'association dans le cas d'une base de textes sert à relier les textes entres eux. Dans le cas des règles avec des propriétés hiérarchisées, cette relation entre les textes peut être interprétée comme le fait qu'un texte mentionne des termes plus spécifiques qu'un autre, ce qui peut aider l'expert à classifier ces textes.

Références

Références

[Agrawal et al., 1993] Agrawal, R., Imielinski, T., and Swami, A. (1993). Mining association rules between sets of items in large database. In *ACM SIGMOD Conference on Managment of Data*, pages 207–216, Washington.

[Agrawal et al., 1996] Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., and Verkamo, A. (1996). Fast discovery of association rules. In *Advances in Knowledge Discovery and Data Mining*, pages 307–328, California.

[Agrawal and Srikant, 1995] Agrawal, R. and Srikant, R. (1995). Mining generalized association rules. In *21st VLDB Conference*, Zurich, Swizerland.

[Barbut and Monjardet, 1970] Barbut, M. and Monjardet, B. (1970). *Ordre et classification – Algèbre et combinatoire (2 tomes)*. Hachette, Paris.

[Bastide et al., 2002] Bastide, Y., Taouil, R., Pasquier, N. ans Stumme, G., and Lakhal, L. (2002). Pascal : un algorithme d'extraction des motifs fréquents. In *Technique et science informatiques*, volume 21 - n°1/2002, pages 65–95.

[Cherfi and Toussaint, 2002] Cherfi, H. and Toussaint, Y. (2002). Fouille de textes par combinaison de règles d'association et d'indices statistiques. In *CIFT. Volume X-n° X*.

[Cornuéjols and Miclet, 2001] Cornuéjols, A. and Miclet, L. (2001). *Apprentissage artificiel*. EYROLLES, deuxième edition.

[Forsyth, 1991] Forsyth, R. (1991). Uci machine learning repository content summary. In *http ://www.ics.uci.edu/ mlearn/MLSummary.html*.

[Ganter and Wille, 1999] Ganter, B. and Wille, R. (1999). *Formal Concept Analysis*. Springer, Berlin.

[Godin et al., 1995] Godin, R., Mineau, R., Missaoui, R., and Mili, H. (1995). Méthodes de classification conceptuelle basées sur les treillis de galois et applications. In *Revue d'intelligence artificielle 9(2)*, pages 105–137.

[Guigues J.L, 1986] Guigues J.L, D. V. (1986). Familles minimales d'implications informatives résultant d'un tableau de données binaires. In *Mathématiques, Informatique et Sciences Humaines*, pages 5–18.

[Guénoche, 1990] Guénoche, A. (1990). Construction du treillis de galois d'une relation binaire. In *Revue Math. Inf. Sci. Hum, 109*, pages 41–53.

[Janetzko D. Cherfi H., 2004] Janetzko D. Cherfi H., Kennke R., N. A. e. T. Y. (2004). Knowledge-based selection of association rules for text mining. In de Mantaras, R. and L.Saitta, editors, *ECAI 2004*, Valencia, spain.

[Maedche and Staab, 2000] Maedche, A. and Staab, S. (2000). Discovering conceptual relation from text. In *Proceeding of the 14th European Conference on artifical intelligence*, pages 321–325, Berlin, Germany.

[Valtchev et al., 2003] Valtchev, P., Godin, R., Missaoui, R., Grosser, D., Roume, C., and Rouane-Hacene, A. (2003). Galicia. In *http ://www.iro.umontreal.ca/ galicia/*, Montréal, Québec.

Summary

Extraction of association rules is widely used as a data mining method. However, one of the limit of this approach comes from the large number of extracted rules and the difficulty for a human expert to deal with the totality of these rules. We propose to solve this problem by structuring the set of rules into hierarchy. The expert can then therefore explore the rules, access from one rule to another one more general when we raise up in the hierarchy, and in other hand, or a more specific rules.

Rules are structured at two levels. The global level aims at building a hierarchy from the set of rules extracted. Thus we define a first type of rule-subsomption relying on Galois lattices. The second level consists in a local and more detailed analysis of each rule. It generate for a given rule a set of generalization rules structured into a

local hierarchy. This leads to the definition of a second type of subsomption. This subsomption comes from inductive logic programming and integrates a terminological model.