Vers l'extraction de motifs rares

Laszlo Szathmary, Sandy Maumus, Petronin Pierre, Yannick Toussaint, Amedeo Napoli

Vers l'extraction de motifs rares

Laszlo Szathmary*, Sandy Maumus*,**, Pierre Petronin***
Yannick Toussaint*, Amedeo Napoli*

*LORIA, 54506 Vandoeuvre-lès-Nancy
{szathmar, maumus, yannick, napoli}@loria.fr
**INSERM U525, 54000 Nancy
Sandy.Maumus@nancy.inserm.fr
***ENSAI, 35172 Bruz Cedex
pierre.petronin@gmail.com

Résumé. Un certain nombre de travaux en fouille de données se sont intéressés à l'extraction de motifs et à la génération de règles d'association à partir de ces motifs. Cependant, ces travaux se sont jusqu'à présent, centrés sur la notion de motifs fréquents. Le premier algorithme à avoir permis l'extraction de tous les motifs fréquents est Apriori mais d'autres ont été mis au point par la suite, certains n'extrayant que des sous-ensembles de ces motifs (motifs fermés fréquents, motifs fréquents maximaux, générateurs minimaux). Dans cet article, nous nous intéressons aux motifs rares qui peuvent également véhiculer des informations importantes. Les motifs rares correspondent au complémentaire des motifs fréquents. A notre connaissance, ces motifs n'ont pas encore été étudiés, malgré l'intérêt que certains domaines pourraient tirer de ce genre de modèle. C'est en particulier le cas de la médecine, où par exemple, il est important pour un praticien de repérer les symptômes non usuels ou les effets indésirables exceptionnels qui peuvent se déclarer chez un patient pour une pathologie ou un traitement donné.

1 Introduction

La fouille de données a pour objectif d'identifier des relations cachées entre les motifs de grandes bases de données. La recherche de règles d'association est une des tâches les plus importantes de la fouille de données. L'extraction de règles d'association est un domaine de l'extraction de connaissances dans les bases de données (ECBD), qui se définit comme un procédé pour trouver des motifs valides, utiles et compréhensibles dans les données (Fayyad et al., 1996). Une règle d'association est une proposition de la forme "80% des étudiants qui suivent le cours *Introduction à Unix* suivent également *Programmation en C*" (Han et Kamber, 2001).

Jusqu'à présent, la littérature s'est intéressée à la recherche des règles d'association valides *fréquentes* (c'est-à-dire les règles d'association avec un support et une confiance suffisamment élevés). Cela requiert d'abord l'extraction des motifs fréquents de l'ensemble des données. Le problème de l'extraction des motifs fréquents était au départ un sous-problème de la fouille de

règles d'association (Agrawal et al., 1996), mais il s'est révélé plus tard utile dans différents domaines, tels que la fouille de motifs séquentiels (Agrawal et Srikant, 1995), de règles d'association spatiales (Koperski et Han, 1995), de règles d'association cycliques (Özden et al., 1998), de règles d'association négatives (Savasere et al., 1998), la recherche de motifs fermés fréquents (voir Section 1.1), de motifs fréquents maximaux (voir Section 1.1), etc.

Nous faisons l'hypothèse que certains phénomènes rares dans les bases de données peuvent également véhiculer une connaissance. C'est donc plus particulièrement l'extraction de motifs rares que nous étudions dans cet article.

1.1 Travaux en relation

L'extraction de motifs rares et la génération de règles d'association rares n'ont pas encore été étudiées en détail dans la littérature. Dans cet article, nous partons d'une vue d'ensemble de la recherche de motifs fréquents pour introduire notre méthode d'extraction des motifs rares.

Plusieurs approches ont été proposées pour trouver les motifs fréquents dans les bases de données. La première est basée sur l'algorithme Apriori, qui fut le premier algorithme par niveau à réaliser cette tâche (Agrawal et al., 1996). Cette méthode identifie les i-motifs à chaque i^{eme} itération puis génère les (i+1)-motifs fréquents à partir des i-motifs[1]. A chaque itération il requiert un passage sur la base de données pour compter le support des motifs candidats et ensuite élague les candidats infréquents. Cet algorithme est très simple et efficace pour des données peu corrélées. Apriori a été suivi par de nombreuses variations dans le but d'en améliorer l'efficacité (Brin et al., 1997; Toivonen, 1996).

La deuxième approche s'intéresse à la recherche de *motifs fermés fréquents* dans la base de données (Pasquier et al., 1999). Les motifs fermés fréquents permettent une représentation condensée et sans perte d'information des motifs fréquents, puisque l'ensemble des motifs fréquents (et leur support) peut être retrouvé à partir des motifs fermés fréquents. Cette idée fut implémentée dans Close (Pasquier et al., 1999), qui est aussi un algorithme par niveau. Depuis Close d'autres algorithmes ont été proposés pour la recherche de motifs fermés fréquents (Stumme et al., 2002; Zaki et Hsiao, 2002; Wang et al., 2003).

Un autre sous-ensemble intéressant de motifs fréquents est l'ensemble des *générateurs minimaux*. Bastide et al. ont montré comment utiliser les générateurs minimaux pour trouver les règles d'association informatives[2] (Bastide et al., 2002, 2000b). Parmi les règles partageant les mêmes individus comme support et ayant la même confiance, les règles construites à partir d'un motif fermé et ayant un motif générateur en partie gauche sont celles qui contiennent le plus d'information (Pasquier, 2000).

Le premier algorithme pour trouver les générateurs minimaux fut Pascal (Bastide et al., 2000a). Pascal peut réduire le nombre de passages sur la base de données et compter le support des candidats plus efficacement. Pascal trouve tous les motifs fréquents et tous les générateurs minimaux, mais ce n'est pas suffisant pour trouver les règles informatives. Pour la génération des règles d'association informatives, il faut identifier parmi les motifs fréquents, les motifs fermés et les associer aux générateurs minimaux. Pour résoudre cette insuffisance, un autre algorithme appelé Zart a été proposé récemment (Szathmary et al., 2005). Zart est un algorithme multifonctionnel d'extraction de motifs qui étend Pascal de manière à ce qu'il soit conforme à

[1] Un i-motif est un motif de taille i. Par exemple {A,C} est un 2-motif.

[2] L'expression "Règles d'association informatives" regroupe la Base Générique et la Base Informative.

la génération de règles d'association informatives. Zart trouve les motifs fréquents, les motifs fermés fréquents et les générateurs minimaux. De plus, les générateurs minimaux sont associés à leur fermeture. En conséquence, la génération des règles informatives peut être réalisée très rapidement et aisément avec Zart.

Une quatrième approche est basée sur l'extraction des *motifs fréquents maximaux*. Un motif fréquent maximal a les propriétés suivantes : tous ses sur-motifs sont infréquents et tous ses sous-motifs sont fréquents. Des expériences ont montré que cette approche est très efficace pour trouver de grands motifs dans les bases de données (Bayardo, 1998; Agarwal et al., 2000; Lin et Kedem, 1998; Gouda et Zaki, 2001). Les algorithmes basés sur cette approche identifient, comme Apriori, l'ensemble des règles d'association.

1.2 Contributions et motivations

Nous présentons une nouvelle méthode pour trouver les motifs rares dans une base de données en deux étapes. La première étape identifie un ensemble générateur minimal appelé ensemble des *motifs rares minimaux*. Dans la seconde étape, ces motifs sont utilisés pour retrouver tous les motifs rares.

La découverte des motifs rares peut se révéler très intéressante, en particulier en médecine et en biologie. Prenons d'abord un exemple simulé d'une base de données médicale où nous nous intéressons à l'identification de la cause des maladies cardio-vasculaires (MCV). Une règle d'association fréquente telle que "{niveau élevé de cholestérol} $\Rightarrow$ {MCV}" peut valider l'hypothèse que les individus qui ont un fort taux de cholestérol ont un risque élevé de MCV. A l'opposé, si notre base de données contient un grand nombre de végétariens, une règle d'association rare "{végétarien} $\Rightarrow$ {MCV}" peut valider l'hypothèse que les végétariens ont un risque faible de contracter une MCV. Dans ce cas, les motifs{végétarien} et {MCV} sont tous deux fréquents, mais le motif {végétarien, MCV} est rare. Un autre exemple est en rapport avec la pharmacovigilance, qui est une partie de la pharmacologie dédiée à la détection et l'étude des effets indésirables des médicaments. L'utilisation de l'extraction des motifs rares dans une base de données des effets indésirables des médicaments pourrait contribuer à un suivi plus efficace des effets indésirables graves et ensuite à prévenir les accidents fatals qui aboutissent au retrait de certains médicaments (par exemple en août 2001, la cérivastatine, médicament hypolipémiant). Finalement, un troisième exemple basé sur les données réelles de la cohorte STANISLAS (Siest et al., 1998; Maumus et al., 2005) montre l'intérêt de l'extraction des motifs rares pour la fouille de données dans des cohortes supposées saines. Cette cohorte est composée d'un millier de familles françaises présumées saines. Son principal objectif est de mettre en évidence l'influence des facteurs génétiques et environnementaux sur la variabilité des risques cardio-vasculaires. Une information intéressante à extraire de cette base de données pour l'expert dans ce domaine consiste en des profils qui associent des données génétiques à des valeurs extrêmes ou limites de paramètres biologiques. Cependant, ces types d'associations sont plutôt rares dans les cohortes saines. Dans ce contexte, l'extraction de motifs rares pourrait être très utile pour atteindre les objectifs de l'expert.

1.3 Organisation de l'article

Dans la section suivante, nous donnons une vue d'ensemble des concepts de base. La Section 3 détaille notre approche pour l'énumération des motifs rares basée sur les treillis, et

contient également les définitions essentielles. Nous décrivons ensuite dans la Section 4 les deux étapes de notre méthode et nous en fournissons les algorithmes, ainsi que des exemples les appliquant. Enfin, les conclusions sont présentées dans la dernière section.

2 Concepts de base

Ci-dessous nous utilisons les définitions usuelles de la fouille de données. Nous considérons un ensemble d'*objets* $O = \{o_1, o_2, \ldots, o_m\}$, un ensemble d'*attributs* $A = \{a_1, a_2, \ldots, a_n\}$ et une relation $R \subseteq O \times A$, où $R(o, a)$ signifie que l'objet o possède l'attribut a. En analyse de concepts formels (Ganter et Wille, 1999), le triplet (O, A, R) est appelé contexte formel. Un ensemble d'attributs est appelé *motif*. Un motif de taille i est appelé i-motif. Nous disons qu'un objet $o \in O$ contient le motif $P \subseteq A$, si $(o, p) \in R$ pour tout $p \in P$. Le *support* d'un motif P indique combien d'objets contiennent le motif. Un motif est dit *fréquent* si son support est supérieur ou égal à un *support minimum* donné (noté *min_supp* par la suite). Un motif est dit *rare* ou *infréquent* si son support est inférieur ou égal à un *support maximum* (noté *max_supp* par la suite). P_2 est un sur-motif de P_1 ssi $P_1 \subseteq P_2$. Dans cet article, nous nous sommes placés dans le cas particulier où $max\_supp = min\_supp - 1$, c'est-à-dire qu'un motif est rare s'il n'est pas fréquent. Cela implique l'existence d'une seule frontière entre motifs rares et fréquents. Boulicaut et al. (2003) fait par ailleurs lui aussi mention de cette frontière. Un motif X est dit *fermé* s'il n'existe pas de sur-motif Y ($X \subset Y$) de même support. L'extraction de motifs fréquents consiste à générer tous les motifs (fermés) fréquents (avec leur support) dont le support est supérieur ou égal à *min_supp*. L'extraction de motifs rares consiste à générer tous les motifs (avec leur support) dont le support est inférieur ou égal à *max_supp*.

3 Une approche basée sur les treillis pour l'énumération des motifs rares

Avant d'exposer nos algorithmes pour trouver les motifs rares, nous présenterons notre méthode du point de vue des treillis (voir Ganter et Wille (1999) pour une description détaillée des treillis).

La Figure 1 montre le treillis de l'ensemble des parties $P(D)$ de l'ensemble des attributs dans notre base de données exemple D[3] (voir Tableau 1). L'ensemble des motifs rares forme un semi-treillis "join" car il est fermé pour l'opération "join", c'est-à-dire que pour tous motifs rares X et Y, $X \cup Y$ est aussi rare. D'un autre côté, il ne forme pas un semi-treillis "meet", car la rareté de X et Y n'implique pas celle de $X \cap Y$. Notons que les motifs fréquents forment un semi-treillis "meet", c'est-à-dire que pour tous motifs fréquents X et Y, $X \cap Y$ est aussi fréquent.

Prenons l'exemple de la base de données D (Tableau 1) et fixons $min\_supp = 3$, ce qui signifie que $max\_supp = 2$. Les motifs peuvent être séparés en deux ensembles formant une partition : les motifs rares et les motifs fréquents. Une frontière peut être dessinée entre ces

[3] L'exemple est emprunté à Pasquier et al. (1999).

	A	B	C	D	E
1	x		x	x	
2		x	x		x
3	x	x	x		x
4		x			x
5	x	x	x		x

TAB. 1 – *Une base de données simple (D) utilisée dans les exemples.*

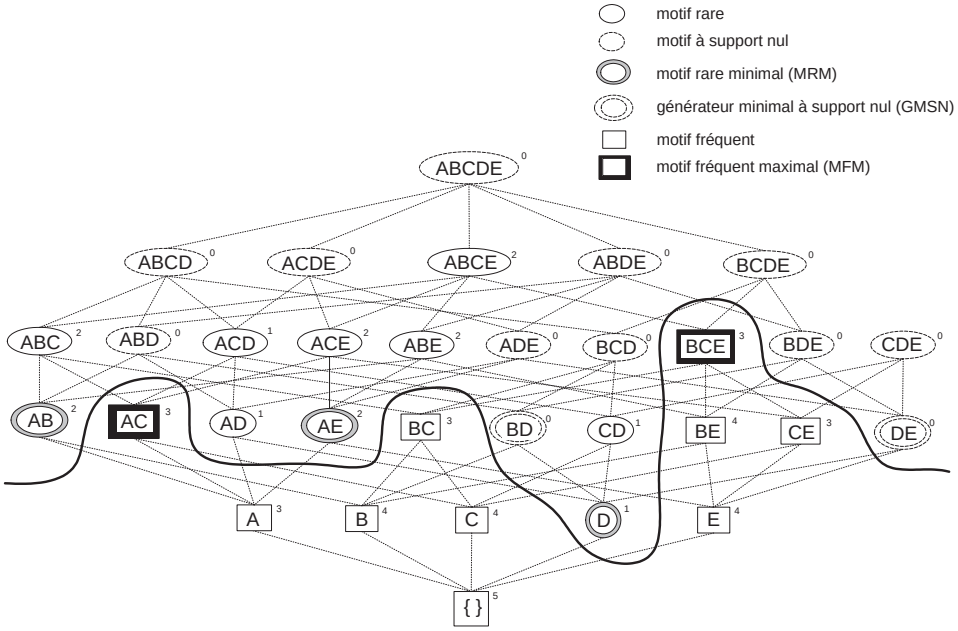

FIG. 1 – *Treillis des parties de la base de données D (voir Tableau 1).*

deux ensembles. En bas du treillis nous trouvons le plus petit motif, l'ensemble vide. A chaque niveau se situent les motifs de même taille. Au sommet du treillis on trouve le motif le plus long qui contient tous les attributs. Le support de chaque motif est indiqué dans le coin en haut à droite (voir Figure 1).

Avant d'énoncer les définitions essentielles, nous empruntons à Apriori (Agrawal et al., 1996) ses deux principes fondamentaux que nous rappelons ici :

Propriété 1 (propriété de *fermeture vers le bas*). Tous les sous-ensembles d'un motif fréquent sont fréquents.

Propriété 2 (propriété d'*anti-monotonocité*). Tous les sur-motifs d'un motif infréquent sont infréquents.

L'ensemble des motifs rares et l'ensemble des motifs fréquents ont tous deux un sous-ensemble minimal générateur. Dans le cas des motifs fréquents, ce sous-ensemble est appelé ensemble des *motifs fréquents maximaux* (MFM).

Définition 1. Un motif est un *MFM* s'il est fréquent (et ainsi tous ses sous-motifs sont fréquents) et si tous ses sur-motifs ne sont pas fréquents.

Ces motifs sont dits *maximaux*, parce qu'ils n'ont pas de sur-motifs fréquents. Du point de vue du nombre de ces motifs ils sont *minimaux*, c'est-à-dire qu'ils forment un ensemble générateur minimal à partir duquel tous les motifs fréquents peuvent être retrouvés[4].

Nous pouvons définir les *motifs rares minimaux* (MRM) en tant que complémentaires des MFMs, de la manière suivante :

Définition 2. Un motif est un *MRM* s'il est rare (et ainsi tous ses sur-motifs sont rares) et si tous ses sous-motifs ne sont pas rares.

Ces motifs forment un ensemble générateur minimal à partir duquel tous les motifs rares peuvent être retrouvés. Tous les motifs fréquents peuvent être retrouvés à partir des MFM. Dans un premier temps, nous devons prendre tous les sous-ensembles possibles des MFM. Dans un deuxième temps, le support des motifs fréquents peut être calculé grâce à un passage sur la base de données. Un processus similaire est mis en œuvre pour retrouver les motifs rares. Nous devons d'abord générer tous les sur-motifs possibles des motifs rares minimaux, puis calculer le support des motifs rares grâce à un passage sur la base de données.

Parmi les motifs rares, nous distinguons deux sous-ensembles : a) les motifs rares de support 0, et b) les motifs rares de support supérieur à 0. Cette distinction est importante, car le nombre total de motifs rares peut être élevé, et ainsi nous avons privilégié les motifs dont le support est non nul.

Définition 3. Un motif est appelé *motif à support nul* si son support est égal à 0. Autrement, il est appelé *motif à support non nul*.

Pour tous les motifs rares nous avons déjà décrit l'ensemble des motifs rares minimaux. Pour les motifs à support nul, un sous-ensemble générateur minimal semblable peut être défini :

Définition 4. Un motif est un *générateur minimal à support nul* (GMSN) si c'est un motif à support nul (ainsi tous ses sur-ensembles sont des motifs à support nul) et si tous ses sous-motifs sont des motifs à support non nul.

Sur la Figure 1 se trouvent deux GMSN : {BD} et {DE}. De plus, les GMSN forment une représentation condensée et sans perte d'information des motifs à support nul, c'est-à-dire qu'à partir des GMSN tous les motifs à support nul peuvent être retrouvés avec leur support (qui est toujours 0). Pour cela, nous avons seulement besoin de générer tous les sur-motifs possibles des GMSN en utilisant les attributs de la base de données.

4 Trouver les motifs rares

Dans cette section nous présentons les deux étapes de notre méthode pour trouver les motifs rares. La première étape trouve seulement les motifs rares minimaux, tandis que la seconde retrouve les motifs rares non nuls à partir de l'ensemble des motifs rares minimaux.

Nous ne générons pas les motifs à support nul à cause de leur grand nombre. Pour éviter les motifs à support nul, nous utiliserons les GMSN. La seconde étape de notre méthode (voir Section 4.2) permet de restaurer tous les motifs rares non nuls à partir des MRM à l'aide d'une approche par niveau. Si un candidat a un sous-motif GMSN, alors ce candidat est de manière

[4] Peut-être devrait-on les appeler plutôt motifs fréquents *les plus longs*, car ils n'ont pas de plus long sur-motif fréquent.

sûre un motif à support nul et peut être ainsi élagué. Autrement dit, à l'aide des GMSN nous pouvons réduire l'espace de recherche pendant que nous retrouvons tous les motifs rares.

4.1 Trouver les motifs rares minimaux

De manière surprenante, les motifs rares minimaux peuvent être trouvés simplement à l'aide de l'algorithme bien connu Apriori. Apriori est basé sur deux principes (voir Propriétés 1 et 2). Il est conçu pour trouver les motifs fréquents, mais, puisque nous sommes dans le cas où non fréquent signifie rare, cela a pour "effet collatéral" d'explorer également les motifs rares minimaux. Quand Apriori trouve un motif rare, il ne générera plus tard aucun de ses sur-motifs car ils sont de manière sûre rares. Puisque Apriori explore le treillis des motifs niveau par niveau du bas vers le haut, il comptera le support des motifs rares minimaux. Ces motifs seront élagués et plus tard l'algorithme peut remarquer qu'un candidat a un sous-motif rare. En fait Apriori vérifie si tous les $(k-1)$-sous-motifs d'un k-candidat sont fréquents. Si l'un d'entre eux n'est pas fréquent, alors le candidat est rare. Autrement dit, cela signifie que le candidat a un sous-motif rare minimal. Grâce à cette technique d'élagage, Apriori peut réduire significativement l'espace de recherche dans le treillis des motifs.

Une légère modification d'Apriori suffit pour conserver les MRM. Si le support d'un candidat est inférieur au support minimum, alors à la place de l'effacer nous l'enregistrons dans l'ensemble des motifs rares minimaux (voir Algorithme 1).

Algorithme 1 (Apriori-Rare) :

Description : modification d'Apriori pour trouver les motifs rares minimaux
Entrée : base de données + min_supp
Sortie : tous les motifs fréquents + motifs rares minimaux

1) $C_1 \leftarrow$ {1-motifs} ;
2) $i \leftarrow 1$;
3) while ($C_i \neq \emptyset$)
4) {
5) SupportCount(C_i) ; // compte le support des motifs candidats
6) $R_i \leftarrow \{r \in C_i \mid \text{support}(r) < min\_supp\}$; // R – pour les motifs rares
7) $F_i \leftarrow \{f \in C_i \mid \text{support}(f) \geq min\_supp\}$; // F – pour les motifs fréquents
8) $C_{i+1} \leftarrow$ Apriori-Gen(F_i) ; // C – pour les candidats
9) $i \leftarrow i + 1$;
10) }
11) $I_{MR} \leftarrow \bigcup R_i$; // motifs rares minimaux
12) $I_F \leftarrow \bigcup F_i$; // motifs fréquents

Fonction Apriori-Gen : à l'aide des k-motifs fréquents, génère les potentiellement fréquent candidats de taille $(k+1)$. Potentiellement fréquent signifie ne pas avoir de sous-motif rare, c'est-à-dire pas de sous-motif rare minimal. Inclure un motif rare implique être rare (voir Propriété 2). Pour une description détaillée de cette fonction consulter Agrawal et al. (1996).

C_1	supp
{A}	3
{B}	4
{C}	4
{D}	1
{E}	4

R_1	supp
{D}	1

F_1	supp
{A}	3
{B}	4
{C}	4
{E}	4

C_2	supp
{AB}	2
{AC}	3
{AE}	2
{BC}	3
{BE}	4
{CE}	3

R_2	supp
{AB}	2
{AE}	2

F_2	supp
{AC}	3
{BC}	3
{BE}	4
{CE}	3

C_3	supp
{BCE}	3

R_3	supp
$\emptyset$	

F_3	supp
{BCE}	3

C_4	supp
$\emptyset$	

TAB. 2 – *Exécution de l'algorithme Apriori-Rare.*

L'exécution de l'algorithme sur la base de données D (Tableau 1) avec un support minimum de 3 (équivalent à un support maximum de 2) est illustrée dans le Tableau 2.

En prenant l'union des R_i, l'algorithme trouve les motifs rares minimaux ({D} avec support 1, {AB} et {AE} avec support 2).

Dans la prochaine sous-section, nous montrons comment restaurer les sur-motifs des MRM (c'est-à-dire comment reconstruire tous les motifs rares) en évitant les motifs à support nul.

4.2 Retrouver les motifs rares

Tous les motifs rares sont retrouvés à partir des motifs rares minimaux. Pour cela nous avons besoin de générer tous les sur-motifs possibles des MRM. Les générateurs minimaux à support nul sont utilisés pour filtrer les motifs à support nul pendant la génération des sur-motifs. De cette manière l'espace de recherche peut être réduit de manière considérable. Dans cette section nous présentons un algorithme prototype pour cette tâche appelé Arima[5] (A Rare Itemset Miner Algorithm, voir Algorithme 2).

L'exécution de l'algorithme sur la base de données D (Tableau 1) avec un support minimum de 3 (équivalent à un support maximum de 2) est illustrée dans le Tableau 3.

L'algorithme prend d'abord le plus court MRM, {D}, qui est rare et ainsi copié dans R_1. Ses sur-motifs de taille 2 sont générés et stockés dans C_2 ({AD}, {BD}, {CD}, et {DE}). Avec un passage sur la base de données leur support peut être compté. Puisque {BD} et {DE} sont des motifs à support nul, ils sont copiés dans la liste des GMSN. A partir des MRM, les

[5] A ne pas confondre avec la méthodologie des modèles ARIMA (Auto Regressive Integrated Moving Average).

Algorithme 2 (Arima) :

Description : retrouve les motifs rares à partir des MRM
Entrée : base de données + MRM
Sortie : tous les motifs rares à support non nul + GMSN

1) $GMSN \leftarrow \emptyset$;
2) $S \leftarrow$ {tous les attributs de D};
3) $i \leftarrow$ {longueur du plus petit MRM};
4) $C_i \leftarrow \{i\text{-}MRM\}$; // c'est à dire les plus courts motifs dans les MRM
5) $GMSN \leftarrow GMSN \cup \{z \in C_i \mid \text{support}(z) = 0\}$;
6) $R_i \leftarrow \{r \in C_i \mid \text{support}(r) > 0\}$;
7) while ($R_i \neq \emptyset$)
8) {
9) boucle sur les éléments de R_i (r) {
10) $Cand \leftarrow$ {tous sur-motifs de r utilisant S}; // aucun doublon permis
11) boucle sur les éléments de $Cand$ (c) {
12) si c a un sous-motif dans $GMSN$ (c'est à dire si c est un sur-motif d'un $GMSN$), alors supprime c de $Cand$;
13) }
14) $C_{i+1} \leftarrow C_{i+1} \cup Cand$; // aucun doublon permis
15) $Cand \leftarrow \emptyset$; // ré-initialise $Cand$
16) }
17) SupportCount(C_{i+1}); // compte le support des motifs candidats
18) $C_{i+1} \leftarrow C_{i+1} \cup \{(i+1)$-motifs de $MRM\}$;
19) $GMSN \leftarrow GMSN \cup \{z \in C_{i+1} \mid \text{support}(z) = 0\}$;
20) $R_{i+1} \leftarrow \{r \in C_{i+1} \mid \text{support}(r) > 0\}$;
21) $i \leftarrow i + 1$;
22) }
23) $I_R \leftarrow \bigcup R_i$; // motifs rares à support non nul

---

$GMSN = \emptyset$
S = {A, B, C, D, E}
MRM = {D(1), AB(2), AE(2)}
$i = 1$

C_1	supp
{D}	1

$GMSN_{avant} = \emptyset$
$GMSN_{apres} = \emptyset$

R_1	supp
{D}	1

C_2	supp
{AD}	1
{BD}	0
{CD}	1
{DE}	0
{AB}	2
{AE}	2

$GMSN_{avant} = \emptyset$
$GMSN_{apres}$ ={BD, DE}

R_2	supp
{AD}	1
{CD}	1
{AB}	2
{AE}	2

C_3	supp
{ACD}	1
{ABC}	2
{ABE}	2
{ACE}	2

$GMSN_{avant}$ ={BD, DE}
$GMSN_{apres}$ ={BD, DE}

R_3	supp
{ACD}	1
{ABC}	2
{ABE}	2
{ACE}	2

C_4	supp
{ABCE}	2

$GMSN_{avant}$ ={BD, DE}
$GMSN_{apres}$ ={BD, DE}

R_4	supp
{ABCE}	2

C_5	supp
$\emptyset$	

$GMSN_{avant}$ ={BD, DE}
$GMSN_{apres}$ ={BD, DE}

R_5	supp
$\emptyset$	

TAB. 3 – *Exécution de l'algorithme Arima.*

2-motifs sont ajoutés à C_2 et les motifs non nuls sont stockés dans R_2. Pour chaque motif rare dans R_2 tous ses sur-motifs sont générés. Par exemple, à partir de {AD} nous pouvons générer les candidats suivants : {ABD}, {ACD} et {ADE}. Si un candidat possède un sous-motif GMSN, alors le candidat est de manière sûre un motif à support nul et peut être élagué ({ABD}, {ADE}). Les candidats potentiels non nuls sont stockés dans C_3. Dans les C_i les doublons ne sont pas permis. L'algorithme s'arrête quand R_i est vide. L'union des R_i donne tous les motifs rares à support non nul. A la fin nous avons aussi collecté tous les GMSN. Ainsi si on a besoin des motifs à support nul, cette liste peut être utilisée pour les retrouver. Le procédé est similaire : nous aurions besoin de générer tous les sur-motifs possibles des GMSN. Dans notre cas nous ne nous sommes intéressés qu'aux motifs non nuls, mais il est possible de travailler avec les motifs à support nul.

5 Conclusions et travaux futurs

Dans cet article, nous avons présenté une méthode pour extraire les motifs rares d'une base de données. Notre méthode est composée de deux étapes : 1) nous trouvons un sous-ensemble générateur minimal des motifs rares appelés MRM (algorithme Apriori-Rare) ; 2) à l'aide des MRM nous retrouvons les motifs rares dont le support est strictement supérieur à 0 (algorithme Arima).

Notre méthode fait partie des premières à s'intéresser spécifiquement aux motifs rares. Apriori fut le premier algorithme pour trouver les motifs fréquents et a été suivi par de nombreux algorithmes plus efficaces. De manière similaire, il ne fait aucun doute que nos algorithmes prototypes pourraient être améliorés de nombreuses manières. Dans le futur nous aimerions travailler sur ce sujet.

Parmi les motifs fréquents un certain nombre de sous-ensembles utiles ont été découverts, parmi lesquels les motifs fermés fréquents, les motifs fréquents maximaux, les générateurs (clés) minimaux, etc. Nous sommes curieux de découvrir si de tels sous-ensembles peuvent être définis pour les motifs rares. Nous connaissons déjà le complémentaire des motifs fréquents maximaux, qui est l'ensemble des motifs rares minimaux. Une autre question intéressante est la suivante. Les motifs fermés fréquents déterminent sans ambiguïté tous les motifs fréquents et leur support. Existe-t-il un sous-ensemble similaire qui déterminerait les autres motifs rares avec leur support ?

Un domaine important de l'utilisation des motifs rares est la génération des règles d'association rares. Par manque de place, nous n'avons pas pu développer ce sujet ici mais nous prévoyons d'étudier cette question en détail dans un autre article.

D'autre part, dans un futur proche, nous prévoyons de décrire des expériences réalisées sur des données réelles de la cohorte STANISLAS, dans le but de fournir un exemple concret de ce nouvel aspect prometteur de la découverte de connaissances dans les bases de données.

Références

Agarwal, R. C., C. C. Aggarwal, et V. V. V. Prasad (2000). Depth first generation of long patterns. In *KDD '00 : Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 108–118. ACM Press.

Agrawal, R., H. Mannila, R. Srikant, H. Toivonen, et A. I. Verkamo (1996). Fast discovery of association rules. In *Advances in knowledge discovery and data mining*, pp. 307–328. American Association for Artificial Intelligence.

Agrawal, R. et R. Srikant (1995). Mining sequential patterns. In *Proc. 1995 Int. Conf. Data Engineering (ICDE '95)*, pp. 3–14.

Bastide, Y., R. Taouil, N. Pasquier, G. Stumme, et L. Lakhal (2000a). Mining frequent patterns with counting inference. *SIGKDD Explor. Newsl. 2*(2), 66–75.

Bastide, Y., R. Taouil, N. Pasquier, G. Stumme, et L. Lakhal (2000b). Mining minimal non-redundant association rules using frequent closed itemsets. In J. *et al.*. Lloyd (Ed.), *Proc. of the Computational Logic (CL'00)*, Volume 1861 of *Lecture Notes in Artificial Intelligence – LNAI*, pp. 972–986. Springer.

Bastide, Y., R. Taouil, N. Pasquier, G. Stumme, et L. Lakhal (2002). Pascal : un algorithme d'extraction des motifs fréquents. *Technique et science informatiques 21*(1), 65–95.

Bayardo, R. J. (1998). Efficiently mining long patterns from databases. In *SIGMOD '98 : Proceedings of the 1998 ACM SIGMOD international conference on Management of data*, pp. 85–93. ACM Press.

Boulicaut, J.-F., A. Bykowski, et C. Rigotti (2003). Free-Sets : A Condensed Representation of Boolean Data for the Approximation of Frequency Queries. *Data Mining and Knowledge Discovery 7*(1), 5–22.

Brin, S., R. Motwani, J. D. Ullman, et S. Tsur (1997). Dynamic itemset counting and implication rules for market basket data. In *SIGMOD '97 : Proceedings of the 1997 ACM SIGMOD international conference on Management of data*, pp. 255–264. ACM Press.

Fayyad, U., G. Piatetsky-Shapiro, et P. Smyth (1996). From data mining to knowledge discovery in databases. *AI Magazine 17*, 37–54.

Ganter, B. et R. Wille (1999). *Formal concept analysis : mathematical foundations*. Berlin/Heidelberg : Springer.

Gouda, K. et M. J. Zaki (2001). Efficiently Mining Maximal Frequent Itemsets. In *ICDM '01 : Proceedings of the 2001 IEEE International Conference on Data Mining*, Washington, DC, USA, pp. 163–170. IEEE Computer Society.

Han, J. et M. Kamber (2001). *Data Mining : Concepts and Techniques*. San Francisco : Morgan Kaufmann Publishers.

Koperski, K. et J. Han (1995). Discovery of spatial association rules in geographic information databases. In *Proc. 4th Int. Symp. Large Spatial Databases (SSD '95)*, pp. 47–66.

Lin, D.-I. et Z. M. Kedem (1998). Pincer Search : A New Algorithm for Discovering the Maximum Frequent Set. In *EDBT '98 : Proceedings of the 6th International Conference on Extending Database Technology*, London, UK, pp. 105–119. Springer-Verlag.

Maumus, S., A. Napoli, L. Szathmary, et S. Visvikis-Siest (2005). Exploitation des données de la cohorte STANISLAS par des techniques de fouille de données numériques et symboliques utilisées seules ou en combinaison. In *5èmes Journées d'Extraction et Gestion des Connaissances, Workshop on Fouille de Données Complexes dans un Processus d'Extraction des Connaissances – EGC 2005, Paris, France*, pp. 73–76.

Özden, B., S. Ramaswamy, et A. Silberschatz (1998). Cyclic association rules. In *Proc. 1998 Int. Conf. Data Engineering (ICDE '98)*, pp. 412–421.

Pasquier, N. (2000). Mining association rules using formal concept analysis. In *Proc. of the 8th International Conf. on Conceptual Structures (ICCS '00)*, pp. 259–264. Shaker-Verlag.

Pasquier, N., Y. Bastide, R. Taouil, et L. Lakhal (1999). Efficient mining of association rules using closed itemset lattices. *Inf. Syst. 24*(1), 25–46.

Savasere, A., E. Omiecinski, et S. B. Navathe (1998). Mining for strong negative associations in a large database of customer transactions. In *ICDE '98 : Proceedings of the Fourteenth International Conference on Data Engineering*, pp. 494–502. IEEE Computer Society.

Siest, G., S. Visvikis, B. Herbeth, R. Gueguen, M. Vincent-Viry, C. Sass, B. Beaud, E. Lecomte, J. Steinmetz, J. Locuty, et P. Chevrier (1998). Objectives, Design and Recruitment of a Familial and Longitudinal Cohort for Studying Gene-Environment Interactions in the Field of Cardiovascular Risk : The Stanislas Cohort. *Clinical Chemistry and Laboratory Medicine (CCLM) 36*(1), 35–42.

Stumme, G., R. Taouil, Y. Bastide, N. Pasquier, et L. Lakhal (2002). Computing Iceberg Concept Lattices with TITANIC. *Data and Knowledge Engineering 42*(2), 189–222.

Szathmary, L., A. Napoli, et S. O. Kuznetsov (2005). ZART : A Multifunctional Itemset Miner Algorithm. LORIA Research Report A05-R-013.

Toivonen, H. (1996). Sampling large databases for association rules. In *Proc. of the 22nd International Conf. on Very Large Data Bases (VLDB '96)*, pp. 134–145.

Wang, J., J. Han, et J. Pei (2003). Closet+ : searching for the best strategies for mining frequent closed itemsets. In *KDD '03 : Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 236–245. ACM Press.

Zaki, M. J. et C.-J. Hsiao (2002). CHARM : An Efficient Algorithm for Closed Itemset Mining. In *SIAM International Conference on Data Mining SDM'02*, pp. 33–43.

Summary

In this paper we address the extraction of rare itemsets. Until now, studies in data mining have concentrated on frequent itemsets and how to generate association rules from frequent itemsets. The first algorithm to find frequent itemsets was Apriori, which has been followed by numerous other algorithms. Most of these algorithms are some kind of optimization of Apriori, and they share in common the fact that they all extract frequent itemsets or a subset of frequent itemsets (frequent closed itemsets, maximal frequent itemsets, minimal generators). In this paper we investigate the complementary of frequent itemsets, namely the rare (or infrequent) itemsets. To the best of our knowledge these itemsets have not yet been studied in detail, though these itemsets also contain important information beside the frequent patterns. One important field of usage for rare itemsets can be medicine, where it can be important for a doctor to notice if some of its patients with the same sickness have unusual symptoms for instance.