Représentation condensée en présence de valeurs manquantes

François Rioult, — Bruno Crémilleux

GREYC CNRS UMR6072
Université de Caen Basse-Normandie
Campus 2 Côte de Nacre
F-14.000 Caen
{Francois.Rioult, Bruno.Cremilleux}@info.unicaen.fr

RÉSUMÉ. *Le problème de la gestion des valeurs manquantes est ancien et classique dans les bases de données réelles. Nous décrivons les dommages causés par les valeurs manquantes sur les représentations condensées des motifs extraits de grands volumes de données. Ces représentations sont très utiles pour améliorer l'efficacité des extractions et autorisent de nouveaux usages des motifs (règle à prémisse minimale, clustering, classification). Nous montrons que ces représentations ne sont malheureusement plus fiables en présence de valeurs manquantes. Nous présentons une méthode de traitement de ces valeurs pour les plus usuelles des représentations condensées, fondées sur les motifs δ-libres ou fermés. Nous montrons la consistance de notre approche, et nous l'illustrons d'un point de vue expérimental. Les expériences sont menées avec notre prototype MVMINER (pour Missing Values miner), qui calcule la collection adéquate de motifs δ-libres.*

ABSTRACT. *Missing values are an old problem that is very common in real data bases. We describe the damages caused by missing values on condensed representations of patterns extracted from large data bases. This is important because condensed representations are very useful to increase the efficiency of the extraction and enable new uses of patterns (e.g., rules with minimal body, clustering, classification). We show that, unfortunately, such condensed representations are unreliable in presence of missing values. We present a method of treatment of missing values for condensed representations based on δ-free or closed patterns, which are the most common condensed representations. This method provides an adequate condensed representation of these patterns. We show the soundness of our approach, both on a formal point of view and experimentally. Experiments are performed with our prototype MVMINER (for Missing Values miner), which computes the collection of appropriate δ-free patterns.*

MOTS-CLÉS : *Fouille de données, motifs fréquents, représentation condensée, valeurs manquantes, données incomplètes.*

KEYWORDS: *Data mining, frequent patterns, condensed representation, missing values, incomplete data.*

1. Introduction

Contexte. La présence de valeurs manquantes dans les bases de données est un problème ancien qui s'est toujours posé lors de l'exploitation de données réelles. Par exemple, dans le domaine médical où le protocole d'expérimentation est à forte variabilité humaine, les données à exploiter sont rarement complètement renseignées. Le temps manque pour pratiquer un examen, ou le patient n'est pas en état de le supporter. En ce qui concerne les sondages d'opinion, il est rare que les interviewés prennent la peine de remplir complètement le questionnaire.

Les techniques classiques de fouille de données (classification, clustering, associations) ne sont pas conçues pour prendre en compte ces données incomplètes. La plupart des solutions mises en oeuvre se contentent de les retirer des exemples, ce qui ne permet pas de prendre en considération l'ensemble de la base, et introduit de nombreux biais dans l'analyse. L'attribution d'une valeur de remplacement commune à toutes les valeurs manquantes (valuation particulière, valeur par défaut, valeur moyenne) n'est pas plus satisfaisante, car cela exagère les corrélations (Grzymala *et al.*, 2000). Enfin des traitements particuliers à certaines bases sont proposés (Jami *et al.*, 1998), mais il est difficile de les appliquer dans le cas général, et il est clair qu'il n'existe pas de méthode universelle de traitement des valeurs manquantes. Nous verrons que le problème se pose également pour les représentations condensées.

Les représentations condensées de motifs fréquents fournissent une synthèse utile dans le cas de grands jeux de données (Boulicaut *et al.*, 2003, Pasquier *et al.*, 1999), mettant en valeur les corrélations présentes dans les données. Cette approche a deux avantages. Premièrement, elle permet d'améliorer l'efficacité des algorithmes pour des tâches usuelles comme la découverte de règles d'association (Agrawal *et al.*, 1993). Même si cette technique est aujourd'hui bien maîtrisée, l'utilisation de représentations condensées permet de réussir l'extraction des règles dans des contextes où les algorithmes à la APRIORI échouent. Deuxièmement, les représentations condensées autorisent des usages multiples des motifs fréquents (Mannila *et al.*, 1996, Durand *et al.*, 2002, Zaki, 2000) (e.g., règles fortes, informatives, à prémisse minimale, clustering, classification), ce qui est un point clé dans de nombreuses applications pratiques. Ces usages sont aujourd'hui plébiscités par les experts, qui savent que le développement de telles méthodes d'analyse leur permet de dégager une large plus-value sur leurs données.

Stratégie. Une grande majorité des réponses au problème des valeurs manquantes concerne leur imputation ou complétion. Il s'agit de proposer une valeur de remplacement pour appliquer les algorithmes connus à la base complète résultante. Nous nous démarquons résolument de cet objectif, car il nous semble plus efficace d'adapter les connaissances extraites à l'incertitude présente dans les données, en nous concentrant sur l'obtention de représentations condensées adéquates. C'est à dire que nous cherchons à qualifier les propriétés des motifs extraits dans la base incomplète relativement à des propriétés valides dans la base complète. Ensuite seulement nous appliquerons les algorithmes d'apprentissage à ces représentations.

La base complète n'est bien sûr pas disponible dans la plupart des situations, sauf lorsque la base incomplète résulte de l'introduction aléatoire de valeurs manquantes, comme ce sera le cas lors de nos expériences à la section 5 Nous verrons en revanche que nous pourrons déterminer, à l'examen de la base incomplète, des propriétés également valables dans la base originale complète, même si celle-ci est inconnue.

Ce principe n'est pas si surprenant. Si l'on considère que les valeurs manquantes occultent la véritable valeur d'une donnée, les fréquences d'apparition de certains motifs vont diminuer, car la décision de présence n'est plus possible pour certaines transactions. Un motif fréquent dans une base incomplète ne peut donc a fortiori qu'être fréquent dans la base complète. C'est cette propriété que nous développerons dans le cadre des représentations condensées.

Contributions. La contribution de ce travail est double. Premièrement, nous décrivons les dommages causés par les valeurs manquantes sur les représentations condensées. Deuxièmement, nous proposons une méthode de traitement des valeurs manquantes pour les représentations condensées fondées sur les motifs δ-libres ou fermés (qui sont les représentations condensées les plus classiques). Nous montrons la consistance de la représentation obtenue, tant d'un point de vue formel qu'expérimental. Nous pensons qu'il s'agit d'une amélioration importante pour la fouille de données et les multiples usages de ces représentations condensées : d'une part les données sont exploitées dans leur intégralité, incertitude comprise, et d'autre part les connaissances extraites sont qualifiées relativement à la base originale.

Organisation de l'article. La section 2 introduit plus précisément nos stratégies face à l'apparition de valeurs manquantes, ainsi que leurs effets sur la représentation condensée des motifs 0-libres. La section 3 rappelle brièvement le principe des représentations condensées des motifs δ-libres et nous formalisons dans la section 3.3 les effets des valeurs manquantes sur ces représentations. Dans la section 4 nous décrivons comment nous adaptons le formalisme des représentations condensées au contexte des valeurs manquantes et donnons un résultat formel sur la consistance de cette adaptation. La section 5 présente des expériences réalisées sur des benchmarks et des données réelles (base de données médicales sur la maladie de Hodgkin) qui mettent en évidence la justesse de notre approche.

2. Motivations : effets des valeurs manquantes

2.1 Définitions

Dans les bases réelles, l'utilisateur doit faire face aux valeurs manquantes, et nous allons voir que les représentations condensées ne sont plus valides en présence de valeurs manquantes, ce qui ne permet plus à l'utilisateur de travailler correctement.

C'est le point de départ de notre travail. Prenons un exemple pédagogique, que nous retrouverons tout au long de cet article. La partie gauche de la table 1 montre un

exemple de base de données (appelée *r* ou table complète de référence) constituée de 7 n-uplets mettant en relation trois attributs V_1, V_2, V_3.

	Attribut / valeur				Transactionnel						
id	V_1	V_2	V_3		*A*	*B*	*C*	*D*	*E*	*F*	*G*
1	+	→	0.2		×		×		×		
2	-	→	0			×	×		×		
3	+	→	0.1		×		×		×		
4	+	←	0.4		×			×		×	
5	-	→	0.6			×	×			×	
6	-	→	0.5			×	×			×	
7	+	←	1		×			×			×
8	-	←	0.8			×		×			×

Table 1. *Exemple d'une base de données r*

Une étape de binarisation fournit les données sous un format transactionnel (table 1 à droite), où chaque transaction est décrite suivant les items (notés de *A* à *G*) qu'elle contient. *A* et *B* codent ainsi la valeur deV_1, etc.

2.2 Codage des valeurs manquantes

Supposons maintenant que certaines variables n'aient pu être mesurées, ne sachant pas par exemple si V_1=*A* ou V_1=*B*. Une valeur manquante apparaît et nous utilisons le caractère `'?'` à la place de la valeur dans le cas attribut-valeur, et dans le format transactionnel pour *chaque* item codant cette variable. Nous avons introduit trois valeurs manquantes dans *r* et indiquons le codage correspondant table 2.

	Attribut / valeur				Transactionnel						
Id	V_1	V_2	V_3		*A*	*B*	*C*	*D*	*E*	*F*	*G*
1	+	→	0.2		×		×		×		
2	-	→	0			×	×		×		
3	+	→	**?**		×		×		**?**	**?**	**?**
4	+	←	0.4		×			×		×	
5	-	→	0.6			×	×			×	
6	**?**	→	0.5		**?**	**?**	×			×	
7	+	←	1		×			×			×
8	-	**?**	0.8			×	**?**	**?**			×

Table 2. *Base vm(r) : introduction de valeurs manquantes dans r*

La nouvelle base obtenue est appelée *vm*(*r*). Nous utilisons cette notation car nous disons qu'il est toujours possible de considérer une base complète *r*, une opération *vm()* qui introduit des valeurs manquantes, et le résultat est une base incomplète *vm*(*r*). L'opérateur *vm()* est par exemple relié au processus réel qui engendre les données, auquel cas *r* est indisponible ; ou bien *vm()* désigne une opération automatique, comme masquer aléatoirement la valeur de certaines variables. Dans tous les cas, nous utiliserons des propriétés issues de *vm*(*r*) pour qualifier celles de *r*.

2.3 Effet des valeurs manquantes

La table 3 fournit la représentation condensée des motifs 0-libres de support supérieur à un seuil de 2 transactions. Pour chaque 0-libre, nous indiquons sa fermeture (ces notions sont détaillées section 3). De cette représentation condensée, nous pouvons par exemple extraire la règle $AC \Rightarrow E$, présente deux fois dans les données, et une confiance de 100%. En effet, l'item *E* est toujours présent avec le motif *A C*. Une première lecture indique donc des règles fortes constituées du motif libre en prémisse et de la fermeture en conclusion. Il s'agit d'une représentation car il est possible de regénérer tous les motifs fréquents et toutes les règles possibles, et elle est condensée car elle contient naturellement bien moins de motifs que le total.

0-libre	fermeture	0-libre	fermeture
A		A C	E
B		A D	
C		A E	C
D		B C	
E	C	B F	C
F		C F	B
G	D		

Table 3. *Représentation condensée de r*

La fouille de données orientée motif s'attache à quantifier la présence d'un motif dans les données, ou l'association entre plusieurs motifs. Quand un item est manquant (repéré par `'?'`) dans une transaction, un motif contenant cet item ne peut être présent. Il y aura donc diminution du support et perte d'associations, ce qui désagrège la représentation condensée.

Comment donc agir avec les valeurs manquantes ? Les méthodes élémentaires retirent ces valeurs (un item manquant est déclaré absent) mais cela conduit à un jeu de données biaisé et des connaissances extraites peu fiables. Prenons un exemple : la table 4 décrit la représentation condensée extraite de *vm*(*r*) en appliquant cette méthode. Elle contient des motifs comme *DG* qui ne sont pas présents dans la représentation issue de la base originale *r*.

0-libre	fermeture	0-libre	fermeture
A		A C	
B		A E	
C		B C	
D	A	B E	C
E	C	B F	C
F		C F	
G		D G	

Table 4. *Représentation condensée de vm(r) en ignorant les valeurs manquantes*

Nous qualifierons plus tard (voir section 3.3) ces motifs de *pollution*. Nous notons également que de nombreux items ont disparu des fermetures. Consécutivement, la règle $AC \Rightarrow E$ n'est plus retrouvée. Clairement, les valeurs manquantes sont responsables de la disparition d'associations et de l'invention de motifs libres. Les expériences de la section 5 montrent que ce résultat est général et il est clair que le calcul de la représentation condensée doit être aménagé en présence de valeurs manquantes. Le but de cet article est de proposer une solution à ce problème ouvert.

3. Représentations condensées

Nous rappelons brièvement la cadre des représentations condensées de motifs fréquents et décrivons les effets des valeurs manquantes sur les représentations obtenues.

3.1 Découverte de motifs

Considérons une base de données transactionnelle r de schéma R, un multi-ensemble de transactions t composées d'items (éléments de R). Sur la figure 1, $r = \{t_1, ..., t_8\}$ où t_1=*A C E* (nous utilisons une notation sous forme de chaîne, e.g. *A C E* pour $\{A,C,E\}$, XA pour $X \cup \{A\}$), t_2=*B C E*, etc. Soit Z un *motif* (i.e. un ensemble d'items de R), une règle d'association fondée sur Z est une expression $X \Rightarrow Y$ avec $X \subset Z$ et $Y = Z \setminus X$.

Le support de X relativement à un ensemble de transactions r est le nombre de transactions de r qui contiennent X, i.e. $supp(X,r) = |\{t \in r \mid X \subseteq t\}|$. Nous notons r_X le sous-ensemble de transactions de r contenant X, nous avons $\text{supp}(X,r) = |r_x|$. Quand r est clair dans le contexte, nous utilisons $supp(X)$ pour $supp(X,r)$. La confiance de $X \Rightarrow Y$ est la proportion de transactions contenant X qui contiennent aussi Y (Agrawal *et al.*, 1993), i.e. $conf(X \Rightarrow Y) = supp(X \cup Y)/supp(X)$.

3.2 Motifs δ-libres

Définition 1 (règle δ-forte) *Une règle δ-forte sur $Z=X \cup Y$ est une règle d'association de la forme $X \Rightarrow Y$ qui admet au plus δ exceptions.*

La confiance d'une telle règle est au moins égale à $1-(\delta / supp(X))$. Quand δ est nul, la confiance atteint le maximum possible : 1 ou 100 %.

Définition 2 (motif δ-libre) *Un motif Z est dit δ-libre s'il n'existe sur Z aucune règle δ-forte $X \Rightarrow Y$ (avec $X \subset Z$ et $Y=Z \setminus X$).*

Le cas où δ=0 (correspondant aux motifs 0-libres) est important : aucune règle de confiance égale à 1 n'existe sur *Z*. Par exemple, *A C* est 0-libre car aucune règle construite avec des sous-ensembles de *A C* n'est exacte (*A* est présent sans *C* et vice-versa). D'un point de vue technique, les règles δ-fortes peuvent être construites à partir des motifs δ-libres, qui en constituent les parties gauches ou prémisses (Boulicaut *et al.*, 2003). Les motifs δ-libres sont reliés au concept de presque fermeture :

Définition 3 (presque fermeture) *Soit δ un entier positif. $AC(X,r)$, la presque-fermeture de X dans r, rassemble les items A tels que :*

$$supp(X,r)-supp(XA,r) \leq \delta \qquad [1]$$

Notons que si $X \Rightarrow Y$ est une règle δ-forte dans *r*, les items de *Y* appartiennent à $AC(X,r)$. En d'autres termes, quand un item appartient à $AC(X,r)$, cela signifie qu'il est présent dans toutes les transactions qui contiennent *X*, avec un nombre d'exceptions borné par δ. En reprenant l'exemple de la figure 1, $C \in AC(B,r)$ avec δ=1 (il y a une unique exception, la transaction t_8). La δ-liberté satisfait la propriété importante d'anti-monotonie et nous disposons alors d'extracteurs efficaces pour ces motifs, qui permettent de puissants élagages (Boulicaut *et al.*, 2003).

La collection des motifs δ-libres et fréquents est une représentation condensée des motifs fréquents : quand on connaît les motifs de cette représentation, on peut regénérer l'ensemble complet des motifs fréquents, en bénéficiant de plus d'une condensation de l'information nécessaire. Plus précisément, si δ=0, il est possible de calculer le support de tous les motifs fréquents. Alors, le concept de presque fermeture correspond au cas particulier de *fermeture*. Les motifs fermés ont des propriétés pertinentes pour le calcul de règles informatives (Bastide *et al.*, 2002) ou non redondantes (Zaki, 2000), proposer un clustering (Durand *et al.*, 2002). Si δ>0, il est possible d'approximer le support de chaque motif fréquent avec une erreur bornée : dans (Boulicaut *et al.*, 2003), il est montré qu'en pratique l'erreur est faible. Les motifs δ-libres ont enfin des propriétés remarquables pour construire des règles δ-fortes avec une confiance élevée ou de caractérisation de classe (Crémilleux *et al.*, 2002).

3.3 Effet des valeurs manquantes sur les représentations condensées

Les valeurs manquantes produisent des effets sur les motifs δ–libres et les items des presque fermetures. Soit *X* un motif δ-libre. Supposons qu'un item A appartienne à la presque fermeture de *X* : cela signifie qu'il est toujours présent avec *X*, exception faite d'un nombre de transactions inférieur à δ. Si des valeurs manquantes se produisent sur *A*, ce nombre d'exceptions ne peut que croître, et peut devenir supérieur à δ : *A* sort alors de la presque fermeture de *X* et *XA* devient libre (voir figure 1). Sur notre exemple, avec $\delta = 0$, *D* est dans la fermeture de *G* dans *r*, tandis que *D* sort de la fermeture dans *vm(r)* à cause de la valeur manquante de la transaction t_8.

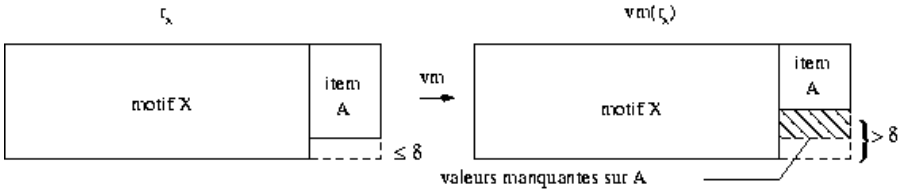

Figure 1. *Valeurs manquantes sur des items de presque fermeture*

4. Représentation adéquate aux valeurs manquantes

Dans cette section, nous montrons d'un point de vue formel que les propriétés découvertes dans la base avec valeurs manquantes restent valides dans la base originale sans valeurs manquantes, même si dans la pratique cette base n'est pas disponible. Ce résultat permet d'utiliser de façon fiable les informations découvertes dans *vm*(*r*), car elles sont consistantes avec celles de *r*.

4.1 Donnée désactivée

En présence de valeurs manquantes, les supports décroissent (Ragel *et al.*, 1998). Sur notre exemple, *supp*(*CE*,*r*)=3 mais *supp*(*CE*,*vm*(*r*))=2. En fait, pour calculer correctement le support d'un motif dans *vm*(*r*), il est nécessaire de distinguer les transactions de *vm*(*r*) qui ont une valeur manquante parmi les items de *X*. Ces transactions vont être temporairement désactivées pour calculer une estimation de *supp*(*X*,*r*) à l'aide de *supp*(*X*,*vm*(*r*)), car il est impossible de décider si oui ou non elles contiennent *X*.

Définition 4 (donnée désactivée) *Une transaction t de vm(r) est désactivée pour X si tous les items de X sont dans t, sauf au moins l'un d'entre eux qui est déclaré manquant dans t. Nous notons Des(X,vm(r)) les transactions de vm(r) désactivées pour X.*

Cette approche permet de retrouver des valeurs pertinentes du support (Ragel *et al.*, 1998), l'estimation des supports réels étant faite relativement aux seules transactions où la décision est possible, en excluant *Des*(*X*,*vm*(*r*)) :

Définition 5 (support estimé) *Le support estimé d'un motif X est*

$$\text{supp}_{est}(X, vm(r)) = \frac{|vm(r)|}{|vm(r)| - |Des(X, vm(r))|}.\text{supp}(X, vm(r))$$

Par exemple, *supp*(*CE*,vm(*r)*)=2 mais la transaction 3 contient *C* et une valeur manquante sur *E* ; elle est désactivée. Le support estimé est donc $\text{supp}_{\text{est}}(CE, vm(r)) = 8/(8-1).2 = 2.3$. Cette valeur est plus proche du support dans la base originale (*supp*(*C E*,*r*)=3) que la valeur calculée dans *vm*(*r*). Le support estimé conserve de plus la propriété d'anti-monotonie utilisée comme critère d'élagage dans les algorithmes d'extraction (Kryskiewicz, 1999).

Avec cette technique, une base de données partielle est utilisée pour évaluer *un* motif *X* en cas de valeurs manquantes sur *X*. Temporairement désactivées, les transactions contenant des valeurs manquantes relatives à un motif ne gênent plus l'évaluation de son support. Quand on change de motif, ces transactions retrouvent leur rôle normal et d'autres sont éventuellement neutralisées. Mais il est important de noter que finalement la base entière est utilisée pour évaluer *tous* les motifs.

4.2 Calcul de support

De façon naturelle, la définition des transactions désactivées permet de relier le support dans la base incomplète et celui dans la base complète :

Proposition 1 (Calcul de support par désactivation)

$$\text{supp}(X, \text{vm}(r)) = \text{supp}(X, r) - |\text{Des}(X, \text{vm}(r_X))|$$

Dans la pratique, seul *supp*(*X*,*vm*(*r*)) est connu, le support de *X* dans la base réelle est inconnu. C'est le cas également du nombre de transactions désactivées dans r_X, dont on connaît en revanche une borne supérieure |*Des*(*X*,*vm*(*r*))|.

Pour bien saisir la différence entre $Des(X, vm(r_X))$ et *Des(X,vm(r))*, nous représentons figure 2 la base vm(r) superposée à r. On suppose que chaque transaction de la moitié supérieure contient X et cette moitié de la base est repérée r_X. La moitié inférieure est repérée $r_{\overline{X}}$.

La zone intermédiaire hachurée désigne les transactions de *vm*(*r*) qui contiennent des valeurs manquantes. Elle est constituée de transactions qui contenaient *X* et constituent désormais $Des(X, vm(r_X))$ mais également de transactions qui ne contenaient pas *X* initialement. Certains items s'excluant les uns les autres quand les données binaires proviennent de données au format attribut-valeur, un attribut manquant cache autant de valeurs manquantes. Une valeur manquante sur un item qui n'appartient pas à *X* peut malgré tout favoriser la désactivation d'une transaction relativement à *X*, mais cette information n'est pas disponible.

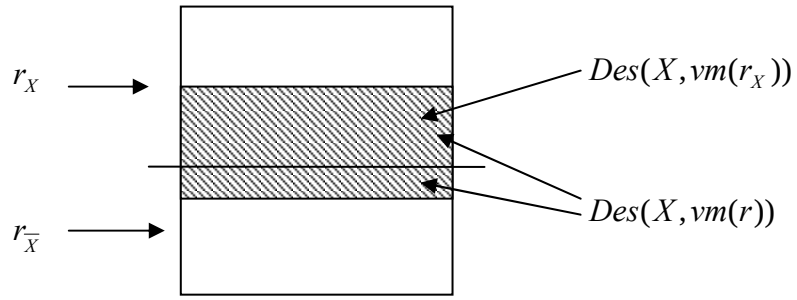

Figure 2. ***Base vm(r) et transactions désactivées pour X***

Détaillons ce principe sur notre exemple de base pour le motif AC : $r_{AC} = \{t_1, t_3\}$. Les valeurs manquantes ajoutées par l'opérateur *vm*() dans r_{AC} concernent les items *E*,*F*,*G*, donc $Des(AC, vm(r_{AC})) = 0$. Nous avons bien l'égalité de la proposition 1. Quand on ne connaît pas la base complète *r*, on ne connaît pas non plus r_X, encore moins $|Des(X, vm(r_X))|$. Mais nous pouvons borner cette quantité en considérant *vm*(*r*) plutôt que $vm(r_X)$ Sur notre exemple $Des(AC, vm(r)) = \{t_6\}$, à cause de la confusion entre *A* et *B*. *supp*(*AC*,*r*) est donc compris entre *supp*(*AC*,*vm*(*r*)) et *supp*(*AC*,*vm*(*r*)) + |*Des*(*AC*,*vm*(*r*))|.

Cette relation s'applique récursivement pour calculer le support de *XA* en fonction des désactivations de *X* et *A*, qui nous sera utile dans une prochaine démonstration :

Proposition 2 (Calcul récursif de support) *Soit X un motif et A un item :*

$$\text{supp}(XA, vm(r)) = \text{supp}(XA, r) - |\text{Des}(X, vm(r_{XA}))| - |\text{Des}(A, (vm(r_{XA}))_X)|$$

En effet, les transactions désactivées pour *XA* le sont à la fois pour *X* (relativement à $vm(r_{XA})$) et pour *A* quand *X* est présent : $Des(A, (vm(r_{XA}))_X)$ désigne les transactions de *r* contenant *XA*, où des valeurs manquantes sont introduites, pour lesquelles *X* subsiste dans son intégralité.

Bien sûr comme précédemment, on ne connaît ni le support théorique ni le nombre de transactions désactivées dans la base complète r_{XA}. On ne peut pas non plus

mesurer le nombre de transactions désactivées dans r_{XA}. Là encore nous obtiendrons des bornes en utilisant $vm(r)$ à la place de $vm(r_{XA})$ car elle contient plus de transactions désactivées.

4.3 Correction des effets des valeurs manquantes

Si la notion de support a été revisitée grâce aux transactions désactivées, nous adaptons maintenant le formalisme des représentations condensées en redéfinissant l'opérateur de fermeture. En présence de valeurs manquantes, il faut prendre en compte les données désactivées relativement à un item A pour tester si $A \in AC(X,vm(r))$. Nous proposons de redéfinir la presque fermeture de X de la manière suivante :

Définition 6 *$AC(X,vm(r))$, la presque fermeture de X dans vm(r), rassemble les items A tels que :*

$$\text{supp}(\text{X}, \text{vm}(r)) - \text{supp}(\text{XA}, \text{vm}(\text{r})) <= \delta + \left|\text{Des}(\text{A}, (\text{vm}(\text{r}))_{\text{X}})\right| \qquad [2]$$

Notons qu'en l'absence de valeurs manquantes, $|Des(A,(vm(r))_X)| = 0$, et $r=vm(r)$, et nous reconnaissons la définition usuelle de la presque fermeture (cf. Définition 3). Cette nouvelle définition est totalement compatible avec l'ancienne quand il n'y a pas de valeurs manquantes. L'inégalité 2 peut être vue comme une généralisation de l'inégalité de la définition originale, et comme une relaxation locale de la contrainte fixée par δ (i.e. δ est ajusté pour chaque A) relativement au nombre de valeurs manquantes sur A dans $(vm(r))_X$. Il s'agit d'une stratégie optimiste : si A devait appartenir à la fermeture malgré les valeurs manquantes, alors nous pouvons valider cette relation. La propriété suivante énonce la consistance de notre nouvelle définition :

Propriété 1 *La Définition 6 de la presque fermeture est consistante en présence de valeurs manquantes, i.e. $A \in AC(X,r) \Rightarrow A \in AC(X,vm(r))$. Alors, en utilisant cette définition, l'effet des valeurs manquantes (perte d'items dans les presque fermetures et pollution des δ-libres (cf. section 3.3) est corrigé.*

Cette propriété est fondamentale, car la définition 6 de la fermeture n'utilise que des calculs que l'on peut effectuer dans la base incomplète, en tenant compte des transactions désactivées. Elle exprime malgré tout une contrainte sur une caractéristique de la base complète, même si celle-ci est indisponible et si les calculs sont menés en présence de valeurs manquantes.

Preuve : Considérons $A \in AC(X,r)$, alors $supp(X,r)-supp(XA,r) \leq \delta$. Or, nous pouvons exprimer cette variation du support à l'aide des propositions 1 et 2 :
$supp(X,r)-supp(XA,r)=supp(XA,vm(r))-supp(X,vm(r))+(|Des(X,vm(r_X))|-|Des(X,vm(r_{XA}))|)$. La quantité entre parenthèses $|Des(X,vm(r_X))|-|Des(X,vm(r_{XA}))|$ est positive, car r_{XA} est plus restrictive que r_X donc contient nécessairement moins de transactions désactivées. Ainsi $supp(X,r)-supp(XA,r) \geq supp(XA,vm(r))-supp(X,vm(r))$

- $|Des(A,(vm(r_{XA}))_X|$. En utilisant la borne supérieure $|Des(A,(vm(r))_X|$ de $|Des(A,(vm(r_{XA}))_X|$, $supp(X,r)-supp(XA,r) \geq supp(XA,vm(r))-supp(X,vm(r)) - |Des(A,(vm(r))_X|$. Cette relation d'ordre assure que $A \in AC(X,r) \Rightarrow A \in AC(X,vm(r))$.

Il est donc possible de retrouver tous les items des presque fermetures en présence de valeurs manquantes. Sur notre exemple, cette méthode permet d'obtenir sur *vm*(*r*) la représentation condensée indiquée table 5.

0-libre	fermeture	0-libre	fermeture
A		A C	E
B	C	A E	C
C		B E	C
D	A	B F	C
E	C	C F	B
F			
G	D		

Table 5. Représentation condensée de vm(r)

La portée de la propriété 1 est clairement soulignée par le corollaire suivant :

Corollaire 1 *Si Z est δ-libre dans vm(r) alors Z est δ-libre dans r.*

Preuve : Soit Z δ-libre dans vm(r). Aucune règle δ-forte ne peut donc être construite sur Z, ou pour tout X motif et A item tels que Z=XA, $\neg(A \in AC(X,vm(r))$. Cette expression est la contraposée de la propriété 1, d'où $\neg(A \in AC(X,r))$, ou Z est δ-libre dans r.

Ainsi, avec la nouvelle définition de la presque fermeture, les motifs δ-libres découverts dans *vm*(*r*) sont également δ-libres dans *r*. L'information révélée par la liberté d'un motif dans *vm*(*r*) est confirmée dans *r*, même inconnue. Sur notre exemple jouet, la prise en compte des transactions désactivées permet d'affirmer que *D* est dans la fermeture de *G*, et *DG* n'est pas libre.

5. Expérimentations et résultats

Le but de cette section est de comparer les représentations condensées obtenues grâce à nos corrections avec celles obtenues sans correction. Les expériences sont réalisées avec notre prototype MVMINER, qui extrait les motifs δ-libres avec leurs presque fermetures, en respectant la nouvelle définition. MVMINER peut être vu comme une instance de l'algorithme par niveau présenté dans (Mannila *et al.*, 1997), extrayant des motifs δ-libres.

5.1 Cadre des expérimentations

Dans la section précédente, nous avons montré que les propriétés de liberté découvertes dans *vm*(*r*) sont consistantes avec *r*. Pour illustrer ce propos, nous devons disposer des deux versions de la base, ce qui n'est pas toujours possible dans une situation réelle où seule *vm*(*r*) est disponible.

Aussi, pour les besoins de l'expérience, nous effectuons des simulations. Une base de données sans valeurs manquantes *r* sert de base de référence. Appelons *représentation condensée de référence* la représentation condensée obtenue à partir de *r*. Puis, des valeurs manquantes sont ajoutées aléatoirement, suivant une distribution uniforme, pour donner *vm*(*r*). Nous discutons des différences entre les représentations obtenues sur *vm*(*r*) en traitant les valeurs manquantes de façon élémentaire (en les ignorant) ou en les intégrant pour désactiver les transactions (en appliquant MVMINER) et nous les comparons par rapport à la représentation condensée de référence. Dans la suite, notre méthode est simplement appelée MVMINER et la méthode élémentaire est appelée "méthode usuelle". La comparaison des représentations s'effectue avec les mesures suivantes : pollution des δ-libres et reconstitution des presque fermetures, différences de support estimé entre méthode usuelle et MVMINER, pollution des 0-libres relativement à la proportion de valeurs manquantes.

Les expérimentations ont été menées sur des benchmarks (bases de données classiques en fouille de données, issues de l'université d'Irvine[1] et sur des données réelles fournies par l'Organisation Européenne de Recherche et Traitement du Cancer (OERTC). Les résultats obtenus sont similaires d'une base à l'autre (Rioult, 2002), et nous avons choisi de ne présenter que les résultats relatifs à la base OERTC. Cette base possède l'avantage de recueillir des données réelles sur un problème qui intéresse beaucoup les médecins. Elle rassemble 576 patients souffrant de la maladie de Hodgkin. Chaque patient est qualifié par 26 attributs multi-valués qui donnent lieu à 75 items binaires.

La problématique étudiée suggère de faire varier trois paramètres : la proportion de valeurs manquantes artificiellement introduites, le support minimum d'extraction et la valeur de δ. En réalité, les expériences ont montré que seules les modifications de δ produisent des renversements de tendance. Nous présentons donc le résultat des expériences menées avec 10 % de valeurs manquantes par attribut, un support minimum de 10 %, et nous faisons varier δ de 0 à 20 transactions (de 0 à 3.5 %).

5.2 Correction des presque fermetures

Les phénomènes mis en évidence dans cette section sont liés aux effets des valeurs manquantes, précédemment décrits dans la section 3.3.

[1] http: //www.ics.uci.edu/~mlearn/MLRepository.html

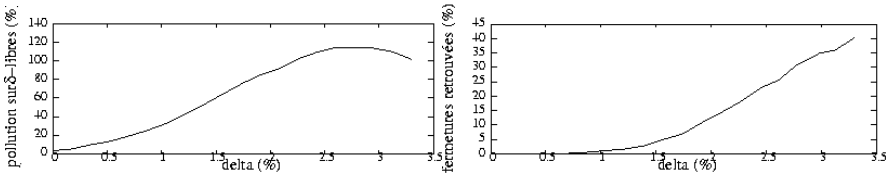

Figure 3. *Inconvénients de la méthode usuelle*

La figure 3 (côté gauche) montre la proportion de motifs δ-libres trouvés dans *vm(r)* par la méthode usuelle et n'appartenant pas à la représentation condensée de référence (i.e. pollution des δ-libres). La méthode usuelle subit une pollution de 50% dès les premières valeurs de δ (1.5% ou 8 transactions). Cela signifie qu'un nombre important de motifs aberrants est découvert et qu'il est évidemment impossible de les différencier des bons motifs. D'autre part, dans les représentations obtenues avec MVMINER, nous n'avons relevé aucune pollution, et ce pour toute valeur de δ. Ce résultat était bien sûr prévisible, suite à la propriété 1, et n'est pas représenté sur la figure.

La figure 3 (côté gauche) indique que le niveau de pollution avec la méthode usuelle est faible quand δ est nul ou très faible, et on pourrait imaginer utiliser cette méthode. La figure 3 (côté droit) détruit cet espoir : les items des presque fermetures ne sont pas retrouvés. Même pour des valeurs importantes de δ, seule la moitié de ces items est retrouvée. Avec les corrections, MVMINER retrouve parfaitement tous les items des presque fermetures. Cela montre qu'il est impossible d'avoir confiance en la représentation condensée obtenue avec la méthode usuelle, même pour de faibles valeurs de δ et souligne à nouveau les avantages de MVMINER.

5.3 Estimation des supports

La figure 4 compare les erreurs sur la valeur estimée des supports des motifs δ-libres, entre la méthode usuelle ($supp(X,vm(r))$) et MVMINER ($supp_{est}(X,vm(r))$). L'amélioration obtenue est précieuse pour calculer des mesures d'intérêt sur les règles qui utilisent le support, comme la confiance, le χ^2, etc (Lenca *et al.*, 2003).

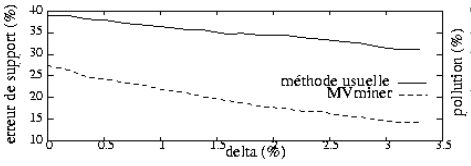

Figure 4. *Erreur de support : méthode usuelle contre MVMINER*

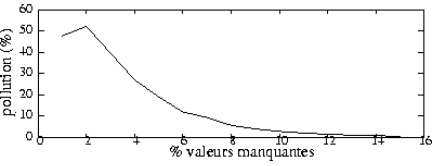

Figure 5. *Pollution des 0-libres (méthode usuelle)*

5.4 Pollution en fonction du taux de valeurs manquantes

Considérons maintenant la pollution apportée par la *méthode usuelle* sur la représentation condensée constituée des motifs 0-libres (un cas très courant de représentation condensée, génératrice des motifs fermés), relativement à la proportion de valeurs manquantes. Cette pollution est la proportion de motifs découverts qui n'appartiennent pas à la représentation de référence. Des valeurs manquantes ont été introduites sur chaque item suivant le pourcentage indiqué à la figure 5. Cette figure montre une pollution importante dès les premières valeurs manquantes introduites (50 % de motifs infondés). Avec beaucoup de valeurs manquantes, cette pollution décroît car trop peu de motifs sont retrouvés. Rappelons qu'il n'y a *aucune pollution* sur les motifs avec MVMINER, car chaque libre trouvé dans $vm(r)$ est un libre de r (cf. section 5.2).

6. Conclusion et travaux futurs

Nous avons présenté les dommages dus aux valeurs manquantes sur les représentations condensées fondées sur les motifs δ-libres ou fermés, une représentation condensée très courante. Sans traitement, les représentations condensées calculées sur des données incomplètes sont inadéquates, ce qui gène les usages multiples des motifs obtenus. Notre analyse clarifie l'effet des valeurs manquantes sur les motifs δ-libres et leurs presque fermetures.

Nous avons proposé des corrections ainsi qu'une nouvelle et consistante définition de la presque fermeture, généralisation de la définition originale et totalement compatible. Sans connaître la base originale sans valeurs manquantes, ces corrections assurent de ne retrouver que des motifs cohérents avec les motifs de la représentation de référence et n'introduisent pas de pollution dans les motifs. Comme elles vérifient les propriétés d'anti-monotonie, ces corrections sont applicables même dans les volumes de données de grande taille, denses ou très corrélées. Des expérimentations sur des données réelles ou des benchmarks confirment qu'il est impossible d'utiliser de façon fiable les représentations condensées obtenues sans correction (e.g. haut niveau de pollution sur les motifs δ-libres même pour de faibles taux de valeurs manquantes) et confirment la pertinence de nos corrections.

Notre objectif futur est de souligner davantage l'intérêt de notre technique en montrant, grâce aux experts, que nous constatons des améliorations lors de l'exploitation de nos représentations corrigées. Nos collaborations en médecine fournissent d'excellentes applications en clustering et classification, méthodes qui utilisent les représentations condensées. Nous cherchons également à améliorer la consistance déjà obtenue en explorant d'autres caractérisations de l'opérateur de fermeture.

Remerciements

Nous remercions le Dr M. Henry-Amar (Centre François Baclesse, France) de nous avoir fourni les données sur la maladie de Hodgkin et pour ses commentaires utiles. François Rioult est financé par l'Unité IRM du CHU de Caen, le Comité de la Manche de la Ligue Contre le Cancer et le Conseil Régional de Basse-Normandie.

References

Agrawal R., Imielinski T., Swami A., Mining association rules between sets of items in large databases, *SIGMOD'93*, 1993, 207-216.

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

Boulicaut J.-F., Bykowski A., Rigotti C., Free-sets : a condensed representation of boolean data for the approximation of frequency queries, *DMKD journal* 7-1, 2003.

Crémilleux B., Boulicaut J.-F., Utilisation de règles δ-fortes pour caractériser des classes, *RFIA'02*, 2002.

Durand N., Crémilleux B., ECCLAT : a new approach of clusters discovery in categorical data, *ES'02*, 2002, 177-190.

Grzymala-Busse J., Hu M., A comparison of several approaches to missing attribute values in data mining, *Int. Conf. on Rough Sets and Current Trends in Computing, Banff,* 2000.

Jami S., Liu X., Loizou G., Learning from an incomplete and uncertain data set: the identification of variant haemoglobins, *Workshop on IDAMP, ECAI'98*, 1998.

Kryszkiewicz M., Association rules in incomplete databases, *PAKDD 99,* 1999, 84-93.

Lenca P., Meyer P., Picouet P., Aide multicritère à la décision pour évaluer les indices de qualité des connaissances, *RSTI-RIA (EGC'03), 1(17): 271-282*, 2003.

Mannila H., Toivonen H., Multiple Uses of Frequent Sets and Condensed Representations (Extended Abstract), *Knowledge Discovery and Data Mining*, 1996, 189-194.

Mannila H.Toivonen H., Levelwise Search and Borders of Theories in Knowledge Discovery, *Data Mining and Knowledge Discovery*, 1, 3, 1997, 241-258.

Pasquier N., Bastide Y., Taouil R., Lakhal L., Efficient Mining of Association Rules Using Closed Itemset Lattices, *Information Systems 24-1*, 24, 1, 1999.

Ragel A., Crémilleux B., Treatment of missing values for association rules, *PAKDD'98,* 1998, 258-270.

Rioult F., Représentation condensée pour les bases de données adéquate aux valeurs manquantes, Technical Report, 60 pages, GREYC, Université de Caen, 2002.

Zaki M. J., Generating Non-Redundant Association Rules, *SIGKDD'00, Boston*, 2000, 34-43.