Vers une nouvelle approche d'extraction des motifs séquentiels non-dérivables

Chedy Raïssi[*,**], Pascal Poncelet[**]

[*]LIRMM, 161 rue Ada,34392 Montpellier Cedex 5, France
raissi@lirmm.fr,
[**]EMA-LGI2P, Parc Scientifique Georges Besse, 30035 Nîmes Cedex, France
prénom.nom@ema.fr

Résumé. L'extraction de motifs séquentiels est un défi important pour la communauté fouille de données. Même si les représentation condensées ont montré leur intérêt dans le domaine des itemsets, à l'heure actuelle peu de travaux considèrent ce type de représentation pour extraire des motifs. Cet article propose d'établir les premières bases formelles pour obtenir les bornes inférieures et supérieures du support d'une séquence S. Nous démontrons que ces bornes peuvent être dérivées à partir des sous-séquences de S et prouvons que ces règles de dérivation permettent la construction d'une nouvelle représentation condensée de l'ensembles des motifs fréquents. Les différentes expérimentations menées montrent que notre approche offre une meilleure représentation condensée que celles des motifs clos et cela sans perte d'information.

1 Introduction

Motivée par de nombreux domaines d'applications (e.g. marketing web, analyses financières, détections d'anomalies dans les réseaux, traitements de données médicales), l'extraction de motifs séquentiels fréquents est un domaine de recherche très actif Mobasher et al. (2002); Ramirez et al. (2000); Lattner et al. (2005). Les travaux menés ces dernières années ont montré que toutes les approches qui visent à extraire l'ensemble des motifs séquentiels deviennent cependant inefficaces dès que le support minimal spécifié par l'utilisateur est trop bas ou lorsque les données sont fortement corrélées. En effet, dans ce cas, et plus encore que pour les itemsets, les recherches sont pénalisées par un espace de recherche trop important. Par exemple, avec i attributs (appelés aussi *items*), il y a potentiellement $O(i^k)$ séquences fréquentes de taille k Zaki (2001). Pour essayer de gérer au mieux ces problèmes de complexités spatiales et temporelles, deux grandes tendances se distinguent à l'heure actuelle. Dans le premier cas, les propositions comme PrefixSPAN Pei et al. (2004) ou SPADE Zaki (2001) se basent sur de nouvelles structures de données et une génération de candidats efficace. Les approches de la seconde tendance considèrent l'extraction d'une représentation condensée Mannila and Toivonen (1996). Même si l'utilisation d'une représentation compacte a montré son intérêt dans le domaine de l'extraction d'itemsets, la complexité structurelle des motifs séquentiels fait qu'il existe cependant peu de travaux utilisant une représentation condensée dans ce contexte.

Ainsi, seuls Clospan Yan et al. (2003) et Bide Wang et Han (2004) ont abordé ce problème en cherchant à extraire des motifs clos. Le problème que nous cherchons à résoudre dans cet article est le suivant : *Est-il possible de trouver une nouvelle représentation condensée pour répondre à la problématique de l'extraction de motifs séquentiels ?* Notre objectif est d'établir les premières bases formelles pour calculer les bornes supérieures et inférieures de la valeur du support d'un motif en utilisant le principe de l'inclusion-exclusion Knuth (1973). Ce principe nous permet d'obtenir des règles de dérivations via lesquelles nous pouvons déduire le support d'une séquence *sans avoir à compter son support dans la base de données*. Nous montrons également que ces règles peuvent être utilisées pour construire une représentation condensée de certains types de motifs.
Cet article est organisé de la manière suivante. La section 2 introduit les concepts liés aux motifs séquentiels ainsi que les notions formelles utilisées dans le reste de l'article. Nous discutons l'utilisation de règles de déductions dans la section 3. L'approche NDSP est introduite dans la section 4. La section 5 présente les premières expérimentations menées qui confirment l'intérêt de notre approche et en discute les limites. Dans la section 6 nous présentons les travaux connexes autour des représentations condensées et des motifs séquentiels. Enfin, la section 7 conclut et présente les principales perspectives associées à ce travail.

2 Concepts préliminaires

Dans cette section, nous définissons *le problème d'extraction des motifs séquentiels* initialement proposé par Srikant (1995); Srikant et Agrawal (1996) et nous introduisons la notion de *S-Apparition* (une notion similaire est proposée dans Calders et Goethals (2002) pour l'extraction des itemsets fréquents).

Soit $\mathcal{D}$ une base de données contenant des transactions regroupées par client où chaque transaction T consiste en : un identifiant de client, noté C_{id} ; une estampille temporelle, notée $time$ et un ensemble d'items (appelé *itemset*) noté it.

Définition 1 (Sequence, Inclusion et Support) *Soit $\mathcal{I} = \{i_1, i_2 ... i_m\}$ un ensemble fini de littéraux appelés items. Un itemset est un ensemble non-vide d'items. Une séquence S est une liste ordonnée (selon les estampilles temporelles) d'itemsets, notée $< it_1\ it_2\ ...it_n >$, où it_j, $j \in 1...n$, est un itemset. Une k-séquence est une séquence de k items (ou de taille k). Une séquence $S' =< s'_1\ s'_2\ ...\ s'_n >$ est une sous-séquence de $S =< s_1\ s_2\ ...\ s_m >$, notée $S' \preceq S$, si $\exists\ i_1 < i_2 < ...\ i_j\ ... < i_n$ tels que $s'_1 \subseteq s_{i_1}$, $s'_2 \subseteq si_2$,... $s'_n \subseteq si_n$. Si $S \not\preceq S'$ et $S' \not\preceq S$, les séquences sont dites incomparables et sont notées $S \prec\succ S'$. De plus, une séquence est dite régulière si chaque itemset it_j contient le même unique item i. Par exemple, $\langle(a)(a)(a)\rangle$ et $\langle(b)(b)\rangle$ sont des motifs séquentiels réguliers. Soit C_{trans} la liste ordonnée des transactions pour un client C (i.e la séquence maximale supportée par C). Le support d'une séquence S dans une base transactionelle $\mathcal{D}$, noté $Support(S, \mathcal{D})$, est défini tel que : $Support(S, \mathcal{D}) = |\{C \in \mathcal{D} | S \preceq C_{trans}\}|$.*

Définition 2 (Extension de séquence) *Soit S une séquence et α un item, $S \cup \alpha$ veut dire que α est rajouté dans le dernier itemset de la séquence S. De même, $S \diamond \alpha$ définit la concaténation de α dans un nouvel itemset introduit à la fin de la séquence. Par exemple, soit $S = \langle(abcd)(ab)\rangle$, $S \cup c = \langle(abcd)(abc)\rangle$ et $S \diamond c = \langle(abcd)(ab)(c)\rangle$*

Avec ces définitions, nous pouvons maintenant décrire formellement le problème d'extraction des motifs séquentiels et sa solution.

Définition 3 (Problème de l'extraction des motifs séquentiels fréquents) *Soit $\mathcal{SP}$ la séquence maximale théorique pouvant être générée à partir des clients dans $\mathcal{D}$. La solution au problème d'extraction des motifs séquentiels fréquents est définie telle que :*

$$FreqSeqSet(S, \mathcal{D}, \sigma) = \{S \preceq \mathcal{SP} | Support(S, \mathcal{D}) \geq \sigma\}$$

Ou σ est un seuil de support minimal défini par l'utilisateur, $0 \leq \sigma \leq |\mathcal{C}|$ et $\mathcal{C}$ est l'ensemble des clients dans $\mathcal{D}$.

Définition 4 (S-Apparition) *Pour chaque séquence S, $A(S, \mathcal{D})$ représente l'ensemble de clients contenant* **exactement** *dans leurs transactions la séquence S :*

$$A(S, \mathcal{D}) = \{C \in \mathcal{D} | C_{trans} = S\}$$

Exemple 1 *Considérons la base de données $\mathcal{D}$ suivante avec $\mathcal{I} = \{a, b, c, d\}$.*

C_1	T_1	*a,b,c,d*
	T_2	*a,b*
C_2	T_3	*a,b*
C_3	T_1	*a,d*
	T_4	*c*

La séquence maximale $\mathcal{SP}$ qui peut être générée est $\langle(abcd)(abcd)\rangle$. Dans $\mathcal{D}$, $A(\langle(ab)\rangle, \mathcal{D}) = \{C_2\}$, $A(\langle(ad)(c)\rangle, \mathcal{D}) = \{C_3\}$, $A(\langle(abcd)(ab)\rangle, \mathcal{D}) = \{C_1\}$. Pour toutes les autres séquences S, $A(S, \mathcal{D}) = \emptyset$.

Soit $a_S^{\mathcal{SP}}$ le cardinal de l'ensemble $A(S, \mathcal{D})$. La notation a_S sera préferée quand le contexte n'est pas ambigue. A partir de la définition 4, nous pouvons exprimer le support d'une séquence en fonction de $a_S,\ S \prec \mathcal{SP}$.

Lemme 1 *Pour chaque séquence $S \preceq \mathcal{SP}$: $Support(S, \mathcal{D}) = \sum_{S \preceq J \preceq \mathcal{SP}} a_J$*

Par manque de place, les preuves de ce lemme, des prochaines propositions et des théorèmes ne sont pas détaillés dans cet article. Le lecteur peut se référer à Raissi et Poncelet (2006). Le lemme 1 sera utilisé dans la prochaine section afin de générer les règles de déductions.

3 Règles de déductions

Dans cette section, nous étendons la définition d'*expression de support* introduite dans Calders et Goethals (2002) pour l'extraction d'itemsets fréquents. Cette expression sert à modéliser les informations du support dans les séquences. Ces informations sont formalisées en implications logiques et nous soulignons les liens entre ces expressions de support et un système d'équations linéaires.

Définition 5 (Expression de support pour les séquences) *Une expression de support pour une séquence S est une égalité $Support(S) = s$ avec $S \preceq \mathcal{SP}$ et $s \in \mathbb{N}$. Une base de données $\mathcal{D}$ "satisfait" l'expression de support $Support(S) = s$ si et seulement si $Support(S, \mathcal{D}) = s$. Soit $\mathcal{S}$ un ensemble d'expressions de support. Une base de données $\mathcal{D}$ "satisfait" $\mathcal{S}$ si et seulement si toutes les expressions dans $\mathcal{S}$ sont satisfaites.*

$$\mathcal{S} = \left\{ \begin{array}{ccc} Support(\{\}) = 3, & Support(a) = 3, & Support(b) = 2 \\ Support(c) = 2, & Support(d) = 2, & Support(\langle ab \rangle) = 2 \\ Support(\langle (a)(a) \rangle) = 1, & Support(\langle (a)(b) \rangle) = 1, & Support(\langle (a)(c) \rangle) = 1 \\ Support(\langle ac \rangle) = 1, & Support(\langle ad \rangle) = 1, & Support(\langle (b)(b) \rangle) = 1 \\ Support(\langle (b)(a) \rangle) = 1, & Support(\langle bc \rangle) = 1, & Support(\langle bd \rangle) = 1 \\ Support(\langle (c)(a) \rangle) = 1, & Support(\langle (c)(b) \rangle) = 1, & Support(\langle cd \rangle) = 1 \\ Support(\langle (d)(a) \rangle) = 1, & Support(\langle (d)(b) \rangle) = 1, & Support(\langle (d)(c) \rangle) = 1 \end{array} \right\}$$

FIG. 1 – *Expressions de support dans la base de données $\mathcal{D}$ de l'exemple 1*

Ces expressions de support sont présentes dans chaque algorithme de type Apriori utilisé pour l'extraction de motifs séquentiels. Ainsi, l'ensemble des expressions de support s'agrandit après chaque étape de comptage des séquences. Néanmoins seul un sous-ensemble de $\mathcal{S}$ est utilisé pour déduire des contraintes sur les supports des séquences candidates : est-il alors possible d'utiliser tout l'ensemble des expressions de support afin d'obtenir de meilleurs contraintes sur les supports ?

Définition 6 *Extension de Calders et Goethals (2002). Soit $S \preceq \mathcal{SP}$ une séquence et $u, l \in \mathbb{R}$. Soit $\mathcal{S}$ un ensemble d'expressions de support.* $\mathcal{S}$ implique *un intervalle $[l, u]$ sur le support de S, noté $\mathcal{S} \models Support(S) \in [l, u]$ si et seulement si pour chaque transaction de la base de données $\mathcal{D}$ satisfaisant $\mathcal{S}$, on a $l \leq Support(S) \leq u$. Cet intervalle $[l, u]$ est minimal, noté $\mathcal{S} \models_{min} Support(S) \in [l, u]$, si et seulement si $\nexists [l', u'] \subset [l, u] \mid \mathcal{S} \models Support(S) \in [l', u']$.*

Nous montrons maintenant comment générer l'intervalle minimal pour chaque séquence candidate en utilisant l'ensemble des expressions de support, c'est à dire utiliser toute l'information disponible dans l'ensemble des expressions de support. Pour cela, nous présentons les liens théoriques entre l'implication logique définie ci-dessus et un systèmes d'équations linéaires.

Lemme 2 *Soit $\mathcal{S}$ une expression de support pour la séquence S. Il existe une base de données $\mathcal{D}$ qui satisfait $\mathcal{S}$ si et seulement le système d'équation linéaire suivant possède une solution dans $\mathbb{N}$ pour chaque variable x_s avec $S \preceq \mathcal{SP}$:*

$$Sys(\mathcal{S}) = \left\{ \begin{array}{ll} x_S \geq 0 & \forall S \preceq \mathcal{SP} \\ \sum_{S \preceq J \preceq \mathcal{SP}} x_J = s_J & \forall (Support(J) = s_J) \in \mathcal{S} \end{array} \right.$$

En utilisant ce type d'inégalités, nous pouvons définir un théorème permettant de déduire directement des règles sur les bornes de l'intervalle de support des séquences candidates S. Ceci est possible car les algorithmes construits niveaux par niveaux contiennent l'information $Support(J) = s_J$ pour chaque sous-séquence de S.

L'utilisation de la séquence théorique maximale $\mathcal{SP}$ n'est pas possible d'un point de vue pratique. Pour cela, nous limitons la portée de notre lemme à la séquence S uniquement en utilisant un principe de projection de séquence sur la base de données $\mathcal{D}$. Contrairement au problème d'extraction d'itemsets, cette projection modifie le support des sous-séquences de S puisque certaines séquences peuvent être incomparables à S tout en ayant des sous-séquences en commun.

Définition 7 (Projection de séquence) *Soit S une séquence et $\mathcal{I}(S)$ l'ensemble des items contenus dans cette séquence.*

La projection des transactions d'un client sur S, notée $\pi_S(C_{trans})$, est une liste ordonnée de transactions définie comme suit :

- *$\mathcal{I}(\pi_S(C_{trans})) \subseteq \mathcal{I}(S)$ (élimination de tous les items de C_{trans} qui ne sont pas dans S).*
- *If $C_{trans} \prec S,\ \pi_S(C_{trans}) = C_{trans}$.*
- *If $S \preceq C_{trans},\ \pi_S(C_{trans}) = S$.*
- *If $S \prec\succ C_{trans},\ \pi_S(C_{trans}) = C_{trans}$ (si un ensemble de transactions d'un client est incomparable avec la séquence projetée, il est inchangé).*

La projection de la base de données $\mathcal{D}$ sur S, notée $\pi_S(\mathcal{D})$, est définie telle que $\pi_S(\mathcal{D}) = \{\pi_S(C_{trans}) | C \in \mathcal{D}\}$.

Lemme 3 *Soit $\mathcal{D}$ une base de données, pour chaque sous-séquence X de S : $Support(X, \mathcal{D}) = Support(X, \pi_S(\mathcal{D}))$.*

D'après le lemme 2, il existe une variable x_S pour chaque sous-séquence $S \preceq \mathcal{SP}$. Le lemme 3 permet de réduire considérablement le système d'équations $Sys(\mathcal{S})$ associé à l'ensemble des expressions de support $\mathcal{S}$. Donc, avec une projection sur la séquence S, nous pouvons restreindre les variables x_X à uniquement celles dont $X \preceq S$.

Lemme 4 *Soit $\mathcal{S}$ l'ensemble des expressions de support pour la séquence S. Il existe une base de données $\mathcal{D}$ qui satisfait $\mathcal{S}$ si et seulement si le système d'équations suivant possède une solution dans $\mathbb{N}$ pour chaque variable x_X avec $X \preceq S$:*

$$Sys(\mathcal{S}) = \begin{cases} x_X \geq 0 & \forall X \preceq S \\ \sum_{X \preceq J \preceq S} x_J = s_J & \forall (Support(J) = s_J) \in \mathcal{S} \end{cases}$$

Pour résoudre $Sys(\mathcal{S})$ nous séparons les coefficients en une matrice booléenne :

$$\begin{bmatrix} a_1 1 & a_1 2 & \dots & a_1 S \\ a_2 1 & a_2 2 & \dots & a_2 S \\ \vdots & \vdots & \ddots & \vdots \\ a_S 1 & a_S 2 & \dots & a_S S \end{bmatrix} \begin{bmatrix} x_X \\ x_J \\ \vdots \\ x_S \end{bmatrix} = \begin{bmatrix} s_X \\ s_J \\ \vdots \\ s_S \end{bmatrix}$$

si nous représentons chaque matrice par un symbole, $Ax = S$ avec A une matrice de contraintes de taille $n \times n$ (n étant le cardinal de l'ensemble contenant S et toutes ses sous-séquences), x un vecteur colonne avec n entrées et S un vecteur colonne avec n entrées. La solution générale en utilisant la méthode d'élimination de Gauss-Jordan ($A^{-1}.S$) est :

$$x_X = \sum_{X \preceq J \preceq S} (-1)^{|J-X|} \Delta_X^J . s_J \tag{1}$$

où Δ_X^J est la valeur absolue de l'élément xj de la matrice inverse A^{-1}.

Soit s_S un entier choisi arbitrairement, d'après le lemme 4 l'ensemble des supports d'expression $\mathcal{S}$ pour la séquence S est satisfiable si et seulement s'il existe une solution entière au système d'équation $Sys(\mathcal{S})$. Donc d'après (1), $Sys(\mathcal{S})$ est satisfiable si et seulement si :

$$x_X = \sum_{X \preceq J \preceq S} (-1)^{|J-X|} \Delta_X^J . s_J \geq 0, \qquad \forall X \preceq S \tag{2}$$

(Notons que (2) est similaire à la formule d'inclusion-exclusion de Knuth (1973)).

En isolant $(-1)^{|S-X|}\Delta_X^S.s_S$ et s_s de la somme nous avons :

$$(-1)^{|S-X|}\Delta_X^S.s_S \geq -\sum_{X \preceq J \prec S} (-1)^{|J-X|}\Delta_X^J.s_J$$

Comme $\forall X \preceq J \prec S$, nous obtenons :

Théorème 1 *Soit $X \preceq S \preceq \mathcal{SP}$:*

$$Support(S) \leq \frac{\sum_{X \preceq J \prec S}(-1)^{|S-J|+1}\Delta_X^J.s_J}{\Delta_X^S}, \quad si|S-X| \ \ impair. \tag{3}$$

$$Support(S) \geq \frac{\sum_{X \preceq J \prec S}(-1)^{|S-J|+1}\Delta_X^J.s_J}{\Delta_X^S}, \quad si|S-X| \ \ pair. \tag{4}$$

Les règles déduites de (3) sont les bornes supérieures de la valeur du support de la séquence S et les règles déduites de (4) sont les bornes inférieures. Ces règles seront notées $\mathcal{R}_X(S)$ comme dans Calders et Goethals (2002) et la borne, notée $\delta_X(S)$ est définie telle que :

$$\delta_X(S) = \frac{\sum_{X \preceq J \prec S}(-1)^{|S-J|+1}\Delta_X^J.s_J}{\Delta_X^S} \tag{5}$$

Nous pouvons alors, pour chaque séquence S, déduire des règles de chaque sous-séquence $X \preceq S$. Ces règles peuvent être utilisées pour définir un intervalle sur la valeur du support de S avec U_S (respectivement L_S) la valeur minimale des bornes supérieures (respectivement la valeur maximale des bornes inférieures) et donc $L_s \leq Support(S) \leq U_s$.

Exemple 2 *Considérons la base de données $\mathcal{D}$ suivante :*

C_1	T_1	*a,b,c*
	T_2	*a,b,c*
C_2	T_3	*a,b*
C_2	T_6	*a,b*
C_3	T_1	*a,c*
	T_4	*a*

$Support(\langle(a)(b)\rangle)$:

1. $\mathcal{R}_{\{\}}(\langle(a)(b)\rangle)$:
   $Support(\langle(a)(b)\rangle) \geq -Support(\{\}) +$
   $Support(a) + Support(b) = 2$
2. $\mathcal{R}_a(\langle(a)(b)\rangle)$:
   $Support(\langle(a)(b)\rangle) \leq Support(a) = 2$

Remarquons pour cette séquence que nous pouvons directement inférer le support sans avoir à le compter. ($\langle(a)(b)\rangle$ est égal à 2 ($2 \leq Support(\langle(a)(b)\rangle) \leq 2$).

Corollaire 1 *Soit $X \preceq S$, la différence entre la borne δ_X^S et la valeur réelle du support de S multipliée par Δ_X^S est égale au cardinal des S-Apparitions de S.*

$$a_X = \Delta_X^S|s_S - \delta_X(S)| \tag{6}$$

L'extraction des motifs séquentiels est fortement limitée par son aspect combinatoire. Afin de résoudre ce problème, il est souvent nécessaire et plus efficace d'extraire un sous-ensemble de motifs contenant ou pouvant permettre l'extraction des mêmes informations que l'ensemble

des motifs séquentiels. Les règles de déductions peuvent être utilisées afin de construire une nouvelle représentation condensée des motifs séquentiels. Si les règles permettent de dériver *exactement* la valeur du support (i.e. $U_S = L_S$) d'une séquence S en utilisant ces sous-séquences (cf. exemple 2 avec la séquence $(\langle(a)(b)\rangle)$), alors il n'est pas nécessaire de garder S. Dans ce cas, S est appelée une séquence dérivable, notée DS. De la même façon, les séquences non-dérivables, notées NDS, sont les séquences qui ne peuvent pas avoir leur support dérivé de manière exacte. Nous allons montrer que l'ensemble des NDS permet la construction de tout l'ensemble des motifs séquentiels.

Théorème 2 *Soit S une séquence et soit α un item, l'intervalle calculé par les règles de dérivations pour la valeur du support de la séquence $S \cup \alpha$ (respectivement $S \diamond \alpha$) est $2\Delta_X^{S\cup\alpha}$ (respectivement $2\Delta_X^{S\diamond\alpha}$) plus petit que l'intervalle calculé pour la valeur du support de la séquence S.*

Corollaire 2 *Monotonie*

Soit $X \preceq S$ une séquence. Si X est une DS, alors S est aussi une DS.

Preuve. Si X est une DS alors $U_X - L_X = 0$. En utilisant le théorème 2 nous savons que : $U_S - L_S \leq \frac{1}{2\Delta_X^S}(U_X - L_X)$, d'où $U_S = L_S$. □

4 L'algorithme NDSP

Dans cette section nous présentons notre approche afin de construire une représentation condensée des motifs séquentiels à partir des règles de déductions extraite du théorème 1 et du corollaire 2. L'avantage d'une représentation condensée est qu'elle est souvent plus petite que l'ensemble des motifs séquentiels, noté $\mathcal{F}$, ce qui rend cette représentation adéquate dans le cadre d'extractions de motifs à partir de données fortement corrélées ou très denses. Les motifs séquentiels non-dérivables sont donc adéquats pour l'extraction de grands ensembles de motifs séquentiels qui ne pourraient pas être obtenus à l'aide d'algorithmes classiques.

Théorème 3 *Soit $\mathcal{D}$ une base de données, σ un seuil de support minimal. Soit $\mathcal{NDSF}(\mathcal{D}, \sigma)$ un ensemble défini tel que : $\mathcal{NDSF}(\mathcal{D}, \sigma) = \{(S, Support(S))|Support(S) \geq \sigma \ \wedge \ U_S \neq L_S\}$*

$\mathcal{NDSF}(\mathcal{D}, \sigma)$ est une représentation condensée de l'ensemble $\mathcal{F}$. Pour chaque séquence $X \notin \mathcal{NDSF}(\mathcal{D}, \sigma)$ nous pouvons dériver $Support(X)$ à partir de ses sous-séquences contenues dans $\mathcal{NDSF}(\mathcal{D}, \sigma)$.

Preuve. Extension de Calders (2003). La preuve est construite par induction sur la taille de la séquence S (Raissi et Poncelet (2006)).

Dans notre approche, la valeur Δ_X^S utilisée afin de calculer nos règles de déductions n'est pas extraite d'une matrice inversée, afin d'optimiser les calculs, mais calculée par la fonction :

Proposition 1 *Let $X \preceq J \prec S$.*

$$\Delta_X^J = \begin{cases} 1 & if|J - X| < 2 \\ \phi - 1 & if|J - X| \geq 2 \end{cases}$$

Où ϕ est le nombre de sous-séquences de taille $|S - 1|$ tel que $X \preceq S$.

Pour le cas spécial où $J = S$, Si $X = \{\}$ alors $\Delta_{\{\}}^S = 1$.

Corollaire 3 *Soit S une séquence régulière alors S est non-dérivable.*

Preuve. Notons qu'une séquence régulière possède une unique sous-séquence X avec $|S - X| = 1$. A partir de l'équation (5) et (1), si $|S - X| > 1$ alors $\Delta_X^S = \phi - 1 = 0$. donc $\delta_X(S)$ est indéfinie. □

Nous avons developpé un algorithme (NDSP : Non-Derivable Sequential Patterns) qui permet de de construire la représentation condensée $\mathcal{NDSF}(\mathcal{D}, \sigma)$. NDSP est un algorithme par niveau et est complétement indépendant de la structure de données utilisée pour la représentation des séquences. L'algorithme 1 est basé sur la stratégie classique du *générer-élaguer* et est divisée en 3 étapes distinctes : (i) la génération de candidats (ligne 1 et 15 avec la fonction $CandidateGeneration()$) ; (ii) Le comptage de support ligne 5 ; (iii) Déterminer les séquences non-dérivables dans F_{level} grâce à la fonction $ComputeBounds()$ (ligne 8), les séquences dérivables sont élaguées ligne 11. Le processus s'arrête quand il n'y a plus de candidats générés.

Algorithme 1 : algorithme NDSP

Data : Une base de données $\mathcal{D}$; σ un seuil de support minimal
Result : $\mathcal{NDSF}(\mathcal{D}, \sigma)$

1 $\mathcal{NDSF}(\mathcal{D}, \sigma) \leftarrow \emptyset$;
2 $level \leftarrow 1$;
3 $C_1 \leftarrow \{\{i\} | i \in \mathcal{I}\}$;
4 **while** $C_{level} \neq \emptyset$ **do**
5 CountSupport(C_{level});
6 $F_{level} \leftarrow \{S \in C_{level} \mid Support(S) \geq \sigma\}$;
7 **foreach** $S \in F_{level}$ **do**
8 $(L_S, U_S) \leftarrow ComputeBounds(S)$;
9 **if** $L_S = U_S$ *and* $L_S = Support(S)$ **then**
10 `//S est une séquence dérivable.`
11 $F_{level} \leftarrow F_{level} \setminus S$;
12 **else**
13 `//S est une séquence non-dérivable //(`$L_S \neq U_S$`).`
14 $\mathcal{NDSF}(\mathcal{D}, \sigma) \leftarrow \mathcal{NDSF}(\mathcal{D}, \sigma) \cup S$;
15 $C_{level+1} = CandidateGeneration(F_{level})$;
16 $level = level + 1$;
17 return $\mathcal{NDSF}(\mathcal{D}, \sigma)$

$ComputeBounds()$ (Algorithm 2) est appelé par $NDSP$ afin de calculer les bornes supérieures et inférieures pour une séquence S. $ComputeBounds()$ vérifie aussi que les règles restent cohérentes par rapport à la base de données $\mathcal{D}$ (lignes 8 et 13). Le test de cohérence est obligatoire car la projection doit prendre en compte les séquences incomparables qui peuvent amener à des règles du type : $U_S < L_S$ or $U_S = L_S \wedge L_S \neq Support(S)$. La dernière fonction $IE()$ (Algorithm 3), appelée par $ComputeBounds()$, est la fonction qui calcule la formule d'inclusion-exclusion (5) d'une manière exhaustive. De plus, l'évaluation de Δ_X^J est faite dans le corps de la fonction IE.

Algorithme 2 : algorithme COMPUTEBOUNDS

Data : Une séquence S
Result : les bornes sur la valeur du support de la séquence S : (l, u)

```
1
2  foreach X ≺ S do
3      l ← 0;
4      u ← |C|;
5      //Calcul des bornes à partir du théorème //(1).
6      delta ← IE(S, X);
7      if |S − X| est impair then
8          //Test de cohérence.
9          if delta ≥ l then
10             u ← min(u, delta);
11     //|S − X| est impair.
12     else
13         //Test de cohérence.
14         if delta < u then
15             l ← max(l, delta);
16     if l = u then
17         return (l,u);
18 return (l,u);
```

Algorithme 3 : algorithme IE

Data : Les séquences S et X pour calculer $\mathcal{R}_X(S)$
Result : V valeur de la règle $\mathcal{R}_X(S)$

```
1  level ← |X|;
2  V ← (−1)^{|S−X|+1} × Δ_X^X × Support(X);
3  while level < |S| − 1 do
4      SuperSeq ← X.getSuperSequences(level + 1);
5      foreach J ≺ SuperSeq do
6          V ← V + (−1)^{|S−J|+1} × Δ_X^J × Support(J);
7  if |X| ≠ 0 then
8      Δ_X^S = S.getSubContain(X) − 1;
9  else
10     Δ_X^S = 1;
11 return V ← V / Δ_X^S;
```

5 Expérimentations

DataSet	Items	Taille moy. des trans.	# de transactions	# de clients
CL10000TR1000IT1000I10	1000	10	1000	10000
CL1000TR10000IT500I40	500	40	10000	1000
CL5000TR10000IT100I20	100	20	10000	5000

FIG. 2 – *Les différents jeux de données pour les expérimentations de NDSP*

Les expérimentations ont été réalisées sur un ordinateur MacBookPro Core-Du cadencé à 2.16 Ghz avec 1Gb de mémoire avec le système d'exploitation Mac OS X 10.4.6. Nous comparons notre algorithme NDSP avec :

1. Un algorithme d'extraction de motifs séquentiels : PrefixSpan Pei et al. (2004).
2. Un algorithme d'extraction de motifs séquentiels clos (représentation condensée), noté $\mathcal{C}los\mathcal{F}$: CloSpan Yan et al. (2003).

Les tests ont été faits sur plusieurs jeux de données synthétiques générés avec l'outil *DatGen*[1] qui est une extension de l'outil IBM QUEST. Les différentes caractéristiques des jeux de données sont représentées dans la figure 2. Les tests se concentrent principalement sur les performances au niveau de la représentation condensée. L'algorithme NDSP a été implementé en JAVA et utilise une structure de données arborescente pour le stockage des séquences et des supports. La figure 3 montre les résultats d'extraction et les performances pour les 2 premiers jeux de données. Pour $0.05 \leq \sigma \leq 0.3$, NDSP dépasse CloSpan et largement PrefixSpan au niveau de la condensation des motifs séquentiels extraits. De plus, pour $\sigma = 0.1$, le nombre des motifs séquentiels non-dérivables décroît plus rapidement que les deux autres approches. L'extraction s'arrête pour l'algorithme NDSP au niveau de profondeur 6 alors que les deux autres algorithmes s'arrêtent à la profondeur 8. Pour CL1000TR10000IT500I40, NDSP teste beaucoup moins de séquences candidates que PrefixSpan ou CloSpan. De plus, le nombre de motifs séquentiels non-dérivables tend à décroître plus rapidement avec le jeu de données CL10000TR1000IT1000I10. En effet ceci est dû principalement à la taille des séquences puisque le jeu de données CL1000TR10000IT500I40 contient un ensemble de motifs séquentiels très long avec très peu d'items différents sachant que les séquences longues ont plus de chance d'être dérivables puisqu'elles contiennent plus d'informations. NDSP réalise donc un meilleur taux de compression avec des données denses contenant de longues séquences. Le reste des jeux de données est documenté dans Raissi et Poncelet (2006).

6 Etat de l'art

Ces dernières années, de nombreuses recherches se sont intéressées à des représentations condensées pour les itemsets. Les représentations concises les plus importantes sont les *itemsets fréquents clos* Boulicaut et Bykowski (2000) qui sont basés sur l'opérateur de fermeture sur le treillis. De nombreux algorithmes ont été développés comme CLOSET Pei et al.

[1] http ://www.datasetgenerator.com/

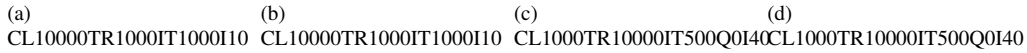

FIG. 3 – *Taux de compression et nombres de séquences fréquentes pour différentes profondeur et supports*

(2000) et CHARM Zaki et Hsiao (2002) qui utilisent une stratégie de profondeur d'abord et considèrent que la base peut être chargée en mémoire. Les *ensembles fréquents libres* Boulicaut et al. (2003) considèrent qu'un itemset I est dit libre si et seulement si $\forall X \subset I$, $Support(X) \neq Support(I)$. Cette propriété est antimonotone ce qui la rend bien adaptée à une représentation concise. Enfin les itemsets non dérivables proposés par Calders et Goethals (2002); Calders (2003)cherche à détecter les redondances dans l'ensemble de tous les itemsets fréquents à l'aide de règles de déduction basées sur le principe d'inclusion-exclusion. La propriété de non dérivabilité étant non monotone, elle permet une représentation concise. Dans le cas des motifs séquentiels, il n'existe à notre connaissance que les algorithmes CloSpanYan et al. (2003) et Bide Wang et Han (2004) qui s'intéressent à une représentation condensée. Cependant Bide ne considère que des séquences dont les itemsets sont réduits à un seul item.

7 Conclusion

Dans cet article, nous avons abordé la problématique des représentations condensées pour les motifs séquentiels.Nos contributions principales sont les suivantes : Premièrement, nous avons jetés les bases formelles pour une nouvelle représentation. Nous introduisons les concepts théoriques des motifs séquentiels non-dérivables et prouvons que les bornes sur la valeur du support d'une séquence peuvent être déduites à partir de règles. Celles-ci sont calculées grâce au principe d'inclusion-exclusion. A notre connaissance, ce travail est le premier travail à introduire une nouvelle représentation condensée autre que la représentation close des motifs séquentiels. Deuxièmement nous avons developpé un algorithme NDSP qui dépasse, en terme de taux de compressions, les algorithmes actuels d'extractions de motifs séquentiels et de motifs séquentiels clos. Ce travail offre de nombreuses perspectives. Tout d'abord, le lemme 4 doit être affiné afin de prouver la complétude et l'adéquation de notre méthode, permettant ainsi l'extraction de motifs non-dérivables sans avoir à tester les cohérences des règles et passer par l'actuelle étape de comptage. De plus, cette approche peut-être couplée à des algorithmes très efficaces comme SPADE, SPAM ou PrefixSPAN. Ce couplage permettrait d'augmenter la vitesse d'extraction puisque ces algorithmes utilisent des structures de données déjà optimi-

sées pour la génération de candidats et leur élagage. La dernière perspective serait d'étendre la théorie de la non-dérivabilité vers d'autres motifs tels que les arbres et les graphes.

Références

Boulicaut, J.-F. et A. Bykowski (2000). Frequent closures as a concise representation for binary data mining. In T. Terano, H. Liu, et A. L. P. Chen (Eds.), *PAKDD*, LNCS, pp. 62–73.

Boulicaut, J.-F., A. Bykowski, et C. Rigotti (2003). Free-sets : A condensed representation of boolean data for the approximation of frequency queries. *Data Min. Knowl. Discov. 7*(1), 5–22.

Calders, T. (2003). *Axiomatization and Deduction Rules for the Frequency of Itemsets*. Ph. D. thesis, University of Antwerp, Belgium.

Calders, T. et B. Goethals (2002). Mining all non-derivable frequent itemsets. In *PKDD*, pp. 74–85. Springer.

Knuth, D. E. (1973). *Fundamental Algorithms* Addison-Wesley.

Lattner, A. D., A. Miene, U. Visser, et O. Herzog (2005). Sequential pattern mining for situation and behavior prediction in simulated robotic soccer. In *RoboCup International Symposium 2005*.

Mannila, H. et H. Toivonen (1996). Multiple uses of frequent sets and condensed representations (extended abstract). In *Knowledge Discovery and Data Mining*, pp. 189–194.

Mobasher, B., H. Dai, T. Luo, et M. Nakagawa (2002). Using sequential and non-sequential patterns in predictive web usage mining tasks. In *ICDM*, pp. 669–672.

Pei, J., J. Han, et R. Mao (2000). CLOSET : An efficient algorithm for mining frequent closed itemsets. In *ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, pp. 21–30.

Pei, J., J. Han, B. Mortazavi-Asl, J. Wang, H. Pinto, Q. Chen, U. Dayal, et M. Hsu (2004). Mining sequential patterns by pattern-growth : The prefixspan approach. *IEEE Trans. Knowl. Data Eng. 16*(11), 1424–1440.

Raissi, C. et P. Poncelet (2006). Mining non-derivable sequential patterns. Technical Report 06037, University of Montpellier 2, LIRMM.

Ramirez, J. C. G., D. J. Cook, L. L. Peterson, et D. M. Peterson (2000). An event set approach to sequence discovery in medical data. *Intell. Data Anal. 4*(6), 513–530.

Srikant, R. et R. Agrawal (1996). Mining sequential patterns : Generalizations and performance improvements. In *EDBT*.

Srikant, R. A. R. (1995). Mining sequential patterns. In *Proceedings of the 11th International Conference on Data Engineering (ICDE 95)*, Tapei, Taiwan, pp. 3–14.

Wang, J. et J. Han (2004). Bide : Efficient mining of frequent closed sequences. In D. Barbará et C. Kamath (Eds.), *ICDE*, pp. 79–90. IEEE Computer Society.

Yan, X., J. Han, et R. Afshar (2003). Clospan : Mining closed sequential patterns in large databases. In D. Barbará et C. Kamath (Eds.), *SDM*. SIAM.

Zaki, M. J. (2001). Spade : An efficient algorithm for mining frequent sequences. *Machine Learning 42*(1/2), 31–60.

Zaki, M. J. et C.-J. Hsiao (2002). Charm : An efficient algorithm for closed itemset mining. In R. L. Grossman, J. Han, V. Kumar, H. Mannila, et R. Motwani (Eds.), *SDM*. SIAM.

Summary

Mining Sequential Patterns is one of the main challenges in data mining. In this paper, we establish the first basis for theoretical upper and lower bounds on the support of a candidate sequential pattern S. We show how these bounds can be derived from S sub-sequences. These rules allow the construction of a concise representation of the frequent sequential patterns. We give the results of experiments and show that our proposal produces a better concise representation than the closed collection while keeping the same expression information.