Extraction de Motifs Séquentiels Multidimensionnels Clos sans Gestion d'Ensemble de Candidats

Marc Plantevit, Anne Laurent, Maguelonne Teisseire

Extraction de Motifs Séquentiels Multidimensionnels Clos sans Gestion d'Ensemble de Candidats

Marc Plantevit, Anne Laurent, Maguelonne Teisseire

LIRMM, Université Montpellier 2, CNRS,
161 Rue Ada 34392 Montpellier, France
nom.prenom@lirmm.fr

Résumé. L'extraction de motifs séquentiels permet de découvrir des corrélations entre événements au cours du temps. En introduisant plusieurs dimensions d'analyse, les motifs séquentiels multidimensionnels permettent de découvrir des motifs plus pertinents. Mais dans ce contexte, le nombre de motifs obtenus peut devenir très important. C'est pourquoi nous proposons, dans cet article, de définir une représentation condensée garantie sans perte d'information : les motifs séquentiels multidimensionnels clos. Nous présentons également des algorithmes, pour l'extraction de tels motifs, sans gestion d'ensemble de candidats. Les expérimentations menées aussi bien sur des données réelles que sur des données synthétiques soulignent l'intérêt de notre proposition.

1 Introduction

Les motifs séquentiels sont étudiés depuis plus de 10 ans (Agrawal et Srikant (1995)). Ils ont donné lieu à de nombreuses applications (*e.g.* comportement des utilisateurs, extraction de motifs à partir des séquences de protéines, détection de fraudes, musique, etc.). Des algorithmes ont été proposés, basés sur le principe d'Apriori (Masseglia et al. (1998); Zaki (2001); Ayres et al. (2002)) ou sur d'autres propositions (Han et al. (2000); Pei et al. (2004)). Récemment, les motifs séquentiels ont été étendus aux motifs séquentiels multidimensionnels par Pinto et al. (2001), Plantevit et al. (2005), et Yu et Chen (2005) dans l'objectif de prendre en compte plusieurs dimensions d'analyse. Par exemple, dans Plantevit et al. (2005), les règles telles que *Un client qui achète une planche de surf avec un sac à NY achète plus tard une combinaison à SF* sont découvertes.

Dans le contexte classique de l'extraction de motifs séquentiels, les ensembles d'analyse, de référence, et temporel sont des singletons (*e.g. produits*, *customer_id* et *date*). Toutefois, le nombre de motifs extraits dans une base de données peut être très important. C'est pourquoi des représentations condensées telles que les motifs *clos* ont été proposées pour l'extraction des itemsets (Pasquier et al. (1999); Pei et al. (2000); Zaki et Hsiao (2002); El-Hajj et Zaïane (2005)) et des séquences (Yan et al. (2003); Wang et Han (2004)). Les clos permettent de disposer à la fois d'une représentation condensée des connaissances extraites et d'un mécanisme d'extraction plus efficace afin d'élaguer significativement l'espace de recherche. Néanmoins, ces propositions ne peuvent pas être directement appliquées aux motifs séquentiels multidimensionnels pour la raison suivante : une super séquence peut être obtenue de deux façons (1)

une plus longue séquence (plus d'items) ou (2) une séquence plus générale (plus de valeurs non spécifiées) ce qui modifie les définitions des méthodes précédemment introduites.

Notre contribution majeure est la définition d'un cadre théorique pour l'extraction de motifs séquentiels multidimensionnels clos ainsi qu'un algorithme permettant de rechercher de tels motifs. Nous adoptons une méthode basée sur le paradigme "pattern growth" (Pei et al. (2004)) afin de proposer une solution d'extraction de motifs séquentiels multidimensionnels clos efficace. De plus, nous souhaitons définir un algorithme qui se dispense de gérer un ensemble de clos candidats, seules les séquences closes étant ajoutées à l'ensembles des clos.

La suite de cet article est organisée de la façon suivante. Tout d'abord, nous proposons, Section 2, une formalisation du problème de l'extraction des motifs séquentiels multidimensionnels clos. Dans la Section 3, nous présentons notre proposition $CMSP$ pour l'extraction de motifs séquentiels multidimensionnels clos. Les expérimentations menées sur des données synthétiques et réelles, détaillées Section 4, soulignent la pertinence de notre proposition aussi bien pour les temps de réponse que pour le nombre de séquences closes par rapport au nombre de séquences fréquentes. Nous situons ensuite notre proposition par rapport aux travaux existants.

2 Motivations et Problématique

Dans cette section, nous définissons le cadre théorique de l'extraction des motifs séquentiels multidimensionnels clos. Pour cela, il est nécessaire de définir au préalable les motifs séquentiels multidimensionnels.

2.1 Motifs Séquentiels Multidimensionnels

Comme dans Plantevit et al. (2005), nous considérons une base de données DB définie sur un ensemble de n dimensions D partitionné en quatre sous-ensembles : (i) D_t pour les dimensions dites temporelles, l'ensemble des dimensions qui permet d'introduire une relation d'ordre entre les événements (e.g. temps) ; (ii) D_A pour les dimensions d'analyse, l'ensemble des dimensions sur lesquelles les motifs vont être extraits ; (iii) D_R pour les dimensions de référence, l'ensemble des dimensions qui va permettre de déterminer le support d'une séquence et donc sa fréquence ; (iv) D_F pour les dimensions ignorées, l'ensemble des dimensions qui ne seront pas prises en compte lors de l'extraction des motifs séquentiels multidimensionnels. Chaque n-uplet $c = (d_1, \ldots, d_n)$ peut s'écrire sous la forme d'un quadruplet $c = (f, r, a, t)$ où f, r, a et t sont respectivement les restrictions de c sur D_F, D_R, D_A et D_t.

Etant donnée une base de données DB, on appelle *bloc* l'ensemble des n-uplets qui partagent la même valeur r sur D_R. On note B_{DB,D_R}, l'ensemble des blocs de DB par rapport aux dimensions de référence D_R. Chaque bloc B de B_{DB,D_R} est identifié par un n-uplet r qui le définit.

Etant donnée la partition des dimensions de la base de données DB, un *item multidimensionnel* e est un m-uplet défini sur l'ensemble D_A des dimensions d'analyse : $e = (d_1, d_2, \ldots, d_m)$ où $d_i \in Dom(D_i) \cup \{*\}, \forall D_i \in D_A$ et $*$ joue le rôle de valeur joker (une valeur sur une dimension non spécifiée). Par exemple, (a_1, b_1, c_1) et $(a_1, *, c_2)$ sont des items multidimensionnels définis sur trois dimensions d'analyse. Un *itemset multidimensionnel* $i = \{e_1, \ldots, e_k\}$ est un ensemble non-vide d'items multidimensionnels alors qu'une *séquence multidimensionnelle* $\varsigma = \langle i_1, i_2, \ldots, i_l \rangle$ est une liste non-vide d'itemsets multidimensionnels. $\langle \{(a_1, b_1, *), (a_2, *, c_2)\}\{(*, b_2, *)\} \rangle$ est une séquence multidimensionnelle composée de deux itemsets. Une séquence multidimensionnelle peut être incluse dans une autre :

Définition 1 (Inclusion de séquence)**.** *Une séquence multidimensionnelle* $s = \langle a_1, a_2, \ldots, a_l \rangle$ *est une sous-séquence de* $s' = \langle b_1, b_2, \ldots, b_{l'} \rangle$ *s'il existe des entiers* $1 \leq j_1 \leq j_2 \leq \ldots \leq j_l \leq l'$ *tel que* $a_1 \subseteq b_{j_1}, a_2 \subseteq b_{j_2}, \ldots, a_l \subseteq b_{j_l}$.

Par exemple, $\langle \{(a_1, b_1, *), (a_2, b_2, c_2\}\{(*, b_2, c_2)\}\rangle$ est une sous-séquence de $\langle \{(a_1, *, *)$,-$(a_2, *, c_2\}\{(*, b_2, *)\}\{(*, *, c_1)\}\rangle$. Chaque bloc défini sur D_R identifie une séquence multidimensionnelle de données. Un bloc *supporte* une séquence s si s est une sous-séquence de la séquence de données identifiée par ce bloc. Ainsi le *support* d'une séquence est le nombre de blocs définis sur D_R qui supportent cette séquence. Etant donné un seuil de support fixé a priori σ, le but de l'extraction des motifs séquentiels multidimensionnels est d'extraire toutes les séquences multidimensionnelles dont le support est supérieur à σ.

2.2 Motifs Séquentiels Multidimensionnels Clos

Défini par Pasquier et al. (1999), un *motif clos* est un motif qui n'a pas le même support que tous ses super-motifs. Les motifs clos permettent de représenter les connaissances extraites de manière compacte sans perte d'information et sont généralement associés à des propriétés qui permettent de réduire sensiblement l'espace de recherche à l'aide d'opérations d'élagage autres que l'élagage élémentaire des motifs infréquents.

Dans un contexte multidimensionnel, une séquence peut être plus spécifique qu'une autre si elle contient plus d'items (séquence plus longue), ou si elle contient des items plus spécifiques (moins de valeurs *).

Définition 2 (Spécialisation/Généralisation)**.** *Un motif séquentiel multidimensionnel* $\alpha = \langle a_1$-$, a_2, \ldots, a_l \rangle$ *est plus général que* $\beta = \langle b_1, b_2, \ldots, b_{l'} \rangle$ *(*$l \leq l'$*) (et* β *plus spécifique que* α*) s'il existe des entiers* $1 \leq j_1 \leq j_2 \leq j_l \leq l'$ *tels que* $b_{j_1} \subseteq a_1, b_{j_2} \subset a_2, \ldots, b_{j_l} \subseteq a_l$.

Si β est plus spécifique que α, nous notons $\alpha \subset_S \beta$ où $\subset_S$ représente la relation de spécialisation. Soient $s_1 = \langle \{(a_1, b_1, c_1), (a_2, *, c_1)\}\{(*, b_2, c_2)\}\rangle$, $s_2 = \langle \{(a_1, *, *), (a_2, *, c_1)\}$-$\{(*, b_2, c_2)\}\rangle$ et $s_3 = \langle \{(a_1, b_1, c_1)\}\{(*, b_2, c_2)\}\rangle$ trois séquences multidimensionnelles. On a $s_2 \subset_S s_1$ et $s_3 \subset_S s_1$. A partir de cette définition, nous pouvons définir une séquence multidimensionnelle close.

Définition 3 (Motif Séquentiel Multidimensionnel Clos)**.** *Une séquence multidimensionnelle* α *est close s'il n'existe pas de séquence* β *telle que* $\alpha \subset_S \beta$ *et* $support(\alpha) = support(\beta)$.

La problématique générale de l'extraction de motifs séquentiels multidimensionnels clos est la suivante : *Etant donné un seuil de support fixé a priori* σ*, le but est d'extraire toutes les séquences multidimensionnelles closes dont le support est supérieur à* σ.

La résolution de ce problème dans un contexte multidimensionnel pose de nombreuses difficultés. Nous allons détailler dans la section suivante les problèmes ainsi que les solutions proposées.

3 CMSP : Extraction de motifs séquentiels multidimensionnels clos sans gestion d'ensemble candidats

Dans cette section, nous présentons successivement : (1) l'adaptation du paradigme "pattern growth" dans un contexte multidimensionnel ; (2) les définitions et les propriétés associées

aux séquences closes dans ce paradigme ; (3) les propriétés permettant un élagage supplémentaire de l'espace de recherche ; et enfin (4) les algorithmes permettant la mise en œuvre de notre approche. De façon plus précise, nous introduisons le problème inhérent à l'absence d'ordre dans les itemsets dans un contexte multidimensionnel. Nous présentons ensuite les propriétés permettant de définir si une séquence est close selon une approche pattern growth ainsi que des propriétés permettant d'élaguer efficacement l'espace de recherche.

3.1 Approche "pattern growth" et ordre dans la séquence

Les méthodes basées sur le paradigme générer-élaguer ne sont pas adaptées à notre contexte multidimensionnel puisque le nombre de combinaisons possibles entre les items est trop important pour pouvoir garantir un passage à l'échelle. Le paradigme "pattern growth" introduit par Pei et al. (2004) permet d'extraire les séquences fréquentes de manière gloutonne en parcourant en profondeur l'espace de recherche. Ainsi, l'extraction des motifs se fait en concaténant à la séquence traitée (préfixe) les items fréquents sur la base de données projetée par rapport à cette séquence préfixe. Nous utilisons le terme de g-k-séquence pour les séquences composées de k items au sein de g itemsets.

Définition 4 (g-k-Séquence)**.** *Une g-k-séquence S est une séquence composée de g itemsets et de k items de la forme :*
$S = \langle\{e_1^1, e_2^1, \ldots, e_{k_1}^1\}, \{e_1^2, e_2^2, \ldots, e_{k_2}^2\}, \ldots, \{e_1^g, e_2^g, \ldots, e_{k_g}^g\}\rangle$ *où* $\sum_1^g(k_i) = k$.

La séquence $\langle\{(a_1, b_1, *), (a_2, b_2, c_2\}\{(*, b_2, c_2)\}\rangle$ *est une* 2-3*-séquence.*

Lorsqu'on considère des séquences d'itemsets, l'opération de concaténation peut s'effectuer de deux façons différentes :

- concaténation inter itemset où l'item est inséré dans un nouvel itemset (le $(g+1)^{ème}$ itemset de la séquence) : $S' = s_1, s_2, \ldots s_g, \{e'\}$.
- concaténation intra itemset où l'item est inséré dans le dernier itemset de la séquence (le $g^{ème}$ itemset de la séquence) : $S' = s_1, s_2, \ldots s_g \cup \{e'\}$.

Ordonner les items au sein des itemsets est un des moyens d'améliorer le processus d'extraction en éliminant de façon efficace des cas déjà examinés. La valeur joker * n'existe pas comme valeur réelle dans la base de données. Cette valeur est un méta-symbole qui est inféré à partir des valeurs réellement présentes dans la base de données. Ainsi, les solutions proposées dans un contexte classique par Yan et al. (2003) (CloSpan) et Wang et Han (2004) (BIDE) ne sont pas directement applicables au contexte multidimentionnel avec valeur joker. Nous illustrons ceci à partir des deux séquences de données présentes dans le Tableau 1.

1	$\langle\{(a_1, b_1), (a_1, b_2)\}\rangle$
2	$\langle\{(a_1, b_2), (a_2, b_1)\}\rangle$

TAB. 1 – *Contre exemple : ordre dans les itemsets*

Puisque la valeur joker n'est pas explicitement présente dans les n-uplets, il n'est pas possible de définir un ordre lexicographique total. Ainsi pour les méthodes indiquées précédemment, il n'est pas possible d'obtenir la séquence $\langle\{(a_1, b_2), (*, b_1)\}\rangle$. CloSpan extrait l'item (a_1, b_2) avec un support de 2 et construit ensuite la base projetée à partir de la séquence

$\langle\{(a_1, b_2)\}\rangle$ qui contient les séquences $\langle\{\}\rangle$ et $\langle\{(a_2, b_1)\}\rangle$. L'item $(*, b_1)$ n'apparaîtra donc pas comme fréquent dans cette base projetée alors qu'il l'est dans la base initiale. Ce contre-exemple trivial illustre bien qu'il est nécessaire d'ordonner les séquences en prenant en compte le caractère joker (*) comme valeur de dimension possible pour les items.

Nous ne souhaitons pas réaliser cette prise en compte par un pré-traitement sur la base de données par extension à l'ensemble des n-uplets contenant la valeur joker. Car, en considérant une base avec m dimensions d'analyse et n_i items dans un itemset i, ceci produirait $(2^m - 1) \times n_i$ items et une base d'une taille de $(2^m - 1) \sum_{t_i \in DB} n_{t_i}$. Nous souhaitons traiter cette particularité à la volée *pendant* le processus d'extraction de motifs séquentiels multidimensionnels clos. C'est pourquoi nous introduisons un ordre lexical et les fonctions associées afin de gérer les items avec valeur joker à la demande. L'ordre *lexico-graphico-specifique*(LGS) est un ordre alpha-numérique par rapport au degré de précision des items (nombre de * dans l'item). La priorité est ainsi donnée aux items les plus spécifiques dans le processus d'extraction. Nous tentons de matérialiser localement cet ordre au sein de chaque transaction à l'aide d'une fonction $LGS\text{-}Closure$ qui est une application d'un itemset i vers la fermeture de i en respectant l'ordre LGS $<_{lgs}$. Par exemple, $LGS\text{-}Closure(\{(a_1, b_1), (a_2, b_1)\}) = \{(a_1, b_1)$,-$(a_2, b_1), (a_1, *), (a_2, *), (*, b_1)\}$. L'extraction des items fréquents s'effectue ainsi sur chaque itemset dans lequel l'ordre LGS est matérialisé. Comme il existe deux types de concaténation applicable à la séquence préfixe, il est important, pour les concaténations intra itemsets, de définir des restrictions afin de filtrer les items déjà présents dans le dernier itemset de la séquence préfixe. Ainsi, on définit la fonction LGS-Closure$_X$ où X représente le dernier itemset de la séquence préfixe.

3.2 Extensions et clos

Actuellement, la plupart des algorithmes d'extraction de motifs clos ont besoin de maintenir l'ensemble des clos (ou juste candidats) en mémoire et vérifier en post traitement si un motif peut être absorbé ou non par un autre motif. Mais la maintenance d'un tel ensemble est très coûteuse (quadratique en la taille de l'ensemble des clos candidats), c'est pourquoi notre objectif est d'éviter une telle gestion.

D'après la définition d'un motif séquentiel multidimensionnel clos, si une g-k-séquence $S = s_1, \ldots, s_g$ n'est pas close alors il existe une séquence S' de même support telle que $S \subset_S S'$. La définition 5 présente les cinq différents types de construction d'une séquence plus spécifique à partir d'une séquence préfixe.

Définition 5. *Une séquence plus spécifique peut être construite de cinq façons différentes à partir d'une g-k-séquence préfixe $\langle s_1, s_2, \ldots, s_g\rangle$: (i) extension vers l'avant inter itemset $S' = \langle s_1, s_2, \ldots, s_g, \{e'\}\rangle$; (ii) extension vers l'avant intra itemset $S' = \langle s_1, s_2, \ldots s_g \cup \{e'\}\rangle$; (iii) extension vers l'arrière inter itemset $S' = \langle s_1, s_2, \ldots, s_i, \{e'\}, s_{i+1}, \ldots, s_g\rangle$; (iv) extension vers l'arrière intra itemset $S' = \langle s_1, s_2, \ldots s_i \cup \{e'\}, s_{i+1}, \ldots, s_g\rangle$; et (v) spécialisation d'un item si $\exists i \in \{1, \ldots, g\}, \exists e, \exists e'$ tq $e \subset_S e'$: $S' = \langle s_1, s_2, \ldots, s_{i-1}, s_i[e'/e], s_{i+1}, \ldots, s_g\rangle$ où $s_i[e'/e]$ correspond à la substitution de e par e' dans s_i.*

Nous verrons que le dernier point peut être facilement détecté grâce à l'ordre de parcours dès lors que les précédents le sont.

Théorème 1 (Extension bi-directionnelle)**.** *Une séquence S est close si et seulement si elle n'accepte aucune extension vers l'avant, extension vers l'arrière, et spécialisation.*

La démonstration découle trivialement de la définition même des motifs clos (Déf. 3).

A partir du théorème 1, nous savons que pour déterminer si une séquence préfixe est close, nous devons vérifier si elle ne peut pas avoir d'extension vers l'avant ou vers l'arrière ainsi que de spécialisation d'item. Il est relativement facile de trouver des extensions vers l'avant à partir du lemme suivant.

Lemme 1 (Extension vers l'avant)**.** *Pour une séquence S, l'ensemble complet des extensions vers l'avant est équivalent à l'ensemble des items localement fréquents sur la base projetée par rapport à S ayant un support égal à $support(S)$.*

Démonstration. Les items localement fréquents sont extraits en parcourant la base de données projetée par rapport à la séquence préfixe S_p. Puisque chaque événement apparaît pendant ou après la séquence préfixe S_p, s'il existe dans toutes les séquences de données projetées de la base de données, alors il forme une extension vers l'avant. Tout événement apparaissant après la première instance de S_p est inclus dans la base de données projetée, ce qui signifie que l'ensemble complet des extensions vers l'avant peut être extrait en parcourant la base de données projetée par rapport à S_p. □

Pour les extensions vers l'arrière, la recherche d'extension est moins triviale. En effet, une extension vers l'arrière peut être réalisée de deux façons différentes :

- $S' = s_1, s_2, \ldots, s_i, \{e'\}, s_{i+1}, \ldots, s_g$
- $S' = s_1, s_2, \ldots s_i \cup \{e'\}, s_{i+1}, \ldots, s_g$

Soit un item s'insère dans un nouvel itemset, entre deux itemset s_i et s_{i+1} existants. Soit il s'insère dans un itemset existant. On peut caractériser ces insertions vers l'arrière par des insertions vers l'arrière *inter*-itemsets ou *intra*-itemsets.

Comme une séquence peut se répéter plusieurs fois à l'intérieur d'une séquence de données, on peut identifier g intervalles pour localiser les possibles insertions vers l'arrière d'une g-k-séquence. La figure Fig. 1 représente ces g intervalles.

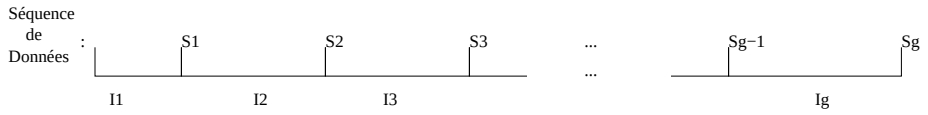


FIG. 1 – *Les différents intervalles d'insertion possibles pour les extensions vers l'arrière d'une g-k-séquence préfixe $S_p = \langle s_1, s_2, \ldots, s_g \rangle$ au sein d'une séquence de données*

Il faut maximiser ces intervalles afin de détecter toutes les extensions possibles vers l'arrière.

Définition 6 ($i^{ème}$ Intervalle maximal)**.** *Etant données une g-k-séquence préfixe $S_p = \langle s_1, s_2, \ldots, s_g \rangle$ et une séquence de données S, le $i^{ème}$ intervalle maximal se définit de la façon suivante :*

- *pour $i = 1$: la sous-séquence du début de S jusqu'à strictement avant $da(s_1)$ la dernière apparition de s_1 dans S telle que $da(s_1) < da(s_2) < \ldots < da(g)$*

– *pour* $1 < i \leq g$ *: la sous-séquence entre la première apparition de la séquence* $\langle s_1, s_2,$-$\ldots, s_{i-1}\rangle$ *notée* $pa(\langle s_1, s_2, \ldots, s_{i-1}\rangle)$ *et strictement avant* $da(s_i)$ *telle que* $da(s_i) < da(s_{i+1}) < \ldots < da(g)$

Lemme 2 (Vérification d'une extension vers l'arrière)**.** *Soit la* g*-*k*-séquence préfixe* $S_p = \langle s_1, s_2, \ldots, s_g\rangle$*, s'il existe un entier* i *tel qu'il existe un item* e *qui apparaît dans chacun des* $i^{èmes}$ *intervalles maximaux de la séquence de préfixe* S_p *dans* DB*, alors* e *est une extension vers l'arrière.*

Autrement, si nous ne pouvons pas exhiber d'item e *qui apparaisse dans chacun des* $i^{ème}$ *intervalles maximaux, alors il ne peut pas y avoir d'extension vers l'arrière de la séquence préfixe* S_p *dans la base de données* DB*.*

Démonstration. Soit la g-k-séquence préfixe $S_p = \langle s_1, s_2, \ldots, s_g\rangle$. S'il existe un item e qui apparaît dans chacun des $i^{èmes}$ intervalles maximaux de la séquence de préfixe S_p dans DB, alors on peut construire la séquence $S' = \langle s_1, s_2, \ldots, s_{i-1} \cup \{e\}, s_i, \ldots, s_g\rangle$ ou $S' = \langle s_1, s_2, \ldots, s_{i-1}, \{e\}, s_i, \ldots, s_g\rangle$ telle que $S_p \subset_S S'$ et $supp(S') = supp(S_p)$. Donc e est une extension vers l'arrière.

Supposons qu'il existe une séquence $S'_p = \langle s_1, s_2, \ldots, s_{i-1} \cup \{e\}, s_i, \ldots, s_g\rangle$ ou $S'_p = \langle s_1, s_2, \ldots, s_{i-1}, \{e\}, s_i, \ldots, s_g\rangle$ qui absorbe S_p. Dans chaque séquence de données de DB contenant S_p, l'item e' doit apparaître après la première apparition de $\langle s_1, \ldots, s_{i-1}\rangle$ et avant la dernière apparition de la sous-séquence $\langle s_i, \ldots, s_g\rangle$, ce qui signifie que e doit apparaître dans chacun des $i^{èmes}$ intervalles maximaux de S_p dans DB. Ainsi, si nous ne pouvons pas exhiber d'items apparaissant dans chaque $i^{ème}$ intervalle de S_p alors il ne peut pas y avoir d'extension vers l'arrière.

□

Une séquence préfixe ne peut pas être close s'il existe une spécialisation d'un item de la séquence préfixe. L'ordre LGS, que nous adoptons, nous permet d'extraire les séquences closes en commençant par celles qui contiennent les items les plus spécifiques (le moins de valeurs $*$). Ainsi, s'il existe une spécialisation possible d'une séquence préfixe considérée, alors la "séquence spécialisée", qui contient au moins un item plus spécifique, sera déjà présente dans l'ensemble des clos déjà extraits. Ainsi, si une séquence est potentiellement close (pas d'extensions vers l'avant ou l'arrière), il suffit de vérifier qu'il n'existe pas de séquence plus spécifique dans l'ensemble des séquences closes déjà extraites. On peut noter que cet ensemble est sensiblement plus petit que l'ensemble des séquences fréquentes. Cette opération de vérification n'est donc pas trop coûteuse. Dans le pire des cas, on doit considérer toutes les séquences closes déjà extraites dont le support est égal à la séquence préfixe examinée.

3.3 Elagage de l'espace de recherche

Tout en recherchant les nouvelles séquences fréquentes avec l'algorithme d'énumération des séquences, nous pouvons utiliser la propriété de fermeture bidirectionnelle pour vérifier si la séquence est close dans le but de générer un ensemble non redondant de connaissances. Bien que la propriété de fermeture retourne un ensemble plus compact, cela ne permet pas d'extraire les séquences plus efficacement. Par exemple, il peut n'y avoir aucun clos au delà d'un certain nœud dans l'arbre des préfixes, il faudrait donc éviter de parcourir inutilement la branche et réduire ainsi significativement l'espace de recherche.

Comme nous l'avons dit précédemment, une séquence peut apparaître plusieurs fois dans une séquence de données. Dans la définition 6, nous avons introduit la notion d'intervalle maximal afin de pouvoir détecter toutes les extensions vers l'arrière. Nous désirons minimiser ces intervalles afin de détecter les séquences "non-prometteuses". Nous définissons ainsi la notion d'$i^{ème}$ intervalle minimal.

Définition 7 ($i^{ème}$ intervalle minimal)**.** *Pour une séquence de données S contenant une g-k-séquence préfixe $S_p = \langle s_1, s_2, \ldots, s_g \rangle$, l'$i^{ème}$ intervalle minimal de S_p dans S se définit de la façon suivante :*

- *Si $i = 1$ alors c'est la sous-séquence située strictement avant la première apparition de s_1.*
- *Si $1 < i \leq g$ alors c'est la sous-séquence comprise entre la première apparition de la séquence $\langle s_1, \ldots, s_{i-1} \rangle$ et strictement avant $pa(s_i)$ telle que $pa(s_i) < pa(s_{i+1}) \leq \ldots \leq pa(s_g)$.*

Théorème 2 (Elagage)**.** *Soit la g-k-séquence préfixe $S_p = \langle s_1, s_2, \ldots, s_g \rangle$, s'il existe un entier i tel qu'il existe un item e qui apparaît dans chacun des $i^{èmes}$ intervalles minimaux de S_p dans la base de données DB, alors il ne peut plus y avoir de séquence close de préfixe S_p.*

Démonstration. Si un item e apparaît dans chacun des intervalles minimaux de la g-k-séquence préfixe S_p alors nous pouvons utiliser la nouvelle séquence préfixe S'_p contenant l'item e. En effet, $S_p \subset_S S'_p$ et $support(S'_p) = support(S_p)$. Ainsi tout item localement fréquent sur la base projetée par S_p est aussi fréquent sur la base projetée par S'_p. Ce qui implique qu'il n'y a aucun espoir d'extraire un motif clos de préfixe S_p, le parcours de la séquence préfixe S_p peut donc être interrompu. □

Grâce aux théorèmes et définitions introduits précédemment, nous pouvons maintenant écrire les algorithmes permettant la mise en œuvre de l'extraction des motifs séquentiels multidimensionnels clos sans gestion d'ensemble de candidats.

3.4 Algorithmes

Les algorithmes 1 et 2 décrivent l'extraction des motifs séquentiels clos sans gestion d'ensembles de candidats. Ces algorithmes conservent la structure des algorithmes d'extraction de séquences fréquentes. En effet, dans le pire des cas, l'espace de recherche est le même. Toutefois, nous introduisons une condition d'élagage qui permet de réduire l'espace de recherche. L'algorithme 2 présente le cœur de l'extraction des motifs séquentiels multidimensionnels. Dans une première étape, si le nombre des extensions vers l'avant et vers l'arrière de la séquence préfixe S_p est nul, alors il faut vérifier qu'il n'existe pas de séquence plus spécifique dans l'ensemble FCS des motifs séquentiels multidimensionnels clos déjà extraits. S'il n'en existe aucune de même support que S_p, alors S_p est ajoutée à FCS. L'ensemble FCS est partitionné en sous-ensembles de motifs séquentiels multidimensionnels en fonction de leur support. Ainsi la recherche dans S_p d'une séquence plus spécifique que S_p s'effectue sur un sous-ensemble des motifs séquentiels clos déjà extraits. Dans le pire des cas, la complexité de cette opération de vérification est $O(l_\sigma)$ où l_σ est le nombre de séquences closes déjà extraites de support σ. Ensuite, chaque item localement fréquent e sur la base projetée est considéré. L'algorithme vérifie s'il est possible d'élaguer l'espace de recherche pointé par la séquence préfixe $S_p.e$ (e ajouté dans un nouvel itemset ou non). Si ce n'est pas possible alors l'algorithme calcule le nombre d'extensions vers l'arrière de la séquence préfixe $S_p.e$ et continue de fouiller l'espace de recherche indiqué par la nouvelle séquence préfixe.

Algorithme 1 : $CMSP$

Entrées : Base de données de séquences DB, seuil de support minimal $minsup$
Sorties : L'ensemble des motifs séquentiels multidimensionnels clos FCS

début
- $FCS = \emptyset$;
- $F1 =$items-fréquents$f(SDB, minsup)$;
- **pour chaque** *1-séquence* $f1 \in F1$ **faire**
  - **si** $!Elagage\_possible(f1, SDB^{f1})$ **alors**
    - `/* Dénombrement des extensions vers l'arrière. */`
    - $BEI =$ *#Extension_vers_arrière*$(f1, SDB^{f1})$;
    - Appel routine $CMSP(SDB^{f1}, f1, minsup, BEI, FCS)$;
- **return** FCS;

fin

Algorithme 2 : routine $CMSP$

Entrées : Une base projetée S_p_SDB, une séquence préfixe S_p, seuil de support minimal $minsup$, le nombre de d'extension vers l'arrière BEI
Sorties : L'ensemble courant des séquences fréquentes closes FCS

début
- `/* Rechercher items fréquents et extensions vers l'avant */`
- $LFI =$items-fréquents$(S_P_SDB, minsup)$;
- $FEI = |\{z \in LFI \,/\, sup(z) = sup^{SDB}(S_P)\}|$;
- **si** $(BEI + FEI) = 0$ **alors**
  - `/* On vérifie s'il n'y a pas de spécialisation déjà présente dans` FCS `*/`
  - **si** $(\nexists \alpha \in FCS \mid S_p \subset_S \alpha \wedge supp(\alpha) = supp(S_p))$ **alors**
    - $FCS = FCS \cup \{S_P\}$;
- **pour chaque** $i \in LFI$ **faire**
  - `/* Ajout de l'item fréquent à la séquence (intra ou inter itemset) et construction base projetée */`
  - $S_p^i = \langle S_p.i \rangle$;
  - $SDB^{S_pi} = pseudo\,projected\,database(S_p\_SDB, S_p^i)$;
- **pour chaque** $i \in LFI$ **faire**
  - `/* On vérifie si un élagage est possible */`
  - **si** $!Elagage\_possible(S_p^i, SDB^{S_pi})$ **alors**
    - $BEI = backward\,extension\,check(S_p^i, SDB^{S_pi})$;
    - Appel routine $CMSP(SDB^{S_pi}, S_p^i, minsup, BEI, FCS)$;

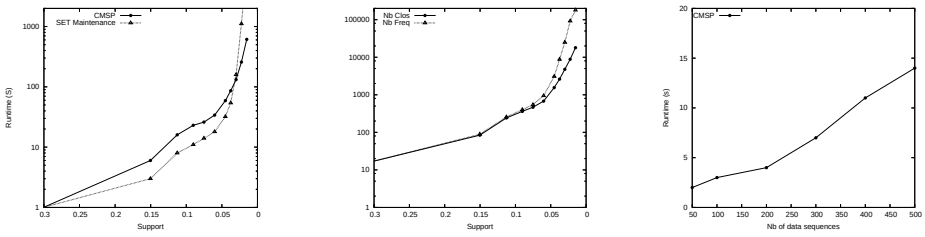
fin

4 Expérimentations

Dans cette section, nous présentons des résultats d'expérimentations menées sur des jeux de données synthétiques et sur des jeux de données réels. Afin de mettre en relief notre proposition, nous comparons les temps d'exécution de notre approche avec un algorithme d'extraction de clos gérant un ensemble de clos candidats. Cette implémentation s'appuie également sur des élagages anticipés introduits par Yan et al. (2003) avec l'approche CloSpan que nous avons adaptée à un contexte multidimensionnel.

Données Synthétiques Une base de données a été générée par *IBM Quest Market-Basket Synthetic Data Generator* (100000 enregistrements), où les items (1 dimension) ont été transformés en des items multidimensionnels (3 dimensions). Puisque notre approche effectue à la

fois des vérifications vers l'avant et l'arrière pour déterminer si une séquence fréquente est close, nous pouvons supposer que lorsque les données examinées sont éparses (nombre de fréquents faible et similaire au nombre de clos) une approche basée sur la gestion d'un ensemble de candidats peut être plus rapide. C'est ce qu'il se passe jusqu'à une certaine valeur du support minimal. Mais, dès que le support considéré entraîne un nombre important de séquences fréquentes, l'approche gérant un ensemble de clos candidats n'est plus adaptée. En effet, le temps d'exécution d'une telle approche est très sensible au nombre de motifs fréquents puisque la plupart d'entre eux sont considérés comme potentiellement candidats ; leur coût de traitement étant quadratique en la taille de l'ensemble des clos candidats. Notre approche est robuste face à ce phénomène puisqu'elle ne considère aucun ensemble de candidats et utilise des propriétés d'élagages supplémentaires qui évite de parcourir inutilement certaines parties de l'espace de recherche. La courbe 2(c) montre le comportement de notre approche en fonction de la taille de la base (en nombre de séquences de données). Le temps d'extraction des motifs multidimensionnels clos est proportionnel à la taille de la base. Ce qui nous permet de considérer le passage à l'échelle de notre approche par rapport à ce paramètre.



(a) Temps d'exécution en fonction du support

(b) # fréquents et clos en fonction du support

(c) Temps d'exécution en fonction de $|DB|$

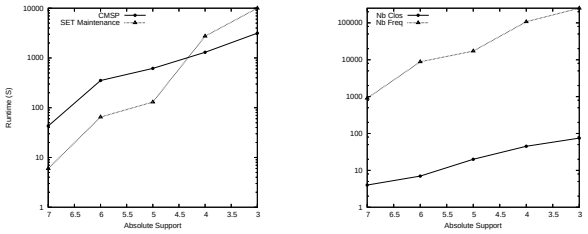
FIG. 2 – *Expérimentations sur des données synthétiques*

Cube de données réel Nous avons mené des expérimentations sur un cube de données issu de la base des clients résidentiels d'EDF. Nous considérons cinq dimensions d'analyse. Ces expérimentations confortent les résultats obtenus sur les jeux de données synthétiques : dès que le nombre de séquences extraites devient trop important, une approche avec gestion d'un ensemble de clos candidats n'est plus adaptée alors que notre approche ne décroche pas et permet d'extraire des connaissances avec des supports très faibles.

5 Travaux Connexes

Nos travaux sont au carrefour de plusieurs problématiques : (1) l'extraction de séquences multidimensionnelles, (2) l'extraction de séquences closes.

Pinto et al. (2001) sont les premiers à aborder le problème de l'extraction de motifs séquentiels dans un contexte multidimensionnel. Les séquences extraites ne contiennent pas plusieurs dimensions puisque la relation d'ordre (temps) concerne uniquement la dimension *produits*. Les autres dimensions sont "statiques" et seulement utilisées pour caractériser le profil des utilisateurs. Yu et Chen (2005) proposent d'extraire des séquences dans un contexte de web usage mining en considérant trois dimensions (pages, sessions, jours) qui appartiennent à une même hiérarchie. Ainsi, les séquences extraites décrivent des corrélations temporelles entre objets en



(a) Temps d'exécution en fonction du support

(b) Temps d'exécution en fonction du support

FIG. 3 – *Expérimentations sur cube de données réel*

considérant une seule dimension (pages). Plantevit et al. (2005) proposent des règles définies sur plusieurs dimensions d'analyse non "statiques".

Même s'il existe de nombreux travaux pour l'extraction d'itemsets clos (Pasquier et al. (1999); Pei et al. (2000); Zaki et Hsiao (2002); El-Hajj et Zaïane (2005)), il n'y a, à notre connaissance, que deux propositions pour les motifs séquentiels clos : BIDE de Wang et Han (2004) et CloSpan de Yan et al. (2003). Elaguer l'espace de recherche dans l'extraction de séquences fréquentes closes est plus fastidieux que pour l'extraction d'itemset clos. Par exemple, un algorithme d'extraction d'itemsets clos basé sur la profondeur arrête l'exploration d'un itemset dès que celui-ci est absorbé par un autre déjà extrait et clos, ce qui n'est pas possible pour les séquences puisqu'un item peut apparaître plusieurs fois dans une séquence. CloSpan et BIDE ne peuvent pas être directement adaptés dans notre contexte multidimensionnel à cause de la valeur joker. De plus CloSpan gère un ensemble de séquences closes candidates et effectue un post-traitement coûteux (quadratique en la taille de l'ensemble) alors que BIDE n'est défini que pour les séquences simples d'items.

Nous pouvons également citer les travaux de Songram et al. (2006) qui abordent le problème des motifs séquentiels clos dans un contexte multidimensionnel en proposant une représentation condensée des motifs définis par Pinto et al. (2001). Dans ce cas, il s'agit de séquences définies sur une seule dimension (*e.g.* product) où les autres dimensions sont "statiques". Il est alors toujours impossible de réaliser des combinaisons des dimensions au cours du temps au sein même de la séquence.

6 Conclusion

Dans cet article, nous avons proposé une approche complète (définitions et algorithmes) pour l'extraction de motifs séquentiels multidimensionnels clos. Ces motifs permettent d'obtenir une représentation condensée de l'ensemble des motifs séquentiels multidimensionnels sans aucune perte d'information. De plus, ceci permet de calculer différentes mesures (*e.g.* la confiance pour les règles séquentielles) sans passe supplémentaire sur la base de données puisque tous les supports sont connus. Outre leur puissance représentative, les motifs multidimensionnels clos permettent d'utiliser des propriétés supplémentaires d'élagage, ce qui est prépondérant pour assurer le passage à l'échelle de telles techniques d'extraction. Bien que des travaux aient été réalisés pour les itemsets clos et les motifs séquentiels clos, nous avons montré que ces approches n'étaient pas adaptées à notre contexte du fait de la valeur joker. Nous avons donc fourni une solution originale au traitement des items multidimensionnels

avec valeur joker * qui ne sont pas réellement matérialisés dans la base de données en adoptant le paradigme *"pattern growth"*. De plus, notre approche ne gère aucun ensemble de candidats. Ce qui permet d'éviter des post-traitements coûteux. Les expérimentations sur les jeux de données réels et synthétiques ont souligné la pertinence de notre proposition.

Les perspectives associées aux motifs séquentiels multidimensionnels clos sont nombreuses. Tout d'abord, nous souhaitons prendre en compte les hiérarchies, ce qui complexifierait d'avantage le problème de l'ordre entre les items. Eventuellement, nous pourrions dans ce cadre proposer de nouvelles représentations condensées véritablement nécessaires dans un contexte multidimensionnel. De telles représentations (non-dérivable Calders et Goethals (2002), k-libre Boulicaut et al. (2003)) sont très présentes dans le contexte des itemsets mais il existe encore trop peu de travaux pour les motifs séquentiels ou les motifs multidimensionnels. L'extraction de motifs séquentiels multidimensionnels sous contraintes (top k) peut aussi nous permettre d'élaguer plus rapidemment l'espace de recherche.

7 Remerciements

Nous remercions particulièrement Françoise Guisnel, Sabine Goutier et Marie-Luce Picard (EDF R&D)[1] pour les jeux de données réels sur lesquels nous avons pu réaliser les expérimentations.

Références

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

Ayres, J., J. Flannick, J. Gehrke, et T. Yiu (2002). Sequential pattern mining using a bitmap representation. In *KDD*, pp. 429–435.

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

El-Hajj, M. et O. R. Zaïane (2005). Finding all frequent patterns starting from the closure. In *ADMA*, pp. 67–74.

Han, J., J. Pei, et Y. Yin (2000). Mining frequent patterns without candidate generation. In *SIGMOD Conference*, pp. 1–12.

Masseglia, F., F. Cathala, et P. Poncelet (1998). The PSP Approach for Mining Sequential Patterns. In *Proc. of PKDD*, Volume 1510 of *LNCS*, pp. 176–184.

Pasquier, N., Y. Bastide, R. Taouil, et L. Lakhal (1999). Discovering frequent closed itemsets for association rules. In *ICDT*, pp. 398–416.

Pei, J., J. Han, et R. Mao (2000). Closet : An effi cient 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.-C. Hsu (2004). Mining sequential patterns by pattern-growth : The prefi xspan approach. *IEEE Transactions on Knowledge and Data Engineering 16*(10).

[1] Dans le cadre du partenariat de recherche OLAP Mining (Détection d'évolutions temporelles atypiques dans des cubes).

Pinto, H., J. Han, J. Pei, K. Wang, Q. Chen, et U. Dayal (2001). Multi-dimensional sequential pattern mining. In *CIKM*, pp. 81–88.

Plantevit, M., Y. W. Choong, A. Laurent, D. Laurent, et M. Teisseire (2005). M^2sp : Mining sequential patterns among several dimensions. In *PKDD*, pp. 205–216.

Songram, P., V. Boonjing, et S. Intakosum (2006). Closed multidimensional sequential pattern mining. In *ITNG*, pp. 512–517.

Wang, J. et J. Han (2004). Bide : Effi cient mining of frequent closed sequences. In *ICDE*, pp. 79–90.

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

Yu, C.-C. et Y.-L. Chen (2005). Mining sequential patterns from multidimensional sequence data. *IEEE Transactions on Knowledge and Data Engineering 17*(1), 136–140.

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

Zaki, M. J. et C.-J. Hsiao (2002). Charm : An effi cient algorithm for closed itemset mining. In *SDM*.

Summary

Sequential pattern mining leads to discovering correlations between events through time. More relevant patterns are discovered by taking several analysis dimensions into account. However, the number of patterns can become too important in a multidimensional framework. This is why we propose to define a condensed representation without lost of information: the closed multidimensional sequential patterns. This representation introduces properties that allow to prune deeply the search space. In this paper, we also define algorithms that do not require candidate set maintenance. Experiments on synthetic and real data are reported and emphasize the interest of our proposal.