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

Marc Plantevit, Anne Laurent, Maguelonne Teisseire

LIRMM, Université Montpellier 2, CNRS,
prenom.nom@lirmm.fr

Résumé. L'extraction de motifs séquentiels permet de découvrir des corrélations entre événements au cours du temps. Introduisant plusieurs dimensions d'analyse, les motifs séquentiels multidimensionnels permettent de découvrir des motifs plus pertinents. Mais 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 extraits ici sans gestion d'ensemble de candidats.

1 Introduction

Les motifs séquentiels sont étudiés depuis plus de 10 ans (Agrawal et Srikant (1995)). Ils ont donné lieu à de nombreuses applications.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 (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. 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.

2 Motivations et Problématique

Nous définissons ici le cadre théorique de l'extraction des motifs séquentiels multidimensionnels clos à partir de l'approche définie par Plantevit et al. (2005).

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 1. *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 2. *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

3.1 Approche "pattern growth" et ordre dans la séquence

Le contexte multidimensionnel rend l'approche générer-élaguer très difficile en raison du très grand nombre de combinaisons d'items possibles. Nous utilisons donc le paradigme "pattern growth" introduit par Pei et al. (2004) s'appuyant sur un parcours en profondeur de l'espace de recherche. 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. Le terme de g-k-séquence désigne les séquences composées de k items au sein de g itemsets.

Définition 3. *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 : (1) concaténation inter itemset où l'item est inséré dans un

nouvel itemset (le $(g+1)^{ème}$ itemset de la séquence) : $S' = s_1, s_2, \ldots s_g, \{e'\}$. (2) 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. 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.

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*

Le Tab. 1 illustre le fait que la valeur joker n'est pas explicitement présente dans les n-uplets, il n'est pas possible de définir un ordre lexicographique total. Ainsi, 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. Il est donc nécessaire d'ordonner les séquences en prenant en compte le caractère joker (*) comme valeur de dimension possible.

Un pré-traitement sur la base de données par extension à l'ensemble des n-uplets contenant la valeur joker étant trop coûteux, 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 *lexico-graphico-specifique*(LGS) qui 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.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 $<_{lgs}$.

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, 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 4 présente les cinq différents types de construction d'une séquence plus spécifique à partir d'une séquence préfixe.

Définition 4. *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, ni extension vers l'arrière, ni spécialisation.*

Pour déterminer si une séquence préfixe est close, nous devons donc vérifier si elle ne peut pas avoir d'extension vers l'avant ou vers l'arrière ainsi que de spécialisation d'item. Le lemme suivant facilite l'étude des extensions vers l'avant.

Lemme 1. *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 $support(S)$.*

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 : (i)$S' = s_1, s_2, \ldots, s_i, \{e'\}$, $s_{i+1}, \ldots, s_g$ et (ii) $S' = s_1, s_2, \ldots s_i \cup \{e'\}, s_{i+1}, \ldots, s_g$. Soit un item s'insère dans un nouvel itemset, entre deux itemset s_i et s_{i+1} existants (*inter-itemsets*), soit il s'insère dans un itemset existant (*intra itemset*). 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. Il faut maximiser ces intervalles afin de détecter toutes les extensions possibles vers l'arrière.

Définition 5. *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. *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.

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.

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 5, 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 6. *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.*

Les algorithmes et expérimentations relatifs à cette approche sont disponibles dans une version étendue de cet article à l'adresse suivante : *www.lirmm.fr/~ plantevi/egc2008.pdf.*

4 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 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). 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).

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). Cependant, il s'agit de séquences définies sur une seule dimension où les autres dimensions sont "statiques".

5 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. Notre approche adopte le paradigme pattern growth et permet l'apparition de valeurs joker * dans les motifs pour une extraction plus pertinente.

Les perspectives associées aux motifs séquentiels multidimensionnels clos sont nombreuses : prise en compte des hiérarchies, autres représentations condensées (non-dérivables Calders et Goethals (2002), k-libres Boulicaut et al. (2003)) et extraction de motifs séquentiels multidimensionnels sous contraintes (top k).

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.

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 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.-C. Hsu (2004). Mining sequential patterns by pattern-growth : The prefixspan approach. *IEEE Transactions on Knowledge and Data Engineering 16*(10).

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 : Efficient 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 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 *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.