HYPE : Prise en compte des hiérarchies lors de l'extraction de motifs séquentiels multidimensionnels

Marc Plantevit, Anne Laurent, Maguelonne Teisseire

LIRMM, Université Montpellier 2, CNRS, 161 Rue Ada 34392 Montpellier, France
nom.prenom@lirmm.fr
http://www.lirmm.fr/~plantevi
http://www.lirmm.fr/~laurent
http://www.lirmm.fr/~teisseir

Résumé. Les entrepôts de données contiennent de grosses masses de données historisées stockées à des fins d'analyse. Si les méthodes et outils d'analyse sont maintenant bien connus (OLAP), il reste difficile de fournir aux utilisateurs des outils de fouille de données permettant la prise en compte des spécificités de ces contextes (e.g. multidimensionnalité, hiérarchies, données historisées). Dans cet article, nous proposons une méthode originale d'extraction de motifs séquentiels prenant en compte les hiérarchies. Cette méthode extrait des connaissances plus précises et étend notre approche précédente M^2SP. Nous définissons les concepts liés à notre problématique ainsi que les algorithmes associés. Les expérimentations que nous avons menés montrent l'intérêt de notre proposition.

1 Introduction

La technologie OLAP et la fouille de données ne sont pas incompatibles (Han (1997)). Les techniques d'extraction de connaissances peuvent apporter une aide non négligeable dans le contexte OLAP où l'utilisateur doit désormais prendre les décisions les mieux adaptées en un minimum de temps. De façon plus précise, la fouille de données constitue une étape clef dans le processus de décision face à de gros volumes de données multidimensionnelles. En effet, les motifs ou règles obtenues permettent une autre appréhension des données sources. Cependant leur découverte nécessite certains paramètres dont en particulier le support minimal. Celui-ci correspond à la fréquence minimale d'apparition des motifs au sein de la base considérée. Si le support minimal choisi est trop élevé, le nombre de règles découvertes est faible mais si le support est trop bas, le nombre de règles obtenues est très important et rend difficile l'analyse de celles-ci. Le décideur est alors confronté au problème suivant : comment baisser le support minimal sans générer la découverte de règles non pertinentes ? Ou comment augmenter le support minimal sans perdre les règles utiles ? Est-il alors nécessaire de faire un compromis entre qualité des connaissances extraites et support ?

L'utilisation des hiérarchies dans l'extraction de connaissances représente un excellent moyen de résoudre ce dilemme. Elle permet de découvrir des règles au sein de plusieurs niveaux de hiérarchies. Ainsi, même si un support élevé est utilisé, les connaissances importantes

dont le support est faible dans les données sources peuvent être *incluses* dans des connaissances plus générales qui, elles, seront comptabilisées comme fréquentes. Nous souhaitons ainsi étendre notre proposition précédente de recherche de motifs séquentiels multidimensionnels (Plantevit et al. (2005)) par la prise en compte des hiérarchies.

L'extraction de motifs séquentiels est devenue depuis son introduction par Agrawal et Srikant (1995), une technique majeure du domaine de l'extraction de connaissances. Ils sont ainsi apparus afin de permettre la découverte de connaissances intégrant la notion de temporalité et séquentialité. Ils permettent de mettre en exergue des corrélations entre évènements en fonction de leur chronologie d'apparition. Même si quelques approches permettent la prise en compte des hiérarchies dans l'extraction de motifs séquentiels, il n'existe pas, à notre connaissance, de travaux conciliant extraction de motifs séquentiels multidimensionnels et prise en compte des hiérarchies. Aucune méthode actuelle ne peut extraire des connaissances du type : *Quand les ventes de boissons gazeuses augmentent en Europe, les exportations de perrier augmentent en France et les exportations de soda augmentent aux USA* où différents niveaux de hiérarchies sont présents dans la séquence multidimensionnelle. Nous proposons donc une approche originale HYPE , extension de notre proposition M^2SP (Plantevit et al. (2005)), permettant d'extraire de telles règles. L'originalité de notre approche réside dans l'idée qu'on ne fixe pas un unique niveau de hiérarchie mais que les motifs séquentiels extraits sont automatiquement associés aux niveaux les plus adéquats.

Dans cet article, nous introduisons les concepts associés aux motifs séquentiels classiques et multidimensionnels ainsi que les approches s'attachant à la prise en compte des hiérarchies lors de l'extraction de connaissances. Nous présentons ensuite les concepts fondamentaux relatifs à notre approche HYPE ainsi que les algorithmes permettant sa mise en œuvre. Des expérimentations effectuées sur des données synthétiques sont reportées et confirment l'interêt de notre approche. Nous montrons aussi que la prise en compte des hiérarchies permet une gestion plus fine des valeurs jokers définies dans l'approche M^2SP.

2 Les hiérarchies dans la fouille de données

Dans cette section, nous présentons les motifs séquentiels ainsi que les approches de la littérature ayant traité le problème de l'extraction de motifs séquentiels dans un contexte multidimensionnel (plusieurs dimensions d'analyse). Ensuite, nous soulignons pourquoi il est pertinent d'utiliser les hiérarchies lors du processus d'extraction de motifs séquentiels et faisons un panorama des travaux associés.

2.1 Motifs séquentiels

L'extraction de motifs séquentiels est devenue depuis son introduction par Agrawal et Srikant (1995), une technique majeure du domaine de l'extraction de connaissances. Ces motifs permettent de mettre en exergue des corrélations entre évènements en prenant compte de leur chronologie d'apparition. Nous présentons ici très brièvement les concepts fondamentaux liés aux motifs séquentiels. Le lecteur désirant plus de détails se réfèrera à Masseglia et al. (2004). Les bases de données sur lesquelles s'appuient l'extraction de motifs séquentiels comportent

trois données étroitement liées au problème du panier de la ménagère : la première représente un identifiant (souvent appelé *client*), le deuxième représente une liste de valeurs (souvent appelée *produits*), la troisième représente la date à laquelle ce client a acheté cet ensemble de produits. On appelle *item* une valeur prise par l'attribut *produit*. Par exemple, DVD ou encore $magnetoscope$ sont deux items possibles. On appelle itemset un ensemble d'items. Par exemple $(DVD, magnetoscope)$ est un itemset. La base de données est donc composée d'itemsets identifiés par une date et un identifiant de client. On appelle séquence une liste ordonnée (selon la date) d'itemsets. La base de données peut donc être vue comme un ensemble de séquences identifiées par le client. On appelle motif séquentiel une séquence qu'un nombre suffisant (au sens du support) de clients partagent au sein de la base de données. Étant donnée une valeur minimale de support (spécifiée par l'utilisateur), on dit qu'un motif séquentiel est *fréquent* si un nombre de clients supérieur au seuil minimal de support ont réalisé cette séquence d'achats. L'enjeu des méthodes de fouille de données est donc l'extraction la plus efficace possible des motifs féquents. Pour cela, plusieurs algorithmes existent dont PSP Masseglia (2002) ou encore PrefixSpan Pei et al. (2004). Ces techniques sont fondées sur le paradigme *générer/élaguer* où des candidats sont générés puis ensuite élagués s'ils ne sont pas fréquents.

Dans le contexte classique (une seule dimension d'analyse) d'extraction de règles d'association ou motifs séquentiels, il existe plusieurs travaux qui prennent en compte les hiérarchies afin de permettre une extraction de connaissances plus fines.

Dans Srikant et Agrawal (1996), les prémisses de la gestion des hiérarchies dans l'extraction de règles d'association et de motifs séquentiels sont proposées. Les auteurs supposent que les relations hiérarchiques entre les items sont représentées par un ensemble de taxonomies sous forme d'un graphe orienté sans cycle. Ils permettent d'extraire des règles d'association ou des motifs séquentiels suivant plusieurs niveaux de hiérarchies. Ils modifient les transactions en ajoutant pour chaque item l'ensemble de ses ancêtres dans la taxonomie associée. Ensuite, ils génèrent les séquences fréquentes tout en essayant de filtrer au maximun l'information redondante et en optimisant le processus à l'aide de plusieurs propriétés. Cette approche peut être difficilement adaptée dans un contexte multidimensionnel. En effet, pour chaque transaction, ajouter sur chaque dimension la liste des ancêtres d'un item dans la taxonomie est impensable. Cela reviendrait, dans le pire des cas, à multiplier la taille de la base de données que l'on souhaite étudier par la profondeur maximale d'une hiérarchie et ceci pour chaque dimension d'analyse, un parcours sur cette base serait alors beaucoup trop coûteux.

L'approche de J. Han est sensiblement différente. Il s'est lui aussi attaché à prendre en compte les hiérarchies dans les processus d'extraction de connaissances. Elle s'applique à la problématique d'extraction de règles d'association, mais cette approche peut être adaptée pour l'extraction de motifs séquentiels. Ainsi, en partant du plus haut niveau de la hiérarchie, il va extraire les règles à chaque niveau tout en abaissant le support lorsqu'il descend d'un niveau dans la hiérarchie. Le processus est itéré jusqu'à ce que l'on ne puisse plus extraire de règles ou que l'on soit au niveau le plus bas de la hiérarchie. Cette méthode ne permet pas d'extraire des règles où des items de niveaux différents cohabiteraient comme par exemple vin et $boisson\ alcoolisée$. Cette méthode propose donc l'extraction de règles d'association *intra niveau de hiérarchie*. Elle ne permet donc pas de répondre à la problématique générale d'extraction des séquences sur différents niveaux de hiérarchies. De plus la mise en œuvre de

cette approche dans un contexte multidimensionnel peut susciter des débats. Dans le cas où plusieurs taxonomies existent (une par dimension), doit-on se déplacer sur les mêmes niveaux de hiérarchies sur les différentes taxonomies ou combiner ces niveaux ? Ce type d'extraction peut être coûteux en temps, car le mécanisme d'extraction de connaissances peut être réitéré plusieurs fois (profondeur de la taxonomie), ce qui n'est pas négligeable.

Nous avons présenté les motifs séquentiels ainsi que des travaux permettant la prise en compte des hiérarchies dans le processus d'extraction de connaissances. Néanmoins les motifs séquentiels sont parfois pauvres par rapport aux données qu'ils décrivent. En effet, les corrélations sont extraites au sein de la seule dimension[1] *produit* alors qu'une base de données peut contenir plusieurs autres dimensions. C'est pourquoi plusieurs travaux tentent de combiner plusieurs dimensions d'analyse dans l'extraction de motifs séquentiels multidimensionnels.

2.2 Motifs séquentiels multidimensionnels

Combiner plusieurs dimensions d'analyse permet d'extraire des connaissances qui décrivent mieux les données. Dans Pinto et al. (2001) les auteurs sont les premiers à rechercher des motifs séquentiels multidimensionnels. Ainsi, les achats ne sont plus décrits en fonction des seuls date et identifiant du client, mais en fonction d'un ensemble de dimensions telles que *Type de consommateur, Ville, Age*. Cette approche permet d'extraire des séquences d'items sur la dimension *produits* et de les caractériser à l'aide des informations fréquentes sur les clients (*Patterns*) qui tendent à supporter les séquences. Cette méthode ne permet pas d'avoir des séquences où plusieurs patterns sont présents. Elle ne permet donc pas d'extraire des connaissances de la forme : $\langle\{(business, *, *, a)(*, chicago, *, b)\}, \{(*, *, young, c)\}\rangle$ alliant différents patterns multidimensionnels.

L'approche M^{2}SP que nous avons proposée Plantevit et al. (2005) permet, quant à elle, l'extraction de motifs séquentiels multidimensionnel *inter pattern*. Nous décrivons plus en détail les concepts associés dans le paragraphe 3.

Dans Yu et Chen (2005), les auteurs proposent d'étendre la recherche de motifs séquentiels au contexte des bases de données décrivant les informations au moyen de plusieurs attributs. Cependant cette approche est restreinte au cas particulier où les dimensions étudiées entretiennent entre elles un très fort lien. En effet, ces dimensions sont organisées en hiérarchie. Ainsi, dans l'exemple pris par les auteurs, les différentes dimensions sont liées au comportement d'internautes dont les visites de pages sont organisées en transactions (dimension 1), elles-mêmes organisées en sessions (dimension 2), elles-mêmes organisées en jours (dimension 3). Ces différentes dimensions sont imbriquées au sein des motifs trouvés et il est impossible de retrouver les valeurs fréquentes le long de ces dimensions, celles-ci n'intervenant que pour organiser le temps de manière hiérarchique. Yu et Chen (2005) propose d'extraire des séquences au sein de séquence de données multidimensionnelles organisées en différents

[1] Nous utilisons le terme de dimension à la place du terme d'attribut car une base de données relationnelle peut être vue comme une table de faits dans un base de données multidimensionnelles.

D (Date)	B ($Bloc_{ID}$)	Pl (Lieu)	P (Produit)
1	1	Allemagne	Bière
1	1	Allemagne	Cacahuètes
2	1	Allemagne	Aspirine
3	1	Allemagne	Chocolat
4	1	Allemagne	Smecta
1	2	France	coca
2	2	France	Vin
2	2	France	Cacahuètes
3	2	France	Aspirine
1	3	UK	Whisky
1	3	UK	Cacahuètes
2	3	UK	Aspirine
1	4	LA	Chocolat
2	4	LA	Smecta
3	4	NY	Whisky
4	4	NY	Coca

FIG. 1 – *Base de données exemple DB*

niveaux de hiérarchie. Néanmoins, les séquences de données ne sont pas réellement multidimensionnelles dans la mesure où les différentes dimensions entretiennent un lien hiérarchique très strict (un jour comporte des sessions qui sont elles-mêmes composées de pages visitées).

Nous pouvons encore citer les travaux de de Amo et al. (2004) qui proposent une approche basée sur la logique temporelle du premier ordre pour l'extraction de motifs séquentiels multidimensionnels, Lee (2005) proposent également une nouvelle méthode de génération des séquences multidimensionnelles présentes dans des bases de transactions.

A notre connaissance, il n'existe aucune approche proposant de prendre en compte les hiérarchies dans l'extraction de motifs séquentiels multidimensionnels. Nous proposons donc d'intégrer la gestion des hiérarchies à M^2SP (Plantevit et al. (2005)) afin de permettre une extraction de connaissances plus complète et dont l'utilisation dans le contexte OLAP peut être envisageable.

2.3 Base exemple

Pour illustrer les différents concepts et définitions, nous proposons la base exemple fig. 1 qui décrit les achats de produit réalisés dans différentes villes du monde. Pour les hiérarchies, nous choisissons deux dimensions, les villes et les produits, dont les taxonomies respectives sont indiquées fig. 2 et fig. 3.

FIG. 2 – *Taxonomie sur la dimension $Lieu$*

FIG. 3 – *Taxonomie sur la dimension $Produit$*

3 Contributions

Dans cette section, nous présentons notre approche permettant la prise en compte des hiérarchies dans l'extraction de motifs séquentiels multidimensionnels. Nous définissons d'abord les concepts relatifs à notre approche. Nous proposons ensuite les algorithmes permettant la mise en œuvre de notre approche.

3.1 Définitions

Partition des dimensions

Nous considérons que *tout est ensemble* dans un contexte multidimensionnel. Les trois données nécessaires pour l'extraction de motifs séquentiels dans un contexte classique (*Client, produits, date*) deviennent dans un contexte multidimensionnel des ensembles[2].

Ainsi, comme dans M^{2}SP (Plantevit et al. (2005)), nous considérons que l'ensemble DB des transactions définies sur un ensemble D de n dimensions est partitionné en trois sous-ensembles :

- l'ensemble des dimensions de référence D_R (*client dans contexte classique*) qui permettent de déterminer si une séquence est fréquente.
- l'ensemble des dimensions D_T (*date dans contexte classique*) permettant d'introduire une relation d'ordre.
- l'ensemble des dimensions d'analyse $D_A = \{D_1, \ldots, D_m$ où $D_i \subset Dom(D_i)\}$ (*produits dans contexte classique*) où sont extraites les corrélations.

Chaque n-uplet $c = (d_1, \ldots, d_n)$ peut s'écrire sous la forme d'un triplet $c = (r, a, t)$ où r, a et t sont les restrictions de c sur respectivement D_R, D_A et D_T.

Définition 1 (Bloc) *Etant donnée une base DB, l'ensemble des n-uplets qui ont la même restriction r sur D_R constitue un bloc.*

Chaque bloc B est identifié par un n-uplet r. Nous notons B_{DB,D_R}, l'ensemble des blocs constituant la base DB.

Cette définition des blocs est nécessaire pour définir le support d'une séquence multidimensionnelle. Son application dans notre base exemple est simple puisque $|D_R| = 1$, les différents blocs obtenus sont décrits fig 8.

Taxonomies

Dans le contexte dans lequel nous nous situons, nous considérons qu'il existe des relations hiérarchiques sur chaque dimension d'analyse[3]. Nous considérons que ces relations hiérarchiques sont matérialisées sous la forme de *taxonomie*. Une taxonomie est un arbre orienté dans lequel les arcs sont des relations de type *is-a*. La relation de *généralisation/spécialisation* s'effectue ainsi de la racine vers les feuilles. Chaque dimension d'analyse possède donc une taxonomie qui permet de représenter les relations hiérarchiques entre les éléments de son domaine.

[2] Dans un contexte classique, ces données sont des singletons.

[3] Dans le pire des cas la hiérarchie minimale se représente par un arbre de profondeur 1 où la racine est etiquetée par * (gestion des valeurs jokers dans M^2SP).

D	B	Pl	P
1	1	Allemagne	Bière
1	1	Allemagne	Cacahuètes
2	1	Allemagne	Aspirine
3	1	Allemagne	Chocolat
4	1	Allemagne	Smecta

FIG. 4 – *bloc* (1)

D	B	Pl	P
1	2	France	Coca
2	2	France	Vin
2	2	France	Cacahuètes
3	2	France	Aspirine

FIG. 5 – *bloc* (2)

D	B	Pl	P
1	3	UK	Whisky
1	3	UK	Cacahuètes
2	3	UK	Aspirine

FIG. 6 – *bloc* (3)

D	B	Pl	P
1	4	LA	Chocolat
2	4	LA	Smecta
3	4	NY	Whisky
4	4	NY	Coca

FIG. 7 – *bloc* (4)

FIG. 8 – *Partition de DB (figure 1) en fonction de $D_R = \{B\}$*

Soit $T_{D_A} = \{T_1, \ldots, T_m\}$ l'ensemble des taxonomies associées aux dimensions d'analyse où :

- T_i est la taxonomie représentant les relations hiérarchiques entre les éléments de la dimension d'analyse D_i.
- T_i est un arbre orienté.
- $\forall$ nœud $n_i \in T_i,\ label(n_i) \in Dom(D_i)$.

On note $\hat{x}$ un ancêtre de x dans la taxonomie et $\check{x}$ un de ses descendants. Par exemple, $boisson = \widehat{soda}$ signifie que $boisson$ est un ancêtre de $soda$ dans la relation *Généralisation/Spécialisation*. Plus précisement, $boisson$ est une instance plus générale que $soda$.

Hiérarchies et Données

Chaque dimension d'analyse D_i d'une transaction b de DB ne peut être instanciée qu'avec une valeur d_i dont le nœud associé à l'étiquette d_i dans la taxonomie T_i est une *feuille*. Plus formellement, $\forall d_i \in \pi_{D_i}(B), \forall$ nœud $n_i\ tq\ label(n_i) = d_i \nexists$nœud $n'\ tq\ n' = \check{n_i}$ (n_i feuille).

Par exemple, la base de transactions DB ne peut pas contenir la valeur $Boisson$ s'il existe des instances plus spécifiques dans la taxonomie comme $soda$.

Item, Itemset, Séquence multidimensionnels h-généralisés

Dans cette section, nous définissons les concepts fondamentaux d'items, d'itemsets et de séquences multidimensionnels h-généralisés.

Définition 2 (Item multidimensionnel h-généralisé)
Un item multidimensionnelh-généralisé $e = (d_1, \ldots, d_m)$ est un m-uplet défini sur les dimensions d'analyse D_A telles que $d_i \in \{label(T_i)\}$.

Contrairement aux transactions de DB, un item multidimensionnel h-généralisé peut être défini avec n'importe quelle valeur d_i dont le nœud associé dans la taxonomie n'est pas nécessairement une feuille.

Exemple 1 $(boisson, USA), (soda, France)$ *sont des items multidimensionnels h-généralisés.*

Comme les items multidimensionnels h-généralisés sont instanciés sur différents niveaux de hiérarchies, il est possible que deux items soient comparables, c'est-à-dire qu'un item soit plus *spécifique ou général* qu'un autre.

Par abus de langage et afin de ne pas alourdir les notations, nous utilisons directement la notion d'*ancêtre* sur l'item et la transaction sans nous situer dans la taxonomie correspondante.

Définition 3 (Inclusion hiérarchique d'items)
Soient deux items multidimensionnels h-généralisés $e = (d_1, \ldots, d_m)$ *et* $e' = (d'_1, \ldots, d'_m)$*, on dit que :*

- e *est plus général que* e' *(*$e >_h e'$*) si* $\forall d_i, d_i = \hat{d'_i}$ *ou* $d_i = d'_i$
- e *est plus spécifique que* e' *(*$e <_h e'$*) si* $\forall d_i, d_i = \check{d'_i}$ *ou* $d_i = d'_i$
- e *et* e' *sont incomparables s'il n'existe pas de relation entre eux (*$e \not>_h e'$ *et* $e' \not>_h e$*)*

Exemple 2 (relations hiérarchiques entre items multidimensionnels h-généralisés)

- $(USA, boisson) >_h (USA, soda)$.
- $(France, vin) <_h (UE, Alcool)$.
- $(France, vin)$ *et* $(USA, soda)$ *sont incomparables.*

Définition 4 *Une transaction* b *supporte un item* e *si* $\Pi_{D_A}(b) <_h e$.

Exemple 3 *La transaction* $(1, 1, France, vin)$ *supporte l'item* $(UE, alcool)$.

Définition 5 (Itemset multidimensionnel h-généralisé)
Un itemset multidimensionnel h-généralisé $i = \{e_1, \ldots, e_k\}$ *est un ensemble non vide d'items multidimensionnels h-généralisés où tous les items sont incomparables entre eux.*

Deux items comparables ne peuvent pas être présents dans le même itemset. Nous adoptons un point de vue ensembliste et préférons ainsi représenter l'information la plus précise possible au sein d'un itemset.

Exemple 4 $\{(France, vin), (USA, soda)\}$ *est un itemset multidimensionnel h-généralisé alors que* $\{(France, vin), (UE, Alcool)\}$ *n'est pas un itemset multidimensionnel h-généralisé car* $(France, vin) <_h (UE, Alcool)$.

La notion de séquence multidimensionnelle h-généralisée découle de la notion d'itemset.

Définition 6 (Séquence multidimensionnelle h-généralisée) *Une séquence multidimensionnelle h-généralisée* $s = \langle i_1, \ldots, i_j \rangle$ *est une liste ordonnée non vide d'itemsets multidimensionnels h-généralisés.*

Exemple 5 $\langle \{(France, vin), (USA, soda)\}, \{(Allemagne, biere)\} \rangle$ *est une séquence multidimensionnelle h-généralisée.*

Définition 7 (Inclusion de séquences) *Une séquence multidimensionnelle h-généralisée* $\varsigma = \langle a_1, \ldots, a_l \rangle$ *est une sous-séquence de la séquence* $\varsigma' = \langle b_1, \ldots, b_{l'} \rangle$ *s'il existe des entiers* $1 \leq j_1 \leq j_2 \leq \ldots \leq j_l \leq l'$ *tel que* $a_1 \subseteq b_{j_1}, a_2 \subseteq b_{j_2}, \ldots, a_l \subseteq b_{j_l}$.

Remarque 1 *L'inclusion des itemsets multidimensionnels doit respecter l'inclusion hiérarchique des items multidimensionnels h-généralisés.*

Exemple 6

- *La séquence* $\langle \{(France, vin)\}, \{(Allemagne, biere)\} \rangle$ *est une sous-séquence de la séquence* $\langle \{(France, vin), (USA, soda)\}, \{(Allemagne, biere)\} \rangle$.
- *La séquence* $\langle \{(France, vin)\}, \{(Allemagne, biere)\} \rangle$ *est une sous-séquence de la séquence* $\langle \{(France, Alcool), (USA, boisson)\}, \{(UE, Alcool)\} \rangle$.
- *La séquence* $\langle \{(UE, vin)\}, \{(Allemagne, biere)\} \rangle$ *n'est pas une sous-séquence de la séquence* $\langle \{(France, vin), (USA, soda)\}, \{(Allemagne, biere)\} \rangle$ *car* $(UE, vin) \not\leq_h (France, vin)$*, l'inclusion hiérarchique n'étant pas respectée.*

Support

Calculer le support d'une séquence multidimensionnelle h-généralisée revient à compter le nombre de blocs définis par les dimensions de référence D_R qui supportent la séquence. Un bloc supporte une séquence multidimensionnelle h-généralisée s'il est possible de trouver un ensemble de n-uplets qui la satisfasse. Pour chaque itemset de la séquence, nous devons exhiber une date du domaine de D_t telle que tous les items multidimensionnels h-généralisés de l'itemset sont supportés par des n-uplets relatifs à cette date. Tous les itemsets doivent être retrouvés à différentes dates appartenant au domaine de D_t tels que l'ordre des itemsets respecte la séquentialité.

Définition 8 *Un bloc supporte une séquence* $\langle i_1, \ldots, i_l \rangle$ *si* $\forall j = 1 \ldots l, \ \exists d_j \in Dom(D_t)$*, pour chaque item* e *de* i_j, $\exists t = (r, e, d_j)$ *ou* $t = (r, \check{e}, d_j) \in T$ *avec* $d_1 < d_2 < \ldots < d_l$.

Définition 9 (Support d'une séquence) *Soient* D_R *l'ensemble des dimensions de référence et* DB *l'ensemble des transactions partitionné en un ensemble de blocs* B_{T,D_R}*. Le support d'une séquence* ς *est :* $support(\varsigma) = \frac{|\{B \in B_{DB,D_R} \ t.q. \ B \ supporte \ \varsigma\}|}{|B_{DB,D_R}|}$

Exemple 7 *Par rapport à notre base de données exemple* DB*, considérons* $D_R = \{B_{id}\}$*,* $D_A = \{Lieu, \ Produit\}$ *et* $D_T = \{Date\}$*, support* $= 2$*, et* $\varsigma = \langle \{(UE, Alcool), (UE, caca\text{-}huetes)\} \ \{(UE, aspirine)\} \rangle$*. Pour que la séquence soit fréquente, au moins deux blocs de la partition de* DB *doivent supporter la séquence.*
1. bloc $(\mathbf{1})$ *(Fig. 4). Si l'on se réfère au taxonomies relatives aux dimensions d'analyse (LABEL),* Allemagne *est une instance plus spécifique de* UE *et* bière *est un* alcool*. Ainsi à la date* 1*, nous avons bien le premier itemset* $\{(UE, Alcool), (UE, cacahuetes)\}$ *de* ς*. A une date postérieure (*2*), le dernier itemset* $\{(UE, aspirine)\}$ *est présent. La séquence* ς *est supportée par ce bloc.*
2. bloc $(\mathbf{2})$ *(Fig. 5).* France *est une instance de* UE *et* vin *est une instance d'*alcool*. Nous retrouvons bien la séquence* ς *dans ce bloc*

3. bloc $(\mathbf{3})$ *(Fig. 6). UK est une instance de UE et whisky est une instance d'alcool. Ce bloc supporte la séquence* ς.
4. bloc $(\mathbf{4})$ *(Fig. 7). Ce bloc ne supporte pas la séquence* ς *puisque la dimension* $Lieu$ *ne contient aucune instance de UE.*

Le support de ς *est donc égal à* 3*. La séquence est fréquente.*

3.2 HYPE : Les algorithmes

Fonctionnement général

Avant de présenter les algorithmes permettant l'extraction de motifs séquentiels multidimensionnels h-généralisés, nous détaillons brièvement le fonctionnement de notre approche.

Le processus d'extraction de motifs séquentiels multidimensionnels h-généralisés se divise en deux phases. Dans un premier temps, les items multidimensionnels h-généralisés maximalement spécifiques sont extraits. Nous pensons que les items maximalement spécifiques sont une alternative à la surabondance de connaissances extraites. En effet, ils permettent de *factoriser* les connaissances, les connaissances plus générales pouvant être inférées en post traitement par l'utilisateur. Ensuite, la deuxième étape vise à extraire les séquences multidimensionnelles h-généralisées fréquentes. Ces séquences sont générées à partir de l'ensemble des items maximalement spécifiques.

Néanmoins, le fait d'utiliser des items maximalement spécifiques pour générer les séquences fréquentes ne nous permet pas d'extraire toutes les connaissances présentes dans la base. En effet, des séquences dont les premiers items ne sont pas maximalement spécifiques ne pourront pas être extraites. Les séquences plus longues ne sont donc pas extraites (les blocs supportent plus rapidement des connaissances plus générales). Toutefois, cette carence est relative car ces séquences non extraites représentent souvent des connaissances trop générales qui n'apportent aucun intérêt à l'utilisateur.

Il n'est pas forcément nécessaire d'effectuer une phase de prétraitement afin d'élaguer les taxonomies. En effet, cette opération peut être facilement effectuée lors de l'extraction des items multidimensionnels h-généralisés fréquents.

Génération des items fréquents

Les items multidimensionnels h-généralisés fréquents sont la base de l'extraction de motifs séquentiels multidimensionnels h-généralisés. Ils représentent les fréquents de taille 1 puisqu'ils correspondent à des séquences composées d'un seul item contenu dans un seul itemset. L'extraction d'items multidimensionnels h-généralisés en une seule passe sur la base n'est pas concevable dans un souci de *passage à l'echelle*. En effet, considérer le produit cartésien des domaines de chaque dimension d'analyse n'est pas envisageable dans des applications où le nombre de dimensions et leurs domaines peuvent être très grands. Si le nombre de dimensions d'analyse est m, alors le nombre d'items générés χ est exponentiel par rapport à m :

$$2^m \leq \chi \leq \sum_{i=1}^{m} \binom{m}{i} i^k \text{ où } k = \max |Dom(D_i)|$$

Nous conviendrons donc qu'avec une telle approche, le passage à l'echelle peut être mis en doute.

Il est donc nécessaire de définir une méthode qui limite à la fois le nombre d'items candidats générés et le nombre de passes sur la base. Afin de limiter le nombre d'items candidats aux seuls items dont la probabilité d'être fréquents est non nulle, nous adoptons une méthode de génération par niveau.

Tout d'abord, nous considérons les items multidimensionnels h-généralisés pour lesquels une seule dimension d'analyse est spécifiée[4], les autres dimensions n'étant pas spécifiées. Les items multidimensionnels fréquents sont alors *joints* entre eux pour obtenir l'ensemble des items candidats pour lesquels deux dimensions d'analyse sont spécifiées. Seuls les fréquents sont retenus. Cette procédure est réitérée m fois jusqu'à l'obtention des items multidimensionnels h-généralisés (toutes les m dimensions d'analyse sont instanciées). Parmi ces items, seuls les plus spécifiques seront retenus.

L'opération de *jointure* entre deux items fréquents suppose que les items soient $\bowtie$-compatibles, c'est-à-dire qu'ils partagent un nombre suffisant de valeurs de dimensions d'analyse (voir définition 10). Pour être $\bowtie$-compatibles, deux items multidimensionnels définis sur n dimensions doivent partager $n-2$ valeurs de dimension. Par exemple, $(a,*,c)$ et $(*,b,c)$ sont deux items définis sur 3 dimensions d'analyse et partagent $3-2=1$ valeur sur la dimension C. Ils sont donc $\bowtie$-compatibles. En revanche, les items $(a_1,b_1,*)$ et $(a_2,b_2,*)$ ne sont pas $\bowtie$-compatibles.

Définition 10 ($\bowtie$-Compatibilité) *Soient deux items multidimensionnels $e_1=(d_1,\cdots,d_n)$ et $e_2=(d'_1,\cdots,d'_n)$ où d_i et $d'_i \in dom(D_i)\cup\{*\}$. On dit que e_1 et e_2 sont $\bowtie$-compatibles si*

- *e_1 et e_2 sont distincts*
- *$\exists\Delta=\{D_{i_1},\ldots,D_{i_{n-2}}\}\subset\{D_1,\ldots,D_n\}$ t.q. $d_{i_1}=d'_{i_1}\neq *$ et $d_{i_2}=d'_{i_2}\neq *\ldots$ et $d_{i_{n-2}}=d'_{i_{n-2}}\neq *$*
- *Pour $\{D_{i_{n-1}},D_{i_n}\}=\{D_1,\ldots,D_n\}\backslash\Delta$, on a $d_{i_{n-1}}=*$ et $d'_{i_{n-1}}\neq *$ et $d_{i_n}\neq *$ et $d'_{i_n}=*$*

L'opération de jointure mise en œuvre pour générer les items multidimensionnels h-généralisés potentiellement fréquents se définit de la façon suivante :

Définition 11 (Jointure) *Soient 2 items multidimensionnels $\bowtie$-compatibles $e_1=(d_1,\cdots,d_n)$ et $e_2=(d'_1,\cdots,d'_n)$. On définit $e_1\bowtie e_2=(v_1,\cdots,v_n)$ avec :*

- *$v_i=d_i$ si $d_i=d'_i$*
- *$v_i=d_i$ si $d'_i=*$*
- *$v_i=d'_i$ si $d_i=*$*

La génération des items multidimensionnels s'effectue donc à l'aide d'un treillis. Néanmoins le nombre de candidats générés reste important, on peut imaginer utiliser la recherche d'items multidimensionnels dérivables pour limiter le calcul du support à un nombre réduit d'items (recherche équivalente à la recherche d'itemsets dérivables).

Génération des séquence fréquentes

Les items multidimensionnels h-généralisés sont donc des séquences multidimensionnelles h-généralisées de taille 1. Ils sont donc des 1-fréquents.

[4]Par définition, un item multidimensionnel h-généralisé est instancié sur la totalité de ses dimensions. Par abus de langage, nous utiliserons aussi item pour les n-uplets fréquents qui seront instanciés niveau par niveau afin d'obtenir des items multidimensionnels h-généralisés conformément à la définition.

Pour extraire les séquences fréquentes, nous adoptons la philosophie *Générer/Elaguer*. En effet, nous conservons la propriété d'antimonotonie du support dans le contexte multidimensionnel (Tout sous-ensemble d'un ensemble fréquent est fréquent, tout sur ensemble d'un ensemble non fréquent est non fréquent).

Une fois les 1-fréquents extraits (items multidimensionnels h-généralisés les plus spécifiques), les k-candidats ($k \geq 2$) sont générés et testés afin de savoir s'ils sont fréquents. Cette opération est itérée tant que des k-candidats fréquents sont extraits.

Pour stocker les séquences candidates, nous utilisons une structure d'*arbre préfixé* (Masseglia et al. (1998)) afin d'éviter toute redondance.

Calcul du support d'une séquence

Les dimensions de référence permettent d'identifier tous les blocs de l'ensemble des données susceptibles de supporter une séquence ς. L'énumération de tous les blocs définis par les dimensions de référence D_R est indispensable pour calculer le support d'une séquence et définir ainsi si la séquence est fréquente ou non.

L'algorithme 1 vérifie pour chaque bloc de DB si la séquence est supportée ou non. Si la séquence est supportée, alors le support est incrémenté. L'algorithme retourne ensuite le ratio des blocs supportant ς.

L'algorithme 2 permet de vérifier si le bloc B supporte la séquence ς. Pour cela, cet algorithme cherche à instancier la séquence itemset par itemset en conjuguant *récursivité* et *ancrage*. L'ancrage correspond à une n-uplet du bloc B à partir duquel la séquence pourra être instanciée. Cet n-uplet correspond donc à une date à laquelle le premier item du premier itemset de la séquence est trouvé. À partir de cet n-uplet, seuls les n-uplets pertinents sont retenus, c'est-à-dire ceux qui partagent la même date. On ne retient donc que les n-uplets partageant la même date. Si le sous-bloc résultant de l'ancrage supporte l'itemset alors on appelle la fonction sur les autres itemsets de ς. Cet appel est effectué en réduisant l'espace de recherche aux seuls n-uplets dont la date est supérieure à la date de l'ancrage précédent, puisque l'on passe à l'itemset suivant, donc à une date ultérieure. Si l'ancrage échoue, on continue la recherche du premier itemset en tentant d'autres ancrages. L'appel récursif s'arrête dès que la séquence placée en paramètre d'entrée est vide. Une telle propriété signifie en effet que tous les itemsets de la séquence ont été trouvés. On retourne donc la valeur $vrai$. La valeur $faux$ est retournée si aucun ancrage n'a réussi et si tout le bloc a été parcouru sans succès.

Complexité

Afin de faciliter l'étude de complexité des algorithmes, nous posons les notations suivantes :

- n_B est le nombre de cellules du bloc B
- $m = |D_A|$ est le nombre de dimensions des items multidimensionnels.
- P_{max}, la profondeur maximale des taxonomies.

supportBloc (algorithme 2)

- Le bloc B étant ordonné par rapport à la dimension D_t, l'opération d'ancrage est réalisable en $O(\log n_C)$. En effet, il suffit de réaliser une recherche à l'aide d'un parcours dichotomique pour trouver tous les n-uplets respectant une certaine condition sur la date.

– Vérifier si un n-uplet supporte un item est réalisable en $O(P_{max} \times m)$. Il suffit de comparer les m dimensions de l'item avec celles du n-uplet.
– Dans le pire des cas, la complexité de l'algorithme est de $O(n_B \times P_{max} \times m \times \log n_B)$.

compterSupport (algorithme 1)

On appelle la fonction précedente pour tous les l blocs B_i de $\{B_{DB,D_R}\}$, l'ensemble des bloc de DB définis suivant D_R. Soit $n_{max} = \max n_{B_i}$. La complexité dans le pire des cas est donc : $O(l) \times O(n_{max} \times P_{max} \times m \times \log n_{max}) = O(l \times n_{max} \times P_{max} \times m \times \log n_{max})$

Fonction compterSupport Données : ς, DB, D_R
Résultat : le support de la séquence ς
début
  $Entier\ support \longleftarrow 0$;
  $Booleen\ seqSupportée$;
  $\mathscr{B}_{DB,D_R} \longleftarrow \{bloc\ de\ DB\ identifiés\ sur\ D_R\}$;
  pour chaque $B \in \mathscr{B}_{DB,D_R}$ **faire**
    $seqSupportée \longleftarrow supportBloc(\varsigma, B)$;
    si $seqSupportée$ **alors**
      $support \longleftarrow support + 1$;
  retourner $\left(\frac{support}{|B_{DB,D_R}|}\right)$
fin

Algorithme 1: Calcul du support d'une séquence (compterSupport)

3.3 Pourquoi les hiérarchies permettent une gestion plus fine de la valeur joker

La prise en compte des hiérarchies peut être vue comme un moyen plus fin de gérer les valeurs jokers. En effet, dans l'approche M^{2}SP, la racine d'une taxonomie représente la valeur joker * sur la dimension associée. Ainsi, si aucune instanciation n'est possible, aucune étiquette feuille ne peut donc convenir, alors on passe directement à la racine de la taxonomie (figure 9).

La prise en compte des hiérarchies, permet d'extraire des connaissances plus fines. En effet, les taxonomies proposent plusieurs alternatives par rapport à l'approche M^{2}SP quand on n'arrive pas à instancier une dimension. En effet, on ne passe pas directement de la feuille à la racine, on essaie d'instancier par l'ancêtre le plus spécifique de la feuille (figure 10).

FIG. 9 – *Gestion de la valeur joker (*)*

FIG. 10 – *Gestion des hiérarchies*

Exemple 8 (Comparaison avec M^{2}SP) *Pour un support fixé à* 2*, la prise en compte des hiérarchies permet d'extraire des connaissances qui ne peuvent pas être extraite par* MSP.

M^{2}SP

```
Fonction supportBloc
Données : ς, B
Résultat : Booléen
début
    *–initialisation–*
    booleen ItemSetTrouvé ⟵ faux
    sequence ⟵ ς
    itemset ⟵ sequence.first()
    item ⟵ itemset.first()
    *–condition d'arrêt de la recursivité –*
    si ς = ∅ alors
        retourner (vrai)
    *–parcours du bloc –*
    tant que tuple ⟵ B.next ≠ ∅ faire
        si supporte(tuple, item) alors
            itemSuivant ⟵ itemset.second()
            si itemSuivant = ∅ alors
                itemsetTrouvé ⟵ vrai
            *–Recherche de tous les items de l'itemset–*
            sinon
                *– On ancre par rapport à l'item (date)–*
                B' ⟵ σ_{date=cell.date}(B)
                tant que tuple' ⟵ B'.next() ≠ ∅ ∧ itemsetTrouvé = faux faire
                    si supporte(cell', itemSuivant) alors
                        itemSuivant ⟵ itemset.next()
                        si itemSuivant = ∅ alors
                            itemsetTrouvé ⟵ vrai
            si itemsetTrouvé = vrai alors
                *– recherche des autres itemsets–*
                retourner (supportBloc(sequence.tail(), σ_{date>tuple.date}(B)))
            sinon
                itemset ⟵ sequence.first()
                *–réduction de l'espace de recherche–*
                C ⟵ σ_{date>cell.date}(B)
    *– ς non supportée –*
    retourner (faux)
fin
```

Algorithme 2: supportBloc : *(Vérifie si une séquence est supportée par un bloc donné)*

– $(*, Chocolat)$, $(*, Cacahuetes)$, $(*, Smecta)$, $(*, Coca)$, $(*, Aspirine)$, $(*, Whisky)$
– $\langle\{(*, Chocolat)\}\{(*, Smecta)\}\rangle$, $\langle\{(*, Cacahuetes)\}\{(*, Aspirine)\}\rangle$

Prise en compte des hiérarchies

– $(Lieu, Chocolat)$,$(UE, Cacahuetes)$, $(Lieu, Smecta)$, $(Lieu, Coca)$, $(UE, Aspirine)$, $(Lieu, Whisky)$, $(UE, Alcool)$,
– $\langle\{(Lieu, chocolat)\}\{(Lieu, Smecta)\}\rangle$
$\langle\{(UE, Cacahuetes)\}\{(UE, Aspirine)\}\rangle$
$\langle\{(UE, Alcool)\}\{(UE, Aspirine)\}\rangle$

– $\langle\{(UE, Alcool), (UE, Cacahuetes)\}\{(UE, Aspirine)\}\rangle$

Le prise en compte des hiérarchies permet ainsi d'extraire des séquences plus complètes que l'approche M²SP.

4 Expérimentations

Des expérimentations ont été effectuées sur des données synthétiques. La base de données générée est composée de 5000 n-uplets. Les tests sont effectués sur 5 dimensions d'analyse. Ces premières expérimentations comparent les résultats obtenus en terme de nombre de fréquents extraits en fonction de la profondeur des taxonomies (degré de spécialisation) et du seuil de support considéré. Nous établissons une comparaison avec M²SP(-α) afin d'étudier la qualité des connaissances extraites.

FIG. 11 – *Nombre de séquences fréquentes par rapport à la profondeur de la taxonomie (minsup=0.3, nb_dim=5, deg = 3)*

FIG. 12 – *Nombre de séquences fréquentes par rapport à la profondeur de la taxonomie (minsup=0.4, nb_dim=5, deg = 4)*

FIG. 13 – *Nombre de séquences fréquentes par rapport au support (nb_dim=5, deg = 3, données denses)*

FIG. 14 – *Nombre de séquences fréquentes par rapport au support (nb_dim=5, deg = 4, profondeur = 4)*

Les figures 11 et 12 montrent le nombre de fréquents extraits en fonction de la profondeur des taxonomies pour un seuil de support fixé. Etendre la taxonomie d'un niveau engendre une spécialisation supplémentaire des données ($Boisson$ devient $Bois.Alcoolise$ ou $Coca$). Ainsi, quand les données se spécialisent, l'approche M²SP extrait moins de fréquents jusqu'à ne plus en extraire à partir d'un certain niveau de spécialisation. La prise en compte des hiérarchies apporte une certaine robustesse face à ce phénomène de spécialisation. En effet des connaissances sont extraites sur plusieurs niveaux de hiérarchies.

La figure 13 montre le nombre de fréquents extraits en fonction du support dans une base de données denses (faible cardinalité des dimensions d'analyse). Quand le support devient trop faible, la méthode M²SP extrait trop de fréquents. En effet beaucoup d'items ont une seule

dimension instanciée (différente de *), et ainsi cette méthode supporte rapidemment des 2-séquences trop générales. La prise en compte des hiérarchies introduit une forte capacité de subsomption qui permet de ne pas extraire un trop grand nombre de séquences inutiles.

Par contre quand les données sont moins denses, figure 14 (plus grande cardinalité des dimensions d'analyse dûe à une spécilisation plus importante des transactions), le nombre de fréquents extraits est similaire aux nombres de fréquents extraits dans des données plus denses alors que l'approche M^2SP extrait très peu de fréquents. Ceci souligne bien la robustesse de notre approche face à la qualité des données (denses, spécialisées).

5 Conclusion et perspectives

Dans cet article, nous définissons les motifs séquentiels multidimensionnels h-généralisés. Nous intégrons la prise en compte des hiérarchies à l'aide de taxonomies présentes sur chaque dimension d'analyse. Ceci permet la construction de séquences multidimensionnelles , de la forme $\langle\{(UE, Alcool), (UE, cacahuetes)\}\{(UE, aspirine)\}\rangle$, définies sur différents niveaux de hiérarchies indiquant que les personnes ayant acheté dans l'union européenne des cacahuètes et des boissons alcoolisées en même temps achètent ensuite dans l'union européenne de l'aspirine.

Nous définissons les différents concepts (item, itemset, motifs séquentiels multidimensionnels h-généralisés) et les algorithmes permettant la mise en œuvre de notre approche sont présentés et validés par des expérimentations effectuées sur des jeux de données synthétiques. Ces expérimentations montrent l'intérêt de HYPE, en particulier sa capacité à subsumer les connaissances ainsi que sa robustesse lors de l'extraction de connaissances devant la diversité des données (densité, spécialisation, ...). Notre approche peut s'appliquer dans le contexte OLAP en représentant un excellent outil pour le décideur.

Ce travail offre de nombreuses perspectives. L'efficacité de l'extraction peut être améliorée en s'appuyant sur des représentations condensées des connaissances extraites (clos, libres). L'utilisation de formes condensées peut permettre des élagages supplémentaire et ainsi améliorer la robustesse de l'extraction. D'autres propositions peuvent être effectuées pour la gestion des hiérarchies. Nous pouvons imaginer une gestion modulaire des hiérarchies où certaines dimensions n'auraient pas le même comportement que les autres afin s'adapter aux besoins de l'utilisateur (interdiction de dépasser le niveau de hiérarchie λ sur la dimension ξ, ...). La gestion des hiérarchies peut nous amener à définir une nouvelle méthode automatisée d'aide à la navigation dans les cubes de données.

Références

Agrawal, R. et R. Srikant (1995). Mining sequential patterns. In P. S. Yu et A. L. P. Chen (Eds.), *Proceedings of the Eleventh International Conference on Data Engineering, March 6-10, 1995, Taipei, Taiwan*, pp. 3–14. IEEE Computer Society.

de Amo, S., D. A. Furtado, A. Giacometti, et D. Laurent (2004). An apriori-based approach for first-order temporal pattern mining. In *XIX Simpósio Brasileiro de Bancos de Dados, 18-20 de Outubro, 2004,Brasília, Distrito Federal, Brasil, Anais/Proceedings*, pp. 48–62.

Han, J. (1997). Olap mining : Integration of olap with data mining. In S. Spaccapietra et F. J. Maryanski (Eds.), *Data Mining and Reverse Engineering : Searching for Semantics, IFIP TC2/WG2.6 Seventh Conference on Database Semantics (DS-7), October 7-10, 1997, Leysin, Switzerland*, Volume 124 of *IFIP Conference Proceedings*, pp. 3–20. Chapman & Hall.

Lee, C.-H. (2005). An entropy-based approach for generating multi-dimensional sequential patterns. In A. Jorge, L. Torgo, P. Brazdil, R. Camacho, et J. Gama (Eds.), *Knowledge Discovery in Databases : PKDD 2005, 9th European Conference on Principles and Practice of Knowledge Discovery in Databases, Porto, Portugal, October 3-7, 2005, Proceedings*, Volume 3721 of *Lecture Notes in Computer Science*, pp. 585–592. Springer.

Masseglia, F. (2002). *Algorithmes et applications pour l'extraction de motifs séquentiels dans le domaine de la fouille de données : de l'incrémental au temps réel*. Thèse de doctorat, Université de Versailles.

Masseglia, F., F. Cathala, et P. Poncelet (1998). The psp approach for mining sequential patterns. In J. M. Zytkow et M. Quafafou (Eds.), *Principles of Data Mining and Knowledge Discovery, Second European Symposium, PKDD '98, Nantes, France, September 23-26, 1998, Proceedings*, Volume 1510 of *Lecture Notes in Computer Science*, pp. 176–184. Springer.

Masseglia, F., M. Teisseire, et P. Poncelet (2004). Recherche des motifs séquentiels. *Revue Ingénierie des Systèmes d'Information (ISI), numéro spécial "Extraction de motifs dans les bases de données" 9*(3-4), pp. 183–210.

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 *Proceedings of the 2001 ACM CIKM International Conference on Information and Knowledge Management, Atlanta, Georgia, USA, November 5-10, 2001*, pp. 81–88. ACM.

Plantevit, M., Y. W. Choong, A. Laurent, D. Laurent, et M. Teisseire (2005). M^2sp : Mining sequential patterns among several dimensions. In A. Jorge, L. Torgo, P. Brazdil, R. Camacho, et J. Gama (Eds.), *Knowledge Discovery in Databases : PKDD 2005, 9th European Conference on Principles and Practice of Knowledge Discovery in Databases, Porto, Portugal, October 3-7, 2005, Proceedings*, Volume 3721 of *Lecture Notes in Computer Science*. Springer.

Srikant, R. et R. Agrawal (1996). Mining sequential patterns : Generalizations and performance improvements. In *Advances in Database Technology - EDBT'96, 5th International Conference on Extending Database Technology, Avignon, France, March 25-29, 1996, Proceedings*, pp. 3–17.

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

Hiérarchies & motifs séquentiels multidimensionnels

Summary

Data warehouses contain large volume of historized data stored at the end of analysis. Despite the evolution of OLAP analysis tools and methods, it is too difficult to provide data mining tools taking into account the specificities of these contexts (e.g. multidimensionnality, hierarchies, data historized). In this article, we propose an original method of extraction of sequential patterns taking into account the hierarchies. This method extracts from more precise knowledge and extends our preceding approach M^2SP. We define the concepts related to our problems as well as the associated algorithms. The experiments which we carried out show the interest of our proposal.