Motifs Séquentiels Multidimensionnels: Principes et Extensions

Marc Plantevit

Motifs Séquentiels Multidimensionnels : Principes et Extensions

Marc Plantevit

LIRMM, Université Montpellier 2, CNRS, 161 Rue Ada 34392
Montpellier, France
marc.plantevit@lirmm.fr
http://www.lirmm.fr/~plantevi

Résumé

Même si les techniques de fouille de données sont de plus en plus évoluées dans les contextes classiques (une seule dimension d'analyse), il reste néanmoins difficile de fournir aux utilisateurs des outils permettant la prise en compte des spécificités des contextes multidimensionnels e.g. multidimensionnalité, hiérarchies, données historisées. Dans cet article, nous présentons M^2SP, une méthode originale d'extraction de motifs séquentiels ainsi que HYPE, extension de M^2SP, qui permet la prise en compte des hiérarchies. Des connaissances plus précises sont ainsi extraites. Les expérimentations que nous avons menées montrent l'intérêt de nos proposition.

1 INTRODUCTION

L'extraction de motifs séquentiels est devenue, depuis son introduction par [1], une technique majeure du domaine de l'extraction de connaissances. Ils sont ainsi apparus afin de permettre la découverte de connaissances intégrant les notions de temporalité et séquentialité. Ils permettent de mettre en exergue des corrélations entre évènements en fonction de leur chronologie d'apparition. De telles règles seront par exemple de la forme : *les clients qui ont acheté un téléviseur et un lecteur DVD achètent plus tard un magnétoscope numérique*. La pertinence des règles et leur découverte est fondée sur la notion de *support* qui, de même que pour les règles d'assocation, spécifie dans quelle proportion les données de la base contiennent les données du motif. Cependant, les propositions existantes ne travaillent que sur une seule dimension d'analyse, nommée $produit$ dans les approches de type *étude du panier de la ménagère*. Ainsi, même si cette dimension peut être modifiée dans des applications de recherche de motifs séquentiels à d'autres domaines que le panier de la ménagère (par exemple dans le cadre de l'étude des comportements d'internautes [12]), il n'en reste pas moins qu'il n'est possible d'analyser qu'une seule dimension à la fois. Ainsi, il n'existe pas à

l'heure actuelle de méthode permettant de mettre en exergue des corrélations entre valeurs de différents attributs, par exemple pour découvrir des règles de la forme $\langle\{(surf, NY, 1), (housse, NY, 1)\}, \{(combi, SF, 1)\}\rangle$ indiquant qu'un nombre suffisant (au sens du support) de personnes ont acheté leur planche de surf et la housse à New York puis qu'un nombre suffisant de personnes ont acheté une combinaison à San Francisco. Si la littérature recense des contributions liées aux motifs séquentiels multidimensionnels proposées par l'équipe de Jiawei Han [8], celles-ci ne permettent pas de combiner plusieurs attributs au sein des motifs extraits pour ce qui est de la partie séquentielle, les multiples attributs n'apparaissant que pour restreindre le cadre dans lequel on trouve la séquence fréquente.

Dans cet article, nous présentons d'abord une méthode d'extraction de motifs séquentiels multidimensionnels (M^2SP). Nous proposons ensuite aussi une extension (HYPE) permettant la prise en compte des hiérarchies dans l'extractions de motifs séquentiels multidimensionnels (h-généralisés). Nous définissons les concepts associés à ces motifs et décrivons les algorithmes permettant leur extraction. Ces algorithmes sont validés par des expérimentations montrant l'intérêt de notre approche.

2 TRAVAUX CONNEXES

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).

2.1 Motifs séquentiels

L'extraction de motifs séquentiels est devenue, depuis son introduction par [1], 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 [5].

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 [4], SPADE [14]. Ces techniques sont fondées sur le paradigme *générer/élaguer* où des candidats sont générés puis ensuite élagués s'ils ne sont pas fréquents. Nous pouvons aussi citer PrefixSpan [7] qui permet l'extraction de motifs séquentiels sans génération de candidats.

Même si ces techniques permettent une extraction efficace des motifs séquentiels extraits, ceux-ci 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 différents 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. Comme le montre la figure 1, dans un contexte multidimensionnel, des connaissances supplémentaires sont extraites. Il n'y a plus seulement des corrélations entre items partageant le même itemset et entre itemsets. L'extraction de séquences dans un contexte multidimensionnel permet de mettre en évidence des corrélations entre les dimensions instanciées d'un item. Dans [8] 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 puisque le *pattern* décrit les clients qui tendent à supporter la totalité de la séquence, une séquence est donc identifiée par un seul pattern. Elle ne permet donc pas d'extraire des connaissances de la forme : $\langle\{(business, *, *, a)(*, chicago, *, b)\}, \{(*, *, young, c)\}\rangle$ alliant différents patterns multidimensionnels.
Dans [13], les auteurs proposent d'extraire des séquences au sein de séquence

[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.

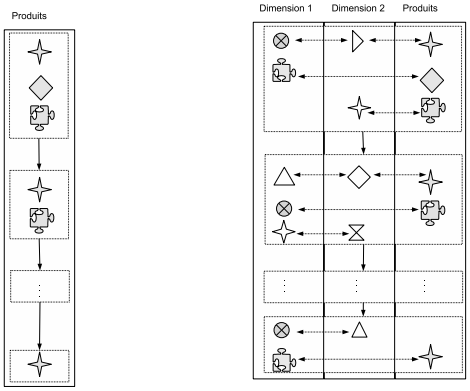

FIG. 1 – Des connaissances extraites dépendantes du contexte

de données multidimensionnelles organisées en différents 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 [2] qui proposent une approche basée sur la logique temporelle du premier ordre pour l'extraction de motifs séquentiels multidimensionnels, [3] proposent également une nouvelle méthode de génération des séquences multidimensionnelles présentes dans des bases de transactions. Cependant celle méthode de génération se réduit à des séquences d'items seulement.

2.3 Base exemple

Pour illustrer les différents concepts et définitions, nous proposons la base exemple fig. 2 qui décrit les achats de produit réalisés dans différentes villes du monde.

3 MOTIFS SÉQUENTIELS MULTIDIMENSIONNELS

Cet article reprend et étend les concepts et les approches développées dans nos travaux présentés dans [9], [10] et [11].

3.1 Données manipulées

Nous étendons les concepts présentés précédemment (client - date - items) en considérant non plus des attributs simples pour décrire les données, mais

D (Date)	B ($Bloc_{ID}$)	Pl (Lieu)	P (Produit)
1	1	Allemagne	B
1	1	Allemagne	Ca
2	1	Allemagne	A
3	1	Allemagne	Ch
4	1	Allemagne	S
1	2	France	Co
2	2	France	V
2	2	France	Ca
3	2	France	A
1	3	UK	W
1	3	UK	Ca
2	3	UK	A
1	4	LA	Ch
2	4	LA	S
3	4	NY	W
4	4	NY	Co

FIG. 2 – Base de données exemple DB

des ensembles d'attributs.

Nous supposons qu'il existe au moins une dimension (*e.g.* temporelle) dont le domaine est totalement ordonné.

Définition 1 (Partition des dimensions)
Pour tout ensemble de transactions DB défini sur un ensemble de n dimensions D, on considère une partition de D en trois sous-ensembles notés respectivement :

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

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

Définition 2 (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.

D	B	Pl	P
1	1	Allemagne	B
1	1	Allemagne	Ca
2	1	Allemagne	A
3	1	Allemagne	Ch
4	1	Allemagne	S

FIG. 3 – bloc (1)

D	B	Pl	P
1	2	France	Co
2	2	France	V
2	2	France	Ca
3	2	France	A

FIG. 4 – bloc (2)

D	B	Pl	P
1	3	UK	W
1	3	UK	Ca
2	3	UK	A

FIG. 5 – bloc (3)

D	B	Pl	P
1	4	LA	Ch
2	4	LA	S
3	4	NY	W
4	4	NY	Co

FIG. 6 – bloc (4)

FIG. 7 – Partition de DB (figure 2) en fonction de $D_R = \{B\}$

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 7.

3.2 Item, itemset, séquence multidimensionnels et valeur joker

Définition 3 (Item multidimensionnel)
Un item multidimensionnel $e = (d_1, \ldots, d_m)$ est un m-uplet défini sur les dimensions d'analyse D_A tel que $d_i \in dom(D_i)$.

Exemple 1
Etant donné $D_A = \{Pl, P\}$, $(LA, Ch), (France, A)$ et (UK, Ca) sont des items multidimensionnels.

D'après la définition précédente, un item ne peut être trouvé que s'il existe une combinaison de valeurs de domaines de D_A se retrouvant fréquemment dans les données de DB. Or il peut arriver qu'aucune combinaison ne soit fréquente. C'est pour cette raison que nous introduisons une valeur *joker* symbolisée par *. Cette valeur signifie que l'on ne tient pas compte de la valeur sur la dimension d'analyse. On appelle de tels items des items α-étoilés.

Définition 4 (Item multidimensionnel α-étoilé)
Soit $e_{[d_i/\delta]}$ la substitution dans e de d_i par δ, e est un item α-étoilé si les conditions suivantes sont vérifiées :
(i) $\forall i \in [1, m], d_i \in Dom(D_i) \cup \{*\}$,
(ii) $\exists i \in [1, m]$ *tel que* $d_i \neq *$,
(iii) $\forall d_i = *, \nexists \delta \in Dom(D_i)$ *tel que* $e_{[d_i/\delta]}$ *est fréquent.*

Exemple 2
Etant donné $D_A = \{Pl, P\}$, $(*, Ch), (France, *)$ *sont des items multidimensionnels α-étoilés.*

Définition 5 (Itemset multidimensionnel)
Un itemset multidimensionnel $i = \{e_1, \ldots, e_k\}$ *est un ensemble non vide d'items multidimensionnels.*

Exemple 3
$\{(*, V), (*, Ca)\}$ *est un itemset multidimensionnel.*

Il est important de remarquer que tous les items d'un même itemset sont deux à deux distincts par définition (un itemset est un ensemble).

Définition 6 (Séquence multidimensionnelle)
Une séquence multidimensionnelle $s = \langle i_1, \ldots, i_j \rangle$ *est une liste ordonnée par rapport à D_t et non vide d'itemsets multidimensionnels.*

Exemple 4
$\langle \{(*, V), (*, Ca)\}\{(*, A)\} \rangle$ *est une séquence multidimensionnelle α-étoilée.*

3.3 Support

Calculer le support d'une séquence multidimensionnelle α–étoilée 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 α-étoilée 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 α-étoilés 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 7
Un bloc B supporte une séquence α-étoilée $\varsigma = \langle i_{s1}, \ldots, i_{sl} \rangle$ *si* $\forall j \in [1, l]$, $\exists \delta_j \in Dom(D_t), \forall e = (d_{i_1}, \ldots, d_{i_m}) \in i_j, \exists t = (f, r, (x_{i_1}, \ldots, x_{i_m}), \delta_j) \in B$ *avec* $d_i = x_i$ *or* $d_i = *$ *et* $\delta_1 < \delta_2 < \ldots < \delta_l$.

Définition 8 (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 : $support(\varsigma) = \frac{|\{B \in B_{DB,D_R} \; t.q. \; B \; supporte \; \varsigma\}|}{|B_{DB,D_R}|}$

Nous avons posé les définitions fondamentales des motifs séquentiels multidimensionnels. Les algorithmes permettant la mise en œuvre de l'extraction de motifs séquentiels multidimensionnels α-étoilés ou non sont décrits dans la section 5.

Il est cependant très difficile d'extraire des connaissance de qualité en fonction du support. 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. L'utilisateur 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 proposons donc une extension de M^{2}SP (*HYPE* : *H*ierarch*Y* *P*attern *E*xtension) permettant la prise en compte des hiérarchies dans l'extraction de motifs séquentiels multidimensionnels.

4 EXTENSION : LA PRISE EN COMPTE DES HIÉRARCHIES

Dans le contexte dans lequel nous nous situons, nous considérons qu'il existe des relations hiérarchiques sur chaque dimension d'analyse[2]. Nous considérons que ces relations hiérarchiques sont matérialisées sous la forme de *taxonomie.*

4.1 Taxonomies

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

[2]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).

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.

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, $Bo = \widehat{Co}$ signifie que Bo est un ancêtre de Co dans la relation *Généralisation/Spécialisation*. Plus précisément, Bo est une instance plus générale que Co.

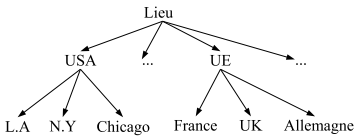

FIG. 8 – Taxonomie sur la dimension $Lieu$

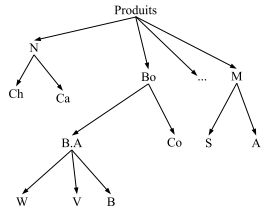

FIG. 9 – Taxonomie sur la dimension $Produit$

Les deux taxonomies associées à la base exemple (Fig. 2) décrivant les relations hiérarchiques entre les éléments de la dimension $Produits$ (resp. $Lieu$) sont représentées dans la figure 9 (resp. Fig. 8).

4.2 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 Bo s'il existe des instances plus spécifiques dans la taxonomie comme Co.

4.3 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 9 (Item multidimensionnel h-généralisé)

Un item multidimensionnel h-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 5
$(USA, Bo), (France, B.A)$ *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 10 (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 6 (relations hiérarchiques entre items h-généralisés)

- $(USA, Bo) >_h (USA, Co)$.
- $(France, V) <_h (UE, B.A)$.
- $(France, V)$ *et* (USA, Co) *sont incomparables.*

Définition 11
Une transaction b *supporte un item* e *si* $\Pi_{D_A}(b) <_h e$.

Exemple 7
La transaction $(1, 1, France, V)$ *supporte l'item* $(UE, B.A)$.

Définition 12 (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 8
$\{(France, V), (USA, Co)\}$ *est un itemset multidimensionnel h-généralisé alors que* $\{(France, V), (UE, B.A)\}$ *n'est pas un itemset multidimensionnel h-généralisé car* $(France, v) <_h (UE, B.A)$.

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

Définition 13 (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 9
$\langle \{(France, V), (USA, Co)\}, \{(Allemagne, B)\} \rangle$ *est une séquence multidimensionnelle h-généralisée.*

Définition 14 (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 10
- *La séquence* $\langle \{(France, V)\}, \{(Allemagne, B)\} \rangle$ *est une sous-séquence de la séquence* $\langle \{(France, V), (USA, Co)\}, \{(Allemagne, B)\} \rangle$.
- *La séquence* $\langle \{(France, V)\}, \{(Allemagne, B)\} \rangle$ *est une sous-séquence de la séquence* $\langle \{(France, B.A), (USA, Bo)\}, \{(UE, B.A)\} \rangle$.
- *La séquence* $\langle \{(UE, V)\}, \{(Allemagne, B)\} \rangle$ *n'est pas une sous-séquence de la séquence* $\langle \{(France, V), (USA, Co)\}, \{(Allemagne, B)\} \rangle$ *car* $(UE, V) \not\leq_h (France, V)$*, l'inclusion hiérarchique n'étant pas respectée.*

4.4 Support

Le calcul du support d'une séquence se définit comme précédemment, il faut compter le nombre de blocs qui supportent la séquence.

Définition 15 (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 11
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 $\varsigma = \langle\{(UE, B.A), (UE, caca\text{-}huetes)\}\ \{(UE, aspirine)\}\rangle$. Pour que la séquence soit fréquente, au moins deux blocs de la partition de DB doivent supporter la séquence.
1. bloc $(\mathbf{1})$ *(Fig. 3). 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 B.A. Ainsi à la date* 1*, nous avons bien le premier itemset* $\{(UE, B.A), (UE, cacahuetes)\}$ *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. 4). France est une instance de UE et V est une instance d'B.A. Nous retrouvons bien la séquence* ς *dans ce bloc*
3. bloc $(\mathbf{3})$ *(Fig. 5). UK est une instance de UE et whisky est une instance d'B.A. Ce bloc supporte la séquence* ς*.*
4. bloc $(\mathbf{4})$ *(Fig. 6). 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.*

5 ALGORITHMES

Nous décrivons la démarche adoptée pour la génération des items candidats et des séquences candidates. Nous décrivons ensuite les algorithmes associés au calcul du support des séquences. Nous nous situons dans le contexte le plus général qui est celui des motifs séquentiels h-généralisés extrait par HYPE, extension de M^{2}SP.

5.1 Fonctionnement général

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 candidats

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[3], 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 $m-1$ fois jusqu'à l'obtention des items multidimensionnels h-généralisés (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 16). Pour être $\bowtie$-compatibles,

[3]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.

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 16 ($\bowtie$-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 17 (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.

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* ([6]) afin d'éviter toute redondance.

5.2 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 : $O(l) \times O(n_{max} \times P_{max} \times m \times \log n_{max}) = O(l \times n_{max} \times P_{max} \times m \times \log n_{max})$

Algorithme 1 – Calcul du support d'une séquence (compterSupport)
Fonction compterSupport Données : ς, DB, D_R
Résultat : le support de la séquence ς
début
  $Entier\ support \longleftarrow 0$;
  $Booleen\ seqSupportée$;
  $\mathscr{B}_{DB,D_R} \longleftarrow \{blocs\ de\ DB\ identifiés\ sur\ D_R\}$;
  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

6 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 10).

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 11).

Exemple 12 (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 extraites par M^{2}SP.

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

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

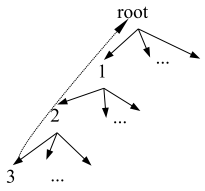

FIG. 10 – Gestion de la valeur joker (*)

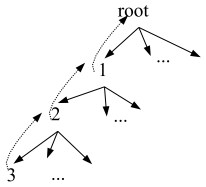

FIG. 11 – Gestion des hiérarchies

M^{2}SP

- $(*, Ch)$, $(*, Ca)$, $(*, S)$, $(*, Co)$, $(*, A)$, $(*, W)$
- $\langle\{(*, Ch)\}\{(*, S)\}\rangle$, $\langle\{(*, Ca)\}\{(*, A)\}\rangle$

Prise en compte des hiérarchies

- $(Lieu, Ch)$,(UE, Ca), $(Lieu, S)$, $(Lieu, Co)$, (UE, A), $(Lieu, W)$, $(UE, B.A)$,
- $\langle\{(Lieu, Ch)\}\{(Lieu, S)\}\rangle$
  $\langle\{(UE, Ca)\}\{(UE, A)\}\rangle$
  $\langle\{(UE, B.A)\}\{(UE, A)\}\rangle$

- $\langle\{(UE, B.A), (UE, Ca)\}\{(UE, A)\}\rangle$

Le prise en compte des hiérarchies permet ainsi d'extraire des séquences plus complètes que l'approche M^{2}SP.

7 EXPÉRIMENTATIONS

Des expérimentations ont été effectuées sur des données synthétiques. Nous avons réalisé deux séries d'expérimentations. Une série complète afin de montrer l'intérêt de M^{2}SP vis à vis de l'approche pionnière du domaine [8]. Ces expérimentations montre le comportement global de notre approche ainsi que sa robustesse. Nous nous sommes ensuite focalisé sur la qualité des connaissances extraites afin de souligner l'intérêt de HYPE, l'extension de M^{2}SP. Nous montrons que HYPE permet de faire face au "dilemme" du support.

7.1 M^{2}SP

Les expérimentations montrent l'intérêt et le passage à l'échelle de notre approche. Comme beaucoup de bases dans le monde réel possède une dimension quantitative, nous distinguons une dimension quantitative. Dans le but de souligner le rôle particulier de cette dimension quantitative, nous considérons deux types d'extraction de motifs séquentiels : (i) valeur joker possible sur toutes les dimensions à l'exception de la dimension quan-

titative (M^2SP-$alpha$), (ii) valeur joker sur toutes les dimensions(M^2SP-$alpha$-mu). Nous notons que le cas(ii) correspond au cas présenté dans les définitions. Nos expérimentations peuvent être vues comme étant conduites sur une table de faits d'une base de données multidimensionnelles où la dimension quantitative représente la mesure. Les expérimentations montrent le nombre de séquences extraite ou le temps d'exécution en fonction de nombreux paramètres ($|D_A|$,$|DB|$, cardinalité moyenne des dimensions d'analyse

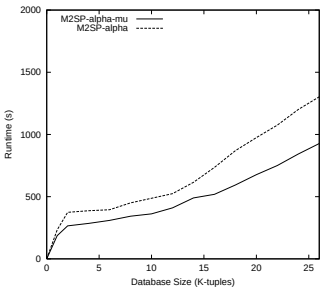

FIG. 12 – Temps d'exécution en fonction de la taille de la base(minsup=0.5, nb_dim=15, avg_card = 20)

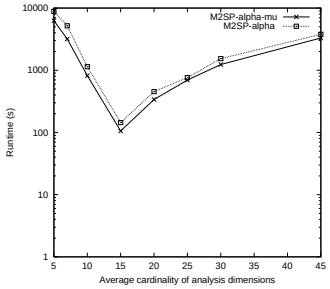

FIG. 13 – Temps d'exécution en fonction de la cardinalité moyenne des dimensions d'analyse (minsup=0.8, DB_size=12000, nb_dim=15)

La figure 12 montre le passage à l'échelle de notre approche puisque le temps d'exécution augmente quasi linéairement par rapport à la taille de la base (de $1,000$ n-uplets à $26,000$ n-uplets). La figure 13 montre le comportement du temps d'exécution lorsque la cardinalité moyenne des dimensions d'analyse varie. Quand la cardinalité est petite, la plupart des candidats sont fréquents. A contratio, lorsque la cardinalité est élévée, de nombreux candidats sont générés et peu d'entre eux sont retenus. Les figures 14 et 15 montrent le comportement de notre approche quand le nombre de dimension d'analyse change. Le nombre d'items fréquents augmente quand le nombre de dimensions d'analyse augmente, ce qui engendre l'augmentation du nombre de séquences fréquentes. Les figures 16 et 17 montrent la différence entre le nombre de séquences extraites par M^{2}SP et par l'approche décrite par [8]. Cette faculté d'extraction souligne l'intérêt de notre approche.

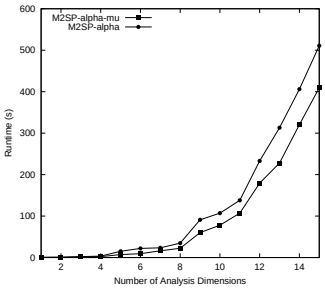

FIG. 14 – Temps d'exécution en fonction du nombre de dimensions d'analyse (minsup=0.5, DB_size=12000, nb_dim=15, avg_card=20)

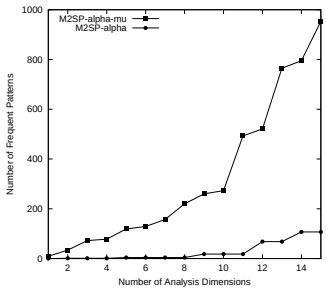

FIG. 15 – Nombre de séquences fréquentes en fonction du nombre de dimensions d'analyse (minsup=0.5, DB_size=12000, nb_dim=15, avg_card=20)

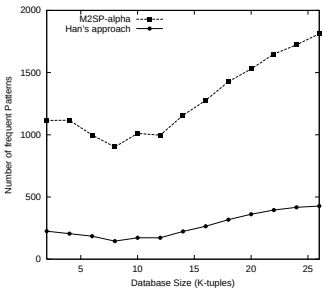

FIG. 16 – Nombre de séquences fréquentes en fonction du nombre de la taille de la base (minsup=0.5, nb_dim=15, avg_card=20)

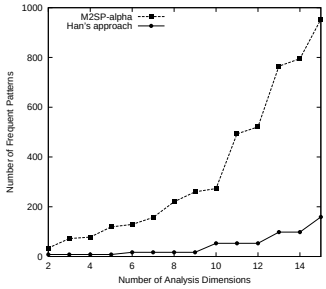

FIG. 17 – Nombre de séquences fréquentes en fonction du nombre dimensions d'analyse(minsup=0.5, DB_size=12000, avg_card=20)

7.2 HYPE

Pour montrer l'intérêt de HYPE, l'extension de M^{2}SP, nous avons simulé une base de données (5000 n-uplets, $|D_A = 5$) où les éléments des dimensions d'analyse sont organisées en différents niveaux de hiérachies. 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 M^{2}SP(-α) afin d'étudier la qualité des connaissances extraites.

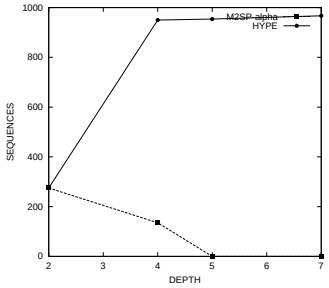

FIG. 18 – Nombre de séquences fréquentes par rapport à la profondeur de la taxonomie (minsup=0.3, nb_dim=5, deg = 3)

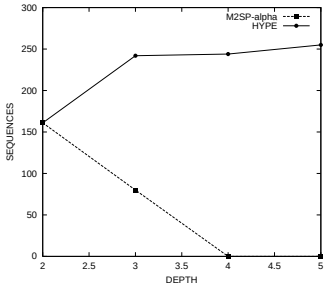

FIG. 19 – Nombre de séquences fréquentes par rapport à la profondeur de la taxonomie (minsup=0.4, nb_dim=5, deg = 4)

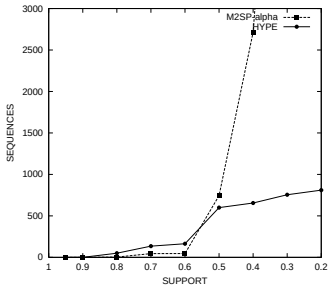

FIG. 20 – Nombre de séquences fréquentes par rapport au support (nb_dim=5, deg = 3, données denses)

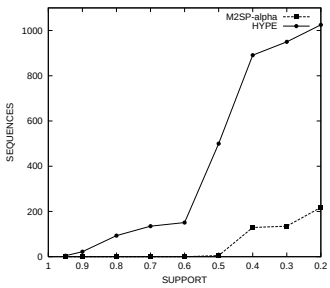

FIG. 21 – Nombre de séquences fréquentes par rapport au support (nb_dim=5, deg = 4, profondeur = 4)

Les figures 18 et 19 montrent le nombre de fréquents extraits en fonction de la profondeur des taxonomies pour un seuil de support fixé. Etendre la taxo-

nomie d'un niveau engendre une spécialisation supplémentaire des données ($Boisson$ devient $Bois.Alcoolisée$ ou $Coca$). Ainsi, quand les données se spécialisent, l'approche M^{2}SP 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 20 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 M^{2}SP 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 21 (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^{2}SP extrait très peu de fréquents. Ceci souligne bien la robustesse de notre approche face à la qualité des données (denses, spécialisées).

8 CONCLUSION ET PERSPECTIVES

Dans cet article, nous définissons les motifs séquentiels multidimensionnels α-étoilés qui sont étendus aux motifs séquentiels multidimensionnels h-généralisés. Ceci permet l'extraction de séquence multidimensionnelle définies sur plusieurs niveaux de hiérarchies. Nous définissons les différents concepts (item, itemset, motifs séquentiels multidimensionnels α-étoilés ou h-généralisés) et les algorithmes permettant la mise en œuvre de nos approches sont présentés et validés par des expérimentations effectuées sur des jeux de données synthétiques. Ces expérimentations montrent l'intérêt de M^{2}SP par rapport aux approches existantes ainsi que sa robustesse. L'intérêt de l'extraction des motifs séquentiels multidimensionnels est accru avec la prise en compte des hiérarchies. Elles montrent aussi la capacité de HYPE à subsumer les connaissances ainsi que sa robustesse d'extraction face à la diversité des données (densité, spécialisation, ...). Nos travaux peuvent 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 hiérar-

chies où certaines dimensions n'auraient pas le même comportement que les autres afin de s'adapter aux besoins de l'utilisateur (interdiction de dépasser le niveau de hiérarchie λ sur la dimension ξ, ...).

RÉFÉRENCES

[1] R. Agrawal et R. Srikant. Mining sequential patterns. In Philip S. Yu et Arbee L. P. Chen, éditeurs, *Proceedings of the Eleventh International Conference on Data Engineering, March 6-10, 1995, Taipei, Taiwan*, pages 3–14. IEEE Computer Society, 1995.

[2] S. de Amo, D. A. Furtado, A. Giacometti et D. Laurent. 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*, pages 48–62. 2004.

[3] Chang-Hwan Lee. An entropy-based approach for generating multi-dimensional sequential patterns. In Alípio Jorge, Luís Torgo, Pavel Brazdil, Rui Camacho et João Gama, éditeurs, *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*, pages 585–592. Springer, 2005.

[4] F. Masseglia. *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, 2002.

[5] F. Masseglia, M. Teisseire et P. Poncelet. 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, 2004.

[6] Florent Masseglia, Fabienne Cathala et Pascal Poncelet. The psp approach for mining sequential patterns. In Jan M. Zytkow et Mohamed Quafafou, éditeurs, *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*, pages 176–184. Springer, 1998.

[7] J. Pei, J. Han, B. Mortazavi-Asl, J. Wang, H. Pinto, Q. Chen, U. Dayal et M.-C. Hsu. Mining sequential patterns by pattern-growth : The prefixspan approach. *IEEE Transactions on Knowledge and Data Engineering*, 16(10), 2004.

[8] H. Pinto, J. Han, J. Pei, K. Wang, Q. Chen et U. Dayal. 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, pages 81–88. ACM, 2001.

[9] Marc Plantevit, Yeow Wei Choong, Anne Laurent, Dominique Laurent et Maguelonne Teisseire. Motifs séquentiels multidimensionnels etoilés. In Véronique Benzaken, éditeur, *BDA 2005, Actes des* 21^{es} *journées de Bases de données avancées, Saint-Malo, 17-20 octobre 2005*.

[10] Marc Plantevit, Yeow Wei Choong, Anne Laurent, Dominique Laurent et Maguelonne Teisseire. M^2sp : Mining sequential patterns among several dimensions. In Alípio Jorge, Luís Torgo, Pavel Brazdil, Rui Camacho et João Gama, éditeurs, *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, 2005.

[11] Marc Plantevit, Anne Laurent et Maguelonne Teisseire. HYPE : Prise en compte des hiérarchies lors de l'extraction de motifs séquentiels multidimensionnels. In *EDA 2006, Actes de la deuxième journée francophone sur les Entrepôts de Données et l'Analyse en ligne, Versailles, 19 juin 2006*. Cépaduès.

[12] D. Tanasa, B. Trousse et Florent Masseglia. *Mesures de l'internet*, chapitre Fouille de données appliquées au logs web : état de l'art sur le Web Usage Mining, pages 126–143. édition Les Canadiens en Europe, 2004.

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

[14] Mohammed Javeed Zaki. Spade : An efficient algorithm for mining frequent sequences. *Machine Learning*, 42(1/2) :31–60, 2001.