Motifs séquentiels multidimensionnels étoilés

M. Plantevit*, Y.W. Choong**,***, A. Laurent*, D. Laurent***, M. Teisseire*

* LIRMM - Université Montpellier 2
CNRS UMR 5506
161, rue Ada
34392 Montpellier Cedex 5

** HELP University College
BZ-2 Pusat Bandar Damansara
50490 Kuala Lumpur - Malaisie

*** LICP - Université de Cergy Pontoise
2, avenue Adolphe Chauvin BP 222
95302 Cergy-Pontoise Cedex

Résumé

L'extraction de motifs séquentiels est un domaine de la fouille de données permettant de rechercher des corrélations fréquentes entre des valeurs en prenant en compte leur chronologie d'apparition. Dans le contexte du panier de la ménagère, ce type de méthodes permet par exemple l'extraction de règles de la forme $\langle (TV, DVD)(magneto\_numerique) \rangle$, indiquant qu'un nombre suffisamment important (au sens du support) de clients ont acheté d'abord un téléviseur et un lecteur DVD puis un magnétoscope numérique. Si de nombreux travaux permettent l'extraction de tels motifs, il n'en reste pas moins que ces motifs sont parfois très pauvres par rapport aux bases de données qu'ils décrivent. En effet, 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 $\langle \{(surf, NY), (housse, NY)\}, \{(combi, SF)\} \rangle$ indiquant qu'un nombre important de personnes ont acheté leur planche de surf et la housse à New York avant de se rendre à San Francisco où ils ont acheté une combinaison. Si la littérature recense des contributions liées aux motifs séquentiels multidimensionnels proposées par l'équipe de Jiawei Han, celles-ci ne permettent pas de combiner plusieurs attributs au sein des motifs extraits. Dans cet article, nous montrons donc les limites des approches existantes et proposons une approche complète d'extraction de motifs multidimensionnels multi-attributs. 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.

Mots clés : Fouille de données, motifs séquentiels, bases de données multidimensionnelles.

1 Introduction

Les motifs séquentiels sont apparus afin de permettre la découverte de règles intégrant la notion de temporalité et d'enchaînement d'événements. 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.*

De nombreux travaux ont traité cette problématique. Les recherches se sont notamment intéressées à l'extraction efficace des motifs face à de gros volumes de données [SA96].

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 des motifs séquentiels à d'autres domaines que le panier de la ménagère (par exemple dans le cadre de l'étude des comportements d'internautes [TTM04]), 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 avant de se rendre à San Francisco où ils ont acheté une combinaison.

Si la littérature recense des contributions liées aux motifs séquentiels multidimensionnels proposées par l'équipe de Jiawei Han [PHP+01], 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 montrons donc les limites des approches existantes et proposons une approche complète d'extraction de motifs multidimensionnels multi-attributs. Nous définissons les concepts associés et présentons les algorithmes permettant leur extraction. Des expérimentations montrent l'intérêt de notre approche. Plusieurs types de motifs séquentiels multidimensionnels sont définis, selon que toutes les dimensions d'analyse sont spécifiées ou non, et selon que la valeur de mesure est spécifiée ou non. On parle de motif multidimensionnel α-étoilé quand une dimension d'analyse au moins n'est pas spécifiée. Par exemple, le motif $\langle\{(surf, NY, 1), (housse, NY, 1)\}, \{(combi, *, 1)\}\rangle$ indique que les personnes ayant acheté une planche de surf et une housse à New York achètent par la suite une combinaison (sans précision de la ville). On parle de motif multidimensionnel μ-étoilé quand la valeur de mesure n'est pas spécifiée. Par exemple, le motif $\langle\{(lecteur\_dvd, FNAC, \circledast)\}, \{(DVD, supermarche, \circledast)\}\rangle$ indique que les personnes achetant un ou plusieurs lecteur(s) DVD à la FNAC achètent ensuite un ou plusieurs DVD dans un supermarché (sans précision de la quantité).

Cet article est organisé de la manière suivante. La section 2 introduit brièvement les concepts liés aux motifs séquentiels, présente les travaux de la littérature traitant le cas où plusieurs attributs sont considérés et introduit les principaux concepts liés aux bases de données multidimensionnelles. La section 3 introduit notre contribution, M^2SP, pour la définition de motifs séquentiels multidimensionnels. La section 4 présente l'extension de cette proposition à la prise en compte de valeurs joker sur les dimensions d'analyse et sur la mesure. La section 5 présente les algorithmes associés à notre proposition. La section 6 présente les premières expérimentations menées qui confirment l'intérêt de notre approche. Enfin, la section 7 conclut et présente les principales perspectives associées à ce travail.

2 Travaux connexes

Dans cette section, nous présentons les principes des motifs séquentiels ainsi que les approches de la littérature ayant traité le cas où plusieurs attributs sont considérés. Dans un dernier temps, nous présentons brièvement les notations sur lesquelles nous nous appuyons dans le contexte des bases de données multidimensionnelles.

2.1 Motifs séquentiels

Nous présentons ici très brièvement les concepts fondamentaux liés aux motifs séquentiels. Le lecteur désirant plus de détails se référera à [MTP04]. Les bases de données sur lesquelles s'appuie la recherche de motifs séquentiels doivent comporter trois données liées à la problématique du panier de la ménagère : la première représente un identifiant (souvent appelé *client*), la deuxième représente une liste de valeurs (souvent appelé *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 techniques existent dont PSP [Mas02] ou encore PrefixSpan [PHMA+04]. Ces techniques sont fondées sur des recherches par niveau *a la* APriori.

2.2 Motifs séquentiels multidimensionnels

Nous présentons ici les trois approches de la littérature ayant abordé la problématique des motifs séquentiels multidimensionnels. et montrons en quoi elles sont limitées par rapport à notre proposition.

2.2.1 Approche de Pinto et al.

Dans [PHP+01] les auteurs sont les tout 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 dimension telles que *Type de consommateur, Ville, Age*, comme illustré par le tableau 1. L'approche proposée repose sur la définition de séquence multidimensionnelle selon le schéma $A_1, ..., A_m, S$ où les A_i correspondent aux dimensions sur lesquelles les données sont décrites et S représente les séquences des achats réalisés par le client ordonnés selon le temps. Une séquence multidimensionnelle possible est donc du type $(id_1,(a_1, ..., a_m),s)$ avec $a_i \in A_i \cup \{*\}$.

Exemple 1 *Le tableau 1 donne un exemple de base de données intégrant des séquences d'achat de produit (dans la dernière colonne) décrites selon plusieurs dimensions. La première colonne correspond à l'identifiant de chaque client (cid). La deuxième colonne correspond au groupe de consommateur (Customer Group) associé. La troisième colonne correspond à la ville où habite le client. La quatrième colonne correspond quant à elle au groupe d'âge du client. Dans cet exemple, la séquence* $((*, Chicago, *),\langle bf\rangle)$ *a un support de* 100%. *Les clients habitant Chicago (cid 20 et 30) ont en effet tous acheté un b puis un f.*

La notion de *pattern multidimensionnel* est introduite et correspond aux co-occurrences fréquentes des valeurs d'attributs. La notation $*$ est

introduite pour signifier *toutes valeurs confondues* et permettre l'écriture des patterns le long de toutes les dimensions. Par exemple, si $Chicago$ se retrouve dans un nombre suffisant de lignes de la base, on notera que $(*, Chicago, *)$ est un pattern multidimensionnel fréquent. Pour trouver les séquences multidimensionnelles fréquentes, deux démarches équivalentes sont proposées :

- Rechercher tous les motifs séquentiels sur la base (sans se soucier des dimensions descriptives) puis, pour les bases réduites aux données associées aux motifs obtenus, rechercher les patterns multidimensionnels associés.
- Rechercher tous les patterns multidimensionnels sur la base (sans se soucier des motifs séquentiels) puis, pour les bases réduites aux données associées aux patterns obtenus, rechercher les motifs séquentiels fréquents.

Pinto et al permettent ainsi d'obtenir des motifs multidimensionnels fréquents *intra pattern* puisque chaque séquence est en fait fréquente sur la sous-base décrite par le pattern multidimensionnel associé.

Cette approche est donc très interessante. Cependant, notre proposition en est une extension et introduit, en plus de ces motifs, des motifs multidimensionnels *inter pattern*. Nous souhaitons en effet obtenir des motifs tels que $\langle\{(business, *, *, a)(*, chicago, *, b)\}, \{(*, *, young, c)\}\rangle$ alliant différents patterns multidimensionnels. Ceux-ci ne peuvent en effet pas être découverts dans la proposition précédente.

2.2.2 Approche de Yu et Chen

Dans [YC05], les auteurs tentent 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 différents 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. De même que dans l'approche proposée par [PHP+01], les séquences ne concernent qu'une seule dimension (les pages internet dans ce cas).

2.2.3 Approche de De Amo et al.

L'approche de [dAFGL04] est basée sur la logique temporelle du premier ordre. Cette proposition est proche de la notre mais présente des limites que nous étendons ici. Notre proposition est une généralisation de [dAFGL04] puisque dans notre approche : (i) la notion de groupe n'est pas codée dans la base mais déterminée par l'utilisateur et (ii) plusieurs attributs peuvent apparaître dans les séquences. Concernant ce dernier point, les auteurs mentionnent la possibilité de traiter plusieurs attributs, cependant le formalisme mis en place n'est pas étendu de manière complète au cas multi-attributs. De plus, dans [dAFGL04] les auteurs construisent leurs séquences candidates à partir d'une opération de jointure sur les séquences, contrairement à notre approche qui s'appuie sur les travaux déjà validés de construction tels que [Mas02].

2.3 Bases de données multidimensionnelles

Il n'existe pas à l'heure actuelle de consensus concernant les modèles de bases de données multidimensionnelles [CD97, Mar98]. Dans cet article, nous adoptons un point de vue relationnel au niveau du stockage des cubes (ROLAP). L'ensemble des notations utilisées dans cet article sont reprises dans le tableau 6.

cid	Cust-Grp	City	Age-grp	product-sequence
10	business	Boston	middle	$\langle(bd)cba\rangle$
20	professional	Chicago	young	$\langle(bf)(ce)(fg)\rangle$
30	business	Chicago	middle	$\langle(ah)abf\rangle$
40	education	New York	retired	$\langle(be)(ce)\rangle$

TAB. 1 – Exemple de base de données incluant plusieurs dimensions d'analyse

Nous considérons les bases de données comme étant organisées en cubes eux-mêmes composés de *cellules*. Dans une représentation relationnelle, chaque cellule correspond à un n-uplet de la base. Nous considérons $D = \{D_1, \cdots, D_n\}$ un ensemble de dimensions définies sur leurs domaines actifs[1] respectifs $Dom(D_1), \ldots, Dom(D_n)$, et une mesure M (correspondant le plus souvent à une valuation numérique) définie sur le domaine $Dom(M)$. Le domaine de la mesure inclut la valeur nulle.

Définition 1 (HyperCube) *Un hypercube (ou simplement cube) de données défini sur les dimensions D_1, ..., D_n est un n-uplet $\langle Dom(D_1), \ldots, Dom(D_n), Dom(M), C\rangle$ où C est une application $C : Dom(D_1) \times \ldots \times Dom(D_n) \rightarrow Dom(M)$. Par abus de langage, on notera C un tel cube.*

Définition 2 (Cellule) *On appelle cellule d'un cube à n dimensions D_1,...,D_n, un n-uplet de la forme : $\langle(d_1, \cdots, d_n), \mu\rangle$ où :*
- $\forall i \in [1 \cdots n],\ d_i \in Dom(D_i)$
- $\mu \in Dom(M)$

Définition 3 (Projection) *Soit $cell = \langle(d_1, \cdots, d_n), \mu\rangle$ une cellule. On note $cell.D_i = d_i$ la restriction de cell sur la dimension D_i.*

Exemple 2 *Si l'on considère l'hypercube du tableau 2 donné sous forme tabulaire, une cellule de cet hypercube sera par exemple : $\langle(1, Educ, Chicago, A, clou), 50\rangle$. Dans cette représentation, les cellules pour lesquelles la valeur de mesure est nulle sont omises. Ainsi, la cellule $\langle(1, Educ, Miami, A, clou), NULL\rangle$ appartient au cube mais n'est pas représentée.*

Définition 4 (Sous-Cube) *Soit C un cube à n dimensions $D = \{D_1, \cdots, D_n\}$, un sous-cube SC de C défini sur les k dimensions $\{D_{j_1}, \cdots, D_{j_k}\} \subseteq D$ est un n-uplet $\langle d_{j_1}, \cdots, d_{j_k}\rangle$. Le sous-cube SC correspond à l'ensemble des cellules telles que : $c = \langle(c_1, \cdots, c_n), \mu\rangle$ avec $\mu \in Dom(M)$ et $\forall j \in [j_1, \cdots, j_k]\ \ c_j = d_j$.*

Un sous-cube est défini à partir de dimensions sur lesquelles les valeurs sont fixées. Un cube peut donc être partitionné en un ensemble de sous-cubes le long d'un ensemble de dimensions, tel que l'opération *split* du modèle multidimensionnel le réalise [CD97, Mar98].

Exemple 3 *Soit le cube C décrit par le tableau 2. Si l'on considère les dimensions Cust-Grp et City, il est possible de diviser ce cube en trois sous-cubes définis par les n-uplets $\langle Educ, Chicago\rangle$ (cf tableau 3), $\langle Educ, LosAngeles\rangle$ (cf tableau 4) et $\langle Reti., Miami\rangle$ (cf tableau 5). Ces trois sous-cubes définissent une partition de C.*

Définition 5 (Opérateur de sélection slice) *Soit θ un prédicat de la forme D_i op val où op est un opérateur de comparaison. On note $\sigma_\theta(C)$ le cube résultant d'une sélection sur les valeurs de dimension (slice) du cube C. Pour toute cellule $cell = \langle(d_1, \cdots, d_n), \mu\rangle$ de C, on a $cell$ dans $\sigma_\theta(C)$ si la condition θ est satisfaite et $cell' = \langle(d_1, \cdots, d_n), NULL\rangle$ dans $\sigma_\theta(C)$ sinon.*

[1] Le domaine actif d'une dimension est, comme dans le modèle relationnel, la partie finie du domaine de la dimension contenant les valeurs présentes dans la base.

Date	Cust-Grp	City	Age	Product	Measure
1	Educ	Chicago	A	clou	50
1	Educ	Chicago	B	pneu	2
1	Educ	Los Angeles	A	clou	30
1	Reti.	Miami	C	clou	20
1	Reti.	Miami	C	marteau	2
2	Educ	Chicago	B	rustine	10
2	Educ	Chicago	B	pneu	3
2	Educ	Los Angeles	A	clou	20
3	Educ	Los Angeles	B	rustine	15

TAB. 2 – cube C

Date	Cust-Grp	City	Age	Product	Measure
1	Educ	Chicago	A	clou	50
1	Educ	Chicago	B	pneu	2
2	Educ	Chicago	B	pneu	3
2	Educ	Chicago	B	rustine	10

TAB. 3 – sous-cube de C défini par $\langle Educ, Chicago \rangle$

Date	Cust-Grp	City	Age	Product	Measure
1	Educ	Los Angeles	A	clou	30
2	Educ	Los Angeles	A	clou	20
3	Educ	Los Angeles	B	rustine	15

TAB. 4 – sous-cube de C défini par $\langle Educ, LosAngeles \rangle$

Date	Cust-Grp	City	Age	Product	Measure
1	Reti.	Miami	C	clou	20
1	Reti.	Miami	C	marteau	2

TAB. 5 – sous-cube de C défini par $\langle Reti., Miami \rangle$

Par exemple, on peut sélectionner les données du cube correspondant à la date 1 en calculant $\sigma_{Date=1}$. Ceci revient à ne conserver que les cellules correspondant aux cinq premières lignes du tableau 2. Les autres cellules voient leur valeur μ mise à NULL, ce qui revient à les éliminer de la mémoire. On notera en effet que du point de vue de l'implémentation, seules les cellules dont la valeur de mesure est non nulle seront conservées pour être traitées, ce qui réduit les temps de parcours de la base de données.

3 M^{2}SP

Dans cette section, nous présentons une nouvelle définition des motifs séquentiels multidimensionnels, nommée M^{2}SP (**M**ining **M**ultidimensional **S**equential **P**atterns).

3.1 Données manipulées

Dans le cadre de ce travail, on suppose que parmi toutes les dimensions définissant un cube, il existe une dimension D_t (dimension temporelle) dont le domaine est totalement ordonné.

Définition 6 (Partition des dimensions) *Pour tout cube défini sur les dimensions D, on considère une partition de D en quatre ensembles notés respectivement :*

- D_t *pour la ou les dimensions temporelles*
- $\mathscr{D}_{\mathscr{A}}$ *pour les dimensions dites d'*analyse
- $\mathscr{D}_{\mathscr{R}}$ *pour les dimensions dites de* référence
- $\mathscr{D}_{\mathscr{F}}$ *pour les dimensions* ignorées.

Il en découle que chaque cellule $cell = \langle (d_1, \cdots, d_n), \mu \rangle$ d'un cube peut être notée $cell = \langle (f, r, a, t), \mu \rangle$ avec f la restriction sur $\mathscr{D}_{\mathscr{F}}$ de cell, r la restriction sur $\mathscr{D}_{\mathscr{R}}$ de cell, a la restriction sur $\mathscr{D}_{\mathscr{A}}$ de cell, t la restriction sur D_t de cell.

Exemple 4 *Dans la suite de cet article, nous considérons la partition suivante des dimensions du cube de données C représenté sur le tableau 2 :*

- $\mathscr{D}_{\mathscr{F}} = \emptyset$
- $\mathscr{D}_{\mathscr{R}} = \{Cust\text{-}Grp,\ City\ \}$
- $\mathscr{D}_{\mathscr{A}} = \{Age,\ Product\}$
- $D_t = \{Date\}$

Dans le cadre de l'extraction des motifs séquentiels multidimensionnels, l'ensemble $\mathscr{D}_{\mathscr{R}}$ permet d'identifier les sous-cubes par rapport auxquels le support sera calculé. Pour cette raison, cet ensemble est nommé *référence*. On note que dans le cadre des motifs séquentiels classiques et des extensions [PHP$^+$01] et [dAFGL04], cet ensemble est réduit à un seul attribut (identifiant du client *cid* du tableau 1 ou identifiant IdG dans [dAFGL04]). Dans nos calculs, le support d'une séquence sera ainsi calculé comme étant la proportion de sous-cubes où cette séquence peut être retrouvée.

L'ensemble $\mathscr{D}_{\mathscr{A}}$ décrit les axes d'*analyse*, c'est-à-dire l'ensemble des dimensions apparaissant explicitement dans les motifs séquentiels multidimensionnels extraits. Dans le cadre des motifs classiques, seule une seule dimension apparaît, correspondant aux produits achetés (ou encore aux pages internet visitées). Dans notre approche, cette dimension est étendue à la prise en compte d'un ensemble de dimensions.

L'ensemble $\mathscr{F}$ décrit les axes *ignorés*, c'est-à-dire ceux qui ne servent ni à définir la date, ni à identifier un sous-cube, ni à définir le motif lui-même.

3.2 Item, itemset et séquence multidimensionnels

Définition 7 (Item multidimensionnel) *Un item multidimensionnel e défini sur les dimensions $\mathscr{D}_{\mathscr{A}} = D_{i_1}, \ldots, D_{i_m}$ et la mesure M est un n-uplet de la forme : $e = (d_{i_1}, \cdots, d_{i_m}, \mu)$ tel que :*

- $\forall k,\ k \in [i_1, \cdots i_m],\ d_{i_k} \in Dom(D_{i_k})$
- $\mu \in dom(M)$

On note $e = \langle a, \mu \rangle$.

$(A, clou, 50)$, $(B, pneu, 2)$, $(B, rustine, 10)$ sont des items multidimensionnels.

Définition 8 (Itemset multidimensionnel) *On appelle itemset multidimensionnel i un ensemble*

non vide d'items multidimensionnels de la forme $i = \{({d_{i1}}^1, \cdots, {d_{im}}^1, \mu^1), \cdots, ({d_{i1}}^p, \cdots, {d_{im}}^p, \mu^p)\}$. *On note* $i = \{e_1, \cdots, e_p\}$.

$\{(A, clou, 50), (B, pneu, 2)\}$ et $\{(B, rustine, 10)\}$ sont des itemsets multidimensionnels.

Remarque 1 *Il est important de remarquer que d'une part tous les items d'un itemset sont définis sur les mêmes dimensions* ($\mathscr{D}_{\mathscr{A}}$), *et que d'autre part les items multidimensionnels d'un même itemset sont deux à deux distincts.*

Définition 9 (Séquence multidimensionnelle) *On appelle séquence multidimensionnelle une liste ordonnée non vide d'itemsets de la forme* $\varsigma = \langle\{({d_{i1}}^1, \cdots, {d_{im}}^1, \mu^1), \cdots, ({d_{i1}}^p, \cdots, {d_{im}}^p, \mu^p)\}, \cdots \{({d_{i1}}^{1'}, \cdots, {d_{im}}^{1'}, \mu^{1'}), \cdots, ({d_{i1}}^{p'}, \cdots, {d_{im}}^{p'}, \mu^{p'})\}\rangle$ *On note* $\varsigma = \langle i_1, \cdots, i_l\rangle$.

$\langle\{(A, clou, 50), (B, pneu, 2)\}\ \{(B, rustine, 10)\}\rangle$ est une séquence.

Définition 10 (Inclusion de séquence) *Une séquence multidimensionnelle* $\varsigma = \langle a_1, \cdots, a_l\rangle$ *est appelée sous-séquence d'une séquence* $\varsigma' = \langle b_1, \cdots, b_{l'}\rangle$ *s'il existe des entiers* $1 \leq j_1 \leq j_2 \leq \cdots \leq j_l \leq l'$ *tels que* $a_1 \subseteq b_{j_1}, a_2 \subseteq b_{j_2}, \cdots, a_l \subseteq b_{j_l}$.
On dit aussi que la séquence ς' *est une sur-séquence de* ς.

Exemple 5 *Soient les séquences multidimensionnelles* $\varsigma = \langle\{(A, clou, 50)\},\ \{(B, rustine, 10)\}\rangle$ *et* $\varsigma' = \langle\{(A, clou, 50), (B, pneu, 2)\}, \{(B, rustine, 10)\}\rangle$. ς *est une sous-séquence de* ς'.

3.3 Support

Calculer le support d'une séquence revient à compter le nombre de sous-cubes de la base qui la supportent, de même qu'il revient à compter le nombre de clients ayant suivi la séquence d'achat dans le contexte du panier de la ménagère. Dans notre approche, un sous-cube *supporte* une séquence s'il est possible de trouver un ensemble de cellules qui la vérifient. Il s'agit, pour chaque itemset de la séquence, de trouver une date de la dimension D_t à laquelle tous les items sont présents. Tous les itemsets doivent être alors trouvés à des dates de D_t respectant l'ordre de la liste des itemsets, d'où la définition ci-dessous.

Définition 11 *Un cube* C *supporte une séquence* $\langle i_1, \cdots, i_l\rangle$ *si*
- $\forall j = 1 \cdots l,\ \exists d_j \in Dom(D_t),\ \forall e = \langle a, \mu_e\rangle \in i_j, \exists cell = \langle(f, a, r, d_j), \mu\rangle \in C$ *et* $\mu_e = \mu$
- $d_1 < d_2 < \cdots < d_l$

Ainsi le sous-cube 1 du tableau 3 supporte la séquence $\langle\{(A, clou, 50), (B, pneu, 2)\}, \{(B, pneu, 3)\}\rangle$ puisque l'on retrouve bien la date 1 à laquelle l'itemset $\{(A, clou, 50), (B, pneu, 2)\}$ est vérifié et la date 2 à laquelle l'itemset $\{(B, pneu)\}$ est vérifié. Le support d'une séquence ς dans un cube C est la proportion de sous-cubes de C qui la supportent.

On note $\mathscr{C}_{C,\mathscr{D}_{\mathscr{R}}}$ l'ensemble des sous-cubes du cube C définis à partir des dimensions $\mathscr{D}_{\mathscr{R}}$ (voir tableau 6). Formellement, on a : $\mathscr{C}_{C,\mathscr{D}_{\mathscr{R}}} = \{r \in \prod_{D_i \in \mathscr{D}_{\mathscr{R}}} Dom(D_i)\ t.q.\ \exists\langle(f, a, r, d_j), \mu\rangle \in C\ et\ \mu \neq NULL\}$. Chaque sous-cube SC de $\mathscr{C}_{C,\mathscr{D}_{\mathscr{R}}}$ est donc défini comme un n-uplet de valeurs des domaines des dimensions de $\mathscr{D}_{\mathscr{R}}$.

Définition 12 (Support d'une séquence) *Soit* $\mathscr{D}_{\mathscr{R}}$ *les dimensions de référence et* C *un cube partitionné en l'ensemble de sous-cubes* $\mathscr{C}_{C,\mathscr{D}_{\mathscr{R}}}$. *Le support d'une séquence* ς *est :* $support(\varsigma) = \frac{|\{SC \in \mathscr{C}_{C,\mathscr{D}_{\mathscr{R}}}\ t.q.\ SC\ supporte\ \varsigma\}|}{|\mathscr{C}_{C,\mathscr{D}_{\mathscr{R}}}|}$

Définition 13 (Séquence fréquente) *Soit* $suppmin \in [0, 1]$, *le support minimal fixé par l'utilisateur. On dit qu'une séquence* ς *est fréquente si* $support(\varsigma) \geq suppmin$.

Exemple 6 *On considère le cube du tableau 2 avec* $\mathscr{D}_{\mathscr{R}} = \{Cust\text{-}Grp,\ City\}$, $\mathscr{D}_{\mathscr{A}} = \{Age, Product\}$, $suppmin = \frac{1}{5}$.

On cherche le support de la séquence $\varsigma = \langle\{\langle(A, clou), 50\rangle, \langle(B, pneu), 2\rangle\}\ \{\langle(B, rustine), 10\rangle\}\rangle$. *Ce calcul nécessite le parcours de tous les sous-cubes définis par les dimensions de référence, c'est-à-dire les trois sous-cubes des tableaux 3, 4 et 5.*

1. *sous-cube défini par* $\langle Educ, Chicago\rangle$ *(cf tableau 3). Dans ce sous-cube, deux dates sont présentes. A la date* 1*, on a bien* $\langle(A, clou), 50\rangle$ *et* $\langle(B, pneu), 2\rangle$. *Puis à la date* 2 *on retrouve bien* $\langle(B, rustine), 10\rangle$. *Donc ce sous-cube supporte la séquence* ς.

2. *sous-cube défini par* $\langle Educ, LosAngeles\rangle$ *(cf tableau 4). Ce sous-cube ne supporte pas l'item* $\langle(B, pneu), 2\rangle$*, et donc pas la séquence* ς.

3. *sous-cube défini par* $\langle Reti., Miami\rangle$ *(cf tableau 5). Ce sous-cube ne contient qu'une seule date, il ne peut donc pas supporter la séquence* ς *qui nécessite des sous-cubes contenant au moins deux dates différentes.*

On a donc : $support(\varsigma) = \frac{1}{3} \geq suppmin$. ς *est une séquence fréquente.*

4 Valeurs *joker*

Nous définissons ici d'une part les raisons pour lesquelles une valeur *joker* peut être intéressante dans le cadre de la recherche de motifs séquentiels multidimensionnels et d'autre part les moyens mis en œuvre pour la définition d'une telle valeur.

4.1 Motifs séquentiels α-étoilés

Dans les définitions précédentes, un item ne peut être trouvé que s'il existe une combinaison de valeurs de domaines de $\mathscr{D}_{\mathscr{A}}$ se retrouvant fréquemment dans les données. Or il peut par exemple arriver que ni $\langle(A, rustine), \mu\rangle$ ni $\langle(B, rustine), \mu\rangle$ ni $\langle(C, rustine), \mu\rangle$ ne soit fréquent alors que la valeur *rustine* est fréquemment associée à la valeur μ. Pour cette raison, 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. Dans le cas précédent, on notera $\langle(*, rustine), \mu\rangle$. Un tel item est dit α-étoilé.

Définition 14 (Item α-étoilé) *Soit* $a_{[d_i/\delta]}$ *la substitution de la valeur* d_i *par* δ *dans* a. *Un item multidimensionnel* α*-étoilé est de la forme* $e = \langle a, \mu\rangle$ *tel que :*

1. $\forall d_i \in a,\ d_i \in Dom(D_i) \cup \{*\}$
2. $\exists d_i \in a$ *t.q.* $d_i \neq *$
3. $\forall d_i = *,\ \nexists \delta \in Dom(D_{i_j})$ *t.q.* $e' = \langle a_{[d_i/\delta]}, \mu\rangle$ *est fréquent*

Ainsi, un item α-étoilé contient au moins une dimension d'analyse spécifiée (non étoilée), et ne contient une étoile que si celle-ci ne peut pas être remplacée par une valeur du domaine tout en restant un item fréquent. On note qu'un item fréquent est une séquence fréquente composée d'un seul itemset lui-même composé d'un seul item.

Une séquence α-étoilée est une séquence dont au moins un item est α-étoilé. Un sous-cube supporte une telle séquence si on peut trouver un ensemble d'ensemble de cellules vérifiant chacun des itemsets dans les contraintes de temps imposées par l'ordre des itemsets.

Définition 15 (Support de séquence α-étoilée) *Un cube de données* C *supporte une séquence* α*-étoilée* $\varsigma = \langle i_1, \cdots, i_l\rangle$ *si :*

- $\forall j = 1 \cdots l,\ \exists d_j \in Dom(D_t),\ \forall e = \langle(a_{i_1}, \cdots, a_{i_m}), \mu_e\rangle \in i_j,\ \exists cell = \langle(f, (x_{i_1}, \cdots, x_{i_m}), r, d_j), \mu\rangle \in C$ *avec* $a_i = x_i$ *ou* $a_i = *$
- $d_1 < d_2 < \cdots < d_l$.

Le support de ς *est la proportion de sous-cubes de* C *supportant* ς *:* $support(\varsigma) = \frac{|\{SC \in \mathscr{C}_{C,\mathscr{D}_{\mathscr{R}}} \text{ t.q. } SC \text{ supporte } \varsigma\}|}{|\mathscr{C}_{C,\mathscr{D}_{\mathscr{R}}}|}$

4.2 Motifs séquentiels μ-étoilés

S'il peut arriver qu'une combinaison de dimensions ne trouve pas d'instanciation fréquente, ce cas de figure est d'autant plus vrai dans le cadre de la mesure. En effet, cette dimension particulière du cube est le plus souvent numérique et prend un nombre de valeurs très important, ce qui rend très

improbable la présence de valeurs fréquentes. Pour cette raison, nous définissons, de manière similaire au caractère joker sur les dimensions d'analyse, un caractère joker sur la valeur de mesure, noté $\circledast$. Un item contenant ce caractère est dit μ-étoilé.

Définition 16 (Item μ-étoilé) *Un item multidimensionnel μ-étoilé est de la forme : $e = \langle a, \mu \rangle$ où $\mu \in Dom(M) \cup \circledast$*

Définition 17 (Support de séquence μ-étoilée) *Un cube de données C supporte une séquence μ-étoilée $\varsigma = \langle i_1, \cdots, i_l \rangle$ si : $\forall j = 1 \cdots l, \quad \exists d_j \in Dom(D_t)$, $\forall e = \langle a, \mu_e \rangle \in i_j$, $\exists cell = \langle (f, a), \mu \rangle \in C$ avec $\mu_e = \circledast$ ou $\mu_e = \mu$.*

Le support de ς est la proportion de sous-cubes de C supportant ς : $support(\varsigma) = \frac{|\{SC \in \mathscr{C}_{C,\mathscr{D}_\mathscr{R}} \text{ t.q. } SC \text{ supporte } \varsigma\}|}{|\mathscr{C}_{C,\mathscr{D}_\mathscr{R}}|}$

Remarque 2 *La présence du symbole $\circledast$ revient en fait à ne considérer que la restriction sur $\mathscr{D}_\mathscr{A}$ sans se soucier de la valeur de μ (seule la présence d'une cellule non nulle suffit).*

Exemple 7 *Sur les trois sous-cubes définis à partir du cube du tableau 2 et de $\mathscr{D}_\mathscr{R}$, seul un vérifie l'item $i_1 = \langle clou, 50 \rangle$, alors que les trois vérifient l'item $i_2 = \langle clou, \circledast \rangle$. On a $support(i_1) = \frac{1}{3}$ et $support(i_2) = 1$.*

5 Algorithmes

Dans cette section, nous définissons tout d'abord les algorithmes permettant d'extraire les items multidimensionnels fréquents puis les algorithmes mis en œuvre pour extraire l'ensemble des séquences fréquentes.

5.1 Génération des 1-fréquents

Les items multidimensionnels fréquents sont la base de l'extraction des motifs séquentiels. Ils représentent les fréquents de taille 1 puisqu'ils correspondent à des séquences composées d'un seul item (contenu dans un seul itemset).

Dans le cas où l'on cherche des items non étoilés, un parcours sur la base permet de recencer les items et leur support (donc les items fréquents), de même que dans le cas classique mono-attribut. Dans le cas où les items peuvent être étoilés, il s'agit de rechercher les items *maximaux* (au sens du nombre de dimensions spécifiées). Afin de limiter le calcul du support aux items dont la probabilité d'être fréquents est non nulle, nous adoptons une méthode de génération par niveau. Cette recherche s'effectue au sein d'un treillis dont la bordure sera retenue pour constituer l'ensemble des items fréquents *maximaux*. Pour ce faire, le premier niveau considère les items multidimensionnels pour lesquels une seule dimension d'analyse est spécifiée, les autres dimensions étant mises à la valeur $*$. Les items multidimensionnels fréquents sont alors *joints* entre eux pour obtenir la liste des items candidats pour lesquels deux dimensions d'analyse spécifiées dont on ne retient que les fréquents. Cette procédure est itérée tant que de nouveaux items fréquents sont trouvés.

L'opération de *jointure* entre deux items fréquents suppose que les items soient compatibles, c'est-à-dire qu'ils partagent un nombre suffisant de valeurs de dimensions d'analyse (voir définition 19).

Exemple 8 *La figure 1 illustre l'approche de spécification par niveau pour les dimensions d'analyse D_1, D_2, D_3 avec :*

- $dom(D_1) = \{a, a'\}$
- $dom(D_2) = \{b, b'\}$
- $dom(D_3) = \{c\}$

Dans une première étape, au niveau 1, les items où une dimension d'analyse est définie sont considérés. Ainsi, il s'agit de tester si les candidats $(a, *, *)$, $(a', *, *)$, ..., $(*, *, c)$ sont fréquents. Imaginons que $(*, b', *)$ ne soit pas fréquent. Les items candidats de niveau 2 sont alors construits en combinant les fréquents de taille 1. Ainsi, $(a, b, *)$, ..., $(*, b, c)$ sont potentiellement fréquents. Il faut donc vérifier leur fréquence. Dans notre exemple, seuls les items non barrés sont trouvés fréquents. On construit alors les candidats de

taille 3. Par exemple, l'item candidat (a, b, c) est construit en combinant $(a, b, *)$ et $(a, *, c)$. Après test des fréquences, on trouve les items maximalement spécifiques (a', b, c), $(a, b, *)$ et $(a, *, c)$. Un élagage est possible avant même la vérification sur la base. Par exemple, si $(*, b, c)$ n'avait pas été fréquent, il aurait été impossible que (a', b, c) le soit, même si $(a', b, *)$ et $(a', *, c)$ l'étaient.

On note que cette exploration est très proche de l'exploration du treillis des cuboïdes dans le cadre des iceberg cubes [BR99, CCL03]. L'amélioration de cette exploration dans le cadre de notre proposition s'effectuera en relation avec ces approches.

La figure 2 met en évidence le fait que ignorer une dimension en la mettant dans $\mathscr{D}_{\mathscr{F}}$ n'est pas équivalent à instancier cette dimension par la valeur joker. En effet, les items multidimensionnels fréquents extraits ne sont pas les même. Par exemple si $D_2 \in \mathscr{D}_{\mathscr{F}}$, nous ne pouvons pas extraire l'item (a', b, c).

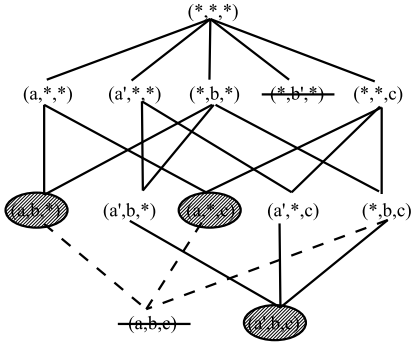

FIG. 1 – Les items multidimensionnels fréquents maximalement spécifiques avec $D_2 \in \mathscr{D}_{\mathscr{A}}$

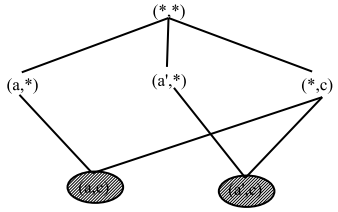

FIG. 2 – Les items multidimensionnels fréquents maximalement spécifiques avec $D_2 \in \mathscr{D}_{\mathscr{F}}$

La mesure M peut être traitée soit en considérant la valeur prise sur cette dimension, soit en ne la prenant pas en compte (valeur = $\circledast$).

Définition 18 ($\bowtie$-compatibilité) *Soient 2 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 $d'_{i_n} = *$*

Pour être $\bowtie$-compatibles, deux items multidimensionnels définis sur n dimensions doivent donc partager $n-2$ valeurs de dimension. Par exemple, $(Chicago, A, *)$ et $(*, A, rustine)$ sont deux items définis sur 3 dimensions d'analyse et partagent $3-2=1$ valeur sur la dimension Age. Ils sont donc $\bowtie$-compatibles. En revanche, les items $(Chicago, B, *)$ et $(NY, A, *)$ ne sont pas $\bowtie$-compatibles.

Définition 19 (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 $d_i = *$*

Soit deux ensembles d'items multidimensionnels de même taille n E et E', on note : $E \bowtie E' = \{e \bowtie e' \text{ t.q. } (e, e') \in E \times E'$ et e et e' sont $\bowtie$-compatibles$\}$

Par exemple, la jointure des deux items $(Chicago, A, *)$ et $(*, A, rustine)$ résultera en $(Chicago, A, rustine)$.

Notation On note F_1^i l'ensemble des 1-fréquents dont i dimensions sont spécifiées (différentes de $*$).

L'ensemble $Cand_1^1$ des items candidats pour lesquels une seule dimension est fixée est calculé grâce à l'algorithme 1.

Données : $C, \mathscr{D}_{\mathscr{A}}, M, \circledast\mu opt$
Résultat : E_1
début
  $Cand_1 \longleftarrow \{\}$;
  si $\circledast\mu opt$ **alors**
    $\mu \longleftarrow \circledast$;
    $D \leftarrow \mathscr{D}_{\mathscr{A}}$;
  sinon
    /*`la mesure est considérée comme une dimension ordinaire` */
    $D \longleftarrow \mathscr{D}_{\mathscr{A}} \cup \{M\}$;
  pour chaque $dim \in D$ **faire**
    pour chaque $v \in dom(dim)$ **faire**
      si $\circledast\mu opt$ **alors**
        $e \longleftarrow \langle (d_1, \cdots, d_{D.length}), \mu \rangle$;
      sinon
        $e \longleftarrow \langle (d_1, \cdots, d_{D.length-1}), d_{D.length} \rangle$;
      pour chaque $i \leq D.length$ **faire**
        si $dim = i$ **alors**
          $d_i = v$;
        sinon
          $d_i = *$;
      $Cand_1 \longleftarrow Cand_1 \cup \{e\}$;
  retourner $Cand_1$
fin

Algorithme 1: génération de $Cand_1^1$

Les fréquents de taille 1 sont alors obtenus en fonction des candidats de taille 1. De manière générale, les candidats de taille i sont obtenus en considérant les fréquents de taille i-1 (*cf.* algorithme 2) dont on extrait les fréquents de taille i :

$F_1^1 = \{f \in Cand_1^1,\ support(f) \geq suppmin\}$
$F_1^i = frequents(F_1^{i-1} \bowtie F_1^{i-1})$

À l'issue de cette génération, il est important de s'interroger sur les items susceptibles d'être examinés. Ainsi, l'utilisateur devra spécifier s'il accepte ou non les items α- et μ-étoilés. Si tel n'est pas le cas, l'utilisateur pourra se trouver plus facilement dans le cas où aucun item multidimensionnel fréquent n'a été identifié, les caractères joker $*$ et $\circledast$ favorisant la présence d'items fréquents.

5.2 Génération de candidats

Une fois les 1-fréquents extraits, 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 trouvés.
Nous nous baserons par exemple sur l'algorithme de [Mas02] en incluant les modifications liées aux jokers sur les valeurs des dimensions (motifs α-étoilés) et sur la valeur de mesure (motifs μ-étoilés).

5.3 Calcul du support des séquences

Nous pouvons maintenant définir un algorithme qui calcule le support d'une séquence ς au sein d'un cube de données C, suivant les dimensions de référence et d'analyse désirés.

Les dimensions de référence permettent d'énumérer tous les sous-cubes de C susceptibles de supporter ς. Cette énumération est indispensable pour calculer le ratio des sous-cubes qui supportent ς, et donc pour définir si la séquence est fréquente ou non.

L'algorithme 3 vérifie pour chaque sous-cube de C si la séquence est supportée ou non. Si la séquence est supportée, alors le support est incrémenté. L'algorithme retourne ensuite le ratio des cubes supportant ς.

Données : $C, \mathscr{D}_{\mathscr{R}}, \mathscr{D}_{\mathscr{A}}, D_t, F_1, *\alpha opt$
Résultat : l'ensemble des items multidimensionnels fréquents
début
  $i \longleftarrow 1$;
  $F_1^m \longleftarrow \emptyset$;
  tant que $F_1^i \neq \emptyset \wedge i \leq m$ **faire**
    $Cand_1^{i+1} \longleftarrow \emptyset$;
    pour chaque *couple* $(e_1, e_2) \in F_1^{i^2}$ *tel que* e_1, e_2 $\bowtie$*-compatibles* **faire**
      $Cand_1^{i+1} \longleftarrow Cand_1^{i+1} \cup \{e_1 \bowtie e_2\}$;
    $F_1^{i+1} \longleftarrow \{f \in Cand_1^{i+1},\ support(f) \geq suppmin\}$;
    $i \longleftarrow i + 1$;
  si $*\alpha opt$ **alors**
    /* `on retourne les items avec le moins d'* possible` */
    retourner ($\{e$ *item* α *étoilé*$\}$);
  sinon
    /*`pas de 1-fréquent avec une *` $\cdots$ `on retourne les 1-frqts dont les` m `dim sont spécifiées` */
    retourner (F_1^m)
fin

Algorithme 2: Génération et extraction des items fréquents

L'algorithme 4 permet de vérifier si le sous-cube SC supporte la séquence ς. Pour cela, cet algorithme cherche à instancier itemset par itemset en conjugant *récursivité* et *ancrage*. L'ancrage correspond à une cellule du cube C d'où il est espéré que la séquence pourra être instanciée. Cette cellule correspond donc à une date à laquelle le premier item du premier itemset de la séquence est trouvé. À partir de cette cellule, seules les cellules pertinentes sont retenues, c'est-à-dire celles qui partagent la même date. On ne retient donc que les cellules partageant la même date. Si le sous-cube 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 seules cellules 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 cube a été parcouru sans succès.

5.4 Complexité

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

- n_C est le nombre de cellules du cube C
- $m = |\mathscr{D}_{\mathscr{A}}|$ est le nombre de dimensions des items multidimensionnels.

supportCube (algorithme 4)

- Le cube C 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 toutes les cellules respectant une certaine condition sur la date.
- Vérifier si une cellule supporte un item est réalisable en $O(m)$. Il suffit de comparer les m dimensions de l'item avec celles de la cellule.
- Dans le pire des cas, la complexité de l'algorithme est de $O(n_C \times m \times \log n_C)$.

supportcount (algorithme 3)

On appelle la fonction précedente pour tous les l sous-cubes C_i de C suivant $\mathscr{D}_{\mathscr{R}}$. Soit $n_{max} = \max n_{C_i}$. La complexité dans le pire des cas est donc $O(l \times n_{max} \times m \times \log n_{max})$.

6 Expérimentations

Nous avons implémenté les algorithmes présents dans ce rapport afin de réaliser un outil d'extraction de motifs séquentiels multidimensionnels étoilés $(*, \circledast)$ ou non. Des expérimentations sont menées sur des bases de données synthétiques représentées sous forme tabulaire (que les cellules non vides présentes). Par défaut le jeu de données utilisé contient 12 000 n-uplets, 15 dimensions d'analyse de cardinalité moyenne 20 et une dimension de mesure dont la cardinalité est 15.

Ces premières expérimentations comparent les résultats obtenus en terme de temps de calcul et de nombre de fréquents trouvés en fonction du nombre de dimensions prises en compte, du seuil de support considéré, de la taille de la base de données. Nous établissons également une comparaison entre notre approche et l'approche de [PHP+01] en fonction du nombre motifs extraits par rapport à la taille de la base ou le nombre de dimensions d'analyse.

Ces expérimentations sont menées sans prise en compte de valeurs jokers (M^2SP), avec prise en compte de valeurs jokers sur les dimensions d'analyse (M^2SP-alpha), avec prise en compte de valeurs jokers sur la mesure (M^2SP-mu) et avec prise en compte de valeurs jokers à la fois sur les dimensions d'analyse et sur la mesure (M^2SP-alpha-mu).

Les figures 3 et 4 le nombre de fréquents extraits ainsi que le temps de calcul en fonction du support considéré. Le fait de ne pas accepter la présence de la valeur joker sur les dimensions d'analyse représente une contrainte importante. En effet, cela est équivalent à rechercher toutes les combinaison de valeurs de domaines de $\mathscr{D}_{\mathscr{A}}$ se retrouvant

Fonction supportcount
Données : $\varsigma, C, \mathscr{D}_{\mathscr{R}}, comptage$
Résultat : le support de la séquence ς
début
  $Entier\ support \longleftarrow 0$;
  $BooleenseqSupportée$;
  $\mathscr{C}_{C,\mathscr{D}_{\mathscr{R}}} \longleftarrow \{sous\ cubes\ de\ C\ identifiés\ sur\ \mathscr{D}_{\mathscr{R}}\}$;
  pour chaque $SC \in \mathscr{C}_{C,\mathscr{D}_{\mathscr{R}}}$ **faire**
    $seqSupportée \longleftarrow supportCube(\varsigma, SC, comptage)$;
    si $seqSupportée$ **alors**
      $support \longleftarrow support + 1$;
  retourner $\left(\frac{support}{|\mathscr{C}_{C,\mathscr{D}_{\mathscr{R}}}|}\right)$
fin

Algorithme 3: Calcul du support d'une séquence (supportcount)

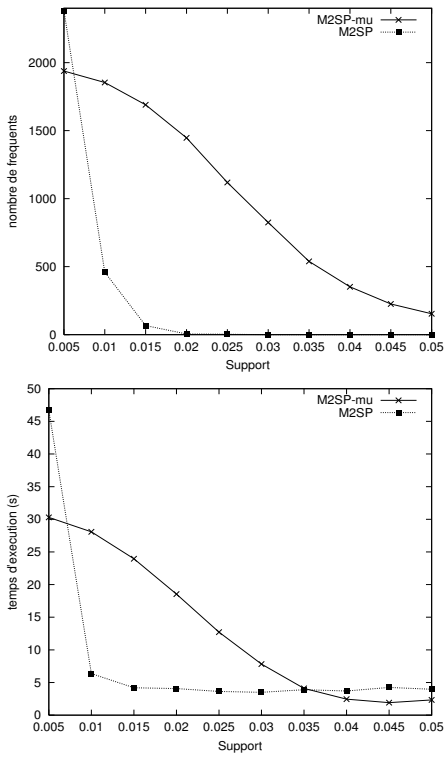

FIG. 3 – Le nombre de motifs extraits et le temps d'exécution en fonction du support sans valeur joker * sur les dimensions d'analyse

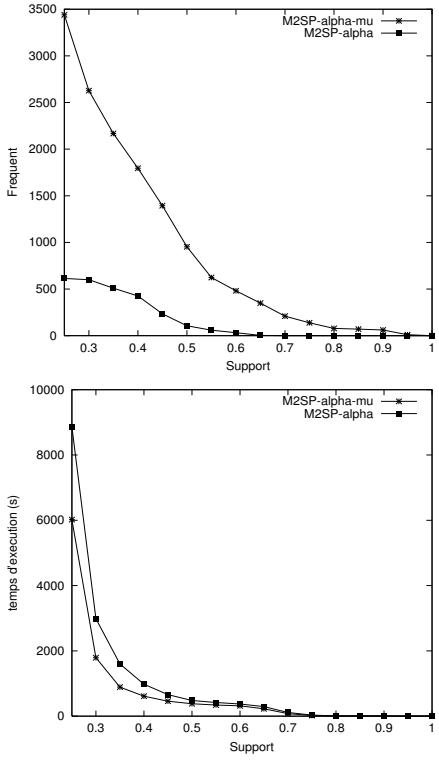

FIG. 4 – Le nombre de motifs extraits et le temps d'exécution en fonction du support avec valeur joker * sur les dimensions d'analyse

```
Fonction supportCube
Données : ς, C, comptage
Résultat : Booleen
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 cube  */
    tant que cell ⟵ C.next ≠ ∅ faire
        si supporte(cell, item, comptage) 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) */
                C' ⟵ σ_{date=cell.date}(C)
                tant que cell' ⟵ C'.next() ≠ ∅ ∧ itemsetTrouvé = faux faire
                    si supporte(cell', itemSuivant, comptage) alors
                        itemSuivant ⟵ itemset.next()
                        si itemSuivant = ∅ alors
                            itemsetTrouvé ⟵ vrai
            si itemsetTrouvé = vrai alors
                /* recherche des autres itemset  */
                retourner (supportCube(sequence.tail(), σ_{date>cell.date}(C), comptage))
            sinon
                itemset ⟵ sequence.first()
                /*on ne veut plus revoir les cellules de C' */
                C ⟵ σ_{date>cell.date}(C)
    /* pas trouvé */
    retourner (faux)
fin
```

Algorithme 4: supportCube *(Vérification si une séquence est supportée par un cube donné)*

fréquemment dans les données. C'est ainsi que nous du utiliser des supports très faibles afin d'obtenir des résultats significatifs pour M^2SP et M^2SP-μ (figure 3). Nous considérerons ainsi par la suite les méthodes M^2SP-alpha et M^2SP-alpha-mu qui autorisent la présence de la valeur joker sur les dimensions d'analyse. On note que la prise en compte de la valeur joker sur la mesure permet d'augmenter le nombre de fréquents trouvés, la contrainte sur la valeur de mesure étant alors relachée.

La figure 5 montre le nombre de motifs séquentiels multidimensionnels extraits ainsi que le temps de calcul en fonction de la taille de la base. Le nombre de fréquents extraits augmente de façon quasi linéaire à partir d'une certaine taille (15 000 n-uplets). Le temps d'éxécution évolue de façon semblable. Ces courbes nous permettent de souligner la robustesse de notre approche ainsi que son caractère « passage à l'échelle ».

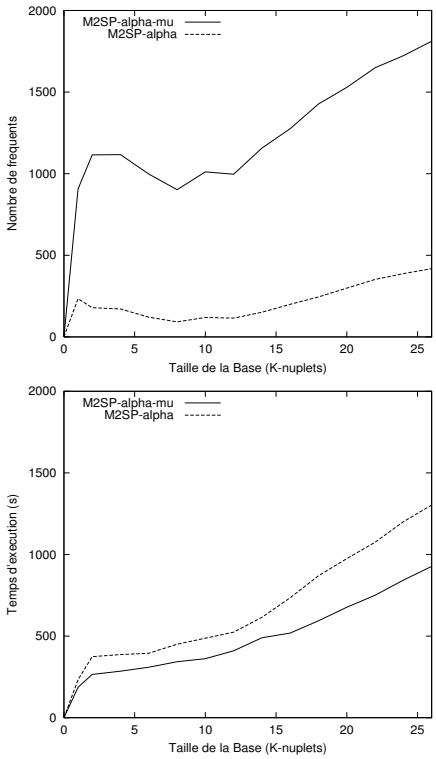

FIG. 5 – Le nombre de motifs extraits et le temps d'éxécution en fonction de la taille de la base

La figure 6 montre le nombre de fréquents extraits et le temps de calcul en fonction du nombre de dimensions d'analyse. On note que lorsque nous avons une seule dimension d'analyse, nous nous situons dans le cas classique de la recherche des motifs séquentiels. En effet, la définition 14 impose qu'au moins une dimension d'analyse soit instanciée par une valeur de son domaine actif. Dans le cas où nous avons plusieurs dimensions d'analyse, le nombre de fréquents extraits croît quand le nombre de dimensions d'analyse augmente. Ceci est dû principalement à l'augmentation des items fréquents. En effet, tout item fréquent sur k dimensions d'analyse permet la construction d'au moins un item fréquent sur $k+1$ dimensions d'analyse (ce qui diffère du cas classique de APriori). Par exemple, si les valeurs NY et $jeune$ apparaissaient ensemble de manière fréquente sur les $k = 2$ dimensions $City$ et Age, celles-ci se retrouveront nécessairement de manière fréquente sur les $k+1 = 3$ dimensions $City$, Age et $Produit$ si l'on considère la valeur joker $*$ sur la dimension $Produit$ et l'item (NY,$jeune$,$*$). Il est même possible que le nombre d'items fréquents augmente si plusieurs valeurs de $Produit$ se retrouvent fréquemment associées à NY et $jeune$ (*e.g.* (NY,$jeune$,$surf$) et (NY,$jeune$,$combi$)). Le nombre d'items fréquents est donc toujours égal (utilisation de la valeur joker) ou supérieur (valeurs spécifiées) quand on ajoute une dimension d'analyse.

Nous avons simulé l'approche de [PHP+01] afin d'établir une comparaison de cette dernière avec notre approche. Une telle comparaison ne peut s'effectuer qu'en autorisant la valeur joker sur les dimensions d'analyse et sur la mesure (M^2Sp-alpha-mu). Les figures figures 7 et 8 relatent ces comparaisons qui sont centrées sur la capacité d'extraction de motifs séquentiels multidimensionnels par rapport à la taille de la base (figure 7) d'une part et le nombre de dimensions d'analyse (figure 8) d'autre part. Notre approche permet l'extraction de plus de motifs séquentiels multidimensionnels. Ceci est principalement dû au fait que notre approche est une généralisation de [PHP+01] qui permet de surcroît l'extraction de motifs séquentiels multidi-

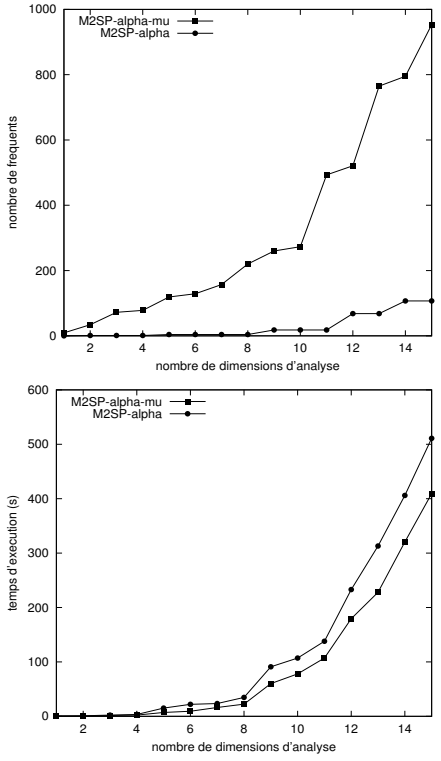

FIG. 6 – Le nombre de motifs extraits et le temps d'exécution en fonction du nombre de dimensions d'analyse

mensionnels *inter pattern*, ce que ne permet pas [PHP+01].

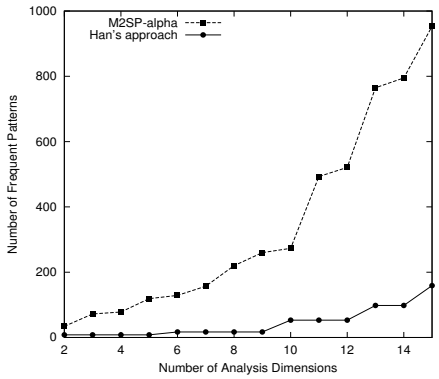

FIG. 7 – Comparaison avec [PHP+01] en fonction du nombre de dimensions d'analyse

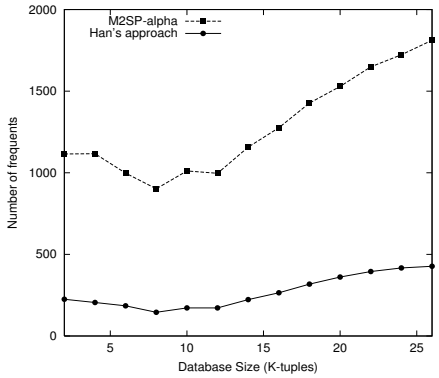

FIG. 8 – Comparaison avec [PHP+01] en fonction de la taille de la base

7 Conclusion

Dans cet article, nous définissons une nouvelle forme de motifs séquentiels, nommés motifs séquentiels multidimensionnels. Contrairement aux propositions présentes dans la littérature [PHP+01], [YC05], [dAFGL04], nous intégrons au sein même de la séquence plusieurs dimensions d'analyse, ce qui permet la construction de motifs de la forme $\langle\{(surf, NY), (housse, NY)\}, \{(combi, LA)\}\rangle$ indiquant que *les personnes ayant acheté leur planche*

notation	sens
C	cube de données
M	Mesure
μ	valeur de mesure
$\mathscr{D}_{\mathscr{R}}$	axes de référence
$\mathscr{D}_{\mathscr{A}}$	axes d'analyse
m	$\lvert\mathscr{D}_{\mathscr{A}}\rvert$
D_t	axe(s) du temps
$\mathscr{F}$	axes ignorés
SC_C	sous-cube de C
$\mathscr{C}_{C,\mathscr{D}_{\mathscr{R}}}$	ensemble des sous-cubes de C définis sur $\mathscr{D}_{\mathscr{R}}$
σ	opérateur de sélection
a	n-uplet défini sur $\mathscr{D}_{\mathscr{A}}$
$e = \langle a, \mu \rangle$	item multidimensionnel
$i = \{e_1, \cdots, e_p\}$	itemset multidimensionnel
$\varsigma = \langle i_1, \cdots, i_l \rangle$	séquence multidimensionnelle
$minsupp$	support minimal

TAB. 6 – Notations utilisées

de surf et la housse à New York ont acheté plus tard leur combinaison à San Francisco.

Nous introduisons également les motifs séquentiels étoilés permettant la prise en compte de valeurs joker $*$ sur les dimensions d'analyse et $\circledast$ sur la mesure.

Nous posons les définitions formelles sous-tendant ces propositions originales. Les algorithmes associés à notre approche sont présentés et validés par des expérimentations effectuées sur des jeux de données synthétiques. Ces expérimentations montrent en particulier l'intérêt de l'introduction des valeurs joker $*$ sur les dimensions d'analyse et $\circledast$ sur la mesure pour traiter les cas où aucun fréquent n'est trouvé.

Ce travail ouvre de nombreuses perspectives, notamment en ce qui concerne la gestion des contraintes de temps pour la définition de motifs multidimensionnels généralisés, ainsi que sur l'intégration d'approximation dans la mesure. Cette dernière approche permettrait en effet de ne pas perdre totalement la connaissance de la valeur de mesure (ce qui est le cas avec $\circledast$) tout en conservant une chance de trouver des motifs fréquents face au grand nombre de valeurs de mesure possibles dans les bases de données issues du monde réél. Ainsi, nous pourrons construire des règles de la forme *Les personnes ayant acheté un lecteur DVD à la FNAC achètent par la suite* environ 3 *DVDs dans un supermarché*. De plus, outre ses applications immédiates au contexte du panier de la ménagère, cette proposition sera utilisée au sein de bases de données multidimensionnelles MOLAP afin de rechercher des enchaînements fréquents de blocs de cellules au sein d'une représentation cubique des données. Des recherches similaires ont été menées dans le cadre de règles d'association [CMLL04], il s'agit de les étendre à la prise en compte de motifs séquentiels. Enfin, les hiérarchies, intérêt majeur du modèle multidimensionnel, seront traitées afin de rechercher des motifs séquentiels multidimensionnels à différents niveaux de granularité.

Références

[BR99] K. Beyer and R. Ramakrishnan. Bottom-up computation of sparse and iceberg cube. In *Proceedings of the 1999 ACM SIGMOD international conference on Management of data*, pages 359–370, 1999.

[CCL03] A. Casali, R. Cicchetti, and L. Lakhal. Cube lattices : A framework for multidimensional data mining. In *Proceedings of the Third SIAM International Conference on Data Mining*, 2003.

[CD97] S. Chaudhuri and U. Dayal. An overview of data warehousing and olap technology. *ACM SIGMOD Record*, 26(1) :65–74, 1997.

[CMLL04] Y.W. Choong, P. Maussion, A. Laurent, and D. Laurent. Summarizing multidimensional databases using fuzzy rules. In *Proc. of the 10th Int. Conf. on Information Processing*

and Management of Uncertainty in Knowledge-Based Systems (IPMU'04), pages 99–106, 2004.

[dAFGL04] S. de Amo, D. A. Furtado, A. Giacometti, and 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.

[Mar98] P. Marcel. *Manipulations de Données Multidimensionnelle et Langages de Règles*. PhD thesis, I.N.S.A. Lyon, 1998.

[Mas02] 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*. PhD thesis, Université de Versailles, 2002.

[MTP04] F. Masseglia, M. Teisseire, and P. Poncelet. Recherche des motifs séquentiels. *Revue Ingénierie des Systèmes d'Information (ISI), numéro spécial "Extraction de motifs dans les bases de données"*, 2004.

[PHMA+04] J. Pei, J. Han, B. Mortazavi-Asl, J. Wang, H. Pinto, Q. Chen, U. Dayal, and M.-C. Hsu. Mining sequential patterns by pattern-growth : The prefixspan approach. *IEEE Transactions on Knowledge and Data Engineering*, 16(10), 2004.

[PHP+01] H. Pinto, J. Han, J. Pei, K. Wang, Q. Chen, and U. Dayal. Multi-dimensional sequential pattern mining. In *CIKM*, pages 81–88. ACM, 2001.

[SA96] R. Srikant and R. Agrawal. Mining sequential patterns : Generalizations and performance improvements. In *EDBT*, pages 3–17, 1996.

[TTM04] D. Tanasa, B. Trousse, and Florent Masseglia. *Mesures de l'internet*, chapter 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.

[YC05] C.-C. Yu and Y.-L. Chen. Mining sequential patterns from multidimensional sequence data. *IEEE Transactions on Knowledge and Data Engineering*, 17(1) :136–140, 2005.