Extraction de Séquences Multidimensionnelles Convergentes et Divergentes

Marc Plantevit, Anne Laurent, Maguelonne Teisseire

Extraction de Séquences Multidimensionnelles Convergentes et Divergentes

Marc Plantevit, Anne Laurent, Maguelonne Teisseire

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

Résumé. Les motifs séquentiels sont un domaine de la fouille de données très étudié depuis leur introduction par Agrawal et Srikant. Même s'il existe de nombreux travaux (algorithmes, domaines d'application), peu d'entre eux se situent dans un contexte multidimensionnel avec la prise en compte de ses spécificités : plusieurs dimensions, relations hiérarchiques entre les éléments de chaque dimension, etc. Dans cet article, nous proposons une méthode originale pour extraire des connaissances multidimensionnelles définies sur plusieurs niveaux de hiérarchies mais selon un certain point de vue : du général au particulier ou vice et versa. Nous définissons ainsi le concept de séquences multidimensionnelles convergentes ou divergentes ainsi que l'algorithme associé, *M2S_CD*, basé sur le paradigme "pattern growth". Des expérimentations, sur des jeux de données synthétiques et réelles, montrent l'intérêt de notre approche aussi bien en terme de robustesse des algorithmes que de pertinence des motifs extraits.

1 Introduction

Les motifs séquentiels sont étudiés depuis plus de dix ans (Agrawal et Srikant (1995)), ils permettent de mettre en exergue des corrélations entre événements suivant leur chronologie d'apparition. Les motifs séquentiels ont été récemment étendus dans un contexte multidimensionnel par Pinto et al. (2001), Plantevit et al. (2005) et Yu et Chen (2005). Ils permettent ainsi de découvrir des motifs définis sur plusieurs dimensions et ordonnés par une relation d'ordre (*e.g.* temporelle). Par exemple, dans Plantevit et al. (2005), des motifs de la forme *"La plupart des consommateurs achètent une planche de surf et un sac à N.Y., puis ensuite une combinaison à SF"* sont découverts. Les motifs séquentiels multidimensionnels sont bien adaptés aux contextes de stockage et de gestion des données actuels (entrepôts de données). En effet, les motifs ou règles obtenus permettent une autre appréhension des données sources. Cependant leur découverte nécessite certains paramètres dont en particulier le support minimal. Celui-ci correspond à la fréquence minimale d'apparition des motifs au sein de la base considérée. Si le support minimal choisi est trop élevé, le nombre de règles découvertes est faible mais si le support est trop bas, le nombre de règles obtenues est très important et rend difficile l'analyse de celles-ci. Un autre problème est la longueur des motifs extraits. Comment ajuster au mieux le support afin d'obtenir des séquences suffisamment longues pour être réellement utilisables ? 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 *"subsumées"* par des connaissances plus générales qui, elles, seront comptabilisées comme fréquentes.

La prise en compte des hiérarchies dans l'extraction de motifs séquentiels multidimensionnels a été proposée par Plantevit et al. (2006b) via l'algorithme HYPE. Néanmoins cette approche ne permet pas d'extraire des motifs de la forme : *"Quand les ventes de Perrier augmentent en France, les ventes de boissons non alcoolisées augmentent en Europe le mois suivant"* où les deux items multidimensionnels présents dans la séquence $(Perrier, France)$ et $(Boisson\ non\ Alcoolisée, Europe)$ sont comparables.

C'est pourquoi nous proposons la possibilité d'extraire de tels motifs en prenant en compte l'une des principales singularités du contexte multidimensionnel : les hiérarchies. Nous introduisons les concepts de séquences multidimensionnelles convergentes et divergentes. Ces concepts permettent d'extraire de longues séquences en modulant le degré de précision/généralisation le long de celles-ci. Une séquence convergente, du général au particulier, est par exemple, *"Quand les ventes de sodas augmentent aux USA, les ventes de coca augmentent sur la côte ouest ainsi que les ventes de pepsi sur la côte est"* alors qu'une séquence divergente, du particulier au général, est *"Quand les ventes de Perrier augmentent à Nice et que les ventes de coca augmentent à Munich, les ventes de boissons non-alcoolisées augmentent dans l'UE"*.

Dans la suite de cet article, nous décrivons les différentes propositions prenant en compte les hiérarchies dans un contexte multidimensionnel. Après avoir rappelé les concepts associés aux motifs séquentiels multidimensionnels, notre contribution est détaillée en définissant les concepts de séquences convergentes et divergentes. Nous décrivons ensuite les algorithmes et les fonctions permettant l'extraction de telles séquences. Des expérimentations, sur des jeux de données synthétiques et réelles, montrent l'intérêt de notre approche $M2S\_CD$ aussi bien en terme de robustesse des algorithmes que de pertinence des motifs extraits.

2 Travaux antérieurs : motifs multidimensionnels et hiérarchies

Combiner plusieurs dimensions d'analyse permet d'extraire des connaissances qui décrivent mieux les données. Dans Pinto et al. (2001), les auteurs sont les premiers à rechercher des motifs séquentiels multidimensionnels. Ainsi les achats ne sont plus simplement décrits en fonction des seuls *date* et *identifiant du client* comme dans contexte classique, mais en fonction d'un ensemble de dimensions telles que *Type de consommateur, Ville, Age.* Cette approche ne permet que l'extraction de séquences définies sur une seule dimension (*e.g. produit*) caractérisées par une motif multidimensionnel. Ainsi, il est impossible d'extraire des combinaisons de motifs multidimensionnels suivant le temps.

L'approche proposée par Yu et Chen (2005) est très singulière puisqu'il existe un très fort lien hiérarchique entre les dimensions d'analyse. Les *pages web* sont visitées durant une *ses-*

sion au cours d'une *journée*. Cette approche multidimensionnelle permet donc une gestion plus fine du temps mais ne permet pas de se situer réellement dans un contexte multidimensionnel.

Dans Plantevit et al. (2005), les règles extraites ne combinent pas seulement plusieurs dimensions d'analyse. Ces dimensions sont également combinées au cours du temps. Par exemple, dans la règle *"Les ventes de pepsi augmentent à NY puis les ventes de coca augmentent à LA"*, NY apparaît avant LA et $pepsi$ avant $coca$.

Il existe très peu de travaux conciliant hiérarchies et multidimensionnalité lors de l'extraction de motifs séquentiels. Les travaux de Yu et Chen (2005) permettent une représentation plus fine du temps, mais ne répondent pas à notre problématique générale d'un nombre quelconque de dimensions. Seule l'approche HYPE, (Plantevit et al. (2006a),Plantevit et al. (2006b)), permet l'extraction de séquences organisées sur différents niveaux de hiérarchies. HYPE permet d'extraire des motifs de la forme : *Quand les ventes de boissons gazeuses augmentent en Europe, les exportations de Perrier augmentent en France et les exportations de soda augmentent aux USA* où différents niveaux de hiérarchies sont présents dans la séquence multidimensionnelle.

Mais cette proposition ne permet pas d'extraire des séquences où des items de même dimension mais de granularité différente cohabitent tels que $(Nice, Coca)$ et $(France, Soda)$. En effet, pour assurer un passage à l'échelle dans un contexte d'explosion du nombre de motifs possibles, le choix de ne conserver que les items maximalement spécifiques a été fait.

A notre connaissance, il n'existe donc aucune approche proposant de prendre en compte les hiérarchies dans un contexte multidimensionnel tel qu'il existe des items comparables dans les séquences extraites. Nous proposons donc les nouveaux concepts de séquences multidimensionnelles *convergentes* et *divergentes* afin de permettre une extraction de connaissances plus complète et adaptée aux spécificités des contextes multidimensionnels. Nous dirigeons ainsi la génération des motifs soit du général au particulier soit du particulier au général afin de limiter le nombre de motifs candidats mais nous ouvrons ainsi la voie à des motifs composés de séquences plus longues.

3 M2S_CD : motifs séquentiels multidimensionnels convergents ou divergents

Dans cette section, nous introduisons un concept original. En effet, l'esprit humain raisonne souvent de deux façons différentes et symétriques. La réflexion s'exécute de l'exemple vers la théorie ou de la théorie vers l'exemple. Nous essayons donc de reproduire ce type de raisonnement dans les connaissances que nous souhaitons extraire. Nous introduisons donc le concept de *séquence multidimensionnelle convergente ou divergente*. Nous présentons les différentes définitions préliminaires associées aux motifs séquentiels multidimensionnels avec prise en compte des hiérarchies pour ensuite détailler les concepts de motifs convergents et divergents ainsi que les algorithmes associés.

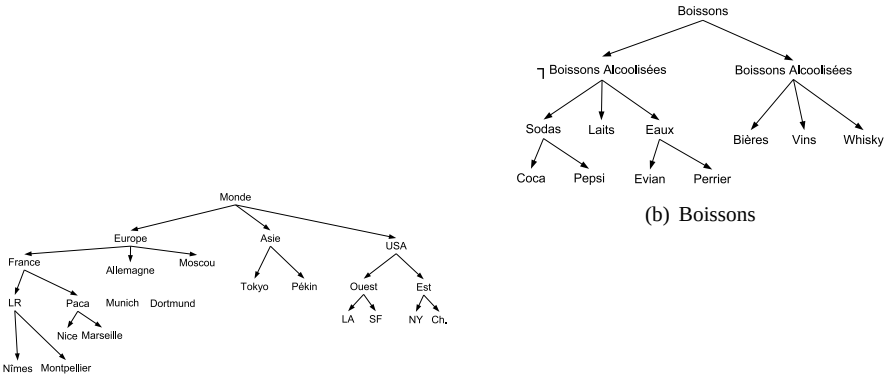
3.1 Base Exemple

Pour illustrer les différents concepts et définitions, nous proposons la base exemple du tableau Tab. 1 qui décrit les ventes réalisées dans différentes villes du monde par différentes

Date	**1**	**2**	**3**	**4**
Enseigne 1	$(Mntp, Pepsi)$	$(Mntp, Pepsi), (Nice, Coca)$	$(Mars., Coca), (Mun., Pepsi)$	$(Moscou, Pepsi)$
Date	**5**	**6**		
Enseigne 1	$(NY, Evian), (Pek., Coca)$	$(Ch., Whisky)$		
Date	**1**	**2**	**3**	**4**
Enseigne 2	$(Mntp, Pepsi)$	$(N\hat{i}m., Pepsi), (Mars., Coca)$	$(Dort., Coca), (N\hat{i}m., Pepsi)$	$(Mun., Coca)$
Date	**5**	**6**		
Enseigne 2	$(Mntp, Evian), (LA, Coca)$	$(Tok., Whisky)$		
Date	**1**	**2**	**3**	**4**
Enseigne 3	$(NY, Bie.), (Ch, Whisky)$	$(LA, Bie.)$	$(SF, Bie.)$	$(Pek., Bie.), (Mntp, Vin)$

TAB. 1 – *Base de données exemple DB*

chaînes de magasins. Deux dimensions sont pourvues de hiérarchies, les villes et les boissons indiquées Fig. 1.



(a) Villes

(b) Boissons

FIG. 1 – *Hiérarchies sur les dimensions d'analyse*

3.2 Définitions préliminaires

Soit une base de données DB où les données sont définies suivant n dimensions, nous considérons une tri-partition de l'ensemble des dimensions :

- L'ensemble des dimensions sur lesquelles sont extraites les règles (*dimensions d'analyse*) noté D_A.
- L'ensemble des dimensions qui permettent de définir si une séquence est fréquente (*dimensions de référence*) noté D_R.
- L'ensemble des dimensions permettant d'introduire une relation d'ordre entre les événements (*e.g.* temps) noté D_T.

La base de données peut se partitionner en *blocs* définis par leur valeur sur les dimensions de référence. Un *item multidimensionnel* e est un m-uplet défini sur l'ensemble des m dimensions d'analyse D_A. Plus précisément $e = (d_1, d_2, \ldots, d_m)$ où $d_i \in Dom(D_i) \cup \{*\}, \forall D_i \in D_A$ et où le méta symbole * joue le rôle de valeur joker. Par exemple, $(Vin, Nice)$ est un item multidimensionnel défini sur les dimensions d'analyse $Boissons$ et $Villes$. Un *itemset multi-*

dimensionnel $i = \{e_1, \ldots, e_k\}$ est un ensemble non vide d'items multidimensionnels. Ainsi, $\{(Vin, Nice), (*, Montpellier)\}$ est un itemset multidimensionnel. Une *séquence multidimensionnelle* $\varsigma = \langle i_1, \ldots, i_l \rangle$ est une liste non-vide d'itemsets multidimensionnels. Par exemple, $\varsigma = \langle \{(Vin, Nice), (Coca, Nice)\}\{(Pepsi, NY)\}\rangle$ est une séquence multidimensionnelle.

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

Ainsi, la séquence $\langle \{(Vin, Nice)\}\{(Perrier, Nice)\}\rangle$ est une sous-séquence de la séquence $\langle \{(Vin, *), (*, Moscou)\}\{(*, Nice)(Perrier, \hat{Nimes})\}\rangle$.

Nous considérons que chaque bloc défini sur D_R contient une séquence de données multidimensionnelles qui est identifiée par ce bloc. Un bloc *supporte* une séquence ς si ς est une sous-séquence de la séquence de données identifiée par ce bloc. Le *support* d'une séquence multidimensionnelle correspond donc au nombre de blocs définis sur D_R qui contiennent cette séquence.

Dans le contexte dans lequel nous nous situons, nous considérons qu'il existe des relations hiérarchiques sur chaque dimension d'analyse matérialisées sous la forme d'*arbres*. Une hiérarchie est donc représentée par un arbre orienté dans lequel les arcs sont de type *is-a*. La relation de *généralisation/spécialisation* s'effectue ainsi de la racine vers les feuilles. Chaque dimension d'analyse possède donc une hiérarchie qui permet de représenter les relations entre les éléments de son domaine. Soit $T_{D_A} = \{T_1, \ldots, T_m\}$ l'ensemble des hiérarchies associées aux dimensions d'analyse où : *(i)* T_i est la hiérarchie représentant les relations entre les éléments de la dimension d'analyse D_i, *(ii)* T_i est un arbre orienté et *(iii)* $\forall$ nœud $n_i \in T_i,\ label(n_i) \in Dom(D_i)$. On note $\hat{x}$ un ancêtre de x dans la hiérarchie et $\check{x}$ un de ses descendants. Par exemple, $boisson = \widehat{soda}$ signifie que $boisson$ est un ancêtre de $soda$ dans la relation *Généralisation/Spécialisation*. Plus précisément, $boisson$ est une instance plus générale que $soda$. Seuls les éléments qui sont feuilles dans l'arbre adéquat sont présents dans la base de données.

Plantevit et al. (2006b) proposent une définition des concepts d'item, itemset et séquence multidimensionnels h-généralisés. Ainsi un *item multidimensionnel h-généralisé* $e = (d_1, \ldots, d_m)$ est un m-uplet défini sur les dimensions d'analyse D_A tel que $d_i \in \{label(T_i)\}$ (d_i existe dans la hiérarchie adéquate). Contrairement aux données de DB, un item multidimensionnel h-généralisé peut être défini avec n'importe quelle valeur d_i dont le nœud associé dans l'arbre hiérarchique n'est pas nécessairement une feuille.

Puisque les items multidimensionnels h-généralisés peuvent être définis sur différents niveaux de hiérarchies, il est nécessaire de définir une relation hiérarchique entre ces items.

Définition 2 (Inclusion hiérarchique) *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* $e' \not>_h e$*)*

L'item $(Coca, France)$ est plus spécifique que l'item $(Soda, UE)$. Les items $(Vin, Nice)$ et $(Pepsi, NY)$ sont incomparables.

3.3 Séquence multidimensionnelle convergente et divergente

Définition 3 (Séquence divergente) *Une séquence* $\varsigma = \langle e_{11}, \ldots, e_{ij}, \ldots, e_{nk} \rangle$ *est divergente si* $\forall e_{ij} \nexists e_{i'j'}, i' < i \; tq \; e_{ij} <_H e''_{ij}$.

En d'autres mots, pour tout item de la séquence, il n'existe pas un item plus général déjà présent à une date antérieure. La séquence $\langle \{(Coca, Montpellier)\}, \{(Coca, LR)(Pepsi, PACA)\}, \{(Soda, France), (Soda, Allemagne)\}\rangle$ est une séquence divergente.

Définition 4 (Séquence convergente) *Une séquence* $\varsigma = \langle e_{11}, \ldots, e_{ij}, \ldots, e_{nk} \rangle$ *est convergente si* $\forall e_{ij} \nexists e_{i'j'}, i' < i \; tq \; e_{ij} >_H e_{i'j'}$.

En d'autres mots, pour chaque item de la séquence, il n'existe pas d'item plus spécifique déjà présent dans la séquence à une date antérieure. La séquence $\langle \{(Boisson, Eurasie)\}, \{(Soda, Europe)(B.A, Asie)\}, \{(Coca, France)(Pepsi, Russie)\}\rangle$ est une séquence convergente.

3.4 Mise en œuvre

3.4.1 Ordre dans les séquences

Ordonner les séquences est une étape fondamentale afin d'améliorer l'implémentation et éviter les cas déjà examinés. Les méthodes existantes, basées sur les différentes philosophies (*pattern growth* (Pei et al. (2004)), *générer/élaguer*(Agrawal et Srikant (1995); Masseglia et al. (1998); Zaki (2001); Ayres et al. (2002))), ne sont pas directement applicables dans un contexte multidimensionnel. En effet, les items ***h-généralisés*** ne sont pas explicités dans la base de données. De tels items sont extraits par inférence puisqu'ils ne sont pas directement associés à un n-uplet dans la base de données.

$\langle \{(Coca, Munich)(Coca, Nice)\}\rangle$
$\langle \{(Coca, Nice)(Pepsi, Munich)\}\rangle$

TAB. 2 – *Contre-exemple*

Le tableau Tab. 2 montre un exemple de séquences de données qui ne peut pas être traité avec les approches existantes dans un contexte classique puisque les items h-généralisés ne sont pas "***explicitement***" présents dans la base. En effet, aucun ordre lexical total, prenant en compte les items h-généralisés, ne peut être directement utilisé. Ainsi, ces méthodes ne peuvent pas extraire la séquence $\langle \{(Coca, Nice)(Soda, Munich)\}\rangle$ (où $Soda$ est un ancêtre de $coca$ et $pepsi$). En effet, les méthodes basées sur le paradigme *pattern growth* trouvent l'item $(Coca, Nice)$ avec un support de 2. Ensuite, elles construisent la base projetée préfixée par la séquence $\langle \{(Coca, Nice)\}\rangle$. Cette base projetée contient les séquences $\langle \{\}\rangle$ et $\langle \{(Pepsi, Munich)\}\rangle$. L'item h-généralisé $(Soda, Munich)$ n'apparaît pas dans cette base projetée alors qu'il est fréquent dans la base initiale. Dans les approches de type *générer-élaguer*, le problème est similaire. Par exemple, dans Masseglia et al. (1998), la projection de la base de données dans l'arbre préfixé des séquences candidates est biaisée.

Il est impossible d'étendre l'ensemble de la base avec tous les item h-généralisés possibles avant le processus d'extraction. Par exemple, considérons une base de données contenant m

dimensions d'analyse et n_i items (feuilles) dans un itemset i, la profondeur moyenne des hiérarchies est d. La transformation d'un itemset va produire $d^m \times n_i$ items au lieu des n_i initiaux, multipliant donc la taille de la base initiale par d^m.

Il est donc nécessaire de prendre en compte les items h-généralisés durant le processus d'extraction et non après un pré-traitement. Nous allons donc introduire un ordre lexical et matérialiser localement les items h-généralisés.

3.4.2 Définitions

Il est primordial de disposer d'un ordre lexicographique lors de l'extraction de motifs fréquents puisque c'est la clef de la non-duplication des items durant le processus.

On dit qu'un itemset est *étendu* s'il est égal à sa fermeture transitive par rapport à la relation de spécialisation ($<_h$). La notion d'itemset étendu permet de prendre en compte tous les items h-généralisés qui peuvent être inférés à partir d'une séquence de données. Afin d'optimiser le traitement des données, nous introduisons un ordre *lexicographico-spécifique* (lgs), qui est un ordre alpha-numérique selon le degré de précision d'un item. Ainsi, les items les plus spécifiques sont prioritaires. Nous devons définir une fonction LGS-Closure qui transforme un itemset (transaction) en un itemset étendu contenant tous les items h-généralisés.

Définition 5 (Fonction LGS-Closure) *La fonction LGS-Closure est une application d'un itemset i vers la fermeture de i avec l'ordre LGS ($<_{lgs}$).*

Par exemple, $LGS\text{-}Closure(\{(USA, B.A)\}) = \{(USA, B.A), (USA, Boisson), (Monde, \neg B.A)\}$. L'item le plus général $(Monde, Boisson)$ n'est pas retourné. En effet, cet item n'est pas nécessaire puisqu'il est une tautologie.

L'extraction des items fréquents peut donc être effectuée sur chaque itemset étendu. Dans les approches *pattern growth*, les séquences sont extraites en ajoutant un item fréquent à une séquence fréquente de manière gloutonne. Il est nécessaire de définir un moyen efficace d'étendre les séquences à partir du dernier itemset de la séquence. Dans ce but, nous définissons une restriction de la fonction LGS-Closure de la façon suivante :

Définition 6 (Fonction LGS-Closure$_X$) *La fonction $LGS\text{-}Closure_X(i)$ est une application d'un itemset i vers la fermeture de i filtrée par rapport à l'itemset $X = \{x_1 <_{lgs} \ldots <_{lgs} x_{k'}\}$ tel que $LGS\text{-}Closure_X(i) = (LGS\text{-}Closure(i) \backslash LGS\text{-}Closure(X)) \cup LGS\text{-}Closure(X \backslash i)$ où $\forall c_j \in LGS\text{-}Closure_X(i), c_j >_{lgs} x_{k'}$.*

Par exemple, $LGS\text{-}Closure_{\{(USA,B.A)\}}(\{(USA, B.A), (USA, \neg B.A)\}) = \{(USA, Boisson),$
$(Monde, B.A), (Monde, \neg B.A)\}$.

3.4.3 Algorithmes

Les séquences divergentes sont extraites en utilisant l'algorithme $M2S\_CD$ (Algorithme 1) suivant une exploration gloutonne en profondeur (paradigme *pattern growth*). Au lieu de parcourir l'intégralité de la base de données, niveau par niveau, comme le font les méthodes de type *générer-élaguer*, la base de données est projetée en fonction de la séquence actuellement explorée. La projection est différente de celle proposée par Pei et al. (2004). En effet, comme

nous devons gérer les items h-généralisés, la projection doit prendre en compte la transaction (itemset) où l'item a été trouvé, et pas seulement l'item lui même comme dans Pei et al. (2004). Pour prendre en compte cette transaction, nous utilisons la fonction LGS-Closure en filtrant les items déjà trouvés.

L'utilisation des bases projetées permet d'éviter des passes inutiles sur des données déjà parcourues. En effet, considérons une séquence fréquente α et la séquence actuellement explorée β tel que $\beta \subseteq \alpha$ or $\alpha \subseteq \beta$, si ces deux séquences partagent la même base projetée alors il est inutile de continuer l'exploration de la séquence β. Nous avons seulement besoin de copier le sous-arbre (déjà extrait) de la séquence α à la séquence β.

L'algorithme 3 permet l'extraction des items localement fréquents sur la base projetée. Il est basé sur la fonction LGS-Closure. La base projetée est parcourue une seule fois pour extraire tous les items fréquents. Deux types d'items peuvent être extraits :

1. Les items qui ne peuvent pas être inclus dans le dernier itemset de la séquence courante ς. Ces items sont donc inclus dans un nouvel itemset de ς. Pour extraire ces items et prendre en compte les items h-généralisés, nous devons étendre les transactions de la base projetée (pas à pas) avec la fonction LGS-CLosure.
2. Les items qui peuvent être inclus dans le dernier itemset de la séquence courante ς. Dans ce cas, nous utilisons la fonction LGS-Closure$_X$ où X représente le dernier itemset de ς.

Algorithme 1: $M2S\_CD$

Data : Base de données DB, support minimum $minsup$
Result : Ensemble des séquences divergentes L
begin
  /*– Initialisation –*/
  $Set\ L \leftarrow \{\}$;
  $Sequence\ \alpha \leftarrow \langle\rangle$;
  /*–Extraction des séquences en profondeur–*/
  $SequenceGrowing(\alpha, DB, L, minsup)$;
  return L;
end

Ces différents algorithmes permettent l'extraction de séquences divergentes. Pour extraire des séquences convergentes, il est nécessaire d'utiliser les mêmes algorithmes mais sur une base de données inversée. En effet, il suffit d'inverser la relation d'ordre (commencer par la fin) au sein de la séquence de données pour permettre un résultat du général au particulier.

4 Expérimentations

Dans cette section, nous reportons les expérimentations effectuées sur des jeux de données synthétiques et réels.

Algorithme 2: SequenceGrowing : Algorithme d'extraction

Data : Séquence α,base projetée $DB|\alpha$, ensemble des fréquents L , support minimum $minsup$

Result : L'ensemble des séquences divergentes fréquentes et préfixées par α

begin

$insérer(\alpha, L)$;

/*– Vérifier si la séquence a déjà été parcourue–*/

if $\exists\beta \mid (\alpha \subseteq \beta \text{ or } \beta \subseteq \alpha) \ \wedge$ α *et* β *partagent la même base projetée* **then**

Copie des descendants de β *dans* α;

return

$Set\ F_l \leftarrow getFrequentItems(DB|\alpha, minsup)$;

foreach $\alpha' \leftarrow \alpha.b$ **do**

Build $DB|\alpha'$;

$SequenceGrowing(\alpha', DB|\alpha', L, minsup)$

end

Algorithme 3: getFrequentItems : extraction des items fréquents

Data : Base projetée $DB|\alpha$, $minsup$

Result : Ensemble F_l des items fréquents dans $DB|\alpha$ et maximalement spécifique

begin

/*–Pour chaque séquence de données S_i de $DB|\alpha$ nous avons : $S_i = LGS\text{-}Closure_{lastItemset(\alpha)}(same).otherTrans$ –*/

/*–Nous devons examiner toutes les séquences de données de $DB|_\alpha$–*/

foreach $S_i \in DB|\alpha$ **do**

foreach *item* $\_e$ *in* $same$ **do**

Gestion de $\_e$;

foreach *itemset* is *in* $other$ **do**

/*–Recherche des items qui peuvent être insérés dans un nouvel itemset de α–*/

SearchOtherTransFrequentItem e in $LGS\text{-}Closure(is)$;

/*–Recherche des items qui peuvent être insérés dans le dernier itemset de α–*/

if is *supports* $lastItemset(\alpha)$ **then**

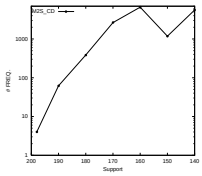
SearchSameTransFrequentItem $\_e$ in $LGS\text{-}Closure_{lastItemset(\alpha)}(is)$;

return $(F_l = \{e | support(e) \geq suppmin \wedge e\ est\ maximalement\ spécifique\})$

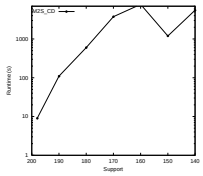
end

4.1 Données synthétiques

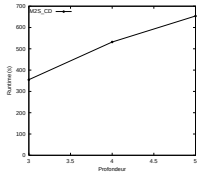
Les expérimentations ont été effectuées sur une base de données synthétiques composée de $10,000$ n-uplets définies sur 5 dimensions d'analyse. Des hiérarchies sont définies sur les dimensions d'analyse. Les expérimentations reportent le nombre de fréquents obtenus et le temps d'exécution en fonction du support, du nombre de dimensions d'analyse, des spécificités des hiérarchies (degré et profondeur).



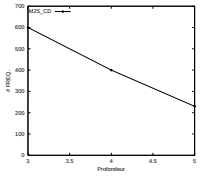
(a) Temps d'exécution en fonction du support (Profondeur =3, Degré =3, $|D_A| = 2$)



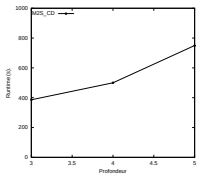
(b) #séquences extraites en fonction du support (Profondeur =3, Degré =3, $|D_A| = 2$)



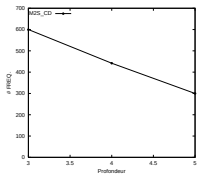
(c) Temps d'exécution en fonction de la profondeur des hiérarchies (Degré =3, $|D_A| = 2$)



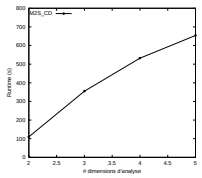
(d) #séquences extraites en fonction de la profondeur des hiérarchies (Degré =3, $|D_A| = 2$)



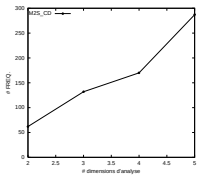
(e) Temps d'exécution en fonction du degré des hiérarchies (Profondeur =4, $|D_A| = 2$)



(f) #séquences extraites en fonction du degré des hiérarchies (Profondeur =4, $|D_A| = 2$)



(g) Temps d'exécution en fonction du nombre de dimensions d'analyse (Profondeur =3, Degré =3)



(h) #séquences extraites en fonction du nombre de dimensions d'analyse (Profondeur =3, Degré =3)

FIG. 2 – *Expérimentations sur jeux de données synthétiques*

Les figures 2(a) et 2(b) montrent le nombre de séquences extraites et le temps d'exécution en fonction du support. Le nombre de fréquents augmente globalement lorsque le support diminue, ainsi que le coût de l'extraction. Néanmoins, il peut arriver que le nombre de fréquents diminue lorsque le support diminue. Ceci est dû au fait que des items plus spécifiques sont fréquents. Or un item plus général est plus rapidement fréquent dans une séquence de données. Il est possible alors d'obtenir moins de séquences.

Les figures 2(c) et 2(d) montrent le nombre de fréquents extraits et le temps d'exécution en fonction de la profondeur des hiérarchies pour un seuil de support fixé. Etendre la hiérarchie d'un niveau engendre une spécialisation supplémentaire des données ($Soda$ devient $pepsi$ ou $coca$). Il y a ainsi plus de valeurs différentes dans la base de données. $M2S\_CD$ apporte une certaine robustesse face à ce phénomène de spécialisation. En effet, même si les données deviennent très détaillées (5 niveaux dans la hiérarchie), notre approche permet d'extraire des séquences définies sur plusieurs niveaux de hiérarchies. On remarque cependant que le temps

de traitement est plus long quand le nombre de niveaux augmente. Ceci est dû au nombre d'items h-généralisés potentiellement fréquents qui augmente.

Les figures 2(e) et 2(f) montrent le nombre de séquences extraites et le temps d'exécution en fonction du degré des hiérarchies. Augmenter le degré d'une hiérarchie équivaut à spécialiser les données (ajout de fils à une instance). Notre approche permet de continuer à extraire des connaissances lorsque la hiérarchie se spécialise. Le temps de traitement devient cependant plus coûteux.

Les figures 2(g) et 2(h) montrent le nombre de séquences extraites et le temps d'exécution en fonction du nombre de dimension d'analyse. Augmenter le nombre de dimensions d'analyse engendre une augmentation du nombre de fréquents et du coût de leur extraction.

Ces expérimentations menées sur des données synthétiques montrent la robustesse de M2S_-CD pour l'extraction des connaissances face à la diversité des données (nombre de dimensions, degré et profondeur des hiérarchies, etc.). Diversifier les données sources engendre un coût de traitement plus important qui reste cependant acceptable.

4.2 Données réelles

Nous avons étudié plusieurs parties du jeu *Eleusis*. Eleusis est un jeu de cartes dont le but est de trouver une règle secrète. Les règles secrètes sont des séquences de cartes contenant une partie droite et une partie gauche. Chaque partie peut contenir plusieurs cartes. Ce jeu permet de simuler la découverte scientifique qui est formée de tests, publications et réfutations. Nous avons donc analysé différentes parties du jeu développé par Dartnell et Sallantin (2005). Nous avons décrit ce problème selon plusieurs dimensions d'analyse :

- une dimension organisant la valeur des cartes (figures, chiffres, impairs, pairs, etc.)
- une dimension organisant la couleur des cartes (rouge, noir, cœur, etc.)
- une dimension positionnant la carte dans la séquence (partie droite ou gauche)
- une dimension pour la réponse de l'oracle (exemple positif ou négatif).

A l'aide de $M2S\_CD$, nous obtenons des séquences convergentes et divergentes sur ce jeu de données. Une des séquences divergentes extraites est : *Pour la règle nénuphar, les joueurs jouent fréquemment le trois de pique, puis l'as de pique avant de jouer un carte impaire de type pique puis une carte impaire de couleur noire*. Une des séquences convergentes extraites est : *Pour la règle lis, les joueurs proposent d'abord une carte rouge puis une carte de cœur et enfin une carte de type chiffre de couleur cœur*. Notons que ces règles, déclarées pertinentes par l'expert, n'auraient jamais pu être extraites à l'aide d'un algorithme classique.

5 Conclusion

Dans cet article, nous proposons une méthode originale pour extraire des connaissances multidimensionnelles définies sur plusieurs niveaux de hiérarchies mais selon un certain point de vue : du général au particulier ou vice et versa. Nous définissons ainsi le concept de séquences multidimensionnelles convergentes ou divergentes ainsi que les algorithmes associés basés sur le paradigme "pattern growth". Des expérimentations, sur des jeux de données synthétiques et réelles, montrent l'intérêt de notre approche $M2S\_CD$.

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 permet des élagages supplémentaires et améliore ainsi la robustesse de l'extraction. D'autres propositions peuvent être effectuées pour la gestion des hiérarchies. Nous pouvons imaginer une gestion modulaire des hiérarchies où certaines dimensions n'auraient pas le même comportement afin de s'adapter aux besoins de l'utilisateur (interdiction de dépasser le niveau de hiérarchie λ sur la dimension ξ, ...) et d'assurer la passage à l'échelle de cette approche.

Références

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

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

Dartnell, C. et J. Sallantin (2005). Assisting scientific discovery with an adaptive problem solver. In *Discovery Science*, pp. 99–112.

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

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

Pinto, H., J. Han, J. Pei, K. Wang, Q. Chen, et U. Dayal (2001). Multi-dimensional sequential pattern mining. In *Proceedings of the 2001 ACM CIKM International Conference on Information and Knowledge Management, Atlanta, Georgia, USA, November 5-10, 2001*, pp. 81–88. ACM.

Plantevit, M., Y. W. Choong, A. Laurent, D. Laurent, et M. Teisseire (2005). M^2SP : Mining sequential patterns among several dimensions. In *Knowledge Discovery in Databases : PKDD 2005, Porto, Portugal, October 3-7, 2005, Proceedings*. Springer.

Plantevit, M., A. Laurent, et M. Teisseire (2006a). HYPE : Mining hierarchical sequential patterns. In I.-Y. Song et P. Vassiliadis (Eds.), *DOLAP 2006, ACM 9th International Workshop on Data Warehousing and OLAP, Arlington, USA, November 9-10, 2006, Proceedings*. ACM.

Plantevit, M., A. Laurent, et M. Teisseire (2006b). 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.

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

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

Summary

Mining sequential patterns aims at discovering correlations between events through time. Multidimensional sequential patterns, recently introduced, consider several dimensions of analysis in order to discover more relevant patterns. Even if several works have been published in a multidimensional framework. No work proposes to take all the specificities of this context (*e.g. hierarchies*). In this paper, we propose a novel approach for mining hierarchical multidimensional sequential patterns. We define the concepts of convergent and divergent sequences. The algorithm *M2S_CD* is *pattern growth* based. Some experiments on synthetic and real data show the relevance of our approach.