HAL
open science

A la Recherche des Motifs Séquentiels Flous

Céline Fiot, Gérard Dray, Anne Laurent, Maguelonne Teisseire

To cite this version:

Céline Fiot, Gérard Dray, Anne Laurent, Maguelonne Teisseire. A la Recherche des Motifs Séquentiels Flous. LFA 2004 - 12èmes Rencontres Francophones sur la Logique Floue et ses Applications, Nov 2004, Nantes, France. pp.131-138. lirmm-00108890

HAL Id: lirmm-00108890

https://hal-lirmm.ccsd.cnrs.fr/lirmm-00108890

Submitted on 5 Nov 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

À la recherche des motifs séquentiels flous

Mining Fuzzy Sequential Patterns

C. Fiot[1] G. Dray[2] A. Laurent[1] M. Teisseire[1]
[1] LIRMM - Université Montpellier II
[2] LGI2P - Ecole des Mines d'Alès

161, rue Ada - 34092 Montpellier Cedex 5 , {fiot, laurent, teisseire}@lirmm.fr
Site EERIE - Parc Scientifique Georges Besse - F 30035 Nîmes Cedex 1, gerard.dray@ema.fr

Résumé :

Les motifs séquentiels ont été étudiés depuis maintenant plusieurs années. Ils permettent de traiter de gros volumes de données et d'en extraire des règles incluant la dimension temporelle de la forme : $<(\text{chocolat pain})(\text{lait})>$ signifiant que les clients qui achètent simultanément du chocolat et du pain achètent plus tard du lait. Cependant, les algorithmes proposés ne travaillent que sur des données binaires qui indiquent l'absence ou la présence d'éléments. Or la plupart des données réelles, intéressantes dans le contexte de données séquentielles, sont numériques (par exemple pour décrire des données issues de capteurs prélevées au cours du temps). Dans ce contexte, nous proposons une extension des méthodes existantes pour permettre l'extraction de motifs séquentiels flous.

Mots-clés :

Fouille de données, Motifs séquentiels flous, Données quantitatives.

Abstract:

Sequential patterns have been studied for several years. They allow the efficient treatment of large datasets in order to discover rules like : $<(\text{chocolate bread})(\text{milk})>$ corresponding to the fact that customers who buy simultaneously chocolate and bred buy later milk. However, these algorithms are built to deal with binary data. However, data from the real worl that are interesting for sequential pattern mining are often quantitative. We propose thus a method based on fuzzy sets in order to mine fuzzy sequential patterns.

Keywords:

Data Mining, Fuzzy Sequential Patterns, Quantitative Data.

1 Introduction

L'explosion des volumes de données disponibles, dans les domaines commerciaux (promotion de ventes, suivi de clientèle par exemple), a récemment transformé la manière dont sont traitées les informations et a rendu nécessaire le développement de méthodes efficaces et pertinentes de production de règles. De nombreux de travaux de recherche se sont focalisés sur la recherche de règles d'association [LAS97, AMS97, AS95] et plus récemment sur les règles d'association flous [KFW98, FWS+98] en particulier dans l'objectif de traiter les données quantitatives. Néanmoins, très peu de travaux ont été réalisés dans le contexte des motifs séquentiels qui sont une extension des règles d'association. Les motifs séquentiels permettent d'extraire des connaissances en y incluant les aspects temporels. Par exemple, des règles du type suivant sont extraites : "*Les clients ayant acheté un téléviseur achètent un lecteur DVD quelque temps plus tard*". Si des algorithmes existent pour extraire des motifs séquentiels, ceux-ci ne permettent pas un traitement efficace de bases de données quantitatives. Il est alors intéressant de proposer une approche basée sur la logique floue pour résoudre cette problématique.

Dans cet article, nous proposons une méthode d'extraction de motifs séquentiels flous afin de traiter les données quantitatives. Un travail similaire a été proposé dans [CTCH01, HCTS03, CH02], cependant il n'est pas satisfaisant car il ne permet pas de différencier les co-occurences d'événements par rapport aux occurences successives. Or ce point est fondamental dans la défintion même d'un motif qui intègre le concept de séquence. Notre proposition est donc la seule, à notre connaissance, à définir une méthode

adaptée et complète de traitement des données quantitatives par les motifs séquentiels flous.

Cet article se présente comme suit : la section 2 présente brièvement les motifs séquentiels, la section 3 présente les travaux existants permettant le traitement de bases quantitatives par l'extraction de règles d'association et de motifs séquentiels ainsi qu'une proposition de formalisation de ces propositions. La section 4 présente notre proposition de motifs séquentiels flous et déroule un exemple complet. Enfin, la section 5 fait le bilan de notre travail et présente les perspectives associées.

2 Motifs séquentiels

Considérons DB une base regroupant l'ensemble des achats réalisés par des clients. Chaque n-uplet T correspond à une transaction financière et consiste en un triplet *(id-client, id-date, itemset)* : l'identifiant du client, la date de l'achat ainsi que l'ensemble des produits (items) achetés .
Soit $I = \{i_1, i_2, ..., i_m\}$ l'ensemble des $items$ (produits). Un *itemset* est un ensemble d'items non vide noté $(i_1 i_2 \ldots i_k)$ où i_j est un *item*, il s'agit d'une représentation non ordonnée. Une *séquence* s est une liste ordonnée, non vide, d'itemsets notée $< \ s_1 s_2 ... s_p \ >$ où s_j est un itemset. Une *n-séquence* est une séquence composée de n items. Par exemple, considérons les achats des produits $1, 2, 3, 4, 5$, réalisés par le client Dupont selon la séquence $s =< \ (1) \ (2 \ 3) \ (4) \ (5) \ >$ indiquée dans le tableau 1. Ceci signifie qu'hormis les achats des produits 2 et 3 qui ont été réalisés ensemble, *i.e.* lors de la même transaction, les autres items de la séquence ont été achetés séparément. Dans notre exemple, s est une 5-séquence.
Une séquence $< \ s_1 \ s_2 ... s_p \ >$ est une sous-séquence d'une autre séquence $< s'_1 \ s'_2 \ ... s'_m >$ s'il existe des entiers $i_1 < i_2 < ... i_j ... < i_n$ tels que $s_1 \subseteq s'_{i_1}, s_2 \subseteq s'_{i_2}, ..., s_p \subseteq s'_{i_n}$. Par exemple, la séquence s' = <(2) (5)> est une sous-séquence de s car (2) $\subseteq$ (2 3) et (5) $\subseteq$ (5). Toutefois, <(2) (3)> n'est pas une sous-séquence de s.

Tous les achats d'un même client sont regroupés et triés par date. Ils constituent la séquence de données du client. Un client *supporte* une séquence s si s est incluse dans la séquence de données de ce client (s est une sous-séquence de la séquence de données). Le *support* d'une séquence s est le pourcentage des clients qui supportent s. Soit $minSupp$ le support minimum fixé par l'utilisateur, une séquence qui vérifie le support minimum (*i.e.* dont le support est supérieur à $minSupp$) est une *séquence fréquente*.

Client	Date	Items
Dupont	12/01	pain (1)
Martin	28/02	chocolat(5)
Dupont	02/03	lait (2) , soda (3)
Dupont	12/03	café (4)
Dupont	26/04	chocolat (5)

Tableau 1 – Exemple d'une base d'achats

Le problème de la recherche de motifs séquentiels dans une base de données consiste à trouver les séquences maximales dont le support est supérieur au support minimum spécifié. Chacune de ces séquences est un motif séquentiel ou plus communément une séquence fréquente. Cependant, comme nous pouvons le constater par cette définition, les items sont traités de façon binaire : absence ou présence.

3 Règles d'association et motifs séquentiels pour les attributs quantitatifs

Les bases de données issues du monde réel sont souvent constituées d'attributs quantitatifs, comme l'illustre le tableau 2. Les algorithmes classiques d'extraction de règles d'association et de motifs séquentiels s'avèrent alors inadaptés pour traiter ce type d'informations. Pour pallier ce problème, les règles d'association floues ont été définies ainsi que des propositions non satisfaisantes dans le cadre des motifs séquentiels.

Client	Date	pain	chocolat	lait	soda	café
Dupont	12/01	1	0	0	0	0
Martin	28/02	0	5	0	0	0
Dupont	02/03	0	0	6	3	0
Dupont	12/03	0	0	0	0	3
Dupont	26/04	0	5	0	0	0

Tableau 2 – Exemple d'une base d'achats

Dans [KFW98], une extension des règles d'association, basée sur la théorie des ensembles flous, permet de raisonner sur des attributs quantitatifs. Il s'agit de rechercher des règles d'association en utilisant les concepts de la théorie des ensembles flous. Soit $\mathcal{T}$ la base des transactions, où chaque transaction t est n-uplet de $\mathcal{T}$. Soit l'ensemble $\mathcal{I}$ des attributs i apparaissant dans $\mathcal{T}$. On note $t[i]$ la valeur de l'attribut i pour la transaction t.
A chaque attribut i, on associe plusieurs sous-ensembles flous, qui définissent une partition floue. Soit l'ensemble $\mathcal{F}_i = \{F_i^1, F_i^2, \ldots, F_i^{l_i}\}$ de sous-ensembles flous associés à l'attribut i. On note $\mu_{F_i^{\lambda_i}}(t[i])$ la fonction d'appartenance de l'attribut i de la transaction t au sous-ensemble flou $F_i^{\lambda_i}$. On considère que ce découpage ainsi que les fonctions d'appartenance aux sous-ensembles flous sont connus, par exemple parce qu'ils sont fournis par un expert du domaine.

Une règle d'association floue se présente sous la forme "*Si X est A, alors Y est B*" où la partie "*X est A*" est l'antécédent (ou condition) de la règle et la partie "*Y est B*" est le conséquent (ou conclusion) de la règle. Cette règle se note $(X, A) \rightarrow (Y, B)$, X et Y sont deux itemsets disjoints, sous-ensembles de $\mathcal{I}$ et A et B sont les ensembles des sous-ensembles flous associés aux éléments de X et Y, tels que pout tout x de X, a de A est un élément de $\mathcal{F}_x$ (i.e. $a = F_x^{\lambda_x}$, $\lambda_x \in [1; l_x]$) et pout tout y de Y, b de B est un élément de $\mathcal{F}_y$ (i.e. $b = F_y^{\lambda_y}$, $\lambda_y \in [1; l_y]$).
Une règle est *satisfaite* si un nombre suffisant de transactions de $\mathcal{T}$ supportent (contiennent) les paires [attribut x, sous-ensemble flou a] et [attribut y, sous-ensemble flou b].

Support d'un itemset (X, A)
Par définition, le support d'un itemset est le pourcentage du nombre de transactions supportant le couple (X, A) par rapport au nombre total de transactions dans la base T.
Une transaction t supporte (X, A) signifie que la valeur $t[x]$ est non nulle pour tout ($x \in X$, $a \in A$), i.e. cette transaction est non nulle pour les attributs de X considérés.
Afin de ne considérer que les attributs significatifs, on introduit un seuil minimum d'appartenance ω, en-dessous duquel on considère que la transaction ne contient pas le couple $[x, a]$.
On calcule alors le support d'un itemset pour une transaction grâce aux fonctions d'appartenance $\mu_a(t[x])$ définies pour chacun des sous-ensembles flous des attributs, on utilisera une $\alpha-$coupe de seuil ω de la fonction d'appartenance :

$$\alpha_a(t[x]) = \begin{cases} \mu_a(t[x]) & \text{si } \mu_a(t[x]) > \omega \\ 0 & \text{sinon} \end{cases}$$

Nous proposons de définir de façon rigoureuse les concepts proposés dans [KFW98, FWS+98]. Pour ceci, nous noterons $\overline{\top}$ (resp. $\underline{\perp}$) la généralisation d'une t-norme $\top$ (resp. t-conorme $\perp$) au cas n-aire.

Pour chaque transaction t, le support de l'itemset est alors donné par la combinaison des supports des éléments de (X, A) :

$$nb_{t_{(X,A)}} = \overline{\top}_{(x,a) \in (X,A)}[\alpha_a(t[x])]$$

Puis le nombre de transactions supportant le couple (X, A) est donné par :

$$\begin{aligned} nb_{(X,A)} &= \textstyle\sum_{t \in \mathcal{T}} nb_{t_{(X,A)}} \\ &= \textstyle\sum_{t \in \mathcal{T}} \overline{\top}_{[x,a] \in (X,A)}[\alpha_a(t[x])] \end{aligned}$$

Le support d'un itemset est alors donné par la relation :

$$\begin{aligned} FSupp_{(X,A)} &= \frac{nb_{(X,A)}}{|\mathcal{T}|} \\ &= \frac{\sum_{t \in \mathcal{T}} \overline{\top}_{[x,a] \in (X,A)} \left[\alpha_a\left(t[x]\right)\right]}{\Theta} \end{aligned}$$

Confiance d'une règle $(X, A) \rightarrow (Y, B)$
La confiance d'une règle $(X, A) \rightarrow (Y, B)$ est définie comme étant le rapport du support du couple $(X \cup Y, A \cup B)$ (nombre de transactions supportant à la fois la condition et la conclusion de la règle) par le support de l'antécédent (X, A). Ce qui donne la formule suivante, d'après la définition du support donnée précédemment :

$$FConf_{(X,A)\rightarrow(Y,B)} = \frac{\sum_{t\in\mathcal{T}} \overline{\top}_{[z,e]\in(Z,E)}\Big[\alpha_e(t[z])\Big]}{\sum_{t\in\mathcal{T}} \overline{\top}_{[x,a]\in(X,A)}\big[\alpha_a(t[x])\big]}$$

où $(Z = X \cup Y, E = A \cup B)$

Pour l'extraction de motifs séquentiels flous, le principe est le même que pour les règles d'association : un motif séquentiel flou consiste en une séquence fréquente de plusieurs attributs quantitatifs. [CTCH01, HCTS03, CH02] propose la recherche de motifs séquentiels flous et plus généralement de connaissances floues relatives au comportement d'achats des consommateurs.
Le principe est le suivant : une séquence floue est une liste ordonnée de sous-ensembles flous fréquents, correspondants à des attributs quantitatifs ("*peu de chocolat*" par exemple). Comme pour les motifs séquentiels flous classiques, on définit un support minimum, afin de déterminer quels sont les fréquents dans la base. Puis on calcule le support des items seuls tout d'abord, puis des itemsets et des séquences.
Toutefois, si l'algorithme présenté dans [HCTS03, CTCH01, CH02] satisfait aux propriétés d'Apriori [SA96a] pour la recherche de sous-ensembles flous et de séquences floues fréquentes, la définition proposée pour le calcul du support ne prend pas en compte la notion d'ordre entre les itemsets, essentielle pour la recherche de séquences (listes ordonnées d'itemsets) fréquentes.

Ainsi, le calcul du support des séquences $< (a)(b) >$ et $< (ab) >$ est identique, ce qui n'est pas satisfaisant dans notre contexte.

Une seconde proposition [HLW01] prend en compte cette nuance pour une succession d'items, mais ne tient pas compte du fait qu'un même client peut avoir acheté plusieurs fois la même séquence. Par ailleurs, cette proposition ne conserve pas la totalité des sous-ensembles flous pour chaque item, une partie de l'information est alors perdue. Nous proposons donc une nouvelle définition du support d'une séquence floue dans la section suivante.

4 Motifs séquentiels flous

Dans cette section, nous présentons une nouvelle définition des motifs séquentiels flous. La contribution essentielle de cette proposition est la définition adaptée du support d'une séquence.

4.1 Définitions

Soit $\mathcal{T}$ une base des transactions contenant Θ transactions t. Chacune des transactions comporte les valeurs des achats pour I attributs i. Soit $\mathcal{I}$ l'ensemble de ces attributs et $t[i]$ la valeur de l'attribut i pour la transaction t. Chaque attribut i est partitionné en sous-ensembles flous. A l'état actuel de nos travaux, ce découpage est supposé connu, soit parce qu'il est construit automatiquement (par exemple par clustering flou), soit parce qu'il est fourni par un expert du domaine. Soient $\mu_{F_i^{\lambda_i}}$ les fonctions d'appartenance pour les attributs i au sous-ensemble $F_i^{\lambda_i}$.
L'utilisateur fixe ω, seuil d'appartenance minimale (pour un comptage seuillé des items).
A la différence des règles d'association, pour les motifs séquentiels on repère les transactions par rapport au client qui les a réalisées. On note $\mathcal{C}$ l'ensemble des Γ clients c et θ_c le nombre de transactions du client c.

La notion d'itemset est modifiée par rapport aux motifs séquentiels classiques :

Définition 1 *Un* item flou *est un couple [item, sous-ensemble flou quantitatif].*

Par exemple, *[chocolat, beaucoup]* est un item flou où *beaucoup* est un sous-ensemble flou défini par sa fonction d'appartenance.

Définition 2 *Un* itemset flou *est un ensemble d'items flous. Il peut être écrit sous la forme d'un couple de deux ensembles (ensemble d'items, ensemble des sous-ensembles flous associés à chaque item) ou sous la forme d'une liste d'items flous.*

Par exemple, *([chocolat, beaucoup][lait, peu])* est un itemset flou où *beaucoup* et *peu* sont deux sous-ensembles flous.

Enfin, on définit le terme de g-k-séquence.

Définition 3 *Une* g-k-séquence S *telle que* $S = < s_1\ s_2\ \ldots\ s_g >$ *est une séquence composée de* g *itemsets flous* $s = (X, A)$ *regroupant au total* k *items flous de la forme* $[x_p, a_p]$.

Prenons la séquence *<([chocolat, beaucoup]) [lait, peu])([lait, peu])>*. Elle regroupe trois items flous dans deux itemsets, il s'agit d'une 2-3-séquence floue.

4.2 Support d'un itemset flou

On définit le support d'un itemset flou comme le pourcentage de clients supportant cet itemset flou par rapport au nombre total de clients dans la base.
Il s'agit donc de compatibiliser, pour chaque client, le nombre de fois où l'on rencontre l'itemset parmi les transactions de ce client, c'est-à-dire le nombre de transactions dans lesquelles on rencontre chacun des couples (items/sous-ensembles flous associés).
Pour cela on définit le degré d'appartenance d'un itemset (X, A) à un client c comme $\underline{\perp}_{j=1}^{\theta_c}[\overline{\top}_{[x,a]\in(X,A)}\alpha_a(t_j[x])]$ où α représente la fonction d'appartenance seuillée, avec

$$\alpha_a(t_j[x]) = \begin{cases} \mu_a(t_j[x]) & \text{si } \mu_a(t_j[x]) > \omega \\ 0 & \text{sinon} \end{cases}$$

La fonction α permet d'utiliser un comptage seuillé des items puisqu'on ne prendra en compte que les valeurs d'attributs dépassant le seuil minimal d'appartenance.
Le support d'un itemset (X, A) est donc donné par la relation

$$FSupp_{(X,A)} = \frac{\sum_{c\in\mathcal{C}}\left[\underline{\perp}_{j=1}^{\theta_c}\overline{\top}_{[x,a]\in(X,A)}\left[\alpha_a(t_j[x])\right]\right]}{|\mathcal{C}|}$$

4.3 Support d'une g-k-séquence

De manière informelle, le support d'une g-k-séquence peut être vu comme la moyenne des meilleures co-occurrences des itemsets. Ainsi, pour chaque client de la base de données, on considère comme support le degré maximum avec lequel l'ensemble des itemsets apparaissent de manière successive.

On considère par exemple la séquence $< \ (a\ b)(c)\ >$. Le degré avec lequel a et b apparaissent ensemble est calculé à partir d'une t-norme (le min dans notre cas), tandis que les degrés de chacun des itemsets seront agrégés à l'aide d'une t-conorme (le max dans notre cas). Ces degrés, calculés pour chacun des clients, sont ensuite agrégés en considérant la moyenne. On obtient donc l'algorithme 1 où FS représente le support flou et *TrouverSequence* est l'algorithme permettant de définir les séquences candidates.

CalculerSupport - Input : g-S ; Ouput : FS ;
Flottant $FS, nbSupp, m, m2, \eta$;
Flottant[g] $Res[\,]$;
Liste de transactions $Trans$;
$FS, nbSupp \leftarrow 0$;
for *chaque client* $c \in \mathcal{C}$ **do**
  $m \leftarrow 0$; $\eta \leftarrow 0$;
  for $i = 0$ *à* $g-1$ **do** $Res[i] \leftarrow 0$;
  for $j = 1$ *à* θ_c **do**
    $Trans \leftarrow \{t_j \ldots t_{\theta_c}\}$;
    TrouverSequence(g-S, $Trans$, Res, η) ;
    $m2 \leftarrow \odot Res$; *// opérateur d'agrégation*
    // pour chaque client, on cherche le meilleur degré
    if $m2 > m$ **then** $m \leftarrow m2$;
  end
  $nbSupp$ += m ;
end
$FS \leftarrow nbSupp/\Gamma$;
return [FS] ;

Algorithme 1: CalculerSupport

4.4 Extraction des g-k-séquences fréquentes

L'extraction des séquences floues suit le même processus que l'extraction des motifs séquentiels classiques. Un algorithme par niveau est utilisé. Les motifs fréquents flous de taille 1 sont d'abord extraits, puis utilisés pour la construction des motifs de taille 2, *etc.*

Dans le cadre des motifs séquentiels flous, deux différences principales apparaissent en comparaison aux motifs classiques :

- les supports sont calculés en considérant le support flou,
- le même item peut se retrouver au sein d'une même séquence mais pas au sein d'un même itemset (il n'est par exemple pas possible de considérer la séquence *<([chocolat,peu][chocolat,beaucoup])>*).

L'algorithme 2 montre comment sont définies les séquences candidates. Ce sont ces séquences qui font ensuite l'objet d'un comptage pour déterminer leur support, comme indiqué dans la section précédente.

TrouverSequence - Input : g-S, T, Res, η ; Ouput : Res ;
Itemset s ;
Transaction t ;
Séquence $sRem$;
Liste de transactions $tRem$;
$s \leftarrow g\text{-}S.first$;
$t \leftarrow T.first$;
$sRem \leftarrow g\text{-}S - g\text{-}S.first$;
$tRem \leftarrow T - T.first$;
if $sRem \neq \emptyset$ & $tRem \neq \emptyset$ **then**
  if $\overline{\top}_{[x,a]\in g-S.first}\{\alpha_a(T.first[x])\} > minSupp$ **then**
    $Res[\eta] \leftarrow \overline{\top}_{[x,a]\in g-S.first}\{\alpha_a(T.first[x])\}$;
    TrouverSequence($sRem, tRem, Res, \eta + 1$) ;
  else
    TrouverSequence(g-S, $tRem$, Res, η) ;
  end
else
  return [Res] ;
end

Algorithme 2: TrouverSequence

Les séquences peuvent être représentées et stockées, ainsi que leur support, grâce à la structure Prefix-Tree, utilisée par l'algorithme d'extraction de motifs séquentiels PSP [MPT03]. L'arbre illustré par la figure 1 représente les séquences *<([chocolat, peu][lait, beaucoup])([lait, beaucoup])>*, *<([chocolat, peu])([pain, moyen])>*, ainsi que les sous-séquences *<([pain, moyen])>* et *<([lait, beaucoup])([lait, beaucoup])>*.

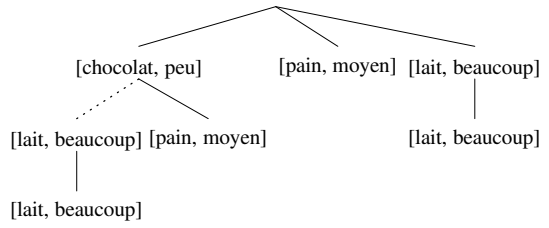

Figure 1 – Stockage de séquences sous forme d'arbre

4.5 Exemple

Dans cette section, nous développons la méthode proposée dans cet article sur un exemple.

Base des transactions. La base d'achats décrite par la figure 2 comporte la quantité achetée pour chaque produit lors des transactions, chacune étant identifiée par un client et une date.
Les cases vides correspondent à une quantité achetée nulle.

Cl.	Date	chocolat	pain	lait	fromage	chips	saucisson
C1	d1	2					
	d2	1	3	1			
	d3	2		1			
	d4				4		
	d5			2			2
C2	d1	2			1		
	d2			2			
	d3		4	1			
	d4	3				5	
C3	d1					2	3
	d2	3	1				
	d3				4		5
	d4			2			
	d5		2				
C4	d1					2	
	d3	2					4
	d4			3			
	d5		2				
	d6			2			

Figure 2 – Transactions

On considère $\omega = 0.49$ et un support minimal $minSupp = 0.5$. L'opérateur $\underline{\perp}$ (resp. $\overline{\top}$) considéré est le max (resp. le min), l'opérateur d'agrégation $\odot$ considéré est la moyenne.

Représentation des fonctions d'appartenance. La première étape consiste à convertir la base de

données quantitative en base de données de degrés d'appartenance. Pour cela, chaque attribut est partitionné en sous-ensembles flous. La figure 3 montre le découpage en sous-ensembles flous et les fonctions d'appartenance à chacun de ces sous-ensembles.

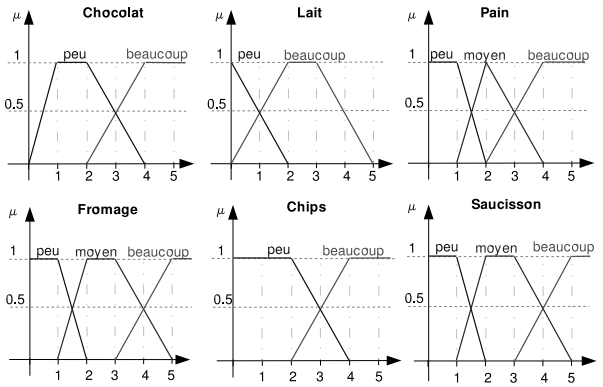

Figure 3 – Partitionnement flou des attributs numériques

L'attribut "Pain" par exemple est découpé en trois parties, "Peu de pain", "Moyen de pain" et "Beaucoup de pain". Un achat de 1 pain par exemple est considéré comme appartenant à "Peu de pain", de 2 comme "moyen", 3 pains est à la fois une quantité "moyenne" et une "grande" quantité et à partir de 4 pains, l'achat comporte "beaucoup de pain".

A partir des fonctions d'appartenance ci-dessus, on définit la base de la figure 4, qui donne les degrés d'appartenance de chaque transaction pour chacun des sous-ensembles flous.

		chocolat		pain			lait		fromage			chips		saucisson		
Cl.	**D.**	P	B	P	M	B	P	B	P	M	B	P	B	P	M	B
C1	d1	1														
	d2	1			0.5	0.5	0.5	0.5								
	d3	1					0.5	0.5								
	d4										1					
	d5							1							1	
C2	d1	1							1							
	d2							1								
	d3					1	0.5	0.5								
	d4	0.5	0.5										1			
C3	d1											1			1	
	d2	0.5	0.5	1												
	d3										1					1
	d4							1								
	d5				1											
C4	d1											1				
	d3	1													0.5	0.5
	d4							1								
	d5				1											
	d6							1								

Figure 4 – Base des appartenances

Recherche des items fréquents. Les items flous correspondants à la base exemple sont les suivants : *[chocolat, peu]*, *[chocolat, beaucoup]*, *[lait, peu]*, *[lait, beaucoup]*, *[pain, peu]*, *[pain, moyen]*, *[pain, beaucoup]*, *[fromage, peu]*, *[fromage, beaucoup]*, *[chips, peu]*, *[chips, beaucoup]*, *[saucisson, moyen]* et *[saucisson, beaucoup]*.

Item	Sous-ensemble	FSupp
Chocolat	Peu	**0.875**
	Beaucoup	0.25
Pain	Peu	0.25
	Moyen	**0.625**
	Beaucoup	0.375
Lait	Peu	0.25
	Beaucoup	**1**
Fromage	Peu	0.25
	Moyen	0
	Beaucoup	**0.5**
Chips	Peu	**0.5**
	Beaucoup	0.25
Saucisson	Peu	0
	Moyen	0.375
	Beaucoup	**0.625**

Figure 5 – Support flou des items flous

Les items flous fréquents sont donc *[chocolat, peu]*, *[lait, beaucoup]*, *[pain, moyen]*, *[fromage, beaucoup]*, *[chips, peu]* et *[saucisson, moyen]*. Ce qui signifie que les achats fréquemment réalisés sont *peu de chocolat, beaucoup de lait, moyen de pain, beaucoup de fromage, peu de chips, moyen de saucisson*.

5 Conclusion et Perspectives

Dans cet article, nous présentons une méthode d'extraction automatique de motifs séquentiels flous à partir de motifs séquentiels. Si une proposition est déjà présente dans la littérature, celle-ci n'est pas pertinente puisqu'elle ne prend pas en compte les aspects temporels pourtant essentiels dans le contexte des motifs séquentiels. Notre proposition permet l'extraction de règle de la forme : *<([chocolat,peu][pain,peu])([lait,moyen])>* signifiant que quand un client achète un peu de chocolat et un peu de pain le même jour, ce client achètera du lait (en quantité moyenne) quelque temps plus tard. L'algorithme présenté

dans cet article fait l'objet actuellement d'une implémentation et de tests.

Les perspectives associées à ce travail sont nombreuses et diverses. Tout d'abord, l'algorithme présenté ici doit être comparé à d'autres algorithmes permettant de prendre en compte différents modes de comptage. De plus, nous souhaitons étendre les motifs séquentiels flous aux motifs séquentiels flous généralisés [SA96b, Mas02]. Les motifs généralisés permettent de paramétrer très finement les limites autorisées entre les transactions (par exemple, deux transactions pourront être considérées comme regroupées si elles sont trop proches ou leur séquence sera ignorée si elles sont trop éloignées). Enfin, nos travaux seront utilisés dans le cadre du traitement des données incomplètes, d'une part pour le remplacement des valeurs manquantes, et d'autre part pour la génération de motifs séquentiels directement sur les bases incomplètes.

Références

[AMS97] K. ALI, S. MANGANARIS et R. SRIKANT : Partial classification using association rules. *In Knowledge Discovery and Data Mining*, pages 115–118, 1997.

[AS95] R. AGRAWAL et R. SRIKANT : Mining sequential patterns. *In Eleventh International Conference on Data Engineering*, pages 3–14, Taipei, Taiwan, 1995. IEEE Computer Society Press.

[CH02] R.-S. CHEN et Y.-C. HU : A novel method for discovering fuzzy sequential patterns using the simple fuzzy partition method. *Journal of the American Society for Information Science*, 54(7):660–670, 2002.

[CTCH01] R.-S. CHEN, G.-H. TZENG, C.-C. CHEN et Y.-C. HU : Discovery of fuzzy sequential patterns for fuzzy partitions in quantitative attributes. *In ACS / IEEE International Conference on Computer Systems and Applications (AICCSA)*, pages 144–150, 2001.

[FWS+98] A. FU, M. WONG, S. SZE, W. WONG, et W. YU : Finding fuzzy sets for the mining of fuzzy association rules for numerical attributes. *In the First International Symposium on Intelligent Data Engineering and Learning (IDEAL)*, pages 263–268, 1998.

[HCTS03] Y.-C. HU, R.-S. CHEN, G.-H. TZENG et J.-H. SHIEH : A fuzzy data mining algorithm for finding sequential patterns. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, 11(2):173–193, 2003.

[HLW01] T.P. HONG, K.Y. LIN et S.L. WANG : Mining fuzzy sequential patterns from multiple-items transactions. *In Proceedings of the Joint 9th IFSA World Congress and 20th NAFIPS International Conference*, pages 1317–1321, 2001.

[KFW98] C. M. KUOK, A. W.-C. FU et M. H. WONG : Mining fuzzy association rules in databases. *SIGMOD Record*, 27(1):41–46, 1998.

[LAS97] B. LENT, R. AGRAWAL et R. SRIKANT : Discovering trends in text databases. *In Proc. 3rd Int. Conf. Knowledge Discovery and Data Mining, KDD*, pages 227–230. AAAI Press, 14–17 1997.

[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, 2002.

[MPT03] F. MASSEGLIA, P. PONCELET et M. TEISSEIRE : Incremental mining of sequential patterns in large databases. *Data and Knowledge Engineering*, 46(1):97–121, 01 2003.

[SA96a] R. SRIKANT et R. AGRAWAL : Mining quantitative association rules in large relational tables. *In Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*, pages 1–12, Montreal, Quebec, Canada, 4–6 1996.

[SA96b] R. SRIKANT et R. AGRAWAL : Mining Sequential Patterns : Generalizations and Performance Improvements. *In Proceedings of the 5th International Conference on Extending Database Technology (EDBT'96)*, pages 3–17, Avignon, France, 9 1996.