Extension de l'algorithme Apriori et des règles d'association aux cas des données symboliques diagrammes et intervalles

Filipe Afonso*,**, Edwin Diday*

*Ceremade-Université Paris Dauphine, Place du Maréchal de Lattre de Tassigny, 75775 Paris Cedex 16, France
afonso@ceremade.dauphine.fr, diday@ceremade.dauphine.fr
**Lamsade-Université Paris Dauphine

Résumé. Nous traitons l'extension de l'algorithme Apriori et des règles d'association aux cas des données symboliques diagrammes et intervalles. La méthode proposée nous permet de découvrir des règles d'association au niveau des concepts. Cette extension implique notamment de nouvelles définitions pour le support et la confiance afin d'exploiter la structure symbolique des données. Au fil de l'article, l'exemple classique du panier de la ménagère est développé. Ainsi, plutôt que d'extraire des règles entre différents articles appartenant à des mêmes transactions enregistrées dans un magasin comme dans le cas classique, nous extrayons des règles d'association au niveau des clients afin d'étudier leurs comportements d'achat.

1 Introduction

L'algorithme Apriori défini dans (Agrawal et Srikant 1994) a pour but d'extraire des règles d'association à partir de données classiques issues du panier de la ménagère. Les règles sont du type lait→beurre traduisant le fait que "si du lait est présent dans le panier de la ménagère alors il y a aussi du beurre". Dès lors, des travaux ont exploité la complexité des données afin d'accélérer l'exécution de l'algorithme Apriori ou d'enrichir les règles d'association. Ainsi, (Wang et al. 2000) et (Cai et al. 1998) découvrent des règles pondérées par l'importance d'un même article dans le panier alors que (Srikant et al. 1997) et (Han et Fu 1995) exploitent les relations de taxonomie dans les données. De plus, les règles d'association sont étendues aux données quantitatives et intervalles, notamment (Srikant et Agrawal 1996) et (Miller et Yang 1997) alors que dans (Kuok et al. 1998), les auteurs s'intéressent aux ensembles flous. Ainsi, cet article s'inscrit dans le prolongement de ces travaux. Nous étendons l'algorithme Apriori aux cas des variables symboliques à valeurs diagrammes et intervalles afin d'extraire des règles d'association non plus au niveau des individus mais au niveau des concepts. Cette extension implique notamment de nouvelles définitions pour le support et la confiance afin de tirer parti de la structure symbolique des données. Finalement, nous terminons par une application où nous étudions les comportements d'achat dans des magasins non plus au niveau des transactions comme dans le cas classique mais au niveau des clients. Pour chaque client, nous agrégeons les articles achetés grâce à un diagramme construit avec la proportion de chaque article par rapport aux achats totaux du client et nous enrichissons nos règles par l'ajout d'une variable intervalle sur les montants dépensés par le client.

Transaction	Client	items=Y	montant=X	Transaction	Client	Y	X
t_1	1	v	50	t_{12}	5	c	10
t_2	1	v,p,c	70	t_{13}	5	c	30
t_3	1	v,p,c	90	t_{14}	6	v,p	50
t_4	1	v	60	t_{15}	6	v,p	100
t_5	2	v,p	60	t_{16}	6	v	75
t_6	2	v,p,c	90	t_{17}	7	p,c	10
t_7	2	v	60	t_{18}	7	p,c	20
t_8	3	v,p	55	t_{19}	7	p	15
t_9	3	v	100	t_{20}	8	v,c	55
t_{10}	4	p,c	80	t_{21}	8	v,c	70
t_{11}	4	p	40	t_{22}	8	v	45

TAB. 1 – Matrice de transactions classique. Les colonnes "Transaction" et "items=Y" constituent les données en entrée de l'algorithme Apriori standard

2 Apriori standard et données symboliques

Nous étendons l'algorithme Apriori afin d'extraire nos règles d'association au niveau des concepts car il s'agit de l'algorithme de référence connu du plus grand nombre.

2.1 Algorithme Apriori et règles d'association classiques

Depuis (Agrawal et al. 1993), la recherche d'algorithmes capables d'extraire des règles d'association dans de grandes bases de données a été un thème très étudié. La découverte de règles d'association entre différents produits présents dans le panier de la ménagère a été un exemple d'application particulièrement exploité. Les articles du panier de la ménagère sont appelés items alors que les sous-ensembles d'items sont appelés itemsets. Une transaction est un sous-ensemble d'items enregistré à la caisse d'un supermarché. Ainsi, en entrée de ces algorithmes, nous avons un ensemble de n items $I = \{i_1, ..., i_n\}$ et un ensemble de m transactions $T = \{t_1, ..., t_m\}$ avec $t_i \in P(I) - \oslash$ (voir table 1 colonnes Transaction et Y=items). Une règle d'association est alors définie par deux itemsets X et Y tels que $X \rightarrow Y$ avec $X \subset I$, $Y \subset I$ et $X \cap Y = \oslash$. Dans (Agrawal et Srikant 1994), les auteurs suggèrent l'algorithme Apriori. L'idée est de générer les règles d'association ayant un support *sup* et une confiance *conf* supérieurs à deux seuils minimaux *minsup* et *minconf* respectivement où : $sup(X \rightarrow Y) = \frac{card(t \in T / X \cup Y \subseteq t)}{card(T)}, conf(X \rightarrow Y) = \frac{card(t \in T / X \cup Y \subseteq t)}{card(t \in T / X \subseteq t)} = \frac{sup(X \cup Y)}{sup(X)}$.

Nous ne présentons pas l'algorithme Apriori standard (Agrawal et Srikant 1994). Nous donnons directement son extension, section 4.2. Nous présentons seulement la propriété, sur laquelle repose Apriori, qui nous permet de supprimer les itemsets non-fréquents avant la génération de plus grands itemsets.

Propriété (*) (Agrawal et al. 1993) : Tout itemset inclus dans un itemset fréquent est lui-même fréquent.

Concepts=Clients	items=Y	montant=X	Clients	Y	X
1	1/2v, 1/4p, 1/4c	[50,90]	5	1c	[10,30]
2	1/2v, 1/3p, 1/6c	[60,90]	6	3/5v,2/5p	[50,100]
3	2/3v, 1/3p	[55,100]	7	3/5p,2/5c	[10,20]
4	2/3p,1/3c	[40,80]	8	3/5v,2/5c	[45,70]

TAB. 2 – Matrice de données symboliques avec des concepts clients décrits par une variable diagramme Y (v=viande, p=poisson, c=céréales) et une variable intervalle X

2.2 Entrée de l'Apriori symbolique : une matrice symbolique

L'intérêt principal de l'analyse des données symboliques (ADS) est de passer de l'étude des individus à l'étude des concepts décrits par des variables intervalles, multi-valuées, diagrammes et histogrammes pour lesquelles les opérateurs numériques standards $\times, +, -$ ne peuvent être appliqués directement (voir (Bock et Diday 2000)). Le logiciel SODAS d'ADS a été développé et est disponible à l'adresse http : // www.ceremade.dauphine.fr /%7Etouati/LISE.htm.

Dans ce papier, nous étendons l'algorithme Apriori aux cas des données diagrammes et intervalles. Concrètement, nous n'avons plus, dans notre matrice de données, une valeur unique par case ou bien un sous-ensemble d'items par transaction comme dans le cas classique. Nous avons, dans le cas qualitatif, un diagramme dans chaque case, i.e. des valeurs multiples pondérées telles que la somme des poids soit égale à un, et un intervalle par case dans le cas quantitatif. Cet "Apriori symbolique" va nous permettre d'étudier des concepts. Nous nous appuierons sur l'exemple du panier de la ménagère afin d'en étudier les concepts clients. Cependant, la méthode s'étend à n'importe quels concepts décrits par des variables à valeurs diagrammes et intervalles.

Ainsi, nous considérons la matrice classique, table 1, avec 22 transactions répertoriées dans un supermarché provenant de 8 clients différents. Dans ces transactions, nous nous intéressons aux associations entre 3 catégories d'items v = viande, p = poisson, c = céréales. Nous notons qu'en pratique le nombre d'items est bien plus important. Pour appliquer l'analyse symbolique sur les concepts clients, nous supprimons la première colonne table 1 et nous créons ces concepts (table 2) qui seront les unités statistiques de notre étude. Pour chaque client, cette matrice agrège tous les items achetés sous forme d'un diagramme construit avec la proportion de chaque article par rapport aux achats totaux du client. A cette variable diagramme, nous ajoutons la variable X=montant qui devient une variable intervalle après la création des concepts. Nous obtenons alors une matrice symbolique où chaque ligne définit la "description " d'un client et chaque colonne est associée à une variable symbolique.

2.3 Objets symboliques et règles d'association symboliques

Un objet symbolique (OS) modélise des concepts. Un concept est généralement défini par un ensemble de propriétés appelé intension et un ensemble d'individus satisfaisant ces propriétés appelé extension (voir (Bock et Diday 2000)).

Définition : Soient Ω, l'ensemble des individus et D, l'ensemble des descriptions d'individus ou de classe d'individus. Un OS est un triplet s=(a,R,d) où d∈D est une description, R est une relation permettant de comparer d à une autre description de D et "a", allant de Ω dans L(défini à la suite) , est une fonction de reconnaissance entre les individus et leurs descriptions.

Nous avons deux sortes d'OS pour deux ensembles L différents. Les OS booléens sont tels que $[y(w)Rd] \in L = \{vrai, faux\}$. Nous donnons, par exemple, a(w)=$\{0 < P_{y_v}(w) \leq 1/3\}$ = {vrai} où $w \epsilon \Omega$ et P_{y_v} désigne la fréquence de la modalité v de la variable diagramme Y table 2. Les OS modaux sont tels que [y(w)Rd] ∈ L=[0,1].

Une assertion est alors un OS défini par $[d'Rd] = \wedge_{i=1..k}[d'_i R_i d_i]$, $k \geq 1$. Nous donnons, par exemple, l'assertion booléenne : a(w)=$\{0 < P_{y_v}(w) \leq 1/3\} \wedge \{1/3 < P_{y_p}(w) \leq 1\} \wedge \{[50, 70] \subseteq X(w)\}$ où X est la variable intervalle table 2.

L'extension d'un OS est donné par $Ext(s) = \{w \in \Omega / a(w) = vrai\}$ dans le cas booléen. Ainsi, nous allons rechercher les assertions fréquentes du type a(w)=$\{0 < P_{y_v}(w) \leq 1/3\} \wedge \{1/3 < P_{y_p}(w) \leq 1\} \wedge \{[50, 70] \subseteq X(w)\}$ et nos règles d'association seront alors du type $\{0 < P_{y_v} \leq 1/3\} \wedge \{[50, 70] \subseteq X\} \rightarrow \{1/3 < P_{y_p} \leq 1\}$ qui signifie : "si pour un client donné, la fréquence d'achat de viande est comprise entre 0 ouvert et 1/3 et si l'intervalle [50,70] est inclus dans l'intervalle du montant dépensé X alors la fréquence d'achat de poisson est comprise entre 1/3 et 1" pour un support et une confiance que nous définissons section 3.

3 Définitions du support et de la confiance dans le cas de nos données symboliques.

Soient Ω un ensemble d'individus (concepts), X et Y deux OS ayant pour intensions $a_x(w)=[\bigwedge_{i_1,u}\{\underline{x}_{i_1,u} < P_{X_{i_1,u}}(w) \leq \bar{x}_{i_1,u}\} \bigwedge_{i_3}\{[\underline{s}_{i_3}, \bar{s}_{i_3}] \subseteq S_{i_3}(w)\}]$ et $a_y(w) = [\bigwedge_{i_2,v}\{\underline{y}_{i_2,v} < P_{Y_{i_2,v}}(w) \leq \bar{y}_{i_2,v}\} \bigwedge_{i_4}\{[\underline{t}_{i_4}, \bar{t}_{i_4}] \subseteq T_{i_4}(w)\}]$ avec $\forall i_1, u, i_2, v \; X_{i_1,u} \neq Y_{i_2,v}$ où $P_{X_{i_1,u}}$ ($P_{Y_{i_2,v}}$) est la fréquence de la catégorie u (v) de la variable diagramme X_{i_1} (Y_{i_2}), $\underline{x}_{i_1,u}$ et $\bar{x}_{i_1,u}$ ($\underline{y}_{i_2,v}$ et $\bar{y}_{i_2,v}$) les bornes des intervalles de fréquences et $\forall i_3, i_4$, $S_{i_3} \neq T_{i_4}$ sont des variables intervalles.

Nous donnons 2 méthodes de calcul du support et de la confiance de la règle $X \rightarrow Y$.

Exemple de règle issue de la table 2 : $\{[70, 90] \subset X\} \wedge \{\frac{1}{3} < P_v \leq \frac{2}{3}\} \rightarrow \{0 < P_p \leq \frac{1}{3}\}$.

3.1 Méthode 1 : cas booléen

Pour les définitions suivantes, les extensions de nos objets symboliques se calculent comme dans le cas booléen (voir section 2.3).

Définitions

A) Support. $Sup(X \rightarrow Y) = \frac{card(ext(X \wedge Y) = \{w \in \Omega / a_x(w) = vrai, a_y(w) = vrai\}}{card(\Omega)}$

Exemple : $Sup(\{[70, 90] \subset X\} \wedge \{\frac{1}{3} < P_v \leq \frac{2}{3}\} \rightarrow \{0 < P_p \leq \frac{1}{3}\}) = \frac{1+1+1+0+...}{8} = 37.5\%$

B) Confiance. $Conf(X \rightarrow Y) = \frac{card(ext(X \wedge Y) = \{w \in \Omega / a_x(w) = vrai, a_y(w) = vrai\}}{card(ext(X) = \{w \in \Omega / a_x(w) = vrai\}} = \frac{sup(X \rightarrow Y)}{sup(X)}$

Exemple : $Conf(\{[70,90] \subset X\} \wedge \{\frac{1}{3} < P_v \leq \frac{2}{3}\} \rightarrow \{0 < P_p \leq \frac{1}{3}\}) = \frac{1+1+1+0+...}{1+1+1+0+0+1+0+0} = 75\%$

3.2 Méthode 2 : cas modal

Dans la première méthode, pour le calcul du support, chaque individu vient ajouter 0 ou 1 au support total. Dans cette deuxième méthode, chaque individu vient ajouter au support une valeur dans l'intervalle [0,1]. Ainsi, nous introduisons la notion de contribution au support d'un individu. Ainsi, si r est une règle d'association, Sup(r) = $\sum_w$Contribution(w,r) où contribution(w,r) désigne la contribution de l'individu w au support de la règle r (pour simplifier, nous calculons les supports absolus, c'est-à-dire sans faire la division par card(Ω)).

Dans le cas diagramme, pour calculer Sup($\underline{x}_{i_1,u}<P_{X_{i_1,u}}\leq\bar{x}_{i_1,u}$), ∀w∈Ω nous avons :

- Si $P_{X_{i_1,u}}$(w)∈[$\underline{x}_{i_1,u}$,$\bar{x}_{i_1,u}$] alors contribution(w,$\underline{x}_{i_1,u}<P_{X_{i_1,u}}\leq\bar{x}_{i_1,u}$) = 1 (nous ajoutons 1 au support) ;
- Si 0<$P_{X_{i_1,u}}$(w)<$\underline{x}_{i_1,u}$ alors contribution(w,$\underline{x}_{i_1,u}<P_{X_{i_1,u}}\leq\bar{x}_{i_1,u}$) = 1-($\underline{x}_{i_1,u}$ - $P_{X_{i_1,u}}$(w)) (nous ajoutons 1-($\underline{x}_{i_1,u}$-$P_{X_{i_1,u}}$(w)) au support) ;
- Si $P_{X_{i_1,u}}$(w)>$\bar{x}_{i_1,u}$ alors contribution(w,$\underline{x}_{i_1,u}<P_{X_{i_1,u}}\leq\bar{x}_{i_1,u}$) = 1-($P_{X_{i_1,u}}$(w)-$\bar{x}_{i_1,u}$) ;
- Si $P_{X_{i_1,u}}$(w)=0 alors contribution(w,$\underline{x}_{i_1,u}<P_{X_{i_1,u}}\leq\bar{x}_{i_1,u}$) = 0.

Exemple : $Sup(0<P_p\leq\frac{1}{3})$={1}+{1}+{1}+{1-(2/3-1/3)}+{0}+{1-(2/5-1/3)}+{1-(3/5-1/3)}+{0} = 5.33. Nous notons que pour la première méthode, ce support serait égal à 1+1+1+0+...=3.

Dans le cas intervalle, pour calculer Sup($[\underline{s}_{i_3},\bar{s}_{i_3}]\subseteq S_{i_3}$), ∀w∈Ω nous avons :

- Contribution(w,$[\underline{s}_{i_3},\bar{s}_{i_3}]\subseteq S_{i_3}$) = 1-$\frac{L([\underline{s}_{i_3},\bar{s}_{i_3}]-([\underline{s}_{i_3},\bar{s}_{i_3}]\cap S_{i_3}(w)))}{L([\underline{s}_{i_3},\bar{s}_{i_3}])}$= 1-la proportion de $[\underline{s}_{i_3},\bar{s}_{i_3}]$ non inclus dans $S_{i_3}(w)$ (où L renvoie la longueur de l'intervalle). Nous ajoutons cette contribution au support ;
- Nous notons que si $[\underline{s}_{i_3},\bar{s}_{i_3}]\subseteq S_{i_3}$(w) alors nous ajoutons 1 au support ;
- Si $[\underline{s}_{i_3},\bar{s}_{i_3}]\cap S_{i_3}$(w) = ∅ alors nous ajoutons 0 au support ;
- Dans tout les autres cas, nous ajoutons au support une valeur dans]0,1[.

Exemple : $Sup([70,100]\subset X) = \{1-\frac{L([90,100])}{L([70,100])}\} + \{1-\frac{L([90,100])}{L([70,100])}\} + \{1\} + \{1-\frac{L([80,100])}{L([70,100])}\}$ + {0} + {1} + {0} + {0} = 3.67. Nous notons que pour la première méthode, ce support serait égal à 0+0+1+0+0+1+0+0...=2.

Précédemment, nous avons montré la méthode de calcul pour des objets symboliques d'une seule propriété. Dans le cas où nous voulons calculer les contributions d'une conjonction de K propriétés K≥2 : Contribution(w,r), $r = \bigwedge_{i_1,u}\{\underline{x}_{i_1,u} < P_{X_{i_1,u}}(w) \leq \bar{x}_{i_1,u}\}\bigwedge_k\{[\underline{s}_{i_3},\bar{s}_{i_3}] \subseteq S_{i_3}(w)\}$), nous devons calculer la contribution de w à chaque propriété r_k, pour $r_k = \{\underline{x}_{i_1,u} < P_{X_{i_1,u}}(w) \leq \bar{x}_{i_1,u}\}$ ∀i_1,u et $r_k = \{[\underline{s}_{i_3},\bar{s}_{i_3}] \subseteq S_{i_3}\}$ ∀i_3. Nous proposons plusieurs possibilités pour le calcul de Contribution(w,r) :

- Contribution(w,r)=$Min_k\{contribution(w,r_k)\}$: le minimum des contributions ;
- Contribution(w,r)=$\prod_k contribution(w,r_k)$: le produit des contributions ;
- Contribution(w,r)=$Max\{\sum_k contribution(w,r_k) - (K-1), 0\}$: méthode de la "somme" des contributions.

Et nous pouvons alors calculer Sup(r)=$\sum_w$Contribution(w,r).

Exemple : Avec la méthode du produit, pour calculer le support de $r = \{0 < P_p \leq \frac{1}{3}\}\wedge\{\frac{1}{3} < P_v \leq \frac{2}{3}\}\wedge\{[70,100] \subset X\}$, nous calculons la liste des contributions de chaque individu à $r_1 = \{0 < P_p \leq \frac{1}{3}\}$, $r_2 = \{\frac{1}{3} < P_v \leq \frac{2}{3}\}$ et $r_3 = \{[70,100] \subset X\}$. Nous

obtenons les listes : contribution(w,r_1) = {1,1,1,0.67,0,0.93,0.733,0}, contribution(w,r_2) = {1,1,1,0,0,1,0,1}, contribution(w,r_3) = {0.67,0.67,1,0.33,0,1,0,0}. Pour obtenir la liste des contributions de r, nous faisons les produits des contributions pour chaque individu. Nous obtenons la liste : contribution(w,r) = {0.67,0.67,1,0,0,0.93,0,0}. Pour calculer le support de r, il ne nous reste plus qu'à sommer les contributions de r, soit $Sup(r) = 3.27$.

Les fonctions proposées sont des T-normes. Nous rappelons que les T-Normes T sont des fonctions de [0,1]×[0,1]→[0,1] avec les propriétés suivantes :

1. Commutativité : xTy = yTx ;
2. Associativité : (xTy)Tz = xT(yTz) ;
3. Non-décroissance par rapport aux arguments : si x≤y, w≤z, alors xTw≤yTz ;
4. 0 comme élément absorbant 0Tx = 0 ;
5. 1 comme élément neutre 1Tx = 1

Ces fonctions sont proposées car la propriété 3 permet de conserver la propriété (*) section 2.1 fondamentale pour l'application de l'algorithme Apriori.

Preuve : Supposons qu'il existe A⊂B tels que support(B)>support(A). Or, par construction, nous avons ∀w contribution(w,B)= (contribution(w,A) T contribution(w,B-A)) ≤ (contribution(w,A) T 1) = contribution(w,A).

Remarques et propriétés intéressantes :

- Les supports obtenus avec la méthode 2 sont supérieurs à ceux de la méthode 1 car ils exploitent mieux la structure symbolique des données. Nous obtiendrons donc plus de règles.
- Les supports ne sont plus forcément à valeurs entières avec la méthode 2.
- Si r est une conjonction de propriétés r_k et s'il existe r_k tel que contribution(w,r_k)=0 alors contribution(w,r)=0.
- La confiance se calcule comme pour la méthode 1 avec les nouveaux supports.

3.3 Définition du support minimum

Nous avons vu section 2.1 que nous voulons extraire les sous-ensembles fréquents, i.e. de support supérieur à un seuil *minsup*. Or, contrairement au cas classique, nous n'avons pas qu'une seule variable mais plusieurs variables. Nous proposons donc de définir un support minimum propre à chaque variable. Ainsi, si A et B sont deux variables de seuils minimaux minsup_A et minsup_B respectivement alors minsup_{AB} = minsup(A,B) sera égal à max(minsup_A,minsup_B). Cette méthode du maximum des seuils *minsup* permet de conserver la propriété (*) section 2.1.

Preuve : Supposons qu'il existe A⊂B tels que support(A) < minsup_A et support(B) ≥ minsup_B. Or support(A) ≥ support(B) ≥ minsup_B = max(minsup_A,minsup_{B-A}) ≥ minsup_A.

4 Algorithme Apriori symbolique (SApriori)

Nous allons dérouler l'algorithme à partir de la matrice table 2 avec huit concepts, une variable diagramme Y et une variable intervalle X. Cependant, la méthode se

généralise en présence de plusieurs variables diagrammes et intervalles. Pour le calcul du support, nous utiliserons la méthode 2 section 3.2 avec la méthode du produit des contributions. Un exemple avec la méthode 1 est donné dans (afonso 2004).

4.1 Principe de la méthode

Pour chaque variable diagramme, nous "discrétisons" les fréquences de chaque catégorie. Nous découpons en intervalles les fréquences P_{Xc} pour chaque catégorie c de chaque variable X. Ainsi, nous regardons les supports des intervalles de fréquences $0 < P_{Xc} \leq 1/h_x, 1/h_x < P_{Xc} \leq 2/h_x, 2/h_x < P_{Xc} \leq 3/h_x, ..., (h_x - 1)/h_x < P_{Xc} \leq 1$ où h_x détermine la précision du découpage de chaque variable X. Pour $h_Y = 3$, nous avons les intervalles 5 à 13 table 3 pour notre exemple et nous calculons leurs supports.

Dans un deuxième temps, nous faisons l'union 2 à 2 des intervalles de poids contigus ayant des supports strictement positifs $0 < P_{Xc} \leq 2/h_x, 1/h_x < P_{Xc} \leq 3/h_x, ..., (h_x - 2)/h_x < P_{Xc} \leq 1$. Nous ne conservons que les plus petits intervalles pour un même support. Ainsi, nous définissons les intervalles 5∪6, 6∪7, 8∪9, 9∪10, 11∪12, 12∪13 et nous conservons uniquement les intervalles 8∪9, 11∪12, 12∪13 car support(8∪9) > max {support(8), support(9)}, support(11∪12) > max {support(11), support(12)} et support(12∪13) > max {support(12), support(13)} (voir table 3 N° 14 à 16).

Nous répétons l'opération jusqu'à obtenir un unique intervalle $0 < P_{Xc} \leq 1$. Dans notre exemple, nous ne construisons plus que l'intervalle 15∪16, table 3 N° 17. Par la suite, tous les intervalles de fréquences de cette taxonomie respectant le support minimum seront considérés.

Pour chaque variable intervalle X, nous projetons les bornes des intervalles sur une droite. Soient $p_1 < p_2 < p_3 ... < p_k$ les points obtenus. Pour notre exemple table 2, si nous projetons les valeurs de la variable intervalle, nous obtenons les points $10 < 20 < 30 < 40 < 45 < 50 < 55 < 60 < 70 < 80 < 90 < 100$.

Nous fixons une longueur d'intervalle minimum l_X qui détermine à partir de quelle longueur un intervalle est intéressant pour l'utilisateur, i.e. la précision du découpage. Nous construisons alors le plus petit intervalle $[p_1, p_{i1}]$ avec $p_{i1} \in \{p_2, p_3, ..., p_k\}$ tel que $p_{i1} - p_1 \geq l_X$, puis l'intervalle $[p_{i1}, p_{i2}]$ avec $p_{i2} \in \{p_3, ..., p_k\}$ tel que $p_{i2} - p_{i1} \geq l_X$ ... jusqu'à l'intervalle $[p_{il}, p_k]$. Dans notre exemple, si nous posons $l_X = 30$, nous obtenons les intervalles [10,40], [40,70] et [70,100] table 3 N° 1 à 3.

Ensuite, nous faisons l'union deux à deux des intervalles contigus et nous conservons ceux respectant le support minimum. Nous considérons $[p_1, p_{i2}]$, $[p_{i1}, p_{i3}]$, $[p_{i2}, p_{i4}]$ ... puis $[p_1, p_{i3}]$, $[p_{i1}, p_{i4}]$ ... tant que nous obtenons des intervalles respectant le support minimum. Pour notre exemple, si nous fixons minsup=40% (i.e. 3.2), nous construisons l'intervalle $[40, 100] = [40, 70] \cup [70, 100]$ et nous ne construisons pas l'intervalle $[10, 70] = [10, 40] \cup [40, 70]$ car il ne respecte pas le support minimum. Par la suite, tous les intervalles fréquents de cette taxonomie seront considérés.

Ainsi, nous ne travaillons plus avec des itemsets mais avec des objets symboliques (OS) qui sont la conjonction des propriétés $\{\frac{\underline{x}}{h_x} < P_{Xc} \leq \frac{\bar{x}}{h_x}\}$ ($\underline{x} = 0..h_x - 1, \bar{x} = 1..h_x, \underline{x} < \bar{x}$) dans le cas diagramme et $\{[\underline{p_j}, \bar{p_j}] \subseteq X\}$ ($p_i, p_j \in \{p_1, p_2, ..., p_k\}$, $p_i < p_j$) dans le cas intervalle. Finalement, un k-OS est une assertion booléenne définie à partir de k propriétés. Par exemple, si P_v et P_p sont les fréquences des catégories v et p de la variable diagramme Y et X la variable intervalle table 2 alors a(w)=[$\{\frac{1}{3} < P_v(w) \leq$

N°	C_1	sup	N°	C_1	sup	N°	C_1	sup
1	[10,40]	1	7	$\frac{2}{3}<P_v\leq 1$	4.53	13	$\frac{2}{3}<P_c\leq 1$	4.22
2	[40,70]	4	8	$0<P_p\leq\frac{1}{3}$	5.33	14=8∪9	$0<P_p\leq\frac{2}{3}$	6
3	[70,100]	3.67	9	$\frac{1}{3}<P_p\leq\frac{2}{3}$	5.92	15=11∪12	$0<P_c\leq\frac{2}{3}$	5.67
4=2∪3	[40,100]	3.83	10	$\frac{2}{3}<P_p\leq 1$	4.58	16=12∪13	$\frac{1}{3}<P_c\leq 1$	5.75
5	$0<P_v\leq\frac{1}{3}$	3.8	11	$0<P_c\leq\frac{1}{3}$	5.2	17=15∪16	$0<P_c\leq 1$	6
6	$\frac{1}{3}<P_v\leq\frac{2}{3}$	5	12	$\frac{1}{3}<P_c\leq\frac{2}{3}$	5.42			

TAB. 3 – 1-OS candidats et 1-OS fréquents.

$\frac{2}{3}\}\wedge\{0 < P_p(w) \leq \frac{1}{3}\}\wedge\{[1,3] \subset X(w)\}]$ est un 3-OS. Ces k-OS ne seront pas totalement traités comme des items de l'Apriori classique. Nous ne croisons pas des intervalles de même catégorie dans le cas diagramme et de même variable dans le cas intervalle.

4.2 Définition de l'algorithme Apriori symbolique (SApriori)

Nous détaillons les différentes étapes de l'algorithme "Apriori symbolique" à l'aide de l'exemple table 2. Nous donnons les précisions h_Y=3 et l_X=30, un support minimum $minsup_Y = minsup_X = 40\%$ (i.e. 3.2 unités) :

L'ensemble C_1 des 1-OS (OS définis à partir d'une seule propriété) candidats est créé table 3, voir section 4.1. Les supports sont calculés à l'aide d'un passage dans les données et les 1-OS de 2 à 17 sont ajoutés au sous-ensemble L_1 des 1-OS fréquents. Le 1-OS candidat N°1 non fréquent est supprimé.

Tant que l'ensemble des k-OS (assertion définie avec la conjonction de k propriétés) fréquents $L_k \neq \emptyset$ $(k \geq 1)$:

1. Générer les k+1-OS candidats en calculant le produit cartésien entre les k-OS de L_k. Dans le cas des diagrammes, nous générons les k+1-OS entre intervalles de catégories différentes (et "non marqués" voir point 2) et dans le cas intervalle, nous ne croisons que les intervalles pour des variables différentes. Ainsi, l'ensemble des candidats C_{k+1} est généré. Du fait de la propriété (*) section 2.1, nous supprimons de C_{k+1} tout k+1-OS I tel qu'il existe un k-OS $J \subset I$ n'appartenant pas à L_k. C'est la première phase d'élagage. Enfin, pour tout $c \in C_{k+1}$, calculer le support avec un passage dans la matrice de données. Tout k+1-OS $I \in C_{k+1}$ fréquent est ajouté à L_{k+1}.

   Pour notre exemple, dans une première itération de la boucle, nous calculons d'abord le produit cartésien entre les OS de L_1. Ainsi, l'algorithme génère les candidats C_2 : (2∧5) à (2∧17) mais ne génère pas (2∧3) et (2∧4) car ces intervalles couvrent la même variable. De plus (5∧8) à (5∧17) sont candidats mais (5∧6) et (5∧7) ne le sont pas car 5, 6 et 7 sont des intervalles de la même fréquence. L'ensemble L_2 des 2-OS fréquents est donné table 4.

   Dans une 2^{eme} itération, SApriori génère les 3-OS candidats C_3 à partir des OS de L_2. L'ensemble L_3 des 3-OS fréquents est donné table 4 colonne L_3.
2. Marquer tout k+1-OS qui n'a pas les plus petits intervalles de fréquences pour un même support. Nous les marquons au lieu de les supprimer car ces k+1-OS

OS	L_2	Sup	OS	L_2	Sup	OS	L_2	Sup	OS	L_3	Sup
18	3∧6	3.33	27	7∧8	3.54	36	9∧16	3.67	43	3∧6∧8	3.27
19	3∧8	3.49	28	7∧9	3.53	37	9∧17	3.92	44	3∧6∧9	3.28
20	3∧9	3.61	29	7∧14	3.6	38	11∧14	3.93	45	3∧6∧14	3.33
21	3∧14	3.67	30	8∧11	3.35	39	12∧14	3.75			
22	4∧9	3.36	31	8∧15	3.4	40	14∧15	4			
23	4∧14	3.42	32	8∧17	3.4	41	14∧16	3.75			
24	6∧8	3.93	33	9∧11	3.85	42	14∧17	4			
25	6∧9	3.92	34	9∧12	3.67						
26	6∧14	4	35	9∧15	3.92						

TAB. 4 – 2-OS et 3-OS fréquents

ne sont pas utilisés pour la génération de k+2-OS mais ils sont utilisés pour la génération de règles. C'est la deuxième phase d'élagage.

Dans notre exemple, (8∧17) n'est pas utilisé pour la génération de 3-OS car 17 et 15 sont des intervalles de fréquences de même catégorie. Or, 17 est plus grand que 15 alors que sup(8∧15) = sup(8∧17). Il n'est donc pas utile de conserver 8∧17 pour la suite. Par contre, les règles 8→17 et 17→8 seront considérées. De même, nous marquons 9∧16, 9∧17, 14∧16, 14∧17 à cause de 9∧12, 9∧15, 12∧14, 14∧15.

3. Générer les règles avec une confiance supérieure à *minconf*. Nous ne traitons pas cette partie dans cet article. Le lecteur intéressé pourra se référer à (afonso 2004).

L'algorithme Apriori a une complexité de $m \times 2^n$ où m est le nombre de transactions et n le nombre d'items. La complexité de SApriori découle directement de ce résultat en remplaçant m par le nombre de concepts c (c<<m) et n par le nombre N d'intervalles discrétisés, i.e $h \times n$ pour une variable diagramme (N>n) et $|\frac{E}{l}| + 1$ pour une variable intervalle où $|\frac{E}{l}|$ désigne la partie entière de la division de l'étendue des données E sur la précision du découpage l. Nous ne rentrons pas plus dans les détails pour ce papier.

5 Applications

5.1 Règles d'association classiques versus symboliques

Nous comparons les règles générées à partir des itemsets fréquents issus de l'algorithme classique appliqué à la matrice table 1 et les règles générées à partir des objets symboliques issus de l'algorithme "symbolique" appliqué à la matrice table 2. Nous donnons les règles obtenues table 5 pour le cas classique avec $minsup = 25\%$ et $minconf = 50\%$ et un extrait des règles obtenues pour le cas symbolique table 6, avec $minsup_Y = minsup_X = 40\%$ (i.e. 3.2 clients) et $minconf = 70\%$ en utilisant la méthode 2 du calcul du support avec le produit des contributions section 3.2.

Dans les deux cas, nous voyons que l'achat de céréales implique l'achat de poisson. En effet, la méthode classique nous donne $c \rightarrow p$ avec une confiance de 60% mais la méthode symbolique nous fournit plus d'informations puisque nous avons en plus les

N°	Règle	Conf%	N°	Règle	Conf%
1	$c \rightarrow p$	60	3	$p \rightarrow c$	50
2	$p \rightarrow v$	58	4	$v \rightarrow p$	47

TAB. 5 – Règles classiques extraites à partir de l'algorithme Apriori standard

N°	Règle	Conf	N°	Règle	Conf
1	$[70,100] \subseteq X \wedge \frac{1}{3} < P_v \leq \frac{2}{3} \rightarrow \frac{1}{3} < P_p \leq \frac{2}{3}$	98%	6	$\frac{1}{3} < P_v \leq \frac{2}{3} \rightarrow \frac{1}{3} < P_p \leq \frac{2}{3}$	78%
2	$[70,100] \subseteq X \wedge \frac{1}{3} < P_v \leq \frac{2}{3} \rightarrow 0 < P_p \leq \frac{1}{3}$	98	7	$\frac{2}{3} < P_v \leq 1 \rightarrow 0 < P_p \leq \frac{1}{3}$	78
3	$[70,100] \subseteq X \wedge 0 < P_p \leq \frac{1}{3} \rightarrow \frac{1}{3} < P_v \leq \frac{2}{3}$	94	8	$\frac{2}{3} < P_v \leq 1 \rightarrow \frac{1}{3} < P_p \leq \frac{2}{3}$	78
4	$[70,100] \subseteq X \wedge \frac{1}{3} < P_v \leq \frac{2}{3} \rightarrow \frac{1}{3} < P_p \leq \frac{2}{3}$	91	9	$0 < P_c \leq \frac{1}{3} \rightarrow \frac{1}{3} < P_p \leq \frac{2}{3}$	74
5	$\frac{1}{3} < P_v \leq \frac{2}{3} \rightarrow 0 < P_p \leq \frac{1}{3}$	79	10	$0 < P_p \leq \frac{1}{3} \rightarrow \frac{1}{3} < P_v \leq \frac{2}{3}$	74

TAB. 6 – Extrait des règles découvertes par l'algorithme Apriori symbolique SApriori

fréquences $0 < P_c \leq \frac{1}{3} \rightarrow \frac{1}{3} < P_p \leq \frac{2}{3}$ (avec une confiance de 74%). Nous avons ainsi des règles plus nombreuses et plus précises grâce à nos intervalles de fréquences. Dans (afonso 2004), nous calculons même une relation linéaire entre les prémisses et la conclusion d'une règle afin d'étudier les variations à l'intérieur de la règle. Deuxièmement, avec l'étude classique, nous obtenons les règles $v \rightarrow p$ avec $conf(v \rightarrow p) = 47\%$ et $p \rightarrow v$ avec $conf(p \rightarrow v) = 58\%$. Par conséquent, la meilleure règle, selon la confiance, serait $p \rightarrow v$ alors que dans le cas symbolique nous obtenons "l'inverse". En effet, les règles $\frac{1}{3} < P_v \leq \frac{2}{3} \rightarrow 0 < P_p \leq \frac{1}{3}$ et $\frac{2}{3} < P_v \leq 1 \rightarrow 0 < P_p \leq \frac{1}{3}$ sont meilleures que la règle $0 < P_p \leq \frac{1}{3} \rightarrow \frac{1}{3} < P_v \leq \frac{2}{3}$ selon la confiance (79%, 78%, 74% resp.). Ainsi, nous voyons que si le "degré d'inclusion" de l'achat de poissons dans l'achat de viandes dans les transactions est grand, l'analyse symbolique nous montre qu'en fait ce sont plutôt les clients de viande qui sont aussi clients de poisson et non l'inverse. Ainsi, les clients de viande sont aussi clients de poisson bien qu'ils achètent plus de viande que de poisson. Si nous prenons un autre exemple, dans un tabac la vente de cigarettes est très importante et par conséquent le "degré d'inclusion" de l'achat de jeux à gratter dans l'achat de cigarettes est grand mais ce sont les clients de cigarettes qui pourront être amenés à acheter des jeux et non l'inverse comme l'aurait suggéré le cas classique.

De plus, nous remarquons que l'ajout de la variable intervalle X=montant améliore les règles obtenues puisque nous avons des confiances allant au maximum jusqu'à 79% sans cette variable alors que les confiances montent jusqu'à 98% avec cette variable. Par conséquent, l'ajout d'autres variables symboliques à la variable diagramme résumant les achats vient enrichir nos règles d'association en sémantique mais aussi en confiance.

5.2 Comparaison des différentes méthodes de calcul du support

Nous prenons l'exemple d'une base de données d'une société comptoir fournie avec le logiciel Access de Microsoft. Cette base répertorie 23705 enregistrements sur des clients ayant consommé parmi 77 produits (items). Nous avons également l'information sur les montants des commandes et sur le pays d'origine du client afin d'enrichir les règles.

Test	h=3, *minsup*=20%				h=6, *minsup*=15%			h=6, *minsup*=12%			
Méthode	1	min	×	+	1	min	×	1	min	×	+
Règles(70%)	3	90	67	59	33	934	578	320	4961	3023	2865
Règles(25%)	41	303	259	241	257	3588	2620	882	16444	11320	10433
Temps (s)	1	2	2	2	2	75	74	6	298	223	193

TAB. 7 – Nombres de règles obtenues et temps d'exécution selon h, $minsup$ et $minconf$

Nous construisons alors les concepts clients, soient 979 concepts. Après la création des concepts, chaque client est décrit par une variable diagramme résumant sa consommation, une variable intervalle montant et une variable classique pays. Nous appliquons SApriori aux 979 clients pour 3 paires (h, *minsup*) différentes (3,20%), (6,15%), (6,12%) et pour des seuils de confiance égaux à 25% et 70%. Nous donnons table 7, le nombre de règles d'associations obtenues et les temps d'exécution pour chaque méthode de calcul du support vue section 3, i.e. la méthode 1 et les méthodes 2 du minimum, notée min, du produit, notée × et de la "somme", notée + des contributions.

Nous remarquons alors que la méthode 1 est bien plus rapide que la méthode 2. Notamment avec h=6 et *minsup*=12%, le temps d'exécution de la méthode 1 n'est que de 6 secondes alors que la méthode min a mis 298 secondes. En contrepartie, nous obtenons beaucoup plus de règles avec la méthode 2. Pour h=6, *minsup*=12% et $minconf$=25%, nous n'avons que 882 règles d'association pour la méthode 1 alors que nous en avons 16444 et 11320 pour les méthodes du minimum et du produit des contributions respectivement. Ainsi, si nous regardons le rapport entre le nombre de règles et le temps d'exécution, nous constatons évidemment que la méthode 1 est plus rapide mais que la différence est tout de même moins importante. De plus, nous avons beaucoup de très bonnes règles d'association (confiance>70%) avec la méthode 2 que nous n'avons pas trouvé avec la méthode 1. Pour h=6 et *minsup*=12%, nous n'avons que 320 règles avec la méthode 1 alors que nous en avons 4961 avec la méthode min. Enfin, la méthode du min donne plus de règles que la méthode du produit et de la "somme". En effet, si x,y∈[0,1] alors min(x,y)>x.y>max(x+y-1,0).

6 Conclusions et perspectives

Nous avons étendu l'algorithme Apriori aux cas des variables symboliques diagrammes et intervalles dans le but d'extraire des règles d'association à partir d'une matrice de concepts. Nous avons pris comme exemple des clients de magasins quelconque où nous trouvons des règles d'association entre les articles achetés au niveau des clients et non plus au niveau des transactions. Nous avons constaté que nous découvrions des informations supplémentaires par rapport aux règles classiques. Aussi, l'ajout d'autres variables symboliques comme des variables intervalles viennent enrichir nos règles. Dans (afonso 2004), une étude sur la qualité de ses règles par la régression linéaire est proposée dans le cas diagramme. Il serait alors intéressant d'étendre cette étude en présence de variables intervalles et de comparer plus précisément les règles

d'association obtenues ainsi que les performances de l'algorithme avec les différentes méthodes de calcul du support présentées dans ce papier.

Références

Afonso F. (2004), Extension de l'algorithme Apriori et des règles d'association au cas des données symboliques diagrammes et sélection des meilleures règles par la régression linéaire symbolique, RNTI, 2004.

Agrawal R. et Srikant R. (1994), Fast algorithms for mining association rules, Proc. of the 20th Int'l Conf. on Very Large Databases, 1994.

Agrawal R.,Imielinski T. et Swami A. (1993), Mining association rules between sets of items in large databases, ACM SIGMOD Records, 1993.

Bock H-H. et Diday E. (2000), Analysis of Symbolic Data, Exploratory methods for extracting statistical information from complex data, Springer Verlag, 2000.

Cai C.H., Fu A.W.C., Cheng C.H. et Kwong W.W. (1998), Mining association rules with weighted items, Proc. of the 1998 Int'l Database Engineering and Applications Symposium (IDEAS'98), pp 68-77.

Han J. et Fu Y. (1995) Discovery of multiple-level association rules from large databases, Proc. of the 21^{th} Int'l Conf. on Very Large Data Bases, 1995.

Kuok C.M., Fu A. et Wong M.H. (1998), Mining fuzzy association rules in databases, ACM SIGMOD Record, Vol. 27, pp 41-46, 1998.

Miller R.J. et Yang Y. (1997) Association rules over interval data, Proc. of the 1997 ACM SIGMOD int'l conf. on Management of data, pp 452-461, 1997.

Srikant R., Vu Q. et Agrawal R. (1997), Mining association rules with item constraints, Proc. of the 3^{rd} Int'l Conf. on Knowledge Discovery in Databases and Data Mining, 1997.

Srikant R. et Agrawal R. (1996) Mining quantitative association rules in large relational tables, Proc. of the ACM-SIGMOD 1996 Conf. on Management of Data.

Wang W., Yang J. et Yu P. (2000), Efficient mining of weighted association rules (WAR), Proc. of the sixth ACM SIGKDD int'l conf. on Knowledge discovery and data mining, pp 270-274, 2000.

Summary

This paper deals with the extension of the Apriori algorithm and of the association rules to the symbolic histogram and interval-valued data. We suggest a method that will enable us to discover rules at the level of the concepts. This extension requires new definitions for the support and the confidence in order to take advantage of the symbolic structure of the data. The market basket data example is developed throughout the paper. Thus, instead of mining rules between different items of some transactions recorded in a retail organization like in the classical case, we discover rules at the level of the customers in order to study their purchase behavior.