CHAPITRE 2

LES REGLES D'ASSOCIATION

2.1 INTRODUCTION :

Dans le domaine du Data Mining la recherche des règles d'association est une méthode populaire étudiée d'une manière approfondie dont le but est de découvrir des relations ayant un intérêt pour le statisticien entre deux ou plusieurs variables stockées dans de très importantes bases de données. Piatetsky-Shapiro présente des règles d'association extrêmement fortes découvertes dans des bases de données en utilisant différentes mesures d'intérêt.

En se basant sur le concept de relations fortes, Rakesh Agrawal et son équipe présente des règles d'association dont le but est de découvrir des similitudes entre des produits dans des données saisies sur une grande échelle dans les systèmes informatiques des points de ventes des chaînes de supermarchés. Par exemple, une règle découverte dans les données de ventes dans un supermarché pourrait indiquer qu'un client achetant des oignons et des pommes de terre simultanément, serait susceptible d'acheter un hamburger. Une telle information peut être utilisée comme base pour prendre des décisions marketing telles que par exemple des promotions ou des emplacements bien choisis pour les produits associés.

En plus des exemples ci-dessus concernant le panier de la ménagère, les règles d'association sont employées aujourd'hui dans plusieurs domaines incluant celui de la fouille du web, de la détection d'intrusion et de la bio-informatique[4].

2.2 Cadre informel :

Le concept d'extraction des règles d'association introduit en 1993 par Agrawal et al. est une méthode qui a vu le jour avec la recherche en bases de données. Agrawal et al. ont été amenés à travailler sur une application de vente de produits dans les supermarchés. Cette application est également appelée "Analyse du panier de la ménagère" et elle est à l'origine des règles d'association. Il s'agit d'obtenir des relations ou des corrélations du type "Si Condition alors Résultat". Pour ce problème, chaque panier n'est significatif que pour un client en fonction de ses besoins et de ses envies, mais si le supermarché s'intéresse à tous les paniers simultanément, des informations utiles peuvent être induites et exploitées. Tous les clients sont différents et achètent des produits différents, en quantités différentes, l'analyse

Dupanier de la ménagère étudie qui sont les clients et pourquoi ils effectuent tel ou tel type d'achat. Cette analyse permet d'étudier quels produits tendent à être achetés en même temps.

Par exemple, on sera capable de dire que 75% des clients qui achètent du lait achètent en même temps des oeufs (lait → oeuf : 0.75), une telle constatation est très intéressante puisqu'elle aide le gestionnaire d'un supermarché à ranger ses rayons de telle sorte que le lait et les œufs soient à proximité.

Dans les bases de données de vente, un tuple consiste en une transaction regroupant l'ensemble des articles achetés appelés items. Ainsi une base de données est un ensemble de transactions, qu'on appelle aussi base transactionnelle ou base de transactions.

Une règle d'association décrit une corrélation entre des ensembles d'items dans une base de transactions. Autrement dit, étant donné un ensemble d'items, le but est de découvrir si l'occurrence de cet ensemble dans une transaction est associée à l'occurrence d'un autre ensemble d'items. Par exemple, "80% des clients qui achètent un ordinateur achètent aussi une imprimante et un abonnement à Internet" est une règle d'association associant l'item ordinateur aux items imprimante et abonnement à Internet [5].

2.3 Cadre formel :

Formellement, le problème de règle d'association est présenté de la façon suivante :

Soit $I = \{i_0, i_1, ..., i_n\}$ un ensemble de n items,

Soit $T = \{t_1, ..., t_m\}$ un ensemble de m transactions telleque chaque transaction t_i soit dotée d'un identifiant unique (noté TID) appelé transaction identificateur, et chaque

Transaction t_i est un sous ensemble d'items $t_i \subseteq I$ [5].

Définition (Itemset):

Un itemset (noté I_0) est un sous ensemble d'items autrement dit c'est un élément de l'ensemble des parties de I ($I_0 \in 2^I$).

Exemple d'itemset :

L'itemset I_0 = {banane, tomate, fromage} est composée de trois items : banane, tomate, fromage [5].

Définition (k-itemset):

Un k-itemset est un sous-ensemble d'items I_0 ($I_0 \subseteq I$) telque : $I_0 = k$.

Exemple d'un k-itemset : L'itemset I_0 = {banane, tomate, fromage} est un 3-itemset [5].

Définition (Support d'un itemset):

Le support d'un itemsetI_k (noté supp(I_k ,T)) est la probabilité qu'une transaction t_i continent cet itemset I_k .

$$Supp(I_k,T)=|\{t_i \in T / I_k \subseteq t_i\}|/|T|$$

Où | | : désigne la cardinalité.

Le support prend sa valeur dans l'intervalle [0,1]. Il est souvent exprimé en pourcentage [5].

Définition (Itemset fréquent):

Etant donné un seuil γ, appelé support minimum (noté minsupp), un itemset I_k est dit fréquent (relativement à γ) dans une base de transactions T, si son support dépasse le seuil γ.

I_k est fréquent si seulement si : $supp(I_k, T) \geq \gamma$. [5]

Définition (Règle d'association):

On appele règle d'association, une relation de la forme $I_1 \Rightarrow I_2$ entre deux itemsets I_1, I_2 telsque :

- $I_1, I_2 \subseteq I$,
- $I_1 \cap I_2 = \phi$,
- $I_1, I_2 \neq \phi$.

La partie gauche de la règle I_1 est appelée la prémisse de la règle et la partie droite I_2 est appelée la conclusion de la règle.

Définition (Support d'une règle d'association):

Le support d'une règle d'association $I_1 \Rightarrow I_2$ (note $Supp(I_1 \Rightarrow I_2,T)$ est le pourcentage des transactions de T qui contiennent $I_1 \cap I_2$

$Supp(I_1 \Rightarrow I_2, T) = |t_i \in T / I_1 \cap I_2 \subseteq t_i| / |T|$

$Supp(I_1 \Rightarrow I_2,T) = Supp(I_1 \cap I_2)$. [5]

Définition (Confiance) :

La confiance d'une règle d'association $I_1 \Rightarrow I_2$, notée ($conf(I_1 \Rightarrow I_2,T)$) représente la proportion de transactions couvrant I_1 et qui couvrent aussi I_2.

$$Conf(I_1 \Rightarrow I_2,T) = (|t_i \in T / I_1 \cap I_2 \subseteq t_i|) / (|t_i \in T / I_1 \subseteq t_i|)$$

$$Conf(I_1 \Rightarrow I_2,T) = Supp(I_1 \cap I_2) / Supp(I_1). [5]$$

Définition (Ensemble de règles d'association valides)

Soit un ensemble F d'itemsets fréquents pour un seuil minimal de support minsupp.

Etant donné un seuil minimal de confiance minconf, l'ensemble AR des règles d'association Valides est :

$$AR = \{R : l_2 \rightarrow (l_1 - l_2) / l_1, l_2 \in F, l_2 \subset l_1, confiance(R) > minconf\}$$

Pour chaque itemset fréquent l_1 dans F, touts les sous-ensembles l_2 de l_1 sont déterminés et la valeur de la confiance de la règle (R) est calculée. Si cette valeur estsupérieure ou égale au seuil minimal de confiance alors la règle d'association $l_2 \rightarrow (l_1 - l_2)$ est générée. [5]

2.4 La découverte des règles d'association :

Le problème de découverte des règles d'association peut être scindée en deux sous problèmes :

- ❖ Le premier sous problème consiste à déterminer l'ensemble des itemsetsfréquents dans la base de transaction T.
- ❖ Le deuxième sous problème consiste à générer les règles d'association à partir de l'ensemble des itemesets fréquents [5].

2.4.1 Extraction des itemsets fréquents :

Cette phase constitue la première partie de problème de recherche de règles d'association, il existe trois grandes approches algorithmiques pour la recherche d'itemsets fréquents pour la génération des règles d'association.

- ❖ Approche d'extraction d'itemsets fréquents,
- ❖ Approche d'extraction d'itemsets fréquents maximaux,
- ❖ Approche d'extraction d'itemsets fermés fréquents.

Dans le cadrede cette approche Agrawal et al. ont proposé dans, le premier algorithme d'extraction des règles d'association dans les bases de données transactionnelles, il s'agit de l'algorithme **Apriori** que nous décrivons ci-après. [5]

Principe de base de l'algorithme Apriori :

L'idée générale de cet algorithme est de générer à chaque itération k, un ensemble d'itemsets potentiels et de le tester (calculer le support). Un balayage est réalisé pour éliminer les itemsets non fréquents. Les k-itemsets fréquents obtenus sont réutilisés lors de l'itération k+1 pour générer les itemsets candidats de taille k+1. Afin de limiter, le nombre d'itemsets fréquents lors de chaque itération, cet algorithme se base sur le principe d'anti-monotonicité de la fréquence.[5]

Propriété 1 (propriété sur les sous-ensembles):

Tous les sous-ensembles d'un itemset fréquent sont fréquents.

Cette propriété permet de limiter le nombre des candidats de taille k générés lors de lak^{ime} itération en réalisant une jointure conditionnelle des itemsets fréquents de taille k-1découverts lors de l'itération précédents.[5]

Propriété 2 (propriété sur les sur-ensembles):

Tous les sur-ensembles d'un itemset non fréquent sont non fréquents.Cette propriété permet de supprimer un candidat de taille k lorsque au moins un de sessous-ensembles de taille k-1 ne fait par partie des itemsets fréquents découverts lors del'itération précédente [5].

Présentation de l'algorithme Apriori :

Durant la première itération de l'algorithme (ligne1), tous les itemsets de taille 1 sontconsidérés et un balayage de la base de transaction T est réalisé afin de déterminer l'ensemble des 1-itemsets fréquents F_1. Les k itérations suivantes se subdivisent en deux étapes :

- Durant la première étape, l'ensemble C_k des k-itemsets candidats est construit en joignant les (k-1)-itemsets fréquents, c'est-à-dire $C_k=F_{k-1}_F_{k-1}$. Cette étape est réalisée dans la procédure Apriori-Gen (F_{k-1}).
- Durant la seconde étape, un balayage de la base de données est réalisé afin dedéterminer le support de chacun des k-itemsets candidats dans C_k et les k-itemsets fréquents sont insérés dans l'ensemble F_k. Lors de ce balayage, pour chaque transaction t de T, l'ensemble C_t des k-itemsets candidats qui sont contenus dans t est déterminé et le support de chacun de ces itemsets est recalculée [5].

La détermination de la fréquence des k-itemsets candidats est réalisée par la fonction Subset(C_k,t). Cette fonction reçoit un ensemble C_k de k-itemsets candidats et une transaction t comme paramètres. Elle retourne un ensemble C_k de k-itemsets candidats de Ck qui sont contenu dans t. Ces itérations s'arrêtent lorsque aucun nouvel itemsets candidat ne peut être généré, c'est-à-dire $F_{k-1} =\varphi$.

La procédure exist_subset_infrequent(c, F_{k-1}) renvoie une booléen afin d'indiquer s'ilexiste un sous-ensemble de c qui n'est pas fréquent.

La procédure Apriori-Gen(F_{k-1}), reçoit un ensemble de F_{k-1} de (k-1)-itemsets fréquents comme paramètre. Elle retourne un ensemble C_k de k-itemsets candidats qui est unsur-ensemble de l'ensemble des k-itemsets fréquents.

Deux (k-1)-itemsets fréquents f_1 et f_2 de F_{k-1} sont joints si et seulement si les k-2premiers items qui les composent sont identiques. La jointure de deux (k-1)-itemsetsrésulteen un k-itemset candidat.

En utilisant la procédure exist_subset_infrequent (1, F_{k-1}) , si le candidat c (k-itemsets) dont l'un des sous-ensembles s de taille k-1 ne se trouve pas dans F_{k-1}, alors c est supprimé deC_k, sinon il est ajouté à C_k. [5]

Entrée	**Base de transactions T ; Support minimum [5]**
Sortie	Ensemble des itemsets fréquents F
01	F1 ←{1− itemsets fréquents}
02	**Pour** (k = 2; $F_{k-1}\neq 0$; k + +)**faire**
03	C ← Apriori-gen(F_{k-1}) ;
04	**pour chaque (**transaction t ∈T **) faire**

05	C_t←subset(C_k ,t) ;
06	**pour chaque** (candidat $_t$ c∈C)**faire**
07	c.support++ ;
08	**fin pour**
09	**fin pour**
10	F_k {c∈C_k\ c.support ≥ misupp}
11	**fin pour**
12	Retourner F_k

Fonction Apriori-Gen(F_{k-1}) [5]:

01	pour **chaque itemset f_1∈ F_{k-1}**faire
02	**pour** chaque itemset f_2∈F_k**faire**
03	**si** ($f_1[1] = f_2[1]$)∨($f_1[2] = f_2[2]$)∨...∨($f_1[k-2] = f_2[k-2]$)∨ ($f_1[k-1]<f_2[k-1]$)**alors**
04	c = $f_1[1]$, ..., $f_1[k-1]$, $f_2[k-1]$ //étape de jointure : générer des candidats
05	**si** exist_subset_infréquent( ) c, f_{k-1}**alors**
06	supprimer c de C_k
07	**sinon**
08	ajouter c à C_k
09	**fin si**
10	**fin si**
11	**fin pour**
12	**fin pour**
13	retourner C_k

Fonction exist_subset_infrequent(c, f_{k-1}) [5]

01	pour chaque **(k −1)-subsets de c** faire
02	**si** s ∈∄$_{k-1}$**alors**
03	retourner TRUE ;
04	**fin si**
05	retourner FALSE ;
06	**fin pour**

Exemple : application de l'algorithme Apriori

Soit une base de transaction T contenant huit items et soit un seuil minimal de supportminsupp égal à 3/8. Le tableau (II.1) présente la base de transactions.

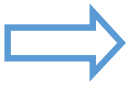
TID	Tomate	Banane	Lait	fromage	Pâte	Riz	Programme tv	Pile
T_1	1	0	0	1	1	0	0	0
T_2	0	0	1	0	0	1	0	0

T_3	0	0	0	0	0	1	1	1
T_4	1	0	0	0	1	0	0	0
T_5	1	0	0	0	1	0	0	0
T_6	0	1	1	0	0	1	0	1
T_7	1	0	0	1	1	0	0	0
T_8	1	0	0	1	1	0	0	0

Tableau2.1 : La base de transaction T [5]

Itemset	Support
Tomate	5/8
Banane	1/8
Lait	2/8
Fromage	3/8
Pâte	5/8
Riz	3/8
Programme tv	1/8
Pile	2/8

Génération de 1-itemsets fréquents



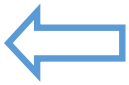
Itemset	Support
Tomate	5/8
Fromage	3/8
Pâte	5/8
Riz	3/8

Génération de 2-itemsets candidats



Itemset
Tomate & Fromage
Tomate & Pâte
Tomate & Riz
Fromage & Pâte
Fromage & Riz
Pâte & Riz

Calcul des supports



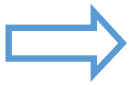
Itemset	Support
Tomate & Fromage	3/8
Tomate & Pâte	5/8
Tomate & Riz	0/8
Fromage & Pâte	3/8
Fromage & Riz	0/8
Pâte & Riz	0/8

Génération de 2-itemsets fréquents



Itemset	Support
Tomate & Fromage	3/8
Tomate & Pâte	5/8
Fromage & Pâte	3/8

Génération de 3-itemsets candidats



Itemset
Tomate & Fromage & pâte

Génération de 3-itemsets fréquents



Itemset	Support
Tomate & Fromage & pâte	3/8

Tableau2.2 : Recherche des itemsets fréquents. [5]

2.5 Génération des règles d'association :

La génération des règles d'association constitue la seconde étape de processus de recherche des règles d'association, elle s'effectue à partir des itemsets fréquents générés précédemment. Agrawal et al. ont proposé un algorithme de génération de règles efficaces.

Le principe général de la génération de l'ensemble des règles d'association est le suivant :

Pour chaque itemset fréquent l_1 dans F de taille supérieure ou égale à deux, tous les sous-ensembles $l_2$sont déterminés et la valeur du rapport support (l_1)/support (l_2) est calculée.

Si ce rapport est supérieur ou égal au seuil de confiance minconffixé par l'utilisateur, la règle d'association $l_2 \rightarrow (l_1 - l_2)$ est générée. L'algorithme proposé dans se base sur la propriété (1) concernant les supports des itemsets pour réduire le nombre d'opérations réalisées par la génération [5].

Etant donné trois itemsets fréquents l_1, l_2 et $l_3$tels que $l_1 \supset l_2 \supset l_3$, il est possible de déduire de la propriété (1) que support (l_3) ≥ support (l_2) ≥ support (l_1). En conséquence, la confiance de la règle R_2 : $l_3 \rightarrow (l_1 - l_3)$ est inférieure ou égale à la confiance de la règle R_1:$l_2 \rightarrow (l_1 - l_2)$. Si la règle R_1 n'est pas valide alors la règle R_2 ne le sera pas non plus.

Cela signifie par exemple, que si la règle d'association AC →DE n'est pas valide, par conséquent les règles A→CDE et C → ADE ne seront pas valides non plus et il n'est pas nécessaire de calculer leur confiance. Cette constatation permet de diminuer le nombre de règles d'association testées par l'algorithme. Réciproquement, la confiance de la règle R_3 :$(l_1 - l_2) \rightarrow l_2$ est supérieure ou égale à la confiance de la règle R_4 : $(l_1 - l_3) \rightarrow l_3$. Si la règle R4 est valide alors la règle R3 le sera également.

Cela signifie que si la règle d'association A→ BC est valide, alors les règles AB →C et AC → B le seront également [5].

Algorithme de génération des règles d'association :

F est un ensemble d'itemsets fréquents dans lequel chaque élément de cet ensemble possède deux champs qui sont l'itemset lui-même et son support. H_m représente l'ensemble des m-itemsets qui sont les conséquences de règles valides générées à partir de l'itemsetl_k.

L'algorithme considère successivement chaque itemset fréquent de F de taille supérieure à 1.

Pour chacun de ces itemsetsl_k, l'ensemble H_1 des itemsets de taille 1 qui sont dessous-ensembles de l_k est généré et pour chaque élément h_1 de H_1, la règle $(l_k - h_1) \rightarrow h_1$estgénérée si sa confiance est supérieure ou égale à minconf. Sinon, si cette règle n'est pasvalide, alors le 1-itemset h_1 est supprimé de H_1. Lorsque tous les 1-itemsets de H_1 ont ététestés, H_1 contient la liste des 1-itemsets qui sont conséquences des règles valides générées àpartir de l_k. Les règles

valides générées à partir de l_k sont les règles dont l'union del'antécédent et de la conséquence donne l'itemsetl_k. La procédure Gen-Rules est alorsappelée afin d'insérer dans AR les règles valides générées à partir de l_k dont la conséquencecontient plus d'un item. L'algorithme se termine lorsque tous les k-itemsets fréquents pourk ≥ 2 ont été considérés. L'ensemble AR renvoyé par l'algorithme contient alors toutes lesrègles d'association valides pour le seuil minimal de confiance minconf générées à partir del'ensemble F [5].

Entrée	**un ensemble F des itemsets fréquents, un seuil minimal de confiance minconf. [5]**
Sortie	Un ensemble AR des règles d'association valides
01	**pour chaque** (k-itemsets fréquents l k ∈F tel que k ≥ 2) faire
02	H1 ←1-itemsets étant des sous ensemble de k l ;
03	**pour chaque** (h_1∈H_1) **faire**
04	confiance (r) ← support (lk)/support (l_k – h_1) ;
05	**si** confiance (r) ≥ minconf**alors**
06	AR ← AR ∪ {r : l_k– h_1→ h_1 }**;**
07	**sinon**
08	H_1← H_1\ **h**$_1$
09	**Fin si**
10	**fin pour**
11	Gen-Rules (lk, H1)
12	**fin pour**
13	retourner AR

Algorithme d'insertion des règles valides :

La procédure Gen-Rules reçoit un k-itemset fréquent l_k, un ensemble H_m qui contient les m-itemsets qui sont les conséquences de règles valides générées à partir de l_k et du seuil minimal de confiance minconf. Elle met à jour l'ensemble AR de règles d'association en y insérant les règles valides générées à partir de l_k dont la conséquence est un (m+1)-itemset.

Cette procédure est récursive et réalise en fin d'exécution un appel afin de générer à partir de l_k les règles valides dont la conséquence est un (m+2)-itemsets. Ces appels ce répète récursivement jusqu'à ce que les règles dont la conséquence est un ($|l_k|$-1)-itemset aient été insérées dans AR [5].

Le premier test correspond au test d'arrêt des appels récursifs de la procédure. Ces appels cessent lorsque l'ensemble H_m reçu comme paramètre contient des itemsetsde taille m=$|l_k|$-1. Dans ce cas, toute les règles valides générées à partir de l_k, ont été insérées dans AR.

Ensuite, l'ensemble H_{m+1} des (m+1)-itemsets qui peuvent être des conséquences des règles valides générées à partir de l_k est crée. Cette création est réalisée en appliquant la procédure

Apriori-Gen, présentée précédemment, à l'ensemble H_m des m-itemsets qui sont les conséquences de règles valides générées à partir de l_k. Chaque règle dont la conséquence est un (m+1)-itemset de Hm+1 est alors testée. Si la règle testée est valide, elle est insérée dans AR [5].

Sinon, le (m+1)-itemset qui est la conséquence est supprimé de H_{m+1}. Cette suppression correspond à la diminution du nombre de règles testées basée sur la propriété 1.

En effet, si la règle d'association AC → DE n'est pas valide, l'itemset DE est supprimé de H_2. Lors de l'appel récursif suivant (avec h_2 en paramètre), les itemsets CDE et ADE ne seront pas crées par Apriori-Gen dans h_3 car DE est un sous-ensemble de CDE et ADE. Les règles A→CDE et C → ADE ne seront donc pas testées.

L'appel récursif de Gen-Rules est réalisé en fin de procédure avec comme paramètre l'itemset l_k et l'ensemble H_{m+1}.

Entrée	• k-itemsets fréquent l_k • ensemble $H_m$dem-itemsets conséquences de règles valides générées à partir de l-k. • un seuil minimal de confiance minconf.
Sortie	un ensemble AR de règles d'association valides augmenté des règles valides générées à partir de $l_k$dont la conséquence est un (m+1)-itemset
01	**si** k >m +1 **alors**
02	H_{m+1}← Apriori-Gen (H_m) ;
03	**pour chaque** h_{m+1} □ H_{m+1}**faire**
04	confiance (r) ← support (l_k) / support (l_k- h_{m+1}) ;
05	**si** confiance (r) ≥ minconf**alors**
06	AR ←ARU {r : (l_k− h_{m+1}) → h_{m+1} };
07	**sinon**
08	supprimer h_{m+1} de H_{m+1} ;
09	**fin si**
10	**fin pour**
11	Gen-Rules(l_k,H_{m+1}) ;
12	**fin si**

Exemple : génération des règles d'association

L'exemple précèdent induit les règles d'association suivants lorsque le seuil de minimum de confiance égal 1/2.

Itemset	Confiance
Tomate → Fromage	3/5
Tomate → Pâte	5/5
Fromage → Pâte	3/3
Tomate → Fromage & Pâte	3/5
Fromage → tomate & Pâte	3/3
Pâte → Tomate & Fromage	3/5

Tableau2.3 : Exemple des règles d'association avec minconf=1/2. [5]

2.6 Les amélioration de l'algorithme Apriori :

L'algorithme Apriori est un algorithme très utilisé afin d'extraire des itemsets fréquents. Ses performances diminuent en présence de données denses ou fortement corrélées.

L'idée de base d'Apriori est de parcourir l'espace de recherche en "largeur d'abord", pour ne retenir que les itemsets fréquents et de générer d'autres éléments dans le niveau suivant par auto jointure. Les supports des itemsets candidats sont calculés et les candidats non fréquents sont supprimés. Cette suppression est basée essentiellement sur la propriété d'anti-monotonicité [5].

Cependant, cette approche souffre de la gestion du nombre de candidats qu'elle pourrait générer, surtout pour des contextes fortement corrélés et/ou des valeurs de support relativement faibles. Quelques algorithmes d'extraction d'itemsets fréquents basés sur l'algorithme Apriori ont été proposés dans le but d'améliorer son efficacité. Nous citons notamment :

- L'algorithme AprioriTid :

Apriori Tid cherche à garder le contexte en mémoire afin de limiter les accès à la base.

A supposer (ce qui est en général le cas sur des données réelles) que les ensembles de candidats décroissent en même temps que la taille des candidats, ce qui permet de diminuer la taille de la base de données progressivement. Dans ce cas, cet algorithme s'avère bien plus efficace qu'Apriori. Une approche mixte de l'algorithme Apriori et Apriori Tida été proposée, s'agit d'une hybridation des deux algorithmes [5].

- L'algorithme Sampling :

Cet algorithme extrait un échantillon de la base qui tient en mémoire. A partir de cetéchantillon, l'ensemble des itemsets fréquents dans l'échantillon, est construit ainsi que sa

bordure négative constituée des itemsets non fréquents minimaux dont toutes les parties sont fréquentes, ce qui limite le risque de non exhaustivité [5].

- L'algorithme Partition :

Cet algorithme partitionne la base de données en plusieurs sous ensembles pour l'extraction d'itemsets fréquents, lorsque les données ne tiennent pas en mémoire par exemple.

- L'algorithme Count Distribution :

Cet algorithme est une version parallélisée de l'algorithme Apriori. Chaque processus traite sa portion locale de la base de transactions. Il minimise le traitement d'une base de données de grande taille impose de coûteuse opérations d'entrée/sortie [5].

2.7 Réduction de l'ensemble de règles d'association :

Le nombre d'itemset fréquents extraits et leur taille moyenne étant élevés dans la plupart des cas, le nombre de règles d'association générées est encore beaucoup trop élevé. Ce nombre important de règles d'association extraites constitue un problème majeur pour la pertinence et l'utilité du résultat, problème qui est accentué par la présence de nombreuses règles redondantes et triviales. La réduction de l'ensemble de règles d'association générées afin d'améliorer la pertinence du résultat a fait l'objet de plusieurs travaux de recherche. Les approches proposées pour résoudre ce problème peuvent se classer en deux catégories : les approches orientées données, et les approches orientées utilisateur, Ces approches seront explicitées dans les sous sections suivantes [5].

2.7.1 Approche orientée données :

Les approches orientées données se basent sur les propriétés structurelles des règles d'association afin de réduire l'ensemble des règles d'association extraites. Nous distinguons quatre catégories parmi ces approches :

- Approche qui utilise une taxonomie, ou hiérarchie de classe des items (règles d'association généralisées)
- Approche qui utilise des mesures statistiques autres que la confiance pour déterminer la précision des relations entre itemsets.
- Approche qui utilise des mesure de déviation des règles d'association autres que le support et la confiance.
- Approche qui consiste à supprimer les règles d'association redondantes de l'ensemble des règles d'association valides extraites [5].

A) Règles d'associations généralisées :

Les règles d'association généralisées, également appelées règles d'association multi niveaux, sont définies en utilisant une taxonomie des items du contexte d'extraction. Cette taxonomie est un **DAG** (Direct Acyclic Graph) dont les sommets sont les items et les arcs sont des relations is-aentre deux items. S'il existe un arc d'un item i vers un item i' dans la taxonomie alors i est appelé père de i' et i' est appelé fils de i. S'il existe un arc d'un item i vers un item i' dans la fermeture transitive de la taxonomie, i est appelé ancêtre de i' et i'est appelé descendant de i. Ce qui signifie que l'item i est une généralisation de l'item i' et que i' est une spécialisation de i. un item n'est pas un ancêtre de lui-même car le graphe est acyclique.

Soit :

T : Base de transaction,

I : Ensemble des items contenus dans les transactions,

R : Relation liant O et I,

τ : Une taxonomie

Les items i (i ∈ I) sont tels que descendant (i) =φ dans la taxonomieτ . Les items de la taxonomie qui possèdent des descendants sont des concepts de généralisation des items du contexte [5].

Exemple :

N°	**Transaction**
1	Baskets, Caleçon
2	T-shirt, Pantalon
3	Baskets, Pantalon, Caleçon
4	T-shirt, Pantalon
5	Pantalon, Caleçon
6	Baskets, T-shirt

Tableau2.4 : Base de transactions

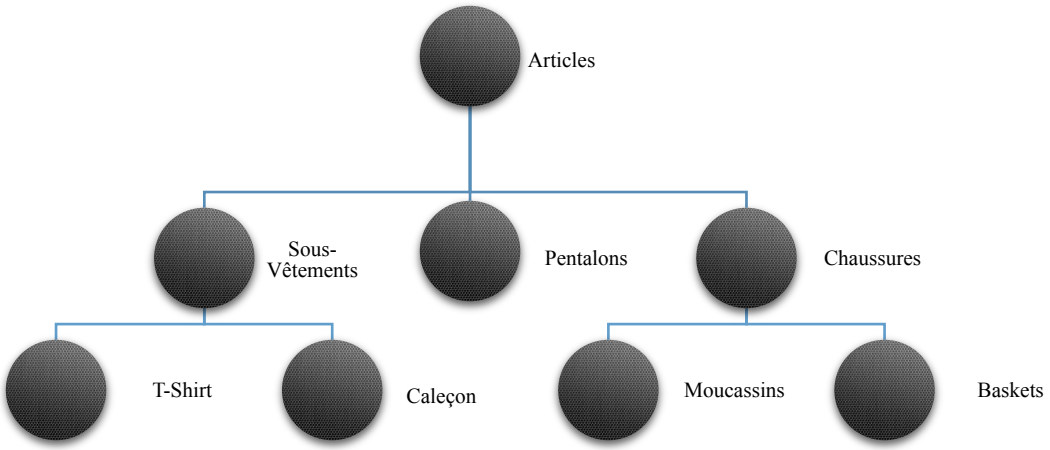


Figure 2.1 : *Taxonomie τdes items de T*

1-Itemset généralisé	Support
{T-shirt}	3/6
{Caleçon}	3/6
{Pantalon}	4/6
{Baskets}	3/6
{Sous-vêtements}	6/6
{Chaussures}	3/6

2-Itemset généralisé	Support
{Caleçon, pantalon}	2/6
{sous-vêtement, pantalon}	4/6
{Caleçon, Baskets}	2/6
{Caleçon, chaussures}	2/6
{sous-vêtements, baskets}	3/6
{sous-vêtements, chaussures}	3/6
{T-shirt, Pantalon}	2/6

Tableau2.5 : Itemsets généralisés fréquents pour minsupp =2/6

Règlesd'associationgénéralisées	Support	Confiance
T-shirt →Pantalon	2/6	2/3
Caleçon→Pantalon	2/6	2/3
Sous-vêtement→Pantalon	4/6	4/6
Pantalon→Sous-vêtement	4/6	4/4
Caleçon→Baskets	2/6	2/3
Baskets →Caleçon	2/6	2/3
Caleçon→chaussures	2/6	2/3
Chaussures→Caleçon	2/6	2/3
Baskets →Sous-vêtement	3/6	3/3
Chaussures→Sous-vêtement	3/6	3/3

Tableau2.6 : Règles d'association généralisées valides pour Minsupp=2/6 et Minconf=2/3

Les règles d'association généralisées sont des règles d'association entre ensembles d'items pouvant appartenir à différents niveaux de la taxonomie. Elles sont de la forme :

R: $l_1 \rightarrow l_2$ Avec $l_1 \cap l_2 = \emptyset$ Telles que aucun item de l_2n'est ancêtre d'un item de l_1(pour éviter les règles triviales du genre I → ancêtre(i)) [5].

Les règles d'association généralisées valides sont celle dont le support et la confiance sont supérieurs ou égaux aux seuils minimaux minsuppet minconfrespectivement. Les itemsets fréquents, à partir desquels les règles d'association généralisées sont générées, sont constituées des items de la taxonomie τ [5].

L'ensemble des itemsets généralisés fréquents est un sur ensemble de l'ensemble des itemsets fréquents. L'ensemble des règles d'association généralisées est donc un sur ensemble de l'ensemble des règles d'association valides.

Toutefois, il est possible de supprimer certaines règles d'association généralisées valides lorsque celles-ci sont résumées par une règle plus générale. Une règle d'association généralisée R' :A'→ C est plus générale qu'une règle R : A → C si les items de A' sont des ancêtres des items de A. La règle R est résumée par la règle R' s'il est possible de réduire R, son support et sa confiance à partir de la règle R'etdes supports des itemsetsA et A'. Dans ce cas la règle R est redondante par rapport à la règle R' [5].

Trois algorithmes d'extraction des itemsets généralisés fréquents nommés Basic,

CumulateetEstMergeont été proposés. Pour l'algorithme Basic, les objets du contexte d'extraction sont remplacés par des objets "étendus" contenant tous les items de l'objet original ainsi que les items ancêtres de ces items. A partir de ces objets étendus, une version modifiée de l'algorithme Apriori extrait les itemsets généralisés fréquents. Dans l'algorithme Cumulate, une liste des items ancêtre de chaque item de la taxonomie est construite et les items les moins généraux sont remplacés au cours de processus par leur item père dans les itemsets déterminés infréquents après un balayage du contexte. Dans l'algorithme EstMerge, les itemsets généralisés fréquents constitués des items les plus généraux et substitue successivement chaque item dans ces derniers par ses items fils jusqu'à ce que l'itemset produit soit infréquent [5].

B) Mesures de déviation :

Les mesures de déviation sont des mesures de distance entre règles d'association définies en fonction de leur support et confiance. Elles sont utilisées de deux manières distinctes afin de supprimer certaines règles de l'ensemble des règles d'association découvertes.

- La première consiste à identifier les règles d'association fortement semblables, caractérisées par une faible distance, et classer ou supprimer certaines de ces règles en fonction des mesures de déviation qui leurs sont associées.
- La seconde permet d'identifier les règles d'association qui sont inattendues pour l'utilisateur et apportent donc une connaissance importante car nouvelle. Les connaissances de l'utilisateur sont représentées en utilisant des modèles probalistiques auxquels sont confrontés les règles d'association extraites. La déviation d'une règle correspond à la différence entre la valeur attendue pour la règle dans le modèle probabiliste et la valeur réelles pour la règle dans le contexte. Nous pouvons citer :

  - Une mesure de déviation basée sur la confiance des règles d'association a été proposée dans. Cette mesure est calculée pour chaque règle d'association R parmi un ensemble **R** de règles extraites. On note moyenne conf(R) la valeur moyenne et déviationconf**(R)** la déviation standard des confiances des règles de l'ensemble **R.** La mesure de distance associée à une règle d'association R : $l_1 \rightarrow l_2$ dans l'ensemble **R.**

    Distance $_{std}$(R, R)=(confiance (R)-moyenneconf(R))-déviationconf(R)

    L'approche proposée dans, consiste à fractionner l'ensemble AR des règles d'association valides extraites en plusieurs sous-ensembles et supprimer dans chacun de ces sous-ensembles **R** les règles R dont la distancestd(R, **R)** est inférieure à un seuil minimal défini par l'utilisateur.

  - Une mesure de déviation spécifiant une distance entre deux règles d'association

possèdant la même conséquence a été proposée dans. La distance entre deux règles est définie en fonction des ensembles d'objets qui vérifient chacune des deux règles. Soit O(I) le nombre d'objet du contexte d'extraction contenant l'itemsetl.

Pour deux régles d'association R:A → C et R': A'→ C la distance entre R et R' est :

$$\mathbf{distance}_{\mathbf{sem}}(R,R') = O(A \cup C) + O(A' \cup C) - 2 \times O(A \cup A' \cup C)$$

La valeur obtenue correspond au nombre d'objets du contexte qui vérifient l'une des deux règles mais pas l'autre. L'auteur a utilisé cette mesure afin de faciliter la visualisation de l'ensemble des règles d'association extraites en formant des groupes de règles qui concernent les mêmes contextes. Les groupes sont construits en minimisant la distance entre les règles de chaque groupe [5].

C) Couverture structurelle :

L'ensemble des règles d'association valides AR est extrait et une couverture structurelle pour les règles d'association est construite en supprimant de cet ensemble les règles redondantes d'un point de vue syntaxique. Cette approche ne tient pas compte de la confiance des règles, ce qui entraîne une perte d'information importante dans de nombreux cas. De plus, la construction de la couverture nécessite pour chaque règle la liste des objets contenant l'itemset union de l'antécédent et de la conséquence de la règle. La couverture structurelle Δ d'un ensemble AR de règles d'association est définie par :

$$\Delta = \{A \rightarrow C \in AR \setminus \exists A' \rightarrow C \in AR \text{ telle que } A' \subset A\}$$

Un algorithme de génération de couvertures structurelles appelé RuleCoverest présenté dans. Afin de générer la couverture structurelle, cet algorithme requiert que pour chaque règle d'association de AR soit associée la liste des objets vérifiant la règle, c'est-à-dire la liste des objets contenant l'itemset union de l'antécédent et de la conséquence de la règle [5].

Définition d'une règle d'association redondante :

Soit AR, l'ensemble de règles d'association générées, Une règle (R_1 : X→ Y ∈AR / conf= 0) est considérée comme redondante par rapport à une autre règlesR: X → Y / conf= c si R_1 vérifie les deux conditions suivantes :

1. Supp (R_1)=Supp(R_2) et conf(R_1)=conf(R_2)=c
2. (X⊂XetY⊂Y) où (X=XetY⊂Y).

2.7.2 Approche orientée utilisateur :

Les approches orientées utilisateur pour la réduction de l'ensemble des règles d'association requièrent l'intervention de l'utilisateur afin de définir des critères de sélection des règles qui figureront dans l'ensemble résultat. Nous distinguons trois catégories différentes parmi ces

approches :

- L'utilisation d'expressions régulières appelées templates utilisées afin de filtrer l'ensemble de règles valides extraites précédemment.
- L'utilisation d'un opérateur appelé MINE RULE qui est une extension du langage SQL.
- L'utilisation de critères de sélection appelés contrainte sur les items pour extraire seulement les itemsets fréquents permettant de générer les règles vérifiant ces contraintes.

A. Templates :

Les templates sont définis comme des expressions régulières spécifiant des critères généraux de sélection d'un sous-ensemble est construit à partir de l'ensemble des règles d'association valides, en conservant les règles qui vérifient les critères spécifiés par les templates parmi cet ensemble. Ces critères stipulent quels items doivent apparaître dans l'antécédent et la conséquence des règles du sous-ensemble. Les templates sont des expressions de la forme: $X_1 \cap ... \cap X_j \rightarrow X_{j+1}$. Dans laquelle chaque Xhest un item i∈I [5].

B. OpérateurMinRule :

Les critères de sélection des règles sont définis par un opérateur nommé MIN RULE. Cet opérateur est une extension du langage de requête SQL implémenté dans les SGBD relationnels. L'extraction des règles d'association est réalisée à partir des tuples des relations de la base de données par une requête qui est une instance de l'opérateur MIN RULE.

Les itemsets fréquents sont des ensembles de valeurs d'attributs de tuples extraits des tables (ou requêtes) qui sont les sources de données de la requête. Les règles d'association sont générées à partir de ces itemsets fréquents selon les critères définis par la requête [5].

C. Contraintes sur les items :

Dans l'approche par contraintes sur les items, l'utilisateur définit des contraintes, spécifiant les règles d'association à extraire, qui sont des expressions portant sur l'antécédent, la conséquence ou l'antécédent et la conséquence simultanément des règles.

Ces contraintes sont utilisées lors de la phase d'extraction des itemsets fréquents afin de limiter l'espace de recherche de cette phase. Elles sont prises en compte lors de la génération des itemsets candidats afin de considérer seulement les candidats permettant de générer des règles satisfaisant les contraintes [5].

2.8. Domaine d'applications

Le fait d'identifier des relations significatives entre les données contenues dans une base peut s'avérer utile à différentes domaines commerciaux, scientifiques est industriels dans leurs but de rentabiliser leur profit.

Plusieurs travaux basés sur la recherche des règles d'association ont été appliqués dans des applications réelles comme :

- La planification commercial.
- Les réseaux de telecommunication.
- La recherchemédicale.
- Les données de web.
- Le marketing bancaire.
- Le multimedia.
- Lepanier de ménagère.

2.9 Conclusion:

Dans Ce chapitre, Nous venons présenter les concepts fondaentaux relatives aux règles d'association, et comment faire pour générer ces règles.

Le prchain chapitre, on va détailler quelques algorithmes de génération des règles d'association selon deux approches : approche classique et l'approche d'expansion des règles.