

ACADÉMIE DE MONTPELLIER
UNIVERSITÉ MONTPELLIER II
— SCIENCES ET TECHNIQUES DU LANGUEDOC —

MÉMOIRE DE STAGE DE MASTER

SPÉCIALITÉ : **Recherche en Informatique**
Mention : **Informatique, Mathématiques, Statistiques**

effectué au laboratoire LIRMM/INFO

—
sous la direction de ANNE LAURENT, MARC PLANTEVIT, MAGUELONNE TEISSEIRE

**Extraction de connaissances atypiques dans les bases de données
multidimensionnelles**

par

Delphine Jouve

Soutenu le 2 juillet 2007

Table des matières

Remerciements	3
Introduction	6
I Etat de l'art	8
1 Les bases de données multidimensionnelles	9
1.1 Du modèle relationnel au modèle multidimensionnel	9
1.2 Une représentation sous forme de cube	10
2 Extraction de connaissances communes dans le contexte multidimensionnel	12
2.1 Les règles d'association multidimensionnelles	12
2.1.1 Les règles d'association intra-dimensionnelles	13
2.1.2 Les règles d'association inter-dimensionnelles	13
2.2 Les motifs séquentiels multidimensionnels	14
2.2.1 Extraction de séquences intra-dimensionnelles	15
2.2.2 Extraction de séquences intra et inter-dimensionnelles	15
3 Extraction de connaissances atypiques	17
3.1 Utilisation des règles d'association classiques	17
3.1.1 Les règles exceptionnelles	17
3.1.2 Les règles anormales	18
3.2 Les règles intéressantes	19
4 Discussion et objectifs	21
II Proposition	23
1 Extraction de connaissances communes	26
1.1 Motifs séquentiels multidimensionnels avec traitement de la mesure	27
1.1.1 La mesure dans la dimension d'analyse après prétraitement	27
1.1.2 La mesure utilisée pour calculer le support	28
1.2 Les règles séquentielles multidimensionnelles	30
1.2.1 Définitions	30
1.2.2 Algorithme d'extraction de règles séquentielles	31
1.2.3 Exemple	32

2	Extraction de connaissances atypiques	34
2.1	Extraction de règles exceptionnelles	34
2.2	Collage des séquences	36
2.2.1	Définition	37
2.2.2	Algorithme RechercheSeqCol	39
2.2.3	Exemple	39
2.3	Recherche de conséquences différentes	41
2.3.1	Définition	42
2.3.2	Algorithme RechercherConseqExc	43
2.3.3	Exemple	44
2.4	Algorithmes généraux	45
III	Expérimentations	47
3	Implémentation	48
3.1	Démarche générale	48
3.2	Partitionnement	49
3.3	Extraction des motifs séquentiels	49
3.4	Génération des règles séquentielles	50
3.5	Collage des séquences et recherche des conséquences différentes	50
4	Expérimentation sur jeu de données réelles	51
4.1	Jeux de données	51
4.2	Prétraitement	51
4.3	Comparaison des résultats	52
4.3.1	Comparaison des techniques de partitionnement	52
4.3.2	Comparaison avec ratio, sans ratio	53
	Conclusion et perspectives	54
A	Connaissances communes	58
A.1	La mesure dans la dimension d'analyse	58
A.1.1	Cas général	58
A.1.2	Motifs séquentiels étoilés	60
B	Table des notations	61

Remerciements

Je souhaite remercier mes encadrants, Marc Plantevit, Anne Laurent et Maguelonne Teisseire pour leur aide et leur accompagnement tout au long de ce stage.

Je tiens également à remercier Céline Fiot pour m'avoir remotivée dans les moments difficiles.

Je remercie certains de mes camarades de la promotions Master2 2007, pour leur convivialité, les discussions enrichissantes que nous avons pu avoir, ainsi que l'aide de certains d'entre eux.

Introduction

Ces dernières années, l'augmentation des capacités de stockage a provoqué une explosion de la masse d'informations. Ainsi, les différents organismes conservent toutes ces informations afin d'analyser et de comprendre leur évolution. Or, la quantité de données est telle que certains organismes ont recours aux *entrepôts de données* permettant le stockage des données sous leur forme agrégée. De telles structures représentent alors non plus des individus mais des groupes d'individus, désignés sous le nom de population. Il s'agit ensuite pour les décideurs de se baser sur les mouvements de ces différentes populations afin d'analyser les comportements et de prendre les décisions associées.

Afin d'assister le décideur, différents algorithmes d'extraction de connaissances ont fait l'objet de nombreux travaux de recherche aboutissant à la création d'algorithmes efficaces. Il s'agit d'extraire des schémas récurrents, valides et utiles à partir de grandes sources de données. Ces méthodes font partie d'un processus plus général désigné sous le nom d'*Extraction de Connaissances* (ECD). La fouille de données regroupe l'ensemble des algorithmes d'extraction de connaissances *communes*.

Les connaissances extraites peuvent cependant être très nombreuses et les informations ont souvent peu d'intérêt. Par exemple, dans un cadre commercial, une grande surface peut extraire des connaissances du type : "*Beaucoup de clients, à chacune de leur venue, achètent du pain*". Ce type de connaissances cache des informations moins singulières : l'occurrence d'un événement fréquent apporte moins d'informations qu'un événement plus rare. Obtenir des connaissances intéressantes devient alors une problématique importante. Nous nous intéressons donc à des comportements dits "*atypiques*" qui permettent d'extraire des connaissances peu fréquentes mais très significatives. La découverte de telles connaissances aurait un rôle important dans de nombreux domaines : sécurité des réseaux, détection d'intrusions, ou encore détection des fraudes aux cartes de crédits.

Les comportements atypiques sont abordés dans la littérature sous plusieurs formes. Tout d'abord, les comportements **anormaux sur des faits** (des *objets* ou des *tuples*) sont souvent nommés "*outliers*" ou "*particularités*" ([BS03], [KNT00], [SCA06]). Un outlier est défini par Hawkins comme "*une observation qui dévie tellement des autres observations qu'on la suspecte d'avoir été générée par un mécanisme différent*". Par exemple, dans le tableau suivant, le troisième tuple est très éloigné des autres puisque les valeurs pour les attributs B et C sont très supérieures aux autres.

A	B	C
1	10	2%
2	15	2.2%
3	120	40%
4	12	2%

Exemple : détection d'outlier

Les comportements peuvent être **anormaux en fonction de connaissances**, les connaissances atypiques sont appelées "*exceptions*", "*anomalies*" ou "*motifs surprenants*". Par exemple une connais-

sance détectée est " *un symptôme A est souvent associé à une maladie M1 pourtant le même symptôme pour une femme est plus souvent associé à la maladies M2*". En prenant en compte une sous population, la règle n'a plus la même finalité.

Nos recherches se situent dans un cadre de détection d'exceptions car nous souhaitons rechercher des comportements atypiques en fonction de connaissances communes. Deux approches sont alors possibles : une approche basé sur les connaissances d'un expert ([AAR96], [Sah99], [LHCM00]) qui est subjective ou une approche basé sur les données elles-mêmes qui est objective ([Suz99], [BCMG04]).

Nous abordons le problème du point de vue des données car si l'on utilise les connaissances d'un expert, moins d'exceptions seront extraites. L'expert n'a qu'une vision partielle d'une base de données très dense. Il se base sur une logique selon ses propres connaissances, les comportements exceptionnels extraits sont alors très spécifiques. Il est plus intéressant d'extraire toutes les connaissances exceptionnelles. Nous utilisons donc des connaissances communes directement extraites de la base de données avec des techniques d'extraction automatique. Une fois les connaissances communes détectées, l'extraction de connaissances atypiques sera effectuée.

Dans une première partie, un aperçu des méthodes existantes sera dressé ainsi qu'une présentation du contexte nous permettant de mieux comprendre la problématique. Dans une seconde partie notre proposition sera exposée.

Première partie

Etat de l'art

1	Les bases de données multidimensionnelles	9
1.1	Du modèle relationnel au modèle multidimensionnel	9
1.2	Une représentation sous forme de cube	10
2	Extraction de connaissances communes dans le contexte multidimensionnel	12
2.1	Les règles d'association multidimensionnelles	12
2.1.1	Les règles d'association intra-dimensionnelles	13
2.1.2	Les règles d'association inter-dimensionnelles	13
2.2	Les motifs séquentiels multidimensionnels	14
2.2.1	Extraction de séquences intra-dimensionnelles	15
2.2.2	Extraction de séquences intra et inter-dimensionnelles	15
3	Extraction de connaissances atypiques	17
3.1	Utilisation des règles d'association classiques	17
3.1.1	Les règles exceptionnelles	17
3.1.2	Les règles anormales	18
3.2	Les règles intéressantes	19
4	Discussion et objectifs	21

Chapitre 1

Les bases de données multidimensionnelles

Dans ce chapitre, nous expliquons la notion de base de données multidimensionnelles ainsi que le vocabulaire associé. Nous mettons en avant les différences entre le modèle relationnel et le modèle multidimensionnel afin de donner un bref aperçu des diverses problématiques associées à de tels modèles.

1.1 Du modèle relationnel au modèle multidimensionnel

Les évolutions des capacités de stockage permettent de nos jours d'enregistrer une vaste quantité de données, et sont actuellement le principal outil de décision des entreprises. Devant la masse d'informations à traiter, il est inutile de considérer les individus un à un, comme avec le modèle relationnel. Il est alors nécessaire de proposer des modèles considérant des groupes d'individus, afin de permettre à l'expert de mesurer rapidement leurs mouvements et de prendre les décisions associées. De tels modèles sont désignés sous le nom d'**entrepôts de données** [Inm90].

Il y a deux façons d'entreposer les données : soit sous leur *forme élémentaire*, ce qui prendra une place importante, soit sous une *forme agrégée* selon des axes ou dimensions d'analyse prévus. L'ajout de données sous forme agrégée est un processus non trivial qui passe par les phases d'extraction des données, de transformation et d'analyse. Il s'agit, dans un premier temps, d'extraire les données et de les rassembler. Ce processus est réalisé à l'aide d'outils d'ETL (*Extract, Transform, Load*), qui permet de mélanger des données provenant de plusieurs sources en résolvant les éventuels conflits. Durant la phase de transformation, les données sont épurées des valeurs manquantes, des duplications et des incohérences, puis elles sont agrégées, donc rendues homogènes. Après cette étape, le niveau détaillé des données est perdu. Enfin, les données sont prêtes pour le processus d'analyse. Pour cela, il faut extraire les données appartenant au champ d'analyse, les agréger dans un **magasin de données** et les présenter à l'utilisateur.

Les **outils OLAP** [CCS93] (*On-Line Analytical Processing*) facilitent l'analyse : ils stockent et restituent des données sous forme multidimensionnelle et représentent une alternative intéressante aux systèmes relationnels classiques, qui sont eux destinés aux traitements individuels des données sans optique d'analyse et de décision via des traitements **OLTP**. Dans les bases OLAP, ce n'est plus l'information individuelle qui est traitée mais plutôt des comportements de groupes. Les principales différences entre OLTP et OLAP sont résumées par le tableau 1.1.

	OLAP	OLTP
Type de requêtes	complexes	simples
But	Optique décisionnelle	Production et mise à jour des données
Niveau de vision	Vision ensembliste	Vision individuelle
Utilisateurs	Peu nombreux	nombreux

TAB. 1.1 – Comparaison de OLAP et OLTP

1.2 Une représentation sous forme de cube

Les bases de données multidimensionnelles sont représentables sous forme de **cubes** ou d'**hyper cubes**, principalement dans le but de permettre l'interaction avec les utilisateurs. Nous expliquons ici le vocabulaire lié aux hypercubes, qui peuvent être représentés de la manière suivante : $D_1 \times \dots \times D_k \rightarrow D_M$ avec $\{D_1, \dots, D_k\} \subseteq D$ et $D_M \subseteq M$, où D est l'ensemble des dimensions et M sont les mesures (données numériques). Par exemple, le cube de la figure 1.1 est défini par Produit \times Ville \times Mois \rightarrow nombre de ventes.

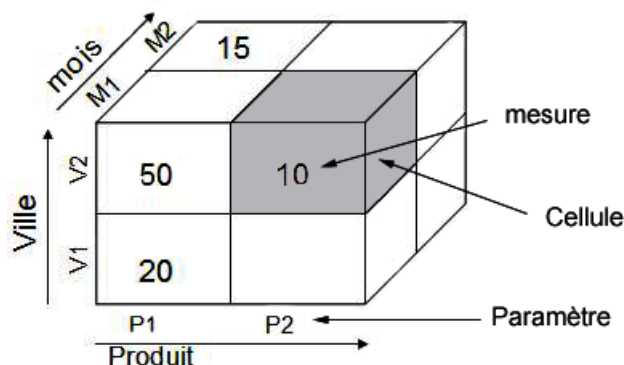


FIG. 1.1 – Exemple de cube

Voici une description des différents éléments d'un cube :

- Une *cellule* est un élément du cube. Par exemple, dans le cube de la figure 1.1, une cellule est une case du cube.
- Les *mesures* sont les valeurs (1 ou +) contenues dans une cellule. Pour le cube de la figure 1.1, il n'y a qu'une seule mesure dans chaque cellule, par exemple la valeur est 50.
- Un *paramètre* ou un *membre* est une graduation d'une dimension. Dans l'exemple 1.1, P1 est un paramètre de la dimension produit.
- Une *référence* est une graduation pour chaque dimension représentant la position de la cellule. La référence, dans le cube 1.1 est par exemple (P1, V1, M1).

Les mesures ont une signification importante au sein de ces cubes. En effet, après l'étape d'agrégation, elles représentent le nombre de transactions qui ont été effectuées car celles-ci n'apparaissent plus dans les bases de données multidimensionnelles. Il est alors primordial que ces données reflètent pour le mieux le mouvement des différentes populations.

Par exemple, dans la figure 1.2 une base relationnelle est représentée à gauche, et son agrégation en base de données multidimensionnelles à droite. On remarque que pour la référence (Marseille, 1, A) la mesure 2 est indiquée. Cette mesure représente le nombre de fois que la référence (Marseille, 1, A) est présente dans la base de données relationnelle. Avec l'agrégation des données, l'individu n'est plus représenté : lorsque nous nous référons à (Marseille, date 1, A) nous ne savons pas quels

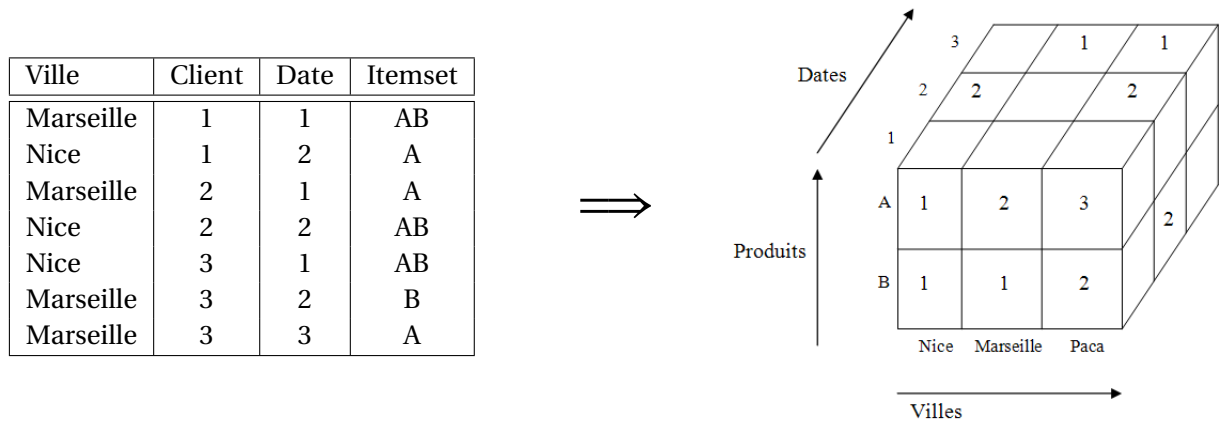


FIG. 1.2 – Agrégation de données

individus sont représentés, ni les informations les concernant.

De plus, dans les bases de données multidimensionnelles, il est possible de représenter une hiérarchie au sein d'une dimension et de calculer la mesure associée. Par exemple, dans la figure 1.2 un niveau de hiérarchie est présent car les villes Marseille et Nice sont incluses dans la région "PACA".

Chapitre 2

Extraction de connaissances communes dans le contexte multidimensionnel

L'extraction de connaissances est un processus non trivial visant à extraire des schémas nouveaux à partir de grandes sources de données. Celui-ci se divise en plusieurs étapes : (1) sélection, (2) préparation (3) fouille de données (4) interprétation. Les deux premiers visent à sélectionner les sources de données que l'on souhaite fouiller, supprimer les données au format invalide et les réécrire au format nécessaire à la fouille souhaitée.

La fouille de données désigne l'ensemble des algorithmes qui ont pour but d'analyser les données. Parmi ceux-ci, nous retrouvons les méthodes d'extraction de schémas récurrents. Nous nous intéressons plus particulièrement à ces dernières méthodes, car elles permettent d'extraire des *connaissances*. Dans ce chapitre, nous décrivons en détails les règles d'associations et les motifs séquentiels appliquées au contexte multidimensionnel.

2.1 Les règles d'association multidimensionnelles

Introduit par [AIS93], la recherche de règles d'association a fait l'objet de nombreux travaux ces dernières années aboutissant sur la création d'algorithmes efficaces. Un exemple de cette problématique couramment cité est celui du panier de la ménagère où le schéma suivant est défini : les paniers représentent les transactions, les items représentent les produits achetés. La découverte de règles d'association consiste à chercher des ensembles d'items, fréquemment liés dans une même transaction, ainsi que des règles les combinant. Les règles d'association permettent alors d'extraire des règles telles que "*une personne qui achète du beurre achète en même temps du lait*" ou "*Dans 80% des cas, une personne obtient un crédit quand elle gagne plus de 3000 euro/mois et travaille dans la même entreprise depuis plus de deux ans*".

Une *transaction* constitue, pour un objet O , l'ensemble des *items* achetés par O à une même date. Dans une base de données client, une *transaction* s'écrit sous la forme (id-client, id-date, ensemble d'items). Il s'agit d'extraire les règles concernant un certain nombre d'objets de la base, respectant donc un *seuil de fréquence minimale* fixé par l'utilisateur. Une règle d'association est alors de la forme $R : X \rightarrow Y (A\%, B\%)$ avec X et Y des ensembles d'items. $A\%$ représente la fréquence d'apparition de la règle, c'est le pourcentage de transactions dans lesquelles l'ensemble $X \cup Y$ est inclus et $B\%$ représente la *confiance* de la règle, c'est-à-dire la proportion d'objets, qui ayant X dans une transaction, ont aussi Y .

Les différents travaux utilisant les règles d'association appliquent les algorithmes proposés par

[AS94]. A partir d'un support minimal (*minsupp*) et d'une confiance minimale (*minconf*) fixés par l'utilisateur, l'algorithme se déroule en deux étapes : rechercher les itemsets fréquents (ceux qui ont une fréquence supérieure au *minsupp*) et extraire les règles à partir des itemsets fréquents (dont la confiance est supérieure à *minconf*). Les algorithmes de [AS94] permettent donc l'extraction efficace de connaissances communes, c'est-à-dire de connaissances concernant un grand nombre d'objets de la base.

Dans une base de données relationnelles, la détection de règles s'applique en fonction de transactions. Cette notion de transaction n'apparaît pas dans les données multidimensionnelles. Il est donc nécessaire d'étendre les règles "traditionnelles" au contexte multidimensionnel en redéfinissant les transactions et les attributs.

[Zhu95] propose l'extraction de deux types de règles en fonction d'une ou plusieurs dimensions : les règles intra-dimensionnelles et les règles inter-dimensionnelles, que nous présentons dans la suite de cette section.

2.1.1 Les règles d'association intra-dimensionnelles

Le premier type de règles considérées par [Zhu95] sont les **règles intra-dimensionnelles** qui sont *définies entre les valeurs d'une même dimension*. Les auteurs proposent de choisir une dimension qui représentera les transactions (chaque valeur de la dimension choisie représentant une transaction différente), et une dimension à analyser en fonction de ces transactions. Par exemple, si l'on choisit la dimension "Ville" du cube 2.1 comme représentant les transactions, on aura trois transactions possibles : "Marseille", "Montpellier" et "Nîmes". La dimension "Produit" sera alors analysée en fonction des différentes villes, et les règles seront extraites en fonction de ces valeurs.

		Produits		
		DVD	CD	iPod
Villes	Marseille	10	5	
	Montpellier		30	10
	Nîmes	1		

TAB. 2.1 – Exemple de cube de deux dimensions Villes et Produits

Dans le cube 2.1, l'item DVD a un support de $\frac{2}{3}$ car il apparaît pour Marseille et Nîmes (les cellules ne sont pas vides) mais pas Montpellier. Le support de la règle $CD \rightarrow iPod$ est de $\frac{1}{3}$ (uniquement la transaction Montpellier) et une confiance de $\frac{1}{2}$ (quand CD apparaît, une fois sur deux iPod apparaît).

Cependant, ce type de règle ne permet pas de prendre en compte plus d'une dimension pour l'analyse, c'est pourquoi [Zhu95] propose un second type de règles.

2.1.2 Les règles d'association inter-dimensionnelles

Les règles inter-dimensionnelles permettent d'extraire des règles *définies entre les valeurs de différentes dimensions*.

Dans les travaux [Zhu95], un *item* peut être une valeur provenant de n'importe quelle dimension. Un *itemset* est un ensemble d'items qui ne peut pas comporter deux valeurs d'une même dimension. Par exemple dans la base du tableau 2.2, les items Livre et CD ne peuvent pas apparaître dans le même itemset car ils font partie de la même dimension.

Le support d'une règle sera calculé grâce à la mesure dont la référence est composée des valeurs présentes dans l'itemset. Si, pour une dimension, une valeur n'est pas spécifiée alors elle sera remplacée par le total (somme des mesures pour les différentes valeurs de la dimension). Par exemple, pour le cube 2.2, le support de l'itemset {Marseille, Livres} est égal à la mesure de la cellule dont la référence est (Marseille, Livres, total) soit un support de $\frac{64}{696}$.

		Produits				
		DVD	Livres	CD	Lecteur	total
Paris	Faible	12	23	43		78
	Moyen		45	10	28	83
	Fort	30		5	56	91
	total	42	68	58	84	252
Montpellier	Faible		58	34	12	104
	Moyen	12		5	15	32
	Fort		9		45	54
	total	12	67	39	72	190
Marseille	Faible	2	30		23	55
	Moyen	45		60	28	133
	Fort		34	24	8	66
	total	47	64	84	59	254
Total	Faible	14	111	77	35	237
	Moyen	57	45	75	71	248
	Fort	30	43	29	109	211
	total	101	199	181	215	696

TAB. 2.2 – Exemple 3 dimensions

[MRBM06] propose une autre façon de trouver des séquences inter-dimensionnelles. Dans un premier temps, les dimensions sont classées selon deux ensembles : les *dimensions de contexte* et les *dimensions d'analyse*.

La règle est définie de la manière suivante : dans un contexte $(\Theta_1, \dots, \Theta_k)$, $(x_1 \wedge \dots \wedge x_s) \rightarrow (y_1 \wedge \dots \wedge y_s)$. Un *itemset* est alors un ensemble des valeurs des dimensions d'analyse qui ne peut pas comporter deux items d'une même dimension. Par exemple, pour le tableau 2.2, si le contexte est composé des villes "Montpellier", "Marseille" et les dimensions d'analyse sont représentées par les produits et le profit alors une règle possible est : dans le contexte $(\{\text{Montpellier}, \text{Marseille}\})$, CD \rightarrow Moyen. Cette règle signifie que, pour Montpellier et Marseille, la vente de CD est associée à un profit moyen. Le support est la somme pour les villes Montpellier et Marseille des mesures dont les paramètres sont CD pour la dimension produit et moyen pour la dimension profit, soit $\frac{65}{696}$.

Les règles d'association ne permettent pas de prendre en compte les notions d'ordre. Pour résoudre ce problème, [AS95] propose les **motifs séquentiels**, pouvant être vus comme une extension des règles d'association intégrant diverses contraintes temporelles. Nous détaillons l'extraction de ces motifs dans un contexte multidimensionnel dans la section suivante.

2.2 Les motifs séquentiels multidimensionnels

La recherche de motifs séquentiels consiste à extraire des ensembles d'items couramment associés sur une période de temps bien spécifiée. Les motifs séquentiels mettent en évidence des associations inter-transactions, contrairement aux règles d'association qui extraient des combinaisons intra-

transactions. Les motifs séquentiels peuvent par exemple montrer que : "60% des gens qui achètent une télévision achètent un magnétoscope dans les deux ans qui suivent" ou "75% des gens achètent des housses de siège après avoir acheté une voiture".

De nombreux algorithmes efficaces permettent d'extraire ce type de connaissances dans le contexte de bases de données relationnelles : APriori [AS95], PSP [MCP98] ou encore PrefixSpan [Zak01].

Dans les bases de données relationnelles, les motifs séquentiels utilisent un format de données particulier qui se traduit par le triplet (client, date, itemset). Chaque n-uplet de la base représente une transaction. Dans les bases de données multidimensionnelles, le format des données est différent et la notion de transaction n'est plus présente. L'extraction de motifs séquentiels a alors été adapté au format des bases de données multidimensionnelles de différentes manières. Nous présentons dans cette section les travaux de [PHP⁺01] et [PCL⁺05].

2.2.1 Extraction de séquences intra-dimensionnelles

[PHP⁺01] sont les premiers à aborder le thème de la recherche des motifs séquentiels dans un contexte multidimensionnel. Les auteurs définissent une séquence multidimensionnelle selon le schéma A_1, \dots, A_m, S avec A_i correspondant aux dimensions sur lesquelles les données sont décrites et S représentant les séquences d'achats réalisées par le client ordonnées selon le temps. La notation $*$ signifie toutes valeurs confondues de la dimension et peut apparaître pour une ou plusieurs valeurs des dimensions A_j .

Par exemple, dans un cadre commercial, les informations concernant un client peuvent être l'âge, la ville et un ensemble d'items DVD, CD et lecteur (mp3). Une séquence possible serait (Jeune, Marseille, \langle (Lecteur)(CD DVD) (CD) \rangle), signifiant que beaucoup de personnes jeunes qui viennent de Marseille achètent un lecteur puis reviennent acheter des CD et des DVD et ils achètent ensuite de nouveau des CD. La séquence étoilée ($*$, Marseille, \langle (Lecteur)(DVD) \rangle) signifie que, quel que soit l'âge, beaucoup d'habitants de Marseille achètent un lecteur puis reviennent acheter des DVD.

Les motifs extraits grâce à cette méthode sont intra-dimensionnels puisque les auteurs recherchent des séquences fréquentes selon des valeurs précises pour les dimensions âge et ville. [PCL⁺05] proposent une nouvelle technique afin de prendre en compte le côté inter-dimensionnel.

2.2.2 Extraction de séquences intra et inter-dimensionnelles

L'extraction de motifs séquentiels présentée par [PCL⁺05], donne une autre vision des motifs séquentiels multidimensionnels. Les auteurs partitionnent l'ensemble de dimensions $U=\{D_1, \dots, D_n\}$ de la base en quatre sous-ensembles : D_T (dimension temporelle), D_A (dimensions d'analyse), D_R (dimensions de référence) et D_F (dimensions ignorées). Par exemple, l'ensemble des dimensions de la base de la figure 2.1 peut être partitionnée de la façon suivante : la dimension temporelle est la date, les dimensions d'analyse sont l'âge, les produits et le nombre de produits et enfin les dimensions de référence sont le groupe client et la ville.

[PCL⁺05] proposent une nouvelle définition de la séquence multidimensionnelle :

- **Item multidimensionnel** : c'est un tuple $e = (d_{i_1}, \dots, d_{i_m})$ tel que pour tout k dans $[1, m]$, d_{i_k} est dans $\text{Dom}(D_{i_k})$ (D_{i_k} dans $D_A = \{D_{i_1}, \dots, D_{i_m}\}$).
Dans l'exemple 2.1 : dans le premier bloc l'item (A, P, 2) est présent.
- **Itemset multidimensionnel** : ensemble non vide d'items multidimensionnels $i = \{e_1, \dots, e_p\}$.
Dans l'exemple 2.1 : l'itemset $\{(J, C, 50)(A, P, 2)\}$ est présent dans le premier bloc.
- **Séquence multidimensionnelle** : liste d'itemset ordonnée $S = \langle i_1, \dots, i_l \rangle$.
Dans l'exemple 2.1 la séquence $\{(J, C, 50)(A, P, 2)\}\{(A, P, 3)\}$ apparaît dans le premier bloc.

- **Support** : D_R dimensions de référence, T une table partitionnée en bloc B_{T,D_R} . Le support d'une séquence est défini par : $\text{support}(s) = \frac{\text{Nbr de blocs supportant } S}{\text{Nbr total de blocs}}$.
Par exemple pour la séquence précédente, le support est de $\frac{1}{3}$ (un bloc sur les trois blocs).

Date	Age	Produit	NB
1	J	C	50
1	A	P	2
2	A	P	3

bloc : Educ, Paris

Date	Age	Produit	NB
1	V	C	20
1	V	M	5
2	V	C	50

bloc : Ret, Marseille

Date	Age	Produit	NB
2	A	P	10
3	J	R	15

bloc : Educ, Marseille

FIG. 2.1 – Exemple : base de données multidimensionnelles partitionnée en blocs.

De plus, [PCL⁺05] proposent d'insérer des valeurs "jokers" (ou *étoiles*) dans les items multidimensionnels, signifiant "*quelle que soit la valeur pour cette dimension*". Un item étoilé est alors un ensemble de valeurs ou chacune d'elle correspond à la valeur d'une dimension d'analyse ou à la valeur joker. Un item ne peut être composé que d'étoiles car cela ne porte aucun sens. **L'item étoilé** est défini par les auteurs de la façon suivante :

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

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

Cette définition permet d'étendre la recherche de séquences fréquentes aux séquences plus générales, susceptibles de contenir de l'information intéressante.

Nous avons proposé dans ce chapitre une brève étude des algorithmes d'extraction de connaissances communes. Cependant, au vue de la masse d'informations stockées, il s'avère que les résultats de ces différentes fouilles sont souvent inintéressants, car ils apportent des informations trop communes, et donc connues des décideurs. Cependant, beaucoup de connaissances intéressantes sont cachées parmi des connaissances évidentes. L'extraction de connaissances atypiques devient alors une problématique importante. Nous expliquons dans le chapitre suivant les méthodes proposant l'extraction de telles méthodes.

Chapitre 3

Extraction de connaissances atypiques

Les algorithmes d'extraction de connaissances actuellement proposés mettent en évidence des comportements récurrents en masse, or ces connaissances communes en trop grand nombre cachent des connaissances plus intéressantes, dites "*atypiques*". Cependant l'extraction de tels motifs implique une baisse de support minimal significative, et les algorithmes se trouvent confrontés au problème de l'explosion combinatoire.

Ainsi, différents chercheurs se sont intéressés au problème de l'extraction de connaissances non fréquentes intéressantes. Nous présentons, dans ce chapitre, ces méthodes, dites objectives, qui ne s'appliquent qu'au contexte relationnel. Toutes ces propositions sont basées sur les connaissances communes extraites selon les méthodes présentées au chapitre précédent.

3.1 Utilisation des règles d'association classiques

Les travaux présentés dans cette section utilisent comme connaissances communes les règles d'association classiques. Les travaux de [Suz99] extraient des connaissances atypiques nommées "*règles exceptionnelles*", définies comme des "*règles contredisant les règles communes*". [BCMG04] propose une définition différente de l'atypicité via la définition de "*règles anormales*" permettant de déceler des comportements alternatifs aux comportements communs.

Ces deux travaux ont en commun leur point de départ, qui sont les connaissances communes, formalisées par les règles d'association. Cependant, les comportements atypiques extraits ont une signification très différente.

3.1.1 Les règles exceptionnelles

Pour [Suz99] et [HLSL00] une règle est exceptionnelle si elle contredit une règle commune. Cela se traduit par un faible support, car cette règle ne concerne pas un grand nombre d'objets (c'est ce qui la rend atypique), et par une forte confiance, ce qui signifie qu'une grande majorité des objets concernés par la règle la vérifient. La définition d'une règle exceptionnelle selon les auteurs est alors la suivante :

$$\frac{\begin{array}{l} Y_u \rightarrow x \\ Y_u \cup Z_v \rightarrow x' \\ Z_v \rightarrow x' \end{array}}{\begin{array}{l} \text{(la confiance et le support sont hauts)} \\ \text{(une confiance haute et un support bas)} \\ \text{(une faible confiance et/ou faible support,} \\ \text{ } Z_v \text{ et } x' \text{ sont peu en relation)} \end{array}}$$

x est x' sont les valeurs différentes d'un même attribut et Y_u et Z_v sont de simples attributs ou des ensembles d'attributs. Par exemple, une exception extraite en fonction de cette définition est "*si*

un patient a un symptôme A_1 alors il a la maladie B_1 pourtant si un patient a les symptômes A_1 et A_2 alors il a la maladie B_2 ". La prise en compte du second symptôme (l'information Z_v) entraîne une conséquence différente.

Afin d'extraire ces règles, [Suz99] propose de fixer cinq seuils. Ces seuils définissent la notion de support faible et de confiance forte. Les règles sont communes si elles respectent deux seuils minimaux fixés pour la confiance et le support. Le support d'une règle exceptionnelle doit être faible mais les auteurs prennent en compte les séquences constituant du bruit, c'est pourquoi ils définissent un autre support minimal pour les règles exceptionnelles. Ils proposent également de fixer un quatrième seuil permettant de définir un seuil de confiance pour les règles exceptionnelles différent de la confiance minimale d'une règle commune. Cela leur permet de s'assurer qu'une règle exceptionnelle a une confiance haute. Un dernier seuil fixé représente le seuil maximal pour la confiance de la règle de condition $Z_v \rightarrow x'$.

Cette technique présente donc l'inconvénient de laisser à l'utilisateur la charge de fixer les cinq seuils cités, qui ne sont pas évidents à déterminer afin d'extraire les règles exceptionnelles les plus intéressantes. Dans [Suz99], les auteurs proposent aux utilisateurs de déterminer ces seuils par un processus automatique.

L'idée appliquée dans [HLSL00] est de détecter les règles exceptionnelles à l'aide de calculs basés sur la confiance et le support. Les deux mesures permettent d'estimer l'intérêt de la règle $AB \rightarrow X$ à partir de $A \rightarrow X$ et $B \rightarrow X$. Une première mesure RI_c^{AB} prend en compte la vraie confiance de la règle $\Pr(X|AB)$ et l'estimation à partir des confiances des règles $A \rightarrow X$ et $B \rightarrow X$. Puis une seconde mesure RI_s^{AB} applique le même traitement mais en fonction du support. L'intérêt d'une règle exceptionnelle est ensuite calculé avec la mesure totale RI . Cette technique permet de n'avoir qu'un seuil fixé par l'utilisateur afin de déterminer si une règle est exceptionnelle.

La définition posée dans les deux articles ([Suz99], [HLSL00]) donne une idée claire de ce qu'est une exception. La règle exceptionnelle est facile à interpréter lorsque la règle commune est indiquée. Ces règles exceptionnelles permettent d'extraire des connaissances très intéressantes comme par exemple que la règle commune "*quand un patient a le syndrome S alors il est atteint de la maladie M1*" est contredite par la règle exceptionnelle "*quand le patient est une femme et qu'elle est atteinte du syndrome S en général elle souffre de la maladies M2*". Ces techniques sont des manières de comparer les deux règles au travers de mesures différentes. Cette définition est donc une base solide pour une multitude de travaux.

3.1.2 Les règles anormales

[BCMG04] propose de définir une anomalie en fonction d'une règle commune. La définition formelle suivante est posée :

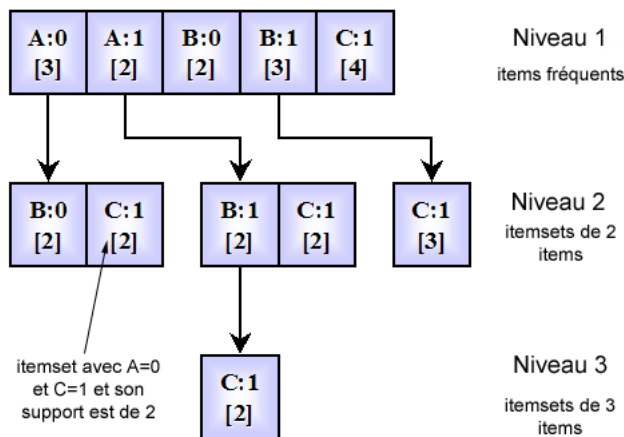
$$\frac{X \rightarrow Y \text{ (le support et la confiance sont hauts)}}{X \neg Y \rightarrow A \text{ (la confiance forte)}} \\ \frac{X \neg Y \rightarrow A \text{ (la confiance forte)}}{XY \rightarrow \neg A \text{ (la confiance forte)}}$$

L'idée est de déterminer à partir d'une règle commune $X \rightarrow Y$ les implications existantes de X sans Y . Cette définition d'anomalie permet de détecter des connaissances du type : "*Si un patient a le symptôme A1 alors il a la maladie B1 cependant si un patient a le symptôme A1 mais pas la maladie B1 alors il a probablement la maladie B2*". Les auteurs ne cherchent pas une règle contradictoire mais plutôt une alternative à la conséquence d'une règle commune.

Pour extraire ces règles, [BCMG04] utilise un algorithme basé sur Apriori [AS94] c'est-à-dire il génère des itemsets fréquents et extrait des règles dérivées de ces itemsets. Ils proposent d'étendre ce modèle afin de découvrir les règles anormales grâce à la définition du calcul de la confiance pour les règles exceptionnelles suivantes :

$$conf_r(X \neg Y \rightarrow A) = \frac{supp(X \cup A) - supp(X \cup Y \cup A)}{supp(Y)supp(X \cup Y)}$$

Ce calcul suppose l'enregistrement de chaque support de la formule. Pour cela, [BCMS01] propose une structure d'arbre TBAR (représentée à la figure 3.1) permettant de gérer efficacement ces supports et représentant les itemsets par niveau. La découverte des règles est alors basée sur le parcours de l'arbre après la génération des règles communes.



A	B	C
0	0	0
0	0	1
0	1	1
1	1	1
1	1	1

FIG. 3.2 – Tableau avec 3 attributs binaires

FIG. 3.1 – Structure TBAR pour un support de 2

La figure 3.1 représente l'arbre construit pour un support de $\frac{2}{5}$ à partir des itemsets fréquents du tableau 3.2. Les itemsets fréquents de l'exemple sont les suivant : (A :0) (A :1) (B :1) ... (A :0, B :0)... (A :1, B :1, C :1).

Les connaissances extraites sont très intéressantes car la détection d'une conséquence moins évidente est difficile à extraire à cause de leurs faibles supports. Cette méthode permet de découvrir des connaissances atypiques parmi une importante masse d'informations.

3.2 Les règles intéressantes

[AL99] proposent de "typer" l'antécédent et la conséquence des règles d'association communes. Ainsi, deux sortes de règles sont recherchées : celles ayant la forme "attributs quantitatifs → attributs quantitatifs" et celles ayant la forme "attributs catégoriques → attributs quantitatifs". Les valeurs quantitatives de la conséquence de la règle représentent dans les deux cas une agrégation (telle que la moyenne, la variance etc...) en fonction des attributs de l'antécédent. Les auteurs ne définissent pas réellement des règles exceptionnelles mais plutôt des règles intéressantes. Ces règles intéressantes sont représentées par la notion de sous-règles définie de la façon suivante :

$$\frac{X \rightarrow mean_j(TX)}{Y \rightarrow mean_j(TY) \text{ avec } X \in Y}$$

$mean_j(TX)$ représente les j attributs numériques calculés en fonction des transactions TX (transactions dans lesquelles l'ensemble d'items X apparaît). La seconde règle est une sous règle intéressante si les valeurs de $mean_j(TY)$ sont très différentes des valeurs trouvées pour $mean_j(TX)$.

Voici un exemple de règles et sous-règles de type "**catégorique** → **quantitatif**" :

- Une personne fumeuse a une espérance de vie de 60 ans
- Une personne fumeuse et qui boit du vin a une espérance de vie de 70 ans

Cela signifie qu'une personne qui fume a une espérance de vie plus forte si elle boit du vin.

Voici un exemple de règles et sous-règles de type "**quantitatif** → **quantitatif**" :

- âge appartient [60,80] → poids en moyenne de 90
- âge appartient [50,80] → poids moyen est de 87.5

Cette seconde règle montre un changement si l'on considère un intervalle plus grand d'âge. Ces deux règles permettent de déduire que les personnes ont tendance à grossir vers 60 ans.

Avec cette technique, la prise en compte d'une information supplémentaire dans l'antécédent entraîne une variation de la valeur de la conséquence importante. Une règle intéressante est alors extraite. Cette définition rappelle le principe proposé par [Suz99] qui ajoute de l'information à une règle commune et vérifie si la conséquence change. Cependant, dans les travaux de [Suz99] la conséquence n'est pas une valeur numérique.

Chapitre 4

Discussion et objectifs

Les bases de données multidimensionnelles reflètent les mouvements d'un grand nombre d'individus, représentés sous une forme agrégée. Dans ce type de base, la date fait souvent office d'identifiant des enregistrements, puisque l'on peut voir ces données agrégées comme une sorte d'archive. L'enjeu est alors d'assister le décideur lors de son expertise en extrayant des informations intéressantes. Cependant, les algorithmes d'extraction de connaissances classiques, tels que les règles d'association ou les motifs séquentiels ne permettent pas de mettre en évidence des connaissances intéressantes, celles-ci étant noyées dans une masse d'informations fréquentes mais non pertinentes.

Différentes méthodes permettant d'extraire des informations potentiellement intéressantes ont été présentées dans le chapitre précédent. Toutes ces méthodes se basent dans un premier temps sur des connaissances communes, plus particulièrement sur l'extraction de règles d'associations classiques. Cependant, aucune des techniques étudiées n'est applicable dans un contexte multidimensionnel, pour de nombreuses raisons. Dans un premier temps, les règles d'association ne prennent pas en compte les notions d'ordre, qui nous permettraient dans notre contexte de conserver les informations relatives à la date. De plus, les règles d'associations adaptées au contexte multidimensionnel présentées extraient des connaissances limitées, puisque les règles sont soit intra-dimensionnelles, soit inter-dimensionnelles, mais pas les deux à la fois, ce qui empêche l'extraction d'items appartenant à différentes transactions en plus de ceux au sein d'une même transaction.

Les motifs séquentiels répondent à ce problème, notamment grâce aux travaux de [PCL⁺05]. Cependant, il n'est plus possible d'utiliser les travaux d'extraction de connaissances atypiques précédent, car la notion de règles n'est pas conservée. De plus, cette méthode est limitée par la façon de considérer les mesures. En effet, ces travaux considèrent les différentes valeurs de la mesure comme symboliques, et détectent donc deux valeurs très proches comme étant différentes, ce qui n'a pas vraiment de sens dans un contexte multidimensionnel. Par exemple le nombre 39 et le nombre 40 sont deux symboles distincts et ne peuvent être comparés. Si les deux items $(a_1, b_1, 39)$ et $(a_1, b_1, 40)$ ne sont pas fréquents, le rapprochement des valeurs 39 et 40 pourrait permettre d'extraire le fait que a_1 et b_1 sont souvent effectués avec des valeurs proches de 40.

Dans leurs travaux, [PCL⁺05] propose la notion de séquence étoilée, séquence composée d'items où une étoile est présente pour une ou plusieurs dimensions. Cela permet d'extraire plus de séquences fréquentes mais beaucoup d'items seront fréquents avec une étoile pour la mesure. Nous n'aurons donc plus aucune idée de la valeur de la mesure. De plus des valeurs très éloignées sont regroupées, les motifs peuvent alors être moins significatifs. L'étoile n'est donc pas une solution adaptée au traitement des données numériques (mesures).

Les définitions de [Suz99] concernant les connaissances atypiques nous semblent justifiées : une connaissance atypique peut être représentée sous la forme d'une règle d'association ayant un faible support mais une forte confiance. La proposition alternative de [BCMG04] en revanche pose des problèmes d'adaptation à notre contexte : il est difficile de définir formellement "l'absence" d'item. Nous proposons donc une méthode d'extraction de connaissance atypiques complète, qui part de la recherche de motifs séquentiels à l'extraction de règles exceptionnelles. Puis, nous définissons formellement des règles exceptionnelles comme règles contredisant des règles communes plus générales, afin de déterminer les algorithmes d'extraction associés.

Deuxième partie

Proposition

1	Extraction de connaissances communes	26
1.1	Motifs séquentiels multidimensionnels avec traitement de la mesure	27
1.1.1	La mesure dans la dimension d'analyse après prétraitement	27
1.1.2	La mesure utilisée pour calculer le support	28
1.2	Les règles séquentielles multidimensionnelles	30
1.2.1	Définitions	30
1.2.2	Algorithme d'extraction de règles séquentielles	31
1.2.3	Exemple	32
2	Extraction de connaissances atypiques	34
2.1	Extraction de règles exceptionnelles	34
2.2	Collage des séquences	36
2.2.1	Définition	37
2.2.2	Algorithme RechercheSeqCol	39
2.2.3	Exemple	39
2.3	Recherche de conséquences différentes	41
2.3.1	Définition	42
2.3.2	Algorithme RechercheConseqExc	43
2.3.3	Exemple	44
2.4	Algorithmes généraux	45

Nous avons pour objectif la détection de comportements atypiques dans des bases de données multidimensionnelles. Les différentes techniques abordées précédemment ne se situent que dans un contexte de bases de données relationnelles pour des règles d'association. C'est pourquoi nous proposons dans cette partie une méthode d'extraction de *règles exceptionnelles* dans un contexte de base de données multidimensionnelles.

La détection de comportements atypiques ne peut se résumer à une simple baisse de support : en effet, les algorithmes d'extraction se retrouvent d'une part confrontés au problème d'explosion combinatoire et d'autre part, les connaissances fréquentes sont extraites en masse et rechercher les connaissances atypiques parmi celles-ci s'avérerait être un travail long, fastidieux et subjectif. Afin de répondre à ce besoin, notre méthode s'articule autour de quatre étapes :

1. Extraction de séquences fréquentes
2. Extraction de règles séquentielles
3. Ajout d'information aux règles
4. Recherche de conséquences différentes

Le schéma représenté à la figure 1 est appliqué.

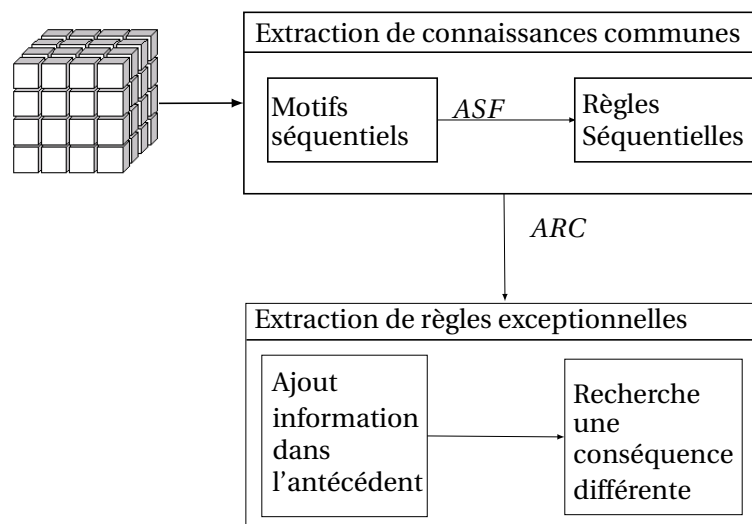


FIG. 1 – Processus d'extraction de règles exceptionnelles

Dans un premier temps, les séquences fréquentes sont extraites et enregistrées dans un arbre (nommé ASF) puis les règles communes sont détectées et stockées dans un arbre des règles communes (ARC). Ensuite, nous recherchons à ajouter de l'information à une séquence. Puis nous cherchons une conséquence différente de la conséquence de la règle commune associée, celles-ci correspondant à des règles exceptionnelles.

Nous proposons dans un premier temps d'extraire les motifs séquentiels à partir de bases de données multidimensionnelles en intégrant la mesure au moment de l'extraction, ce qui va permettre la découverte de motifs pertinents. Cette étape peut être vue comme une extension des travaux de [PCL⁺05]. Afin de traiter cette mesure, nous proposons deux techniques : l'une utilise la mesure au moment du calcul du support d'une séquence, ce qui a pour effet de soustraire la mesure des dimensions d'analyse, et l'autre utilise un partitionnement strict et flou pour les mesures, tout en conservant la mesure dans les dimensions d'analyse.

L'étude de l'existant nous a permis de valider les principes de l'approche de [Suz99] : un comportement atypique pertinent est un comportement qui a une faible fréquence, mais une forte confiance. Ainsi, nous proposons la création de règles séquentielles, définies formellement à la section 1.2 afin de mieux représenter une connaissance exceptionnelle, ainsi que les algorithmes d'extraction associés. Ce format nous permettra alors de définir une confiance pour un motif séquentiel.

L'idée générale présentée dans [Suz99] est alors adaptée dans nos travaux à un contexte multidimensionnel. Le principe est d'ajouter une information à l'antécédent d'une règle commune et de vérifier si la conséquence résultante est différente. Dans ce cas, nous avons découvert une règle exceptionnelle. Cette définition impose dans un premier temps la définition de l'ajout d'information à l'antécédent d'une règle séquentielle, ce qui est rendu possible grâce à la définition formelle de *séquences collables* puis dans un second temps la définition de *conséquences différentes* dans notre contexte.

Chapitre 1

Extraction de connaissances communes

Afin d'extraire des comportements exceptionnels, nous utilisons une méthode objective qui consiste à extraire des connaissances communes à partir desquelles nous recherchons des comportements atypiques. Dans cette section le type de connaissances utilisées et les traitements nécessaires pour obtenir des connaissances communes intéressantes sont abordés.

Dans la section précédente, nous avons décrits deux types de connaissances existantes sur les bases de données multidimensionnelles : les règles d'association et les motifs séquentiels multidimensionnelles. Dans notre contexte, une notion de temporalité est très présente puisque les données sont identifiées par une date. Les motifs séquentiels, au contrairement aux règles d'association, traitent les connaissances avec une notion de temporalité. Dans les travaux de [Suz99], les connaissances atypiques sont définies comme des règles exceptionnelles. Afin d'étendre ces travaux, nous avons défini ces règles dans le contexte multidimensionnel. Les connaissances utilisées sont alors des règles séquentielles multidimensionnelles.

Notre méthode consiste à extraire dans un premier temps des séquences fréquentes et en fonction de celles-ci générer des règles séquentielles associées en calculant leurs taux d'implication (appelé aussi confiance). La détection de connaissances communes s'effectue alors selon le schéma suivant :

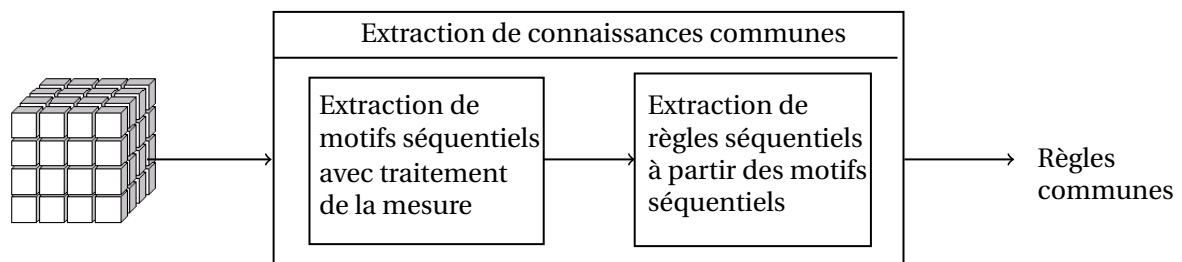


FIG. 1.1 – Détection de connaissances communes

[PCL⁺05] propose une méthode d'extraction de motifs séquentiels multidimensionnels efficace cependant la mesure est traitée dans les dimensions d'analyse comme une valeur symbolique. A la section 1.1, nous détaillerons une méthode pour extraire les motifs séquentiels en prenant en compte la mesure. Puis à la section 1.2.2, nous définirons formellement les règles séquentielles ainsi que les algorithmes de découverte associés.

1.1 Motifs séquentiels multidimensionnels avec traitement de la mesure

Les méthodes d'extraction de motifs séquentiels classique incrémentent le support d'une séquence dès que celle-ci apparaît dans une ou plusieurs transactions d'un client. Dans les bases de données multidimensionnelles, la mesure a un rôle très important, car c'est la seule information qui permet de quantifier les tendances d'une population à une date donnée. Pour [PCL⁺05], elle fait partie des dimensions d'analyse ce qui aboutit à l'extraction de motifs ayant les mêmes valeurs pour toutes les dimensions et des valeurs très proches pour la mesure. Par exemple les items $(a_1, b_1, 34)$ et $(a_1, b_1, 35)$ sont deux items différents. Pourtant si cette mesure représente un nombre de ventes alors 34 et 35 sont des valeurs très proches. Une simple incrémentation du support à l'appartition d'une mesure n'est donc pas suffisamment représentatif, surtout dans notre contexte, où les connaissances communes serviront de base à l'extraction de connaissances exceptionnelles.

Il est donc primordial d'extraire des motifs où la mesure est pleinement considérée. Nous proposons pour cela deux solutions : soit prétraiter la mesure en remplaçant les valeurs numérique par une valeur symbolique représentative, soit en calculant le support d'une séquence à partir de la mesure. C'est ce que nous détaillons dans cette section.

1.1.1 La mesure dans la dimension d'analyse après prétraitement

Le principe de cette seconde technique prétraite la mesure afin de trouver un partitionnement et utilise une valeur symbolique plutôt que la mesure elle-même. Afin de traiter les données numériques pour les processus d'extraction de connaissances, de nombreux travaux utilisent une technique de partitionnement. Le partitionnement de valeurs numériques dans l'extraction de connaissances a été abordée dans de nombreux travaux.

[SA96], [AL99] utilisent des techniques de partitionnement strict afin de pouvoir transformer les données numériques en données binaires. Par exemple, l'âge est une donnée numérique, elle est partitionnée en trois ensembles [12-25], [26-55] et [56-100], si l'âge du client est de 54 alors on aura pour l'attribut age[12-25] la valeur 0, pour age[26-55] la valeur 1 et pour age[56-100] la valeur 0.

[KFW98] et [Gye00] proposent d'utiliser un partitionnement flou pour les données numériques dans le cadre de règle d'association. Pour les motifs séquentiels, un partitionnement flou est aussi effectué dans les travaux de [FLT04]. Cependant, ces différents travaux s'appliquent uniquement dans un contexte de bases de données relationnelles.

Une première solution consiste alors à adapter ces techniques à notre contexte. Les mesures sont remplacées par la valeur de son sous ensemble et elles sont conservées dans les dimensions d'analyse. Par exemple les items $(a_1, b_1, 34)$ et $(a_1, b_1, 35)$ apparaissent sous la forme (a_1, b_1, Peu) . Dans le cas d'un partitionnement flou, chaque item est associé à une degré d'appartenance et le support peut être calculé de différentes façons : calcul flou ou binaire ([FLT04], [Gye00]).

Cette méthode prend en compte tout type de mesure, cependant les petites mesures représentent des événements rares. Il serait alors intéressant de mettre en avant les événements qui ont lieu en forte quantité. Pour cela nous proposons une seconde technique dans laquelle la mesure n'est plus dans les dimensions d'analyse mais elle permet de calculer le support d'une séquence.

1.1.2 La mesure utilisée pour calculer le support

Dans les bases de données multidimensionnelles, les données sont agrégées. L'extraction de motifs séquentiels pertinents implique la prise en compte de la mesure au moment du calcul du support. En effet les méthodes classiques incrémentent le support d'une séquence à chaque apparition de celle-ci dans un bloc. Dans cette méthode nous proposons de ne plus incrémenter de un le support mais de la valeur de la mesure, ce qui le rend plus représentatif.

Les calculs effectués sur la mesure doivent conserver l'apparition de tous les évènements d'une séquence, c'est pourquoi nous proposons l'utilisation d'un opérateur t-norme. Prenons par exemple la séquence $\{(a_1, b_1)\}\{(a_1, b_2)\}$ qui met en évidence que le premier évènement (a_1, b_1) *et* le second évènement (a_1, b_2) ont eu lieu (à des unités de temps différentes).

Pour chaque séquence s_1 apparaissant dans le bloc, nous prenons la mesure minimale de tous les items de la séquence. Comme pour les motifs séquentiels classiques, la séquence peut apparaître plusieurs fois dans le même bloc, nous considérons alors le support maximum de cette séquence dans un même bloc selon la définition 1. Le support d'une séquence est exprimé de la façon suivante :

Définition 1 Soit s une séquence (ensemble des itemsets de la séquence), B un ensemble de blocs, IS l'ensemble des items d'un itemset, $m[i]$ la mesure de l'item i alors le support de s est :

$$support(s) = \sum_{b \in B} [\underline{\theta}_b \overline{\Gamma}_{IS \in s} \overline{\Gamma}_{i \in IS}(m[i])]$$

Par souci de clarté, nous représentons le support sous la forme d'un pourcentage. Le support ne doit plus être divisé par le nombre de bloc mais par une quantité définie selon la mesure. Deux types de division sont proposés : macro ou micro count.

La technique **Micro count (sans ratio)** consiste à diviser le support (pour l'ensemble des blocs) par la mesure totale de la base (moyenne pour tous les blocs). Le support est alors exprimé de la façon suivante :

Définition 2 Soit s une séquence, B un ensemble de blocs, IS l'ensemble des items de la séquence, $m[i]$ la mesure de l'item i et Γ_1 la mesure totale dans la base alors le support de s est :

$$support(s) = \frac{\sum_{b \in B} [\underline{\theta}_b \overline{\Gamma}_{IS \in s} \overline{\Gamma}_{i \in IS}(m[i])]}{\Gamma_1}$$

La technique **Macro count (avec ratio)** consiste à diviser le support de la séquence pour chaque bloc par la mesure totale du bloc. Puis la somme des supports de chaque bloc est divisée par le nombre de bloc (une moyenne des moyennes par bloc). Le support est alors exprimé de la façon suivante :

Définition 3 Soit s une séquence, B un ensemble de blocs, IS l'ensemble des items de la séquence, $m[i]$ la mesure de l'item i et $\Gamma_{2,b}$ la mesure totale du bloc b alors le support de s est :

$$support(s) = \frac{(\sum_{b \in B} [\underline{\theta}_b \overline{\Gamma}_{IS \in s} \overline{\Gamma}_{i \in IS}(m[i])])}{nbBlocs}$$

Afin de mieux comprendre le calcul du support, prenons la base de données de la figure 1.2 dont les dimensions sont partitionnées de la façon suivante :

- Les dimensions d'analyse sont : A, B, et C.
- La dimension de référence est Geo (situation géographique).

Date	A	B	C	M
1	a_1	b_1	c_1	128
	a_1	b_1	c_2	152
	a_2	b_1	c_1	202
2	a_1	b_1	c_1	100
	a_1	b_1	c_2	200
	a_2	b_1	c_1	77

GEO=CRCNO, nb d'actions
total du bloc =220

Date	A	B	C	M
1	a_1	b_1	c_1	50
	a_1	b_1	c_2	111
	a_2	b_1	c_1	70
2	a_1	b_1	c_1	108
	a_1	b_1	c_2	100
	a_2	b_1	c_1	80

GEO=CRCNA nb d'actions
total du bloc =180

Date	A	B	C	M
1	a_1	b_1	c_1	45
	a_1	b_1	c_2	300
	a_2	b_1	c_1	130
2	a_1	b_1	c_1	106
	a_1	b_1	c_2	200
	a_2	b_1	c_1	125

GEO=CRCBO, nombre d'actions
total du bloc = 500

FIG. 1.2 – Exemple de bases de données multidimensionnelles

- M est la mesure et représente un nombre d'actions.
- Date est la dimension temporelle.

Pour la séquence $\langle\langle(a_1, b_1, c_1) (a_1, b_1, c_2)\rangle\rangle$, le calcul effectué est le suivant :

1. Dans le premier bloc, cette séquence apparaît pour la date 1, et son support est le minimum des mesures pour chaque item : $\min(128, 152) = 128$
2. Cette séquence apparaît plusieurs fois dans la même bloc (date 2), le support de la séquence pour ce bloc est le maximum des supports des séquences possibles du bloc : $\max(\min(128, 152), \min(100, 200)) = 128$
3. Le support de la séquence est la somme des supports de la séquence pour chaque bloc : $128 + 100 + 106 = 334$ (même schéma pour calculer le support des autres blocs).
4. Le support dans le cas d'un micro count est de : $128 + 100 + 106 = 334$ ($\frac{334}{900} = 37\%$)
5. Le support dans le cas d'un macro count est de : $(\frac{128}{220} + \frac{100}{180} + \frac{106}{500}) / 3 = 45\%$

L'étoile dans les motifs séquentiels permet de trouver des séquences plus générales. Par exemple pour les séquences $\langle\langle(a_1, b_1, c_2)\rangle\rangle$ et $\langle\langle(a_1, b_1, c_1)\rangle\rangle$ avec un faible support, il est intéressant de détecter que $\langle\langle(a_1, b_1, *)\rangle\rangle$ est une séquence fréquente. Cela signifie que quelle que soit la valeur de la troisième dimension, l'item avec les valeurs a_1 (pour la dimension A) et b_1 (pour la dimension B) est fréquent.

Ce calcul est possible car nous considérons que tous les items possèdent une mesure. Cependant, il est possible d'extraire des séquences étoilées pour lesquels il n'existe pas de mesure associée à l'item. Une étoile sur une dimension signifie que telle **ou** telle valeur apparaît pour la dimension. Afin de représenter cette notion, nous utilisons un opérateur $t_conorme$. Nous proposons alors deux façons de considérer la mesure pour un item étoilé : le maximum ou la somme.

Pour une séquence, nous proposons de prendre la mesure **maximum** pour les différentes apparitions d'un item quelle que soit la valeur de l'étoile pour une même date. Puis nous appliquons le traitement expliqué dans la section précédente pour calculer le support de la séquence. Par exemple le support de l'item $(*, b_1, c_1)$ pour le tableau 1.2 est de $\frac{128+102+125}{900} = 40\%$. Le support de la séquence $\langle\langle(*, b_1, c_1)\rangle\rangle$ est de $(100+70+45)/900=23.8\%$. Dans les bases de données multidimensionnelles, il peut y avoir pour certaines dimensions une valeur dominante (c'est à dire qu'elle est souvent associée aux plus fortes mesures). L'utilisation du maximum peut entraîner qu'une séquence a un support souvent identique ou proche qu'une séquence plus générale. Prendre le maximum pour les mesures possibles de l'étoile, extrait alors peu de séquences en plus.

Une solution alternative est alors de prendre **la somme des mesures** pour une même date des items avec des valeurs différentes pour l'étoile. Par exemple en fonction de l'exemple 1.2, le support de l'item $(*, b_1, c_1)$ est de $\frac{(128+59)+(102+90)+(105+125)}{900} = 67\%$. Le problème principal rencontré lorsque nous prenons la somme est que certaines dimensions ne sont pas additionnables. Par exemple une dimension "loisirs" est présente dans une base de données concernant des abonnés à un service quelconque, certains individus peuvent avoir plusieurs loisirs. Si nous mettons une étoile sur la dimension "loisirs", l'addition des différentes valeurs n'a pas de sens car les clients peuvent être comptabilisés plusieurs fois, ce qui entraîne le risque de trouver un support supérieur à 100%.

Cette méthode met en avant la mesure à travers le support. Elle permet d'extraire des motifs avec des mesures importantes ou qui apparaissent dans beaucoup de blocs. Un événement rare dans cette technique est un événement qui a des mesures très faibles ou qui apparaît dans peu de blocs.

La mesure n'est plus dans la dimension d'analyse, nous n'avons donc plus d'idée précise de la mesure pour chaque item multidimensionnel. De plus, une cellule dans une base de données multidimensionnelle peut contenir plusieurs mesures. La méthode proposée permet de calculer un support en fonction d'une seule mesure.

Nous avons proposé dans cette section un ensemble de méthodes permettant la prise en compte de la mesure au moment de l'extraction des motifs. Nous avons donc extrait des motifs séquentiels pertinents. Nous souhaitons maintenant formaliser les connaissances communes sous la forme de règles afin d'extraire par la suite des règles exceptionnelles. Dans la prochaine section, nous définissons de manière formelle les règles séquentielles.

1.2 Les règles séquentielles multidimensionnelles

Les règles séquentielles sont une extension des motifs séquentiels et des règles d'association. Elles permettent de conserver à la fois la notion d'implication des règles d'association, et la notion de temporalité des motifs séquentiels. Les règles d'association classiques sont extraites à partir d'un seuil minimum de confiance et de support. Il s'agira alors de redéfinir la notion de confiance et d'implication, dans le cadre des règles séquentielles multidimensionnelles.

1.2.1 Définitions

Les règles d'association classiques mettent en avant la notion d'implication. Dans notre contexte, il est primordial de conserver la notion de temporalité. Nous retrouvons celle-ci dans la définition des motifs séquentiels, c'est pourquoi nous proposons d'étendre cette définition de la manière suivante.

Définition 4 Soit une séquence $s = \langle IS_1 \dots IS_n \rangle$, alors une règle séquentielle est notée $\langle IS_1 \dots IS_k \rangle \rightarrow \langle IS_{k+1} \dots IS_n \rangle$ signifiant que $\langle IS_1 \dots IS_k \rangle$ implique et précède $\langle IS_{k+1} \dots IS_n \rangle$. $\langle IS_1 \dots IS_k \rangle$ est appelé l'antécédent et $\langle IS_{k+1} \dots IS_n \rangle$ est nommé la conséquence de la règle.

Comme pour les règles d'association, un règle séquentielle doit respecter deux calculs permettant d'estimer l'intérêt d'une règle : le support (notion de fréquence) et la confiance. Deux seuils sont alors fixés afin de savoir si une règle est commune : le support minimum (*minsupp*) et la confiance minimum (*minconf*). La principale différence entre les règles séquentielles et les motifs séquentiels est la confiance qui représente dans les règles séquentielles la probabilité d'avoir la séquence d'événements $IS_{k+1} \dots IS_{k+n}$ alors que la séquence d'événement $\langle IS_1 \dots IS_k \rangle$ a eu lieu.

Une règle séquentielle commune est définie comme une séquence fréquente avec une forte probabilité d'obtenir les $n-k$ derniers itemsets sachant que les autres itemsets ont eu lieu. Pour l'itemset IS_k , k représente donc la position de l'itemset dans la séquence. k ne représente pas forcément une unité de temps, mais une notion d'ordre.

1.2.2 Algorithme d'extraction de règles séquentielles

Les règles communes sont détectées à partir de l'arbre des séquences fréquentes (ASF). Cet arbre représente les séquences fréquentes sous leurs formes préfixées. Il existe deux types de liens : "en même temps" et "autre". Les feuilles représentent alors les séquences, et chaque père d'un noeud son préfixe. La sous section suivante illustre un exemple. A chaque noeud, le support est enregistré. Le lecteur intéressé pourra se référer à [MCP98]. Détecter des règles séquentielles consiste à tester toutes les possibilités de règles à partir des séquences fréquentes maximales. Pour chaque lien de type "autre" d'une séquence fréquente maximale nous testons si ce lien est à la fois "autre" et "implique". Pour cela nous utilisons les supports enregistrés pour chaque séquence maximale et sous séquence afin de calculer la confiance.

Afin de stocker de manière efficace les règles générées, nous avons créé un troisième type de branche permettant de détecter rapidement où se trouve l'implication d'une règle. Nous nommons ce nouveau lien "other implique".

Algorithme 1 : RechercheRC : Extraction des règles séquentielles

```

Entrées : minConf                                     /* La confiance minimum */
1      ASF                                             /* L'arbre des séquences fréquentes */
Sorties : ARC                                       /* arbre des règles communes */
2 début
3   conf ← 0                                           /* Confiance de la séquence */
4   pour chaque s ∈ SequencesMaximales(ASF) faire
5       pour chaque lo ∈ lienOther(s) faire
6           conf ← CalculConfiance(s, lo)
7           si conf ≥ minConf alors
8               ARC.AjoutRègle(S, lo) /* Ajout de la règle avec implication sur le
9                   lien lo */
9           fin
10        fin
11    fin
12    retourner ARC
13 fin

```

L'algorithme 1 cherche pour chaque séquence maximale (ligne 4) toutes les règles qui dérivent de celle-ci. Pour cela nous regardons pour chaque lien autre de la séquence (ligne 5) si ce lien est un lien de type 'other implique'. Nous calculons la confiance (ligne 6) en considérant le lien comme une implication et si la confiance est supérieure à *minConf* (ligne 7) alors la règle est commune. Nous l'insérons alors dans l'arbre des règles communes (ligne 8) au travers de la création d'une branche "other implique".

Bloc	Date	A	B
1	1	A ₁	B ₁
1	1	A ₁	B ₂
1	2	A ₂	B ₂
1	2	A ₂	B ₁
1	3	A ₃	B ₂
2	1	A ₁	B ₃
2	2	A ₂	B ₂
2	3	A ₃	B ₂

Bloc	Date	A	B
3	1	A ₁	B ₂
3	2	A ₁	B ₂
3	3	A ₂	B ₃
3	4	A ₂	B ₁
4	1	A ₁	B ₃
4	1	A ₂	B ₃
4	2	A ₂	B ₂
4	3	A ₃	B ₂

Bloc	Date	A	B
5	1	A ₂	B ₂
5	2	A ₃	B ₃
5	3	A ₃	B ₁
6	1	A ₁	B ₁
6	2	A ₃	B ₂
6	2	A ₂	B ₃
6	3	A ₃	B ₂

FIG. 1.3 – Exemple : base de données multidimensionnelle

Items fréquents	(A ₃ ,*)(support 5/6), (A ₃ ,B ₂)(support 4/6), (*,B ₁)(support 4/6), (A ₂ ,*)(support 6/6) (* ,B ₃)(support 5/6), (A ₁ ,*)(support 5/6), (*,B ₂)(support 6/6)
Second niveau	⟨(A ₂ ,*){(A ₃ ,*)}⟩(support 5/6),⟨(A ₂ ,*){(A ₃ ,B ₂)}⟩(support 4/6) ⟨(A ₂ ,*){(*,B ₂)}⟩(support 4/6),⟨(*,B ₃){(A ₃ ,*)}⟩(support 4/6) ⟨(A ₁ ,*){(A ₃ ,B ₂)}⟩(support 4/6),⟨(A ₁ ,*){(*,B ₂)}⟩(support 4/6) ⟨(A ₁ ,*){(A ₃ ,*)}⟩(support 4/6)
Troisième niveau	⟨(A ₁ ,*){(*,B ₂)}{(A ₃ ,B ₂)}⟩(support 4/6)

TAB. 1.1 – Ensemble des séquences fréquentes

1.2.3 Exemple

Un exemple de base de données multidimensionnelle est représenté par la figure 1.3. Dans cet exemple, nous n'avons pas mis de dimension "mesures" mais nous pouvons considérer que toutes mesures sont à 1 ou que la mesure a été partitionnée et qu'elle est représentée par A ou B. Le support d'une séquence est alors le pourcentage de blocs qui respectent cette séquence.

Pour un support minimum de 66% et une confiance minimum à 80%, nous avons extrait les séquences fréquentes représentées dans le tableau 1.1. Les séquences fréquentes sont représentées dans l'arbre 1.4 qui est un arbre de type PSP. Un noeud représente un item et nous avons deux types de branche same (ligne continue) et other (ligne discontinue).

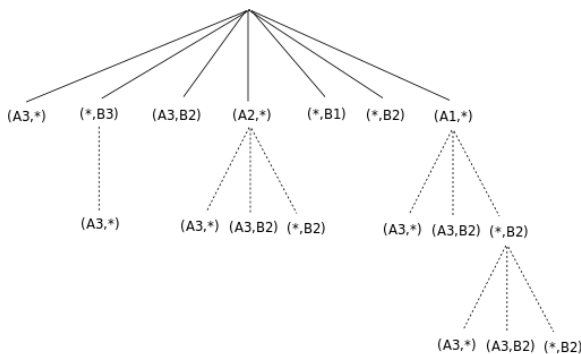


FIG. 1.4 – Arbre des séquences fréquentes

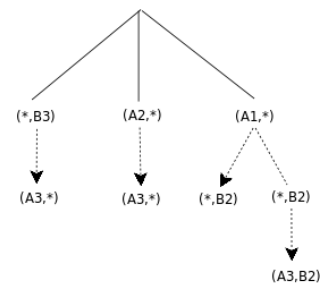


FIG. 1.5 – Arbre des règles séquentielles communes

Pour chaque lien other des séquences maximales de l'arbre 1.4 nous recherchons les règles sé-

quentielles en calculant la confiance. Par exemple la séquence $\langle\{(*, B_3)\} \{(A_3, *)\}\rangle$ est fréquente, la règle $\langle\{(*, B_3)\} \rightarrow \{(A_3, *)\}\rangle$ est alors testée. Sa confiance est de $\frac{\text{support}(\langle\{(*, B_3)\} \{(A_3, *)\}\rangle)}{\text{support}(\langle\{(*, B_3)\}\rangle)} = 80\%$. La règle est donc commune car elle a une confiance supérieure au seuil minConf et elle est insérée dans l'arbre des règles communes avec un support de 83% et une confiance de 80%. Le traitement est ainsi effectué sur l'ensemble des séquences maximales. L'arbre de la figure 1.5 représente les règles séquentielles communes obtenues à la fin du processus.

Les règles séquentielles communes extraites servent à la recherche des règles exceptionnelles. Le processus d'extraction de règles séquentielles passe par deux étapes dont la première est l'ajout d'information à l'antécédent d'une règle commune. Dans le chapitre suivant, nous décrivons cette étape.

Chapitre 2

Extraction de connaissances atypiques

Dans les travaux de [Suz99], les connaissances exceptionnelles sont des règles qui contredisent des connaissances communes. Par exemple, en voiture la ceinture de sécurité sauve des vies mais pour un enfant de moins de douze ans, la ceinture est dangereuse.

La notion de règle contradictoire semble appropriée à la définition d'une règle exceptionnelle, c'est pourquoi nous proposons dans ce rapport une extension des règles contradictoires au contexte multidimensionnel. Nous considérons alors qu'une règle exceptionnelle est une règle commune dont l'ajout d'information plus spécifique dans l'antécédent change la conséquence. Deux notions sont alors abordées : l'ajout d'information plus spécifique d'une part, et la contradiction de séquence d'autre part. Il s'agira alors de formaliser ces deux notions et de définir les algorithmes associés.

Dans le chapitre précédent, nous avons expliqué comment extraire les connaissances communes, ainsi que le type de connaissances considérées. Nous utilisons des règles séquentielles multidimensionnelles étoilées signifiant que l'on ne prend pas en compte certaines informations de la règle : nous conservons en effet des informations plus générales.

Nous proposons de nous orienter vers cet aspect pour l'étape d'ajout d'information. Par exemple considérons les dimensions age, loisir et événements dans une base de données, un item étoilé est (jeune, *, permis B). En spécifiant la dimension loisir nous pouvons obtenir par exemple les items (jeune, Foot, permis B), (jeune, surf, permis B), etc... Pour une séquence, un ou plusieurs items peuvent être spécifiés. Dans ce cas, si le fait d'ajouter une information à un ou plusieurs items étoilés de la séquence entraîne une conséquence différente, nous avons détecté une règle exceptionnelle. Par exemple la règle commune $\langle \{(jeune, *, permis B)\} \{(jeune, *, travail)\} \rangle \rightarrow \langle \{(jeune, *, Achat petite voiture)\} \rangle$ est fréquente cependant si nous spécifions le premier item en prenant uniquement les jeunes qui font du surf cela peut entraîner l'achat d'une grande voiture. Nous obtenons alors la règle exceptionnelle $\langle \{(jeune, surf, permis B)\} \{(jeune, *, travail)\} \rangle \rightarrow \langle \{(jeune, surf, Achat grande voiture)\} \rangle$.

Cette partie est organisée de la manière suivante : dans un premier temps, nous définissons formellement les règles exceptionnelles et décrivons l'algorithme général (section 2). Nous formalisons ensuite l'ajout d'information avec la notion de séquences collables (section 2.2) puis les conséquences contradictoires (section 2.3).

2.1 Extraction de règles exceptionnelles

L'extraction de règles exceptionnelles est effectuée selon le processus présenté par la figure 2.1. Dans l'étape précédente, l'ensemble des règles communes a été généré. Il s'agit maintenant de dé-

couvrir les règles exceptionnelles. Ce processus s'effectue selon deux étapes : ajout d'information à l'antécédent d'une règle commune puis recherche des règles n'impliquant plus la même conséquence.

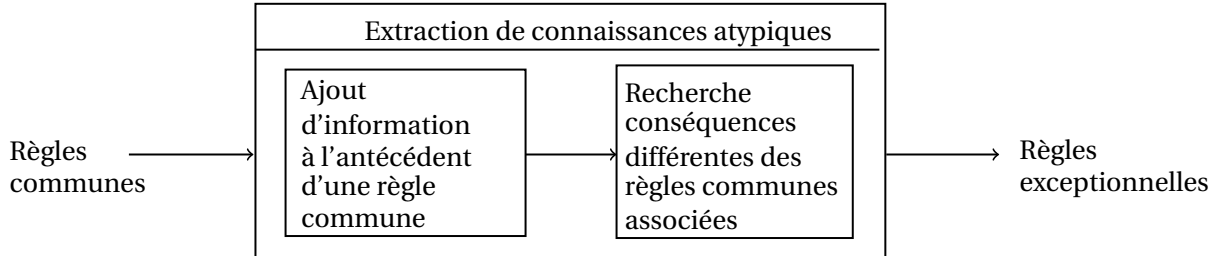


FIG. 2.1 – Détection de règles exceptionnelles

Nous définissons une règle exceptionnelle de la façon suivante :

$RC : S_1 \rightarrow S_c$	(confiance et support hauts)
S_2 collable à S_1	
$RC_o : S_2 \uplus_c S_1 \rightarrow S_c$	(confiance et support faible)
$RE : S_2 \uplus_c S_1 \rightarrow S_{c'}$	(confiance haute et support faible)
$RV : S_1 \rightarrow S_{c'}$	(confiance et support faible)

RC est une règle commune, S_2 est une séquence collable à l'antécédent de la règle RC . La notion de séquences collables, signifiant que S_2 apporte de l'information à S_1 , sera définie dans la section 11.

L'opération \uplus_c représente le collage des séquences dont le résultat donne une séquence plus spécifique que S_1 . RC_o est une règle de condition, RE une règle exceptionnelle, RV une règle de vérification et $S_{c'}$ représente une conséquence différente de S_c .

Pour extraire des règles séquentielles exceptionnelles, l'utilisateur doit fixer quatre seuils afin de déterminer l'importance du support et de la confiance :

1. **minRc** représente le support minimum d'une règle commune. Une règle est fréquente si son support est supérieur à *minsupp*.
2. **minConf** représente la confiance minimum d'une règle. Une règle est intéressante si sa confiance est supérieure à *minconf*. Une confiance forte est une confiance supérieure ou égal à ce seuil et une confiance faible est inférieure à ce seuil.
3. **suppMaxExc** représente le support maximum que peut atteindre une règle exceptionnelle. Ce support est inférieur au support minimal *minsupp*. Une séquence est exceptionnelle si elle a un faible support. Ce seuil permet de vérifier que la règle est assez rare pour être exceptionnelle.
4. **suppBruit** représente le support minimum d'une règle exceptionnelle. La règle doit avoir un support faible mais supérieur à ce seuil afin de vérifier que ça ne soit pas du bruit.

La définition 5 exprime le problème en fonction du support et de façon plus formelle :

Définition 5 Soit la règle commune $S_1 \rightarrow S_c$, les séquences S_2 collables à S_1 sont recherchées telles que :

- $R1 : S_2 \uplus_c S_1 \rightarrow S_c, \text{supp}(R1) < \text{suppMaxExc}, \text{conf}(R1) < \text{minconf}$
- $\exists S_{c'} \neq S_c$ tels que $R2 : S_2 \uplus_c S_1 \rightarrow S_{c'}, \text{suppBruit} < \text{supp}(R2) \leq \text{suppMaxExc}$ et $\text{conf}(R2) > \text{minconf}$

– $R3 : S_1 \rightarrow S_c', \text{conf}(R3) < \text{minConf}$ et $\text{supp}(R3) < \text{suppMaxExc}$

Les règles communes ou exceptionnelles se situent en fonction des quatre supports selon le schéma suivant de la figure 2.2. En effet, on remarque que la fréquence d'une règle exceptionnelle se situe au delà d'un certain bruit et au dessous d'un seuil où les règles sont considérées comme communes.

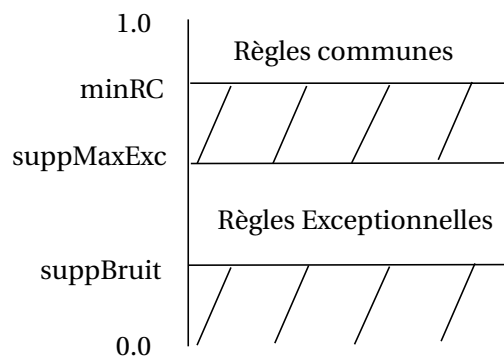


FIG. 2.2 – Règles exceptionnelles et communes en fonction des supports

Si une séquence S_2 collée à la séquence précédente ne vérifie pas la règle de condition RC_0 , car son support et sa fréquence sont faibles alors les règles dérivées ne sont pas exceptionnelles. La condition est en fait une règle permettant de vérifier que l'implication n'est plus vraie si l'on ajoute de l'information à S_1 .

La recherche d'une règle exceptionnelle passe ensuite par la recherche d'une conséquence notée S_c' , différente de toutes les conséquences qu'impliquent l'antécédent de la règle commune. Cela revient à vérifier la règle RV ce qui signifie vérifier que son support et sa confiance sont faibles. La règle RV a alors un support inférieur à suppMaxExc et une confiance inférieure à minconf .

Afin de mieux comprendre la suite de ce rapport, nous rappelons la définition de l'inclusion entre séquences multidimensionnelles étoilées présentée dans [PCL⁺05].

Définition 6 Une séquence multidimensionnelle $s = \langle a_1 \dots a_l \rangle$ est appelée sous-séquence d'une séquence $s' = \langle b_1 \dots b_{l'} \rangle$ s'il existe des entiers $1 \leq j_1 \leq \dots \leq j_l \leq l'$ tels que $a_i \subseteq b_{j_i}$.

Par exemple, si nous avons la séquence $S1 = \langle \{(a_1, b_1, *)\} \{(a_2, *, *)\} \{(a_3, b_1, *)\} \{(a_2, *, *)\} \rangle$, les séquences suivantes sont incluses dans $S1$

- $S2 = \langle \{(a_1, b_1, *)\} \{(a_2, *, *)\} \rangle$
- $S2 = \langle \{(a_1, b_1, *)\} \{(a_2, *, c_2)\} \{(a_3, b_1, c_2)\} \rangle$
- $S2 = \langle \{(a_2, b_2, *)\} \rangle$

2.2 Collage des séquences

L'ajout d'information à l'antécédent d'une règle se traduit par la spécification d'une ou plusieurs dimensions de cette séquence. Afin de répondre à ce besoin, la notion de séquences collables a été définie. Dans notre contexte, nous avons définis le fait de coller une séquence à une autre revient à spécifier cette dernière.

2.2.1 Définition

Une séquence S_2 collable à une séquence S_1 est une séquence qui apporte une information pour certaines dimensions étoilées de la séquence S_1 et qui est incluse dans celle-ci. Le collage entre deux séquences peut être vu comme un processus de spécification d'une séquence.

Par soucis de clarté, nous définissons le problème par étape, c'est-à-dire en premier nous abordons le problème pour un item puis pour un itemset, et enfin pour une séquence. Pour l'antécédent d'une règle commune, plusieurs séquences sont collables. L'opérateur \uplus_c représente l'opération "collée à" ou "spécifiée". $S_2 \uplus_c S_1$ signifie alors que S_2 est collé à S_1 , le résultat est une séquence S_3 plus spécifique que la séquence S_1 .

La notion d'items collables représente une spécification possible des dimensions, c'est-à-dire qu'un item i_1 est collable à un second item i_2 si il est plus spécifique que l'item i_2 .

Définition 7 Soit i_1 un item multidimensionnel étoilé et i_2 un item multidimensionnel (étoilé ou non), i_2 est collable à i_1 si $i_2 \subset i_1$.

Cette définition signifie que i_2 est inclus dans i_1 mais i_2 n'est pas équivalent à i_1 . Cette définition n'est pas commutative. Si nous avons un item i_2 collable à i_1 alors i_1 n'est pas collable à i_2 . L'opération "collé à" pour deux items i_2 et i_1 consiste à ajouter les dimensions de i_2 qui ne sont pas dans i_1 . Elle est alors définie de la façon suivante :

Définition 8 Soient deux items i_2 et i_1 tels que i_2 est collable à i_1 , $i_r = i_2 \uplus_c i_1$ signifie $\forall i_{1k} \in i_1$, si $i_{1k} = *$ et $i_{2k} \neq *$ alors $i_{rk} = i_{2k}$ sinon $i_{rk} = i_{1k}$.

L'item i_r est en fait identique à i_2 car pour toutes les dimensions de i_1 , i_2 a soit la même valeur soit une valeur plus spécifique.

Par exemple :

- $i_1 = (a_1, *, \text{Peu})$ et $i_2 = (a_1, b_1, \text{Peu})$. i_2 est collable à i_1 , mais pas l'inverse. $i_2 \uplus i_1 = (a_1, b_1, \text{Peu})$.
- $i_1 = (a_1, *, \text{Peu})$ et $i_2 = (a_1, b_1, *)$. i_2 n'est pas collable à i_1 .
- $i_1 = (*, *, \text{Peu})$ et $i_2 = (a_2, b_1, \text{Peu})$. i_2 est collable à i_1 . $i_2 \uplus i_1 = (a_2, b_1, \text{Peu})$

La spécification d'un item signifie le remplacement de cet item par celui que l'on colle. L'étape suivante consiste à définir la notion d'*itemsets collables* que nous posons au travers de la définition 9.

Définition 9 Soient IS_1 et IS_2 deux itemsets multidimensionnels tels que $|IS_2| \leq |IS_1|$, IS_2 est collable à IS_1 si $IS_1 \neq IS_2$ et s'il existe une permutation ρ des items dans IS_1 tel que : $\forall i \in IS_2, IS_2[i] = \rho(IS_1)[i]$ ou $IS_2[i]$ est collable à $\rho(IS_1)[i]$

Cette définition signifie qu'il existe un ordre pour les items de IS_1 pour lequel tous les items de IS_2 sont égaux ou collables aux items de IS_1 (un item ne peut pas être comparé à plusieurs items). De plus, pour que IS_2 soit collable à IS_1 , il faut qu'au moins un item de IS_2 soit collable à un item de IS_1 selon la même permutation de IS_1 .

L'opération de spécification entre itemsets est alors exprimée de la façon suivante :

Définition 10 Deux itemsets IS_2 et IS_1 , $\rho(IS_1)$ est une permutation des items de IS_1 tels que IS_2 est collable à IS_1 , $IS_r = IS_2 \uplus_c \rho(IS_1)$ signifie $\forall IS_{2k}$ collable à $\rho(IS_1)_j$, $k \leq j$, $IS_{rj} = IS_{2k}$ sinon $IS_{rj} = \rho(IS_1)_j$.

Comme nous l'avons vu précédemment, l'itemset IS_2 est collable à IS_1 selon une certaine permutation de IS_1 , nous avons alors ici la même permutation à faire afin de pouvoir coller les itemsets.

Par exemple, nous recherchons les séquences collables à l'itemset $IS_1 = \{(a_1, *, \text{Peu}) (a_2, *, *)\}$:

- Si $IS_2 = \{(a_1, b_1, \text{Peu})\}$ alors IS_2 est collable à IS_1 . IS_2 est inclus dans IS_1 et que l'on apporte une information sur le premier item (b_1 sur la seconde dimension). $IS_2 \cup IS_1 = \{(a_1, b_1, \text{Peu}) (a_2, *, *)\}$.
- Si $IS_2 = \{(a_2, *, *) (a_1, b_1, \text{Peu})\}$ alors IS_2 est collable à IS_1 . Le premier item de IS_2 est collable au second item de IS_1 et le second item de IS_2 est égal au premier de IS_1 . $IS_2 \cup IS_1 = \{(a_2, *, *) (a_1, b_1, \text{Peu})\}$.
- Si $IS_2 = \{(a_1, b_1, *) (a_2, b_1, *)\}$ alors IS_2 n'est pas collable à IS_1 car le premier item de IS_2 n'est pas inclus dans IS_1 .
- Si $IS_2 = \{(a_1, *, \text{Peu})\}$ alors IS_2 n'est pas collable à IS_1 car IS_2 n'apporte aucune information à IS_1 .

Avec ces définitions, nous pouvons maintenant définir la notion de séquences collables (S_2 est collable à S_1).

Définition 11 Soit $S_1 = \langle IS_1 \dots IS_n \rangle$ une séquence multidimensionnelle étoilée, $S_2 = \langle IS_1 \dots IS_k \rangle$ une séquence multidimensionnelle (étoilée ou non) tel que $k \leq n$ et $S_1 \neq S_2$, $S[i]$ représente le $i^{\text{ème}}$ itemset de la séquence S . S_2 est collable à S_1 si $\forall S_1[i] \ i \in [1, n], \exists j_1 < \dots < j_i < \dots < j_n \leq k \mid S_1[j_i] = S_2[i]$ ou $S_1[i]$ est collable à $S_2[j_i]$ avec au moins un item collable.

Une séquence S_2 collable à une séquence S_1 est une séquence incluse dans S_1 et S_2 a au moins un itemset collable à un itemset de S_1 .

Pour finir, nous définissons l'opération "collée à" entre deux séquences :

Définition 12 Deux séquences S_2 et S_1 tels que S_2 est collable à S_1 , $S_r = S_2 \cup_c S_1$ signifie $\forall S_2_k$ collable à $S_1_j, k \leq j, S_r_j = S_2_j \cup_c S_1_k$ sinon $S_r_j = S_1_j$.

Par exemple, si nous avons la séquence $S_1 = \{(a_1, *, \text{Peu}) (a_2, b_1, \text{Peu})\} \{(\text{Pull}, \text{agé}, *)\}$ alors :

- Si $S_2 = \{(a_1, \mathbf{b}_1, \text{Peu}) (a_2, b_1, \text{Peu})\} \{(\text{Pull}, \text{agé}, *)\}$ alors S_2 est collable à S_1 . Le premier itemset de S_2 est collable au premier itemset de S_1 et le second est équivalent. $S_2 \cup S_1 = \{(a_1, \mathbf{b}_1, \text{Peu}) (a_2, b_1, \text{Peu})\} \{(\text{Pull}, \text{agé}, \mathbf{Peu})\}$
- Si $S_2 = \{(a_1, b_1, \text{Peu})\} \{(\text{Pull}, \text{agé}, \mathbf{Peu})\}$ alors S_2 est collable à S_1 . Les deux itemsets de S_2 sont collables aux deux itemsets de S_1 (dans l'ordre). $S_2 \cup S_1 = \{(a_1, b_1, \text{Peu}) (a_2, b_1, \text{Peu})\} \{(\text{Pull}, \text{agé}, *)\}$.
- Si $S_2 = \{(a_1, *, \text{Peu}) (a_2, b_1, \text{Peu})\}$ n'est pas collable à S_1 car $S_2 \subset S_1$ mais S_2 n'a aucun itemset collable à un itemset de S_1 .

Coller deux séquences peut entraîner plusieurs résultats. Par exemple, si $S_1 = \langle \{(a_1, *, c_1)\} \{(a_1, b_1, *)\} \rangle$ et $S_2 = \langle \{(a_1, b_1, c_1)\} \rangle$ alors S_2 est collable à S_1 . Un ajout d'information peut être fait sur le premier itemset de S_1 ou sur le second. Nous obtenons alors deux séquences possibles $\langle \{(a_1, b_1, c_1)\} \{(a_1, b_1, *)\} \rangle$ ou $\langle \{(a_1, *, c_1)\} \{(a_1, b_1, c_1)\} \rangle$.

Dans la définition 12, S_r est un ensemble de séquences. Nous présentons alors la méthode pour découvrir des séquences collées à partir d'une séquence donnée.

2.2.2 Algorithme RechercheSeqCol

Pour chaque item de la séquence, nous cherchons l'ensemble des items collables. Nous proposons un algorithme récursif afin de tester toutes les combinaisons.

Il s'agit dans un premier temps de coller un item au premier item de la séquence antécédente de la règle, puis de coller un item au second item, etc... Lorsque le support de la séquence est trop faible (inférieur *suppBruit*) le collage s'arrête. En effet, la séquence a un support inférieur ou égal à une séquence plus générale (donc inférieur *suppBruit*).

Dans l'algorithme 2, la fonction RechercheItCollable (ligne 8) permet de récupérer pour un item passé en paramètre les items fréquents collables à celui-ci. Ensuite pour chaque item collable, si le collage de la séquence implique un support inférieur à *suppBruit*, alors on ne colle plus d'items à cette séquence. Si le support est supérieur à *suppBruit* la séquence collée est enregistrée et un lien vers la séquence plus générale de la règle commune de départ est ajouté via la fonction addLienRC. Dans cas, le processus de collage est réitéré pour l'item suivant de la séquence. Les lignes 27 à 29 permettent de coller un item au $k + 1^{ème}$ item sans que le précédent soit spécifié.

Par exemple, avec la séquence suivante $\langle\{(A_1, *)\}(A_2, *)\}\{(*, B_1)\}\rangle$, pour chaque item les items collables sont recherchées :

- Pour l'item $(A_1, *)$ il y a deux items fréquents collables : (A_1, B_1) , (A_1, B_2)
- Pour l'item $(A_2, *)$ il y a uniquement l'item fréquent collable : (A_2, B_2)
- Pour l'item $(*, B_1)$ il y a deux items fréquents collables : (A_1, B_1) , (A_3, B_1)

Un item au premier item de la séquence sera donc collé :

- $\langle\{(A_1, B_1)\}(A_2, *)\}\{(*, B_1)\}\rangle$ (support de 70%), la séquence a un support supérieur à *suppBruit*, alors le collage continue
- $\langle\{(A_1, B_2)\}(A_2, *)\}\{(*, B_1)\}\rangle$ (support de 5%), la séquence a un support inférieur à *suppBruit*, alors le collage s'arrête et cette séquence n'est pas un antécédent de règle possible (donc pas enregistré dans le résultat).

Un item sur le deuxième items de la séquence est ensuite collé :

- $\langle\{(A_1, B_1)\}(A_2, B_2)\}\{(*, B_1)\}\rangle$ (support de 50%).
- $\langle\{(A_1, *)\}(A_2, *)\}\{(*, B_1)\}\rangle$ (support de 72%) sans avoir collé d'item au premier item

Nous avons uniquement deux séquences car nous prenons les séquences qui ont déjà un item collé et un support supérieur à *suppBruit*. Nous cherchons ensuite à spécifier le dernier item de la séquence $\langle\{(A_1, *)\}(A_2, *)\}\{(*, B_1)\}\rangle$ (en prenant en compte tous les cas de collages précédent ou sans avoir collé d'item précédemment).

2.2.3 Exemple

Reprenons l'exemple 1.3 : après avoir découvert les règles séquentielles communes, nous recherchons pour chaque antécédent des séquences plus spécifiques.

Nous devons avoir à notre disposition les items qui ont un support supérieur à *suppBruit* sans forcément être fréquents (inférieur à *minSupp*). Ces items peuvent être calculés et enregistrés au moment de la génération des séquences fréquentes sans être utilisées pour ce processus. Nous avons alors l'ensemble des items suivants :

Nous avons préalablement extrait les quatre règles communes :

1. $\langle\{(*, B_3)\}\rangle \rightarrow \langle\{(A_3, *)\}\rangle$
2. $\langle\{(A_2, *)\}\rangle \rightarrow \langle\{(A_3, *)\}\rangle$
3. $\langle\{(A_1, *)\}\rangle \rightarrow \langle\{(*, B_2)\}\rangle$

Algorithme 2 : RechercheSeqCol : recherche des séquences collées pour une séquence

```

Entrées :  $k$  /* Position de l'item auquel on souhaite ajouter une information
*/
1     ARC /* Arbre des règles communes */
2     ARE /* Arbre des règles exceptionnelles */
3     suppBruit, suppMaxExc, minConf /* Seuils */
4     Seqcol /* Séquence de départ ou séquence avec des informations déjà
collées */
5     Rc /* Règle commune */
6     ESC /* Ensemble des séquences collées */
7 début
    /* Eic : ensemble des items collables au  $k^{ieme}$  item de seqCol */
8     Eic ← RechercheItCollable(Rc.getItem( $k$ ), ARC)
9     pour chaque  $ic \in Eic$  faire
    /* On colle  $ic$  à la séquence seqCol en remplaçant le  $k^{ieme}$  item par  $ic$ 
*/
10    seqColle ← remplace(seqCol,  $ic$ ,  $k$ )
11    RegleCol = seqColle & RC.getConseq() /* Concatène la séquence collée et la
conséquence */
12    si (support(seqColle) < suppBruit) alors
13    |   retourner Esc
14    sinon si ARE.have(seqColle) alors
    /* AddlienRC est le lien vers le noeud de la règle commune
associée */
15    |   ARE.getNode(seqColle).addLienRC(RC.getNodeAnt())
16    sinon
17    |   Esc.add(seqColle) /* Ajout dans la liste des séquences collées */
18    |   ARE.add(seqColle) /* Ajout de la séquence antécédente dans l'arbre
ARE */
19    |   ARE.getNode(seqColle).addLienRC(RC.getNodeAnt())
20    fin
21    si  $k < seqCol.getnbItem()$  alors
22    |   Esc ← RechercheSeqCol( $k+1$ , ARC, ARE, suppBruit, suppMaxExc, minConf,
seqColle, RC)
23    sinon
24    |   retourner Esc
25    fin
26  fin
27  si  $k < seqCol.getnbItem()$  alors
28  |   Esc ← RechercheSeqCol( $k+1$ , ARC, ARE, suppBruit, suppMaxExc, minConf,
seqCol, RC)
29  fin
30  retourner Esc
31 fin

```

Item	Support	Item	Support
$(A_3, *)$	5/6	$(*, B_2)$	6/6
(A_3, B_2)	4/6	(A_1, B_1)	2/6
$(*, B_1)$	4/6	(A_1, B_2)	2/6
$(A_2, *)$	6/6	(A_1, B_3)	2/6
$(*, B_3)$	5/6	(A_2, B_2)	3/6
$(A_1, *)$	5/6	(A_2, B_3)	3/6

TAB. 2.1 – Ensemble des items qui ont un support supérieur à $minBruit$

$$4. \langle (A_1, *) \rangle \{ (*, B_2) \} \rightarrow \langle (A_3, B_2) \rangle$$

Pour chaque séquence antécédente des règles, nous recherchons les items collables au premier item de la séquence puis au second etc...

Prenons la première règle : $\langle (*, B_3) \rangle \rightarrow \langle (A_3, *) \rangle$, nous recherchons des items collables à la séquence antécédente. Nous calculons le support des séquences collées, cela donne deux séquences antécédentes différentes $\langle (A_1, B_3) \rangle$ (support 2/6) ou $\langle (A_2, B_3) \rangle$ (support 3/6)

Comme ces deux séquences ont un support supérieur au support $suppBruit$, elles sont insérées dans l'arbre des règles exceptionnelles. De plus nous ajoutons un lien vers l'antécédent de la règle commune à laquelle elles sont associées.

Nous effectuons ce traitement sur toutes les séquences antécédentes. Pour une séquence plus longue, nous allons tenter de coller des items sur un ou plusieurs items de la séquence. Si nous prenons la dernière règle, et que nous recherchons les séquences antécédentes, nous effectuons le traitement suivant :

1. Sur le premier item, nous collons l'item (A_1, B_1) : $\langle (A_1, B_1) \rangle \{ (*, B_2) \}$ (support 2/6)
2. Nous collons ensuite l'item (A_1, B_2) au second item : $\langle (A_1, B_1) \rangle \{ (A_1, B_2) \}$ (support 0/6), cette séquence a un support inférieur au $suppBruit$, elle est élaguée et n'apparaît donc pas dans l'arbre des règles exceptionnelles.
3. Nous collons ensuite l'item (A_2, B_2) au second item $\langle (A_1, B_1) \rangle \{ (A_2, B_2) \}$ (support 1/6) élagué
4. Nous collons ensuite l'item (A_3, B_2) au second item $\langle (A_1, B_1) \rangle \{ (A_3, B_2) \}$ (support 2/6)
5. Sur le premier item, nous collons l'item (A_1, B_2) : $\langle (A_1, B_2) \rangle \{ (*, B_2) \}$ (support 2/6)
6. etc...

Nous insérons alors dans l'arbre des règles exceptionnelles les séquences spécifiées qui représente **les antécédents des règles exceptionnelles**. La figure 2.3 illustre l'ensemble des antécédents extraits pour l'exemple 1.3.

Nous avons obtenu à la fin du processus un ensemble d'antécédents spécifiés. Nous recherchons dans la section suivante les conséquences des règles exceptionnelles.

2.3 Recherche de conséquences différentes

Rechercher une règle exceptionnelle passe par une phase de recherche d'une conséquence différente. Dans la section précédente, nous avons défini l'ajout d'information pour l'antécédent d'une

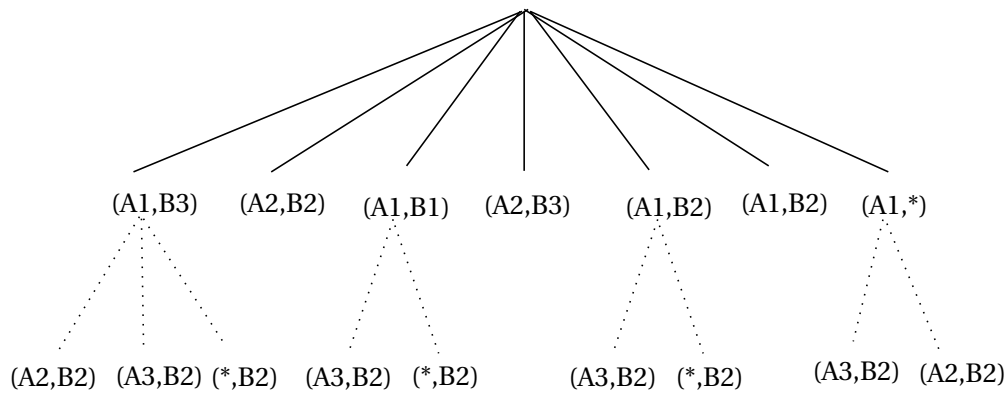


FIG. 2.3 – Arbre contenant les antécédents des règles exceptionnelles

règle. Nous nous concentrons donc dans cette section à la définition d’une conséquence différente.

2.3.1 Définition

Nous définissons dans un premier temps la différence entre deux items, puis entre deux itemsets, avant de déterminer sur la différence entre deux séquences multidimensionnelles.

Nous avons dans la conséquence un item e_c et nous souhaitons trouver un item différent noté e_d . e_{c_k} représente la $k^{ième}$ valeur de l’item e_c . Un item est différent d’un autre s’il respecte les conditions de la définition suivante :

Définition 13 $e_c \neq e_d$ si $e_c \not\subseteq e_d$ et $e_d \not\subseteq e_c$.

Cela signifie que deux items sont différents si ils sont incomparables. Par exemple :

- $e_c=(a1, \mathbf{b1}, c1)$ et $e_d=(a1, \mathbf{b2}, c1)$ sont différents.
- $e_c=(a1, b1, c1)$ et $e_d=(a1, b1, *)$ ne sont pas différents.
- $e_c=(a1, *, c1)$ et $e_d=(a1, b1, *)$ sont différents.

Un itemset est différent d’un second itemset à partir du moment où l’un n’est pas inclus dans l’autre est inversement. La notion de différence entre itemsets est alors exprimée de la façon suivante :

Définition 14 Soit IS_1 et IS_2 deux itemsets, $IS_1 \neq IS_2$ si $IS_1 \not\subseteq IS_2$ et $IS_2 \not\subseteq IS_1$.

Par exemple prenons l’itemset IS_1 suivant $\{(a1, b1, *) (a2, *, c2) (*, b2, c3) (a2, b3, c1)\}$:

- $IS_2 = \{(a1, b1, c1) (a2, *, c2) (*, b2, c3)\}$ n’est pas différent de IS_1 puisque $IS_2 \subseteq IS_1$
- $IS_2 = \{(a1, b1, *) (a2, *, c3)\}$ est différent de IS_1 car le second item n’est inclus dans aucun item

Nous étendons cette notion à la différence entre deux séquences :

Définition 15 Soit S_1 et S_2 deux itemsets, $S_1 \neq S_2$ si $S_1 \not\subseteq S_2$ et $S_2 \not\subseteq S_1$.

Prenons, par exemple, la séquence $S_1 = \{(a1, b1, *) \} \{(a2, *, c2) (*, b2, c3)\} \{(a2, b3, c1)\}$:

- $S_2 = \{(a1, b1, *) \} \{(a2, *, c2) (a1, b2, c3)\}$ n’est pas différent de S_1 car S_2 est encore inclu dans S_1 .
- $S_2 = \{(a1, b1, *) \} \{(a3, *, c3)\}$ est différent de S_1 car l’item $(a3, *, c3)$ dans S_2 n’est pas présent dans S_1 .

2.3.2 Algorithme RechercherConseqExc

L'algorithme de recherche de conséquences différentes s'effectue niveau par niveau. Nous générons les conséquences d'un item pour lequel le support de la règle ne serait pas du bruit (supérieur à *suppBruit*), puis nous recherchons ensuite une conséquence de deux items. Après cette étape, nous élaguons les règles qui ont un support trop faible. Après cet étape, toutes les conséquences générées nous vérifions que chaque règle "maximale" a une confiance inférieure au *suppMaxExc*, et ces conséquences ne sont pas incluses et n'incluent pas la conséquence de la règle commune à laquelle la règle exceptionnelle se rapporte.

Algorithme 3 : RechercherConseqExc : Recherche les conséquences pour trouver les règles exceptionnelles

```

Entrées : ARC                                     /* Arbre des règles communes */
1         ARE                                       /* Arbre des règles exceptionnelles */
2         suppBruit, suppMaxExc, minConf           /* Seuils */
3 début
   /* Vérifie la règles de condition                */
4 VerifRegleCond(); /* Génère les règles candidates avec une conséquence d'un
   item                                             */
5 Ecand←getItemConseq(ARE)
   /* Calcul du support des règles et élagation selon suppBruit et
   minConf                                       */
6 Elagage(Ecand, ARE, suppBruit, suppMaxExc, minConf)

   /* Génère les règles candidates avec une conséquence de deux items selon
   les principes de VPSP                          */
7 Ecand←getDeuxItemsConseq(ARE)
8 Elagage(Ecand, ARE, suppBruit, suppMaxExc, minConf)

   /* Générer les règles candidates avec une conséquence avec plus de deux
   items                                           */
9 tant que Ecand ≠ ∅ faire
10   | Ecand←genererConseq(ARE)
11   | Elagage(Ecand, ARE, suppBruit, suppMaxExc, minConf)
12 fin
   /* Elimination les conséquences incluses ou qui incluent la conséquence
   de la règle commune associée et élimine si le support de la règle >
   suppMaxExc                                    */ ElaguerRE(ARE)
13 fin

```

L'algorithme 3 est de type générer élaguer, il permet de trouver des conséquences pour toutes les séquences collées enregistrées dans l'arbre des règles exceptionnelles. Lorsque nous exécutons cette fonction, nous avons préalablement enregistré tous les antécédents des règles. La fonction VerifRegleCond permet de vérifier la règle de condition. Nous vérifions alors que l'antécédent n'implique plus la conséquence de la règle commune associée. Si cela est vérifié, nous générons les conséquences possibles. Nous avons conservé tous les items fréquents dans l'arbre des règles communes. Pour générer les conséquences d'une règle, nous nous servons d'abord (ligne 5 et 6) des items fréquents pour générer des conséquences de longueur 1 (un seul item).

Ensuite pour générer les conséquences candidates de longueur 2, nous utilisons les conséquences découvertes au niveau précédent (conséquences d'un item). Par exemple si une séquence $\langle\{a_1, b_1\}\{a_2, *\}\rangle$ implique la séquence $\langle\{a_1, b_2\}\rangle$ et $\langle\{a_2, b_2\}\rangle$ alors nous allons chercher si $\langle\{a_1, b_1\}\{a_2, *\}\rangle$ implique : $\langle\{a_1, b_2\}\{a_2, b_2\}\rangle$, $\langle\{a_2, b_2\}\{a_1, b_2\}\rangle$ et/ou $\langle\{a_1, b_2\} (a_2, b_2)\rangle$.

Pour finir nous générons les candidats à partir des conséquences de longueur 1 et k-1 les conséquences de longueur k. Par exemple si $\langle\{a_1, b_1\}\{a_2, *\}\rangle$ implique $\langle\{a_1, b_2\} (a_2, b_2)\rangle$ et l'item (a_1, b_2) alors ça implique peut être : $\langle\{a_1, b_2\}\{a_1, b_2\} (a_2, b_2)\rangle$ ou $\langle\{a_1, b_2\} (a_2, b_2)\{a_1, b_2\}\rangle$

La fonction *élaguerRE* vérifie sur l'ensemble des règles exceptionnelles trouvées que l'on supprime toutes les règles exceptionnelles ayant une conséquence incluse ou incluant la conséquence de la règle commune associée. Nous vérifions donc que le support de la règle soit compris entre les seuils *suppMaxExc* et *suppBruit*.

Nous avons par la suite extrait toutes les règles exceptionnelles qui sont contenues dans l'arbre *ARE*. Dans notre définition, une règle est exceptionnelle en fonction d'une règle commune. Les résultats sont écrits sous la forme (règle commune, règle séquentielle).

2.3.3 Exemple

Reprenons l'exemple 1.3 où nous avons extrait les antécédents des règles exceptionnelles. Nous recherchons maintenant toutes les conséquences que l'antécédent peut impliquer. Pour cela nous commençons par tester la règle de condition, c'est-à-dire vérifier que l'antécédent n'implique pas la même conséquence que la règle commune à laquelle il est associé. Si la règle est vérifiée, nous recherchons des conséquences d'un item avec les items qui ont un support supérieur à *suppBruit* puis les conséquences de deux items etc.

Par exemple si nous considérons l'antécédent $\langle(A_1, B_3)\rangle$, associé à la règle commune $\langle(*, B_3)\rangle \rightarrow \langle\{A_3, *\}\rangle$ nous commençons par tester la règle $\langle(A_1, B_3)\rangle \rightarrow \langle\{A_3, *\}\rangle$. Celle-ci a un support de 2/3 et un confiance de 100%. La règle de condition n'est pas vérifiée, cette règle n'est pas exceptionnelle et nous arrêtons de rechercher une conséquence différente. Aucune règle ne sera extraite avec cet antécédent.

Si nous prenons la règle $\langle(A_1, *)\{(*, B_2)\}\rangle \rightarrow \langle\{A_3, B_2\}\rangle$. Nous avons plusieurs séquences collées associées. Si nous recherchons une conséquence à la séquence collées $\langle(A_1, B_2)\{(*, B_2)\}\rangle$ nous avons :

Règle	Support	Confiance
$\langle(A_1, B_2)\{(*, B_2)\}\rangle \rightarrow \langle\{A_2, B_1\}\rangle$	2/6	100%
$\langle(A_1, B_2)\{(*, B_2)\}\rangle \rightarrow \langle\{(*, B_1)\}\rangle$	2/6	100%
$\langle(A_1, B_2)\{(*, B_2)\}\rangle \rightarrow \langle\{A_2, *\}\rangle$	2/6	100%

Une fois spécifiée la règle n'implique plus la conséquence $\langle\{A_3, B_2\}\rangle$, nous recherchons alors une conséquence de deux items, nous avons les possibilités suivantes :

Pour cet antécédent, nous avons alors la conséquence maximale et la plus spécifique suivante : $\langle(A_1, B_2)\{(*, B_2)\}\rangle \rightarrow \langle\{A_2, B_1\}\rangle$ (2/6)

Règle	Support
$\langle (A_1, B_2) \{ (*, B_2) \} \rangle \rightarrow \langle \{ (A_2, B_1) \} \{ (A_2, B_1) \} \rangle$	0/6
$\langle (A_1, B_2) \{ (*, B_2) \} \rangle \rightarrow \langle \{ (A_2, B_1) \} \{ (*, B_1) \} \rangle$	0/6
$\langle (A_1, B_2) \{ (*, B_2) \} \rangle \rightarrow \langle \{ (A_2, B_1) \} \{ (A_2, *) \} \rangle$	0/6
$\langle (A_1, B_2) \{ (*, B_2) \} \rangle \rightarrow \langle \{ (*, B_1) \} \{ (A_2, B_1) \} \rangle$	0/6
$\langle (A_1, B_2) \{ (*, B_2) \} \rangle \rightarrow \langle \{ (*, B_1) \} \{ (A_2, *) \} \rangle$	0/6
$\langle (A_1, B_2) \{ (*, B_2) \} \rangle \rightarrow \langle \{ (*, B_1) \} \{ (*, B_1) \} \rangle$	0/6
$\langle (A_1, B_2) \{ (*, B_2) \} \rangle \rightarrow \langle \{ (A_2, *) \} \{ (A_2, B_1) \} \rangle$	0/6
$\langle (A_1, B_2) \{ (*, B_2) \} \rangle \rightarrow \langle \{ (A_2, *) \} \{ (A_2, *) \} \rangle$	0/6
$\langle (A_1, B_2) \{ (*, B_2) \} \rangle \rightarrow \langle \{ (A_2, *) \} \{ (*, B_1) \} \rangle$	0/6

Ce traitement est effectué pour tous les antécédents, nous obtenons à la fin de ce processus une seule règle exceptionnelle : $\langle (A_1, B_2) \{ (*, B_2) \} \rangle \rightarrow \langle \{ (A_2, B_1) \} \rangle$ (2/6) en fonction de la règle commune $\langle (A_1, *) \{ (*, B_2) \} \rangle \rightarrow \langle \{ (A_3, B_2) \} \rangle$.

2.4 Algorithmes généraux

L'algorithme ExtRE (algorithme 4) est l'algorithme principal permettant l'extraction des règles exceptionnelles dans lequel nous séparons alors l'extraction de connaissances communes de l'extraction de connaissances atypiques. La première étape consiste à extraire des règles séquentielles (ligne 6), puis les règles exceptionnelles en fonction des règles communes sont découvertes (ligne 7).

Algorithme 4 : ExtRE : Extraction des règles exceptionnelles

```

Entrées : minConf /* La confiance minimum */
1 minSupp /* Le support minimum */
2 suppMaxExc /* Le support maximum d'une règle exceptionnelle */
3 suppBruit /* Le support minimum d'une règle exceptionnelle */
4 FichierDatabase /* Base de données */
Sorties : ARE /* arbre des règles exceptionnelles */
5 début
   /* Extraire des règles séquentielles */
6 ARC ← GenRegSeq(minSupp, minConf, FichierDatabase)
   /* Extraire des règles exceptionnelles */
7 ARE ← RechercheRegExc(ARC, minConf, suppMaxExc, suppBruit,
   FichierDatabase)
8 retourner ARE
9 fin

```

La fonction RechercheRegExc extrait des règles exceptionnelles en fonction des connaissances communes préalablement extraites. Les règles exceptionnelles sont alors découvertes selon deux phases : dans un premier temps les séquences collables pour chaque antécédent des règles communes sont détectées, dans un second temps une conséquence différente pour chaque séquence antécédente collée est recherchée. L'algorithme général pour détecter des règles exceptionnelles est décrit par l'algorithme RechercheRegleExc (algorithme 5).

L'algorithme 5 prend en entrée l'ensemble des supports nécessaires pour déterminer si une règle

est exceptionnelle, ainsi que l'arbre contenant les règles séquentielles (nommé *ARC*). De la ligne 4 à 7, les séquences antécédentes collables sont recherchées pour chaque règle commune maximale et les séquences collées sont stockées dans l'arbre *ARE*, qui contient alors à la fin de cette étape uniquement les antécédents des règles exceptionnelles ayant un support supérieur à *suppBruit*.

Lorsque toutes les séquences spécifiées sont dans *ARE*, les conséquences possibles sont recherchées (ligne 8) et ajoutées dans l'arbre *ARE*. Dans le même temps, il est vérifié pour chaque règle exceptionnelle que la conséquence de la règle commune associée est incomparable à la conséquence trouvée. Deux séquences S_1 et S_2 sont incomparables si $S_1 \not\subseteq S_2$ et $S_2 \not\subseteq S_1$.

Algorithme 5 : RechercheRegExc : Extraction des règles exceptionnelles

```

Entrées : minConf, suppMaxExc, suppBruit /* une confiance et deux supports */
1      ARC                               /* Arbre des règles communes */
2      fichierRes                         /* Fichier contenant les résultats */
Sorties : ARE                           /* Arbre des règles exceptionnelles */
3 début
4   | ARE ← new arbre()
5   | pour chaque Rc ∈ RegleMaximum(ARE) faire
6   | | /* Esc : ensemble des séquences collées */
6   | | Esc ← RechercheSeqCol(1, ARC, ARE, suppBruit, suppMaxExc, minConf,
6   | | RC.getAntécédent(), RC)
7   | fin
8   | RechercheConseqExc(ARC, ARE)
9   | AfficherResultats(fichierRes, ARE);
10 fin

```

Troisième partie

Expérimentations

3	Implémentation	48
3.1	Démarche générale	48
3.2	Partitionnement	49
3.3	Extraction des motifs séquentiels	49
3.4	Génération des règles séquentielles	50
3.5	Collage des séquences et recherche des conséquences différentes	50
4	Expérimentation sur jeu de données réelles	51
4.1	Jeux de données	51
4.2	Prétraitement	51
4.3	Comparaison des résultats	52
4.3.1	Comparaison des techniques de partitionnement	52
4.3.2	Comparaison avec ratio, sans ratio	53
	Conclusion et perspectives	54
A	Connaissances communes	58
A.1	La mesure dans la dimension d'analyse	58
A.1.1	Cas général	58
A.1.2	Motifs séquentiels étoilés	60
B	Table des notations	61

Chapitre 3

Implémentation

3.1 Démarche générale

La figure 3.1 illustre le schéma général pour l'implémentation :

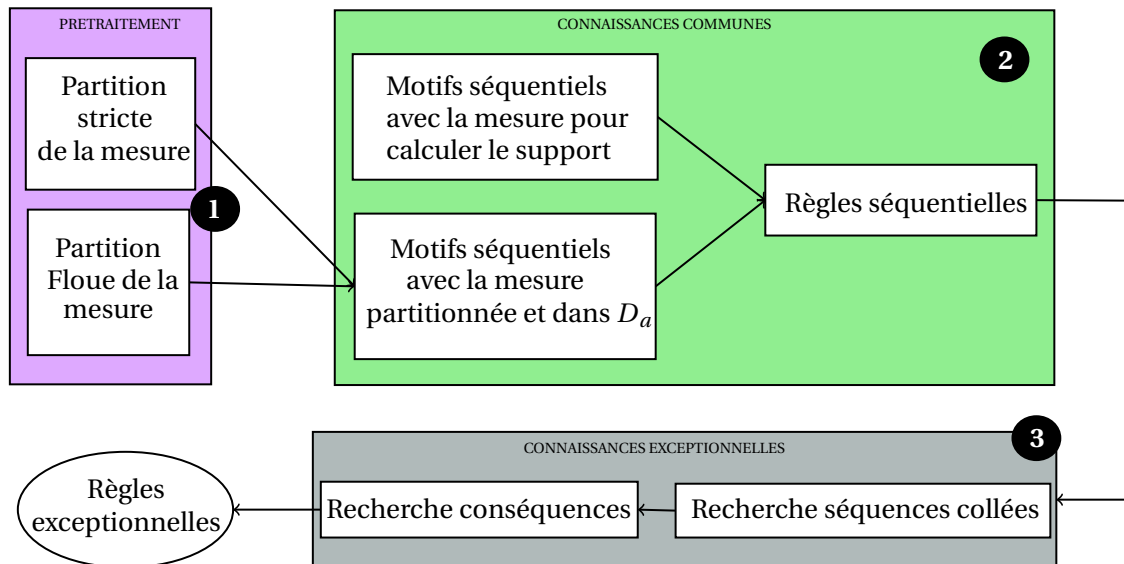


FIG. 3.1 – Protocole d'extraction de règles exceptionnelles

Notre démarche peut se décomposer en trois étapes. Dans un premier temps, il est nécessaire de prétraiter les données : il s'agit ici de partitionner la mesure. Nous avons décrit à la section 1.1 du chapitre 1 deux méthodes de partitionnement : stricte et flou. Cette étape est primordiale si nous souhaitons obtenir des motifs de qualité.

La seconde étape consiste à extraire les règles séquentielles, ce qui est réalisé lors du processus de fouille de données, après la découverte des motifs séquentiels. L'implémentation est décrite en détail à la section 3.4.

Nous recherchons ensuite des règles contredisant les règles communes. Cela passe par la recherche de séquences collables, puis par la découverte de conséquences différentes. Le détail de cette étape est décrit à la section 2. Ce processus aboutit à l'extraction de règles exceptionnelles.

3.2 Partitionnement

Nous avons implémenté et testé le partitionnement flou triangulaire et le partitionnement flou trapézoïdal. La première technique consiste à assigner une valeur d'appartenance à 1 uniquement pour les valeurs correspondant aux bornes des intervalles choisis. Par exemple, les trois intervalles $[11 - 20]$, $[21 - 30]$ et $[31 - 40]$. Seules les valeurs 11, 21, 31 et 40 ont un degré d'appartenance de 1 pour les sous-ensembles 1, 2, 3 et 4 respectivement. Les valeurs entre 21 et 29 ont un degré d'appartenance différents de 1 et de 0 pour les deux sous-ensembles 2 et 3.

Le partitionnement trapézoïdale consiste à prendre un intervalle flou sur deux. Par exemple, pour trois intervalles de $[11 - 20]$, $[21 - 30]$ et $[31 - 40]$ alors toutes les valeurs entre 11 et 20 appartiennent au premier sous-ensemble, et toutes les valeurs entre 31 et 41 appartiennent au second sous-ensemble. Les valeurs comprises entre 21 et 30 appartiennent avec un certain degré aux deux sous-ensembles (par exemple 25 appartient à 0.5 au premier sous ensemble et 0.5 au second). Les figures 4.2 et 4.3 du chapitre suivant illustre ces différents partitionnement.

Ces deux techniques de partitionnement sont dites floues, car elles assignent un degrés d'appartenance d'une valeur à un ou deux sous ensembles. Nous avons également testé le partitionnement binaire, qui consiste à affecter 0 ou 1 à une valeur, en fonction de si elle appartient à un intervalle ou non. Par exemple, pour les intervalles $[11 - 20]$, $[21 - 30]$ et $[31 - 40]$, la valeur 25 sera remplacée par les attributs 0 1 0.

3.3 Extraction des motifs séquentiels

Afin d'extraire des motifs séquentiels pertinents et de prendre en compte l'étoile, nous avons utilisé les motifs convergent proposés dans [PLT07]. Le principe est d'extraire des motifs qui sont au cour du temps de plus en plus précis. Dans notre contexte nous obtenons des séquences composées de beaucoup d'items étoilés au départ et d'items de plus en plus spécifiques vers la fin de la séquence. En effet si l'ajout d'un item très spécifique à une séquence s ne la rend pas fréquente, nous devons vérifier si avec un item plus général (l'étoile) elle est fréquente. Par exemple, la séquence $s = \langle \{(A_1, B_2)\} \{(A_2, B_2)\} \rangle$ est fréquente, si la séquence $s_2 = \langle \{(A_1, B_2)\} \{(A_2, B_2) (A_1, B_1)\} \rangle$ n'est pas fréquente, nous testons les séquences $\langle \{(A_1, B_2)\} \{(A_2, B_2) (A_1, *)\} \rangle$ et $\langle \{(A_1, B_2)\} \{(A_2, B_2) (*, B_1)\} \rangle$. Pour rechercher des séquences convergentes, il suffit alors d'extraire les séquences divergentes puis de retourner les séquences. Par exemple $\langle \{(A_1, B_2)\} \{(A_2, B_2) (A_1, *)\} \rangle$ donne la séquence $\langle \{(A_1, *) (A_2, B_2)\} \{(A_1, B_2)\} \rangle$.

Pour cela, nous avons utilisé une version de l'algorithme de type VPSP [DJJK⁺06] adaptée au contexte multidimensionnel ainsi qu'au calcul de support avec la mesure et à la génération de séquences divergentes. VPSP est un algorithme d'extraction de motifs séquentiels classique de la catégorie "générer-élaguer". Le principe est d'utiliser les $(k - 1)$ séquences pour générer les séquences candidates de longueur k , puis les séquences non fréquentes du niveau k sont élaguées en fonction du support minimum.

VPSP allie la structure d'arbre préfixée de PSP et la représentation de la base en mémoire de SPADE sous la forme de vecteurs d'apparitions, c'est-à-dire pour chaque séquence la liste des couples (client, date) qui supportent la séquence est chargée en mémoire. Pour générer les candidats du niveau k , seuls les vecteurs d'apparition du niveau 1 et du niveau $(k - 1)$ sont gardés. Cette représentation permet de calculer le support des séquences rapidement car il est inutile de faire une passe dans la base de données. Le support est alors le nombre de clients présents dans le vecteur.

Nous avons adapté cette structure à notre contexte, y compris la représentation, car le vecteur des couples (clients, date) ne permettait pas la mise en oeuvre de notre méthode. En effet les vecteurs

d'apparition permettent de calculer rapidement le support, dans notre cas, la mesure est utilisée pour le calculer, une adaptation est alors nécessaire. De plus certains problèmes peuvent apparaître avec la prise en compte de l'étoile car nous ne pouvons pas distinguer l'apparition de l'itemset $\{(A_1, *)(*, b_1)\}$ de l'item simple (A_1, B_1) . Nous avons alors adapté notre présentation afin de faire face à ces problèmes. Nous avons donc pu à la fin de cette étape générer des motifs séquentiels étoilés divergents, susceptibles de contenir des connaissances exceptionnelles.

3.4 Génération des règles séquentielles

La génération des règles séquentielles est faite à partir des motifs extraits. Il s'agit de générer pour chaque motif séquentiel la liste des règles séquentielles associées. Nous testons donc successivement les règles en coupant après chaque itemset. Pour stocker ces règles, nous avons ajouté un nouveau type de lien dans l'arbre des préfixes symbolisant l'implication.

Les règles doivent respecter un seuil de confiance minimale fixé par l'utilisateur, et trivialement implémenté car il s'agit de la division du support de l'antécédent de la règle par la séquence considérée. Toutes ces informations sont disponibles dans l'arbre des séquences fréquentes.

3.5 Collage des séquences et recherche des conséquences différentes

Nous n'avons pas fini l'implémentation de cette partie. Elle est en cours et nous espérons pouvoir obtenir des résultats prochainement.

Chapitre 4

Expérimentation sur jeu de données réelles

4.1 Jeux de données

L'étude réalisée ici utilise un jeu de données industrielles confidentiel. Ces données ont été collectées au cours d'une année et sont organisées sous la forme d'un sous-cube à 6 dimensions G , $date$, A , TO , SO , TC . La mesure, dimension numérique à valeurs entières correspond au nombre d'observations de la combinaison des valeurs des différentes dimensions. Les données utilisées lors des expérimentations commentées ici couvrent la totalité de ce sous-cube soit environ 83000 cellules comportant une mesure. Toutes les dimensions sont utilisées : la dimension de référence est la dimension G et les dimensions d'analyse sont les dimensions A , TO , TC et SO . La dimension $date$ est utilisée comme dimension temporelle.

4.2 Prétraitement

Les mesures de la base étudiée sont comprises dans l'intervalle $[1, 1000[$. Cependant, elles ne sont pas réparties de façon homogène, la figure 4.1 illustre cette distribution.

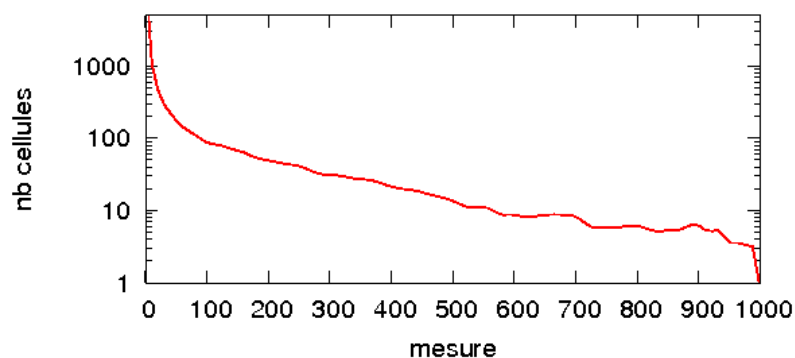


FIG. 4.1 – Distribution des valeurs de la mesure dans le sous-cube analysé

Considérant cette répartition, dans ces premières expérimentations, nous avons choisi de réaliser un partitionnement par équi-répartition des cellules dans chaque intervalle (partition stricte) ou sous-ensemble flou (partition floue). Différents tests ont été ensuite réalisés avec des partitions comportant 3, 9, 12 ou 24 sous-ensembles/intervalles. Les figures 4.2 et 4.3 montrent des exemple de ces

partitions.

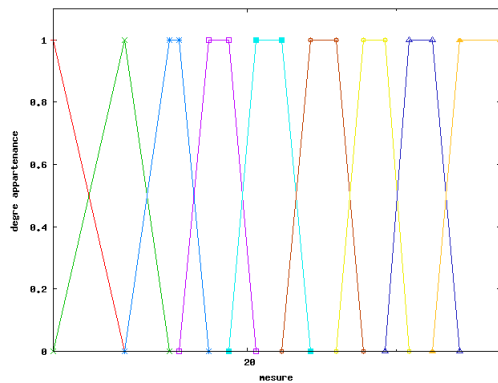


FIG. 4.2 – Partitionnement trapézoïdal

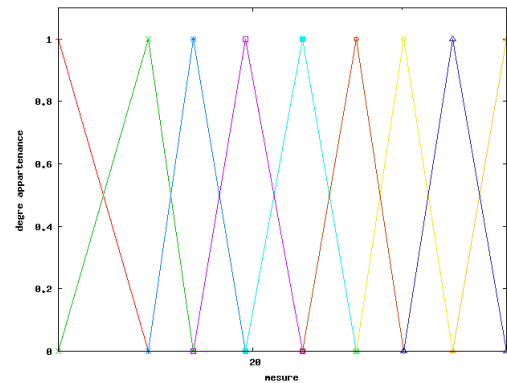


FIG. 4.3 – Partitionnement triangulaire

L'analyse des motifs extraits avec 3 intervalles montre qu'un tel découpage n'est pas pertinent pour notre analyse. En effet, l'ensemble des motifs séquentiels multidimensionnels extraits sont les mêmes quelle que soit la méthode utilisée : comptage binaire, avec partition stricte ou floue, ou comptage flou. De plus, le nombre d'items fréquents et donc de séquences fréquentes est trop élevé pour que ceux-ci puissent être analysés par un expert. En ce qui concerne les résultats obtenus avec les partitions en 12 ou 24, ils sont très similaires à ceux obtenus avec les partitions strictes ou floues en 9 intervalles ou sous-ensembles flous. Par ailleurs, le nombre de séquences fréquentes croit de façon beaucoup plus progressive pour ces découpages, il est alors plus simple pour l'utilisateur d'obtenir des motifs pertinents, en quantité raisonnable, puis de les analyser.

4.3 Comparaison des résultats

4.3.1 Comparaison des techniques de partitionnement

Nous comparons tout d'abord les résultats obtenus avec un partitionnement strict ou un partitionnement flou avec comptage binaire. Le principe du comptage binaire est d'incrémenter la fréquence de la séquence si le degré d'appartenance de tous ses items est supérieur à 0.

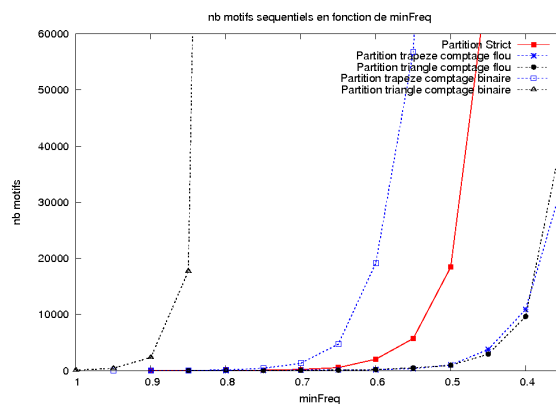


FIG. 4.4 – Nombres de séquences fréquentes en fonction de minFreq

Selon la figure 4.4, le nombre de motifs avec une partition stricte est moins important qu'avec une partition floue, qu'elle soit triangulaire ou trapézoïdale.

L'utilisation de sous-ensembles triangulaires conduit à l'extraction d'un nombre beaucoup plus élevé de séquences fréquentes. En effet, les surfaces de recouvrement entre deux sous-ensembles sont plus nombreuses, plus de combinaisons des itemsets multidimensionnels sont alors contenues dans la base.

Les motifs séquentiels multidimensionnels extraits par comptage binaire sont beaucoup plus nombreux que par comptage flou. En effet, le comptage flou pondère le nombre d'apparitions de chaque item par le degré d'appartenance de la mesure. Le comptage binaire est donc moins sélectif.

Le comptage binaire des apparitions d'une séquence sur une partition floue apparaît trop peu sélectif. Trop de connaissances sont extraites pour pouvoir être analysées facilement pour un expert humain. Ainsi, une fréquence minimum de 65% (encore très élevée si on considère le nombre d'enregistrements analysés), de l'ordre de 2500 séquences sont extraites. Il apparaît donc que cette combinaison partition floue/comptage binaire n'est pas pertinente pour notre jeu de données. De même, un découpage en intervalles ne semble pas suffisant pour sélectionner au mieux les séquences fréquentes. En effet, d'une part, le nombre de motifs extraits reste très élevés ; d'autre part, certains motifs porteurs d'informations intéressantes pour l'expert, n'apparaissent pas avec une partition stricte mais sont découverts avec une partition floue.

4.3.2 Comparaison avec ratio, sans ratio

Nous comparons dans cette section le nombre de résultats obtenues en fonction d'un partitionnement flou avec comptage flou et des deux techniques pour lesquels la mesure est utilisée pour calculer le support.

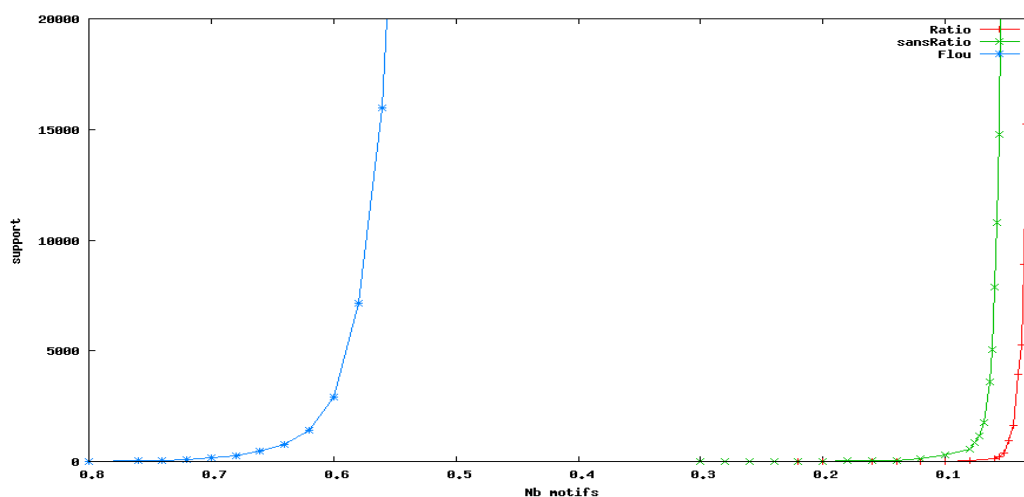


FIG. 4.5 – Comparaison du nombre de motifs extraits selon la partition

La figure ci-dessus (figure 4.5) montre le nombre de motifs extraits en fonction du support et de la technique utilisée. On remarque que le partitionnement flou est beaucoup plus sensible à la baisse de support que les autres partitions, notamment car cette technique ajoute une dimension d'analyse, et conserve une trop grande majorité d'items. Les techniques de ratio et sans ratio conservent les mesures les plus représentatives, c'est pour cela que le nombre de motifs est moins important.

Les résultats obtenus avec ou sans ratio apparaissent dans les motifs pour lesquels la mesure est paritionnée avec pour mesure les sous-ensembles représentant les plus fortes mesures. Ces résultats sont dû au fait qu'en utilisant la mesure pour calculer le support seuls les items avec les mesures les plus fortes seront extraits.

Conclusion et perspectives

En conclusion, nous avons dans ce rapport proposé un modèle complet d'extraction de connaissances atypiques. Nous nous situons dans un processus objectif, nos travaux se basent donc sur l'extraction de connaissances communes à l'extraction de comportements atypiques.

Le contexte multidimensionnel a été exploité dans son intégralité puisque les connaissances allient à la fois la notion de temporalité, la mise en valeur de la mesure et des connaissances multidimensionnelles. De plus, nous avons proposé la notion de confiance au travers de la définition de règles séquentielles multidimensionnelles.

Dans notre contexte, les règles séquentielles sont alors des motifs séquentiels étoilés qui ont une confiance importante. Ce type de connaissances est utilisé pour la détection de règles exceptionnelles. Une fois de plus nous avons profité de tous les aspects de ces connaissances : d'une part l'étoile dans les règles communes est au coeur de l'ajout d'information, et d'autre part la règle est utilisée dans son intégralité en séparant la notion de conséquences et d'antécédents.

De nombreuses perspectives sont envisageable à la suite de ce travail. Tout d'abord l'utilisateur doit fixer 4 seuils qui ne sont pas évident à déterminer. Cela constitue la première limite de notre modèle. Une solution consiste à l'utilisation d'une mesure plus fine pour calculer le taux d'intérêt d'une règle. Cela permettrait alors à l'utilisateur de n'avoir aucun seuil ou un seuil à fixer.

Ensuite, la méthode d'extraction de comportement atypiques est très naïve. En effet les séquences fréquentes servent de base à la génération de règles séquentielles et en fonction de celle-ci les règles exceptionnelles sont extraites. Une méthode moins naïve serait alors d'extraire en même temps les règles séquentielles et les règles exceptionnelles comme propose les travaux de [Suz99] dans leur contexte.

Une dernière perspective est d'utiliser une hiérarchie de valeurs de dimensions au lieu d'utiliser un seul niveau c'est-à-dire le total (l'étoile). En effet si nous avons l'item (Europe, Pain), une spécification possible pour cet item est (France, Pain). L'ajout d'information serait alors effectuée en fonction de la hiérarchie des valeurs des dimensions et non plus uniquement entre des items étoilés.

Bibliographie

- [AAR96] Soumen A. ARNING, R. AGRAWAL et P. RAGHAVAN : A linear method for deviation detection in large databases. *In Mining Intl Conference on Knowledge Discovery in Databases and Data Mining (KDD-95)*, 1996.
- [AIS93] Rakesh AGRAWAL, Tomasz IMIELINSKI et Arun SWAMI : Mining association rules between sets of items in large databases. *In SIGMOD '93 : Proceedings of the 1993 ACM SIGMOD international conference on Management of data*, pages 207–216, New York, NY, USA, 1993. ACM Press.
- [AL99] Yonatan AUMANN et Yehuda LINDELL : A statistical theory for quantitative association rules. *In Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, 1999.
- [AS94] Rakesh AGRAWAL et Ramakrishnan SRIKANT : Fast algorithms for mining association rules in large databases. *In VLDB '94 : Proceedings of the 20th International Conference on Very Large Data Bases*, pages 487–499, San Francisco, CA, USA, 1994. Morgan Kaufmann Publishers Inc.
- [AS95] R. AGRAWAL et R. SRIKANT : Mining Sequential Patterns. *In the 11th IEEE International Conference on Data Engineering*, pages 3–14, 1995.
- [BCMG04] Fernando BERZAL, Juan-Carlos CUBERO, Nicolas MARIN et Matias GAMEZ : Anomalous Association Rules. *In IEEE ICDM 2004 Workshop on Alternative Techniques for Data Mining and Knowledge Discovery*, November 2004.
- [BCMS01] Fernando BERZAL, Juan-Carlos CUBERO, Nicolas MARIN et Jose-Maria SERRANO : Tbar : An efficient method for association rule mining in relational databases. *Data Knowl. Eng.*, 37(1):47–64, 2001.
- [BS03] Stephen D. BAY et Mark SCHWABACHER : Mining distance-based outliers in near linear time with randomization and a simple pruning rule. *In KDD '03 : Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 29–38, New York, NY, USA, 2003. ACM Press.
- [CCS93] E.F. COOD, S.B. COOD et C.T. SALLEY : Providing olap(on-line analytical processing) to user-analysts : An it mandate. *In Arbor Software Corporation. Available at http://www.arborsoft.com/essbase/wht_ppr/coddToc.html*, page 31, 1993.
- [DJJK⁺06] L. DI-JORIO, D. JOUVE, D. KRAEMER, A. SERRA, C. RAISSI, A. LAURENT, M. TEISSEIRE et P. PONCELET : VPSP : extraction de motifs séquentiels dans weka. *In Démonstrations dans les 22èmes journées “Bases de Données Avancées” (BDA'06)*, 2006.
- [FLT04] Celine FIOT, Anne LAURENT et Maguelonne TEISSEIRE : A la recherche des motifs séquentiels flous. *In 12èmes rencontres francophones sur la Logique Floue et ses Applications, novembre 2004*, France, 2004.
- [Gye00] Attila GYENESEI : A fuzzy approach for mining quantitative association rules. Rapport technique TUCS-TR-336, Turku Centre for Computer Science, 2000.

- [HLSL00] Farhad HUSSAIN, Huan LIU, Einoshin SUZUKI et Hongjun LU : Exception rule mining with a relative interestingness measure. In *PADKK '00 : Proceedings of the 4th Pacific-Asia Conference on Knowledge Discovery and Data Mining, Current Issues and New Applications*, pages 86–97, London, UK, 2000. Springer-Verlag.
- [Inm90] W.H. INMON : *Building the Data Warehouse*. New York, NY : John Wiley & Sons, 1990.
- [KFW98] Chan Man KUOK, Ada Wai-Chee FU et Man Hon WONG : Mining fuzzy association rules in databases. *ACM SIGMOD Record*, 27(1):41–46, 1998.
- [KNT00] Edwin M. KNORR, Raymond T. NG et Vladimir TUCAKOV : Distance-based outliers : algorithms and applications. *The VLDB Journal*, 8(3-4):237–253, 2000.
- [LHCM00] Bing LIU, Wynne HSU, Shu CHEN et Yiming MA : Analyzing the subjective interestingness of association rules. *IEEE Intelligent Systems*, 15(5):47–55, 2000.
- [MCP98] F. MASSEGLIA, F. CATHALA et P. PONCELET : The PSP approach for mining sequential patterns. In *the Second European Conference on Principles of Data Mining and Knowledge Discovery*, pages 176–184, 1998.
- [MRBM06] Riadh Ben MESSAOUD, Sabine Loudcher RABASEDA, Omar BOUSSAID et Rokia MISSAOUI : Enhanced mining of association rules from data cubes. In *DOLAP '06 : Proceedings of the 9th ACM international workshop on Data warehousing and OLAP*, pages 11–18, New York, NY, USA, 2006. ACM Press.
- [PCL⁺05] Marc PLANTEVIT, Yeow Wei CHOONG, Anne LAURENT, Dominique LAURENT et Maguelonne TEISSEIRE : M2sp : Mining sequential patterns among several dimensions. In *Principles of Knowledge Discovery in Databases, PKDD*, volume 3721, pages 205–216. Springer Verlag Lecture Notes in Artificial Intelligence, 2005.
- [PHP⁺01] Helen PINTO, Jiawei HAN, Jian PEI, Ke WANG, Qiming CHEN et Umeshwar DAYAL : Multi-dimensional sequential pattern mining. In *CIKM '01 : Proceedings of the tenth international conference on Information and knowledge management*, pages 81–88, New York, NY, USA, 2001. ACM Press.
- [PLT07] M. PLANTEVIT, A. LAURENT et M. TEISSEIRE : Motifs séquentiels multidimensionnels convergents et divergents. In *Actes de : Extraction et Gestion des Connaissances (EGC07), Namur (Belgique)*, 2007.
- [SA96] Ramakrishnan SRIKANT et Rakesh AGRAWAL : Mining quantitative association rules in large relational tables. In H. V. JAGADISH et Inderpal Singh MUMICK, éditeurs : *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*, pages 1–12, Montreal, Quebec, Canada, 4–6 1996.
- [Sah99] Sigal SAHAR : Interestingness via what is not interesting. In *KDD '99 : Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 332–336, New York, NY, USA, 1999. ACM Press.
- [SCA06] Pei SUN, Sanjay CHAWLA et Bavani ARUNASALAM : Mining for outliers in sequential databases. In *In Proceedings of the 2006 SIAM Conference on Data Mining SDM'06*, 2006.
- [Suz99] Einoshin SUZUKI : Scheduled Discovery of Exception Rules. In *Proceedings of the Second International Conference on Discovery Science*, pages 184–195, 1999.
- [Zak01] M. J. ZAKI : SPADE : An efficient algorithm for mining frequent sequences. *Machine Learning*, 42(1/2):31–60, 2001.
- [Zhu95] Hua ZHU : *On-line analytical mining association rules*. Thèse de doctorat, University of science and technology of China, 1995.

Annexe A

Connaissances communes

A.1 La mesure dans la dimension d'analyse

Le principe de cette technique est de pré-traiter la mesure afin de trouver un partitionnement et d'utiliser une valeur symbolique plutôt que la mesure elle-même. Afin de traiter les données numériques pour les processus d'extraction de connaissances, de nombreux travaux utilisent une technique de partitionnement.

Nous souhaitons adapter les deux types de partitionnement à notre contexte. nous expliquons les motifs séquentiels non étoilés et présente la méthode de calcul de support. Puis nous verrons comment traiter le cas de l'étoile avec cette méthode.

A.1.1 Cas général

Le calcul du support diffère en fonction du type de partitionnement utilisé. Avec un partitionnement strict, le support est incrémenté par la présence de la séquence dans un bloc.

Dans le second cas, on établit un partitionnement flou, les items auront la même forme. Le support peut se calculer avec un calcul flou ou binaire.

Calcul flou : Le degré d'appartenance est une valeur numérique comprise entre 0 et 1. Une solution possible consiste à appliquer le même calcul que la formule (1). Pour chaque apparition d'une séquence dans un bloc nous prenons le minimum des degrés d'appartenance. Le maximum des degrés d'appartenance est pris pour les différentes apparitions de la séquence pour un bloc. La somme des degrés d'appartenance de tous les blocs est ensuite effectuée. Ensuite le support est divisé par le nombre de bloc.

$$supp(X, A) = \frac{\sum_{b \in B} [\min_{j=1}^{\theta_b} \overline{1}_{[x,a] \in (X,A)} [\alpha_a(t_j[x])]]}{nbBlocs} \quad (A.1)$$

Calcul binaire : La seconde façon de calculer est d'incrémenter le support si le degré d'appartenance est supérieur à 0 (ou à un seuil). Puis le support est divisé par le nombre de bloc.

En fonction de l'exemple A.1, voici quelques exemples de calculs de support dans le cas de partitionnement flou et strict.

1. $\langle \{(a_1, b_1, c_1, M)\} \rangle$:

Voici un exemple de calcul dans le cas d'un partitionnement flou et strict, la mesure a été préalablement partitionnée. Le partitionnement des dimensions utilisées est le suivant :

- dimension d'analyse : A, B, C, Valeur symbolique
- dimension de référence : GEO.

Date	A	B	C	M	Partition flou			Partition Strict
					P	M	B	
1	a_1	b_1	c_1	128		1		M
	a_1	b_1	c_2	152		1		M
	a_2	b_1	c_1	202		0.49	0.51	B
2	a_1	b_1	c_1	100	0.5	0.5		M
	a_1	b_1	c_2	200		0.5	0.5	B
	a_2	b_1	c_1	77	1			P

Date	A	B	C	M	Partition flou			Partition Strict
					P	M	B	
1	a_1	b_1	c_1	50	1			P
	a_1	b_1	c_2	111		1		M
	a_2	b_1	c_1	70	1			P
2	a_1	b_1	c_1	108	0.46	0.54		M
	a_1	b_1	c_2	100	0.5	0.5		M
	a_2	b_1	c_1	80	1			P

Date	A	B	C	M	Valeur symbolique			Partition Strict
					P	M	B	
1	a_1	b_1	c_1	45	1			P
	a_1	b_1	c_2	300			1	B
	a_2	b_1	c_1	130		1		M
2	a_1	b_1	c_1	106	0.47	0.53		M
	a_1	b_1	c_2	200		0.5	0.5	B
	a_2	b_1	c_1	125		1		M

FIG. A.1 – Exemple de bases de données multidimensionnelles avec la mesure partitionnée

- partition floue et calcul flou : $(1 + 0.54 + 0.53) / 3 = 0.69$
 - partition floue et calcul binaire : $(1 + 1 + 1) / 3 = 1$
 - partition stricte : $(1 + 1 + 1) / 3 = 1$
2. $\{(a_1, b_1, c_1, M) (a_1, b_1, c_2, M)\}$
- partition floue et calcul flou : $(1 + 0.5 + 0.5) / 3 = 0.67$
 - partition floue et calcul binaire : $(1 + 1 + 1) / 3 = 1$
 - partition stricte : $(1 + 0 + 1) / 3 = 0.67$
3. $\{(a_1, b_1, c_1, M) (a_1, b_1, c_2, M)\} \{(a_1, b_1, c_1, M)\}$:
- partition floue et calcul flou : $(0.5 + 0 + 0) / 3 = 0.17$
 - partition floue et calcul binaire : $(1 + 0 + 0) / 3 = 0.33$
 - partition stricte : $1 / 3 = 0.33$

A.1.2 Motifs séquentiels étoilés

Nous utilisons des opérateurs t-norme dans le cas d'une étoile dans un item. La somme n'est pas une solution envisageable car nous ne pouvons pas pour un bloc avoir un support plus fort que 1. Pour la valeur de l'étoile, nous avons alors comme possibilité, le maximum entre les degrés d'appartenance dans le cas d'un partitionnement flou, et pour un partitionnement strict, la valeur prise est toujours 1 si on trouve l'une des valeurs possibles de l'étoile.

Donc le principe lorsqu'on a un partitionnement flou est de prendre le maximum des degrés d'appartenance pour les valeurs possibles de l'étoile. Voici un exemple, pour les blocs de données de la figure A.1 :

1. $\langle \{(*, b_1, c_1, M)\} \rangle : (1 + 0,51 + 1)/3 = 0,84$
2. $\langle \{(a_1, b_1, *, M)\} \{(*, b_1, c_1, M)\} \rangle : (0,5 + 0,51 + 0)/3 = 0,37$
3. $\langle \{(*, b_1, c_1, M)\} \{(*, b_1, c_1, M)\} \rangle : (0,5 + 0 + 0)/3 = 0,17$

Lorsque nous utilisons la mesure partitionnée dans les dimensions d'analyse, les séquences extraites donne une information sur le type de mesures qui apparait pour chaque item de la séquence. Nous pouvons mettre en évidence les évolutions des mesures à travers le temps. Le principal problème est la difficulté de trouver un partitionnement adéquat pour un jeu de données car si il n'y a pas assez de partition, beaucoup trop de séquences sont fréquentes. En revanche si il y a trop de partition, il peut y avoir des résultats peu intéressants et très peu de connaissances extraites (cela revient quasiment à considérer la mesure en tant que valeur symbolique).

Annexe B

Table des notations

Dans ce chapitre, les notations suivantes sont utilisées :

Total de la mesure pour toute la base	Γ_1
Total de la mesure pour le bloc b	$\Gamma_{2,b}$
Itemset de la séquence s	IS
Item d'un itemset	i
Ensemble des blocs de la base	B
Mesure de l'item i	$m[i]$
Minimum (t-norme)	$\overline{\top}$
Maximum (t-conorme)	$\underline{\perp}$