



**HAL**  
open science

## Autour et alentours des motifs séquentiels

Maguelonne Teisseire

► **To cite this version:**

Maguelonne Teisseire. Autour et alentours des motifs séquentiels. Interface homme-machine [cs.HC].  
Université Montpellier II - Sciences et Techniques du Languedoc, 2007. tel-00203628

**HAL Id: tel-00203628**

**<https://theses.hal.science/tel-00203628>**

Submitted on 10 Jan 2008

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

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

UNIVERSITÉ MONTPELLIER II  
— Sciences et Techniques du Languedoc —

## HABILITATION À DIRIGER DES RECHERCHES

DISCIPLINE : INFORMATIQUE  
*Spécialité Doctorale* : *Informatique*  
*Ecole Doctorale* : *Information, Structure, Systèmes*

présentée et soutenue publiquement par

Maguelonne TEISSEIRE

le 5 décembre 2007

## Autour et alentour des motifs séquentiels

### JURY

Elisa BERTINO, Professor, Purdue University, USA, ..... Rapporteur  
Georges GARDARIN, Professeur, Université de Versailles Saint Quentin, ..... Rapporteur  
Mohand-Said HACID, Professeur, Université Claude Bernard Lyon 1, ..... Rapporteur  
Danièle HÉRIN, Professeur, Université Montpellier II, ..... Examineur  
Dominique LAURENT, Professeur, Université de Cergy-Pontoise, ..... Président  
Jocelyne NANARD, Professeur, Université Montpellier II, ..... Examineur  
Nicolas SPYRATOS, Professeur, Université Paris Sud, ..... Examineur



# Table des matières

<b>1</b>	<b>Introduction</b>	<b>7</b>
1.1	Le processus d'extraction de connaissances . . . . .	7
1.2	Vers l'extraction de comportements : les motifs séquentiels . . . . .	8
1.3	Des motifs séquentiels! oui, mais ... . . . .	9
1.4	Organisation du mémoire . . . . .	11
<b>2</b>	<b>Problématique et définitions</b>	<b>13</b>
2.1	Définitions et problématique . . . . .	13
2.2	Le point . . . . .	15
2.2.1	Méthodes basées sur un parcours en largeur d'abord . . . . .	15
2.2.2	Méthodes basées sur une projection de la base . . . . .	17
2.2.3	Recherche des motifs séquentiels fermés . . . . .	18
2.3	Discussion . . . . .	19
<b>3</b>	<b>Approche incrémentale</b>	<b>21</b>
3.1	Introduction . . . . .	22
3.2	Le point . . . . .	22
3.3	Vers une approche incrémentale : ISE . . . . .	23
3.3.1	Principe . . . . .	23
3.3.2	L'algorithme ISE . . . . .	26
3.4	Discussion . . . . .	26
<b>4</b>	<b>Les motifs séquentiels flous</b>	<b>29</b>
4.1	Introduction . . . . .	30
4.2	Théorie des sous-ensembles flous : quelques rappels . . . . .	30
4.3	Le point . . . . .	32
4.4	Vers des motifs séquentiels flous . . . . .	33
4.4.1	Principe . . . . .	33
4.4.2	Les algorithmes SPEEDYFUZZY, MINIFUZZY et TOTALLYFUZZY .	36
4.5	Discussion . . . . .	38
<b>5</b>	<b>Les contraintes de temps</b>	<b>41</b>
5.1	Introduction . . . . .	42
5.2	Le point . . . . .	42
5.3	Extension des contraintes de temps . . . . .	43
5.4	Vers des contraintes de temps étendues : GETC . . . . .	46
5.4.1	Principe . . . . .	47
5.4.2	L'algorithme GETC . . . . .	47
5.4.3	Illustration . . . . .	52
5.5	Discussion . . . . .	54

<b>6</b>	<b>Données textuelles</b>	<b>57</b>
6.1	Introduction . . . . .	58
6.2	Le point . . . . .	58
6.3	Vers une catégorisation par motif séquentiels : SPAC . . . . .	61
6.3.1	Première étape - Des textes aux motifs séquentiels . . . . .	61
6.3.2	Deuxième étape - Des motifs séquentiels aux catégories . . . . .	64
6.4	La classification de documents : le MCT . . . . .	65
6.4.1	Représentation des documents . . . . .	66
6.4.2	Catégorisation et catégoriseur . . . . .	68
6.5	Discussion . . . . .	70
<b>7</b>	<b>Données multidimensionnelles</b>	<b>73</b>
7.1	Introduction . . . . .	74
7.2	Le point . . . . .	74
7.3	Vers des motifs séquentiels multidimensionnels : $M^2SP$ , $HYPE$ . . . . .	75
7.3.1	Principes . . . . .	75
7.3.2	Les algorithmes $M^2SP$ et $HYPE$ . . . . .	81
7.4	Discussion . . . . .	86
<b>8</b>	<b>Conclusions et Perspectives</b>	<b>89</b>
8.1	Un bref historique . . . . .	89
8.1.1	Synthèse . . . . .	89
8.1.2	Plus précisément, sur la fouille de données . . . . .	90
8.2	Publications . . . . .	92
8.3	Encadrements . . . . .	99
8.3.1	Encadrement de Thèses . . . . .	99
8.3.2	Encadrement de Stages Recherche de Master R (ou ex DEA) . . . . .	101
8.3.3	Encadrement de Mémoires d'Ingénieur CNAM . . . . .	102
8.4	Transferts de Technologie . . . . .	102
8.5	Perspectives . . . . .	103
8.5.1	De la préservation de la vie privée . . . . .	103
8.5.2	Des données disponibles de plus en plus rapidement . . . . .	104
8.5.3	Des motifs fréquents? oui, mais ... . . . . .	105
8.5.4	Le processus d'extraction revisité . . . . .	106

# Synopsis

La modélisation du comportement a guidé mes travaux de thèse (définition du modèle comportemental  $IFO_2$  et ses mécanismes de vérification et validation des spécifications réalisées). Ces travaux se sont poursuivis avec un objectif bien précis : ne plus imposer une structure de description du comportement mais le détecter de façon automatique en utilisant les données déjà modélisées et stockées sous un format a priori quelconque. Une telle démarche s'inscrit dans une double problématique : la représentation des données et des comportements associés et enfin l'extraction (et validation) de tels comportements.

Ce mémoire se focalise sur ces deux problématiques. Tout d'abord, nous nous basons sur une représentation des données très "classique" et développons les définitions de motifs séquentiels que nous avons choisis comme format de description des comportements extraits (Chap. 2) ainsi que nos propositions réalisées pour l'extraction de tels motifs : découverte et mises-à-jour (Chap. 3). Ensuite nous proposons une représentation moins stricte des comportements et définissons les méthodes d'extraction associées (Chap. 4). Adoptant la même philosophie de fouille approximative, nous étendons la gestion des contraintes de temps (Chap. 5).

Même lorsque les données sont plus complexes, les motifs séquentiels s'avèrent également une représentation adaptée. Nous nous attachons à décrire certaines de nos propositions sur deux types de données plus complexes : les documents textuels (Chap. 6) et les données multidimensionnelles (Chap. 7).

Enfin le dernier chapitre de ce mémoire (Chap. 8) est dédié aux bilans et aux nombreuses perspectives offertes par ces travaux.

Pour chacune de nos propositions, nous nous attachons à définir les concepts introduits et présenter les algorithmes permettant leur mise en œuvre. Tous ces travaux ont donné lieu à des évaluations sur des jeux de données réels ou synthétiques afin de souligner la pertinence, le passage à l'échelle ou l'adéquation à des types de données particuliers. Dans ce mémoire, nous n'abordons pas ces expérimentations et nous encourageons le lecteur intéressé à se reporter aux articles associés.

---



# Chapitre 1

## Introduction

Motivés par la modélisation du comportement, les travaux de recherche effectués lors de mon doctorat, se sont focalisés sur la définition de la partie comportementale du modèle *IFO*<sub>2</sub>. Dans ce cadre, il s'agissait d'offrir à l'utilisateur un nouveau modèle conceptuel permettant de décrire le comportement des applications avancées. Au travers de mes propositions, le concepteur de l'application pouvait, non seulement, spécifier le comportement de l'application mais aussi vérifier et valider un certain nombre de propriétés. Bien entendu, de manière à avoir une approche complète, l'approche proposée permettait également de dériver de manière automatique vers des systèmes cibles : les règles actives.

Les recherches que j'ai menées par la suite s'inscrivent toujours dans la modélisation du comportement mais en considérant cette fois-ci les données représentant l'univers réel. L'objectif dans ce cadre est d'offrir à l'utilisateur final, i.e. le décideur, des mécanismes pour mieux appréhender le comportement des systèmes sous-jacents. Pour illustrer ce propos, considérons le cas d'un serveur Web où de nombreux utilisateurs se connectent régulièrement. Nous nous retrouvons devant un ensemble de données à partir desquelles nous souhaitons répondre aux requêtes suivantes : quelles sont les pages les plus utilisées ? quelles sont les pages avec problème ? quel est le comportement des utilisateurs sur ce site Web ? Alors que les deux premières questions peuvent facilement être traitées via une requête, la dernière question soulève un nouveau problème : à quoi correspond un comportement d'utilisateur ? Intuitivement, nous pouvons dire qu'il s'agit de l'enchaînement des différentes pages sur le site et notre objectif est alors d'extraire, à partir des données sources, les comportements typiques. Ce problème entre tout à fait dans le cadre du processus d'extraction de connaissances et c'est dans ce contexte que se situent les travaux de recherches que j'ai menés ces dernières années.

### 1.1 Le processus d'extraction de connaissances

Motivés par des problèmes d'Aide à la Décision, les chercheurs de différentes communautés (Intelligence Artificielle, Statistiques, Bases de Données, Interface Homme Machine) se sont intéressés à la conception et au développement d'une nouvelle génération d'outils permettant d'extraire automatiquement de la connaissance de grandes bases de données. Ces outils, techniques et approches sont le sujet d'un thème de recherche connu sous le nom de *Knowledge Discovery in Databases* (Extraction de Connaissances dans les Bases de Données) dont le *Data Mining* (Fouille de Données) est une étape spécifique. Elles sont utilisées dans de nombreux domaines d'applications. Les exemples les plus courants sont les compagnies d'assurance, les compagnies



bancaires (crédit, prédiction du marché, détection de fraudes), le marketing (comportement des consommateurs, "mailing" personnalisé), la recherche médicale (aide au diagnostic, au traitement, surveillance de population sensible), les réseaux de communication (détection de situations alarmantes, détection d'intrusions, prédiction d'incidents), l'analyse de données spatiales, etc.

Ces besoins variés nécessitent des approches différentes dont nous décrivons brièvement les plus significatives :

- **Recherche de règles d'association.** Le problème de la recherche de règles d'association introduit par R. Agrawal *et al.* en 1993 [AIS93a], est souvent appelé "problème du panier de la ménagère" (*Market Basket Problem*) car les transactions opérées par les clients d'un magasin et dont la trace est stockée représentent une application typique pour le processus de découverte de connaissances. Dans un tel contexte, une règle d'association peut être par exemple : "85% des clients achètent du beurre et du café achètent aussi du lait". La recherche de règles couvre un large champ d'applications telles que la conception de catalogues en ligne dans un contexte de e-commerce, la promotion de ventes, le suivi de clientèle, la gestion des stocks, etc.
- **Le clustering.** Le problème du clustering (ou segmentation) consiste à regrouper des enregistrements qui semblent similaires dans une même classe. Il est complémentaire à celui de la classification car le but ici est de rechercher les différentes classes possibles d'appartenance en fonction des différents attributs ou critères qui caractérisent les données. Les applications concernées incluent notamment la segmentation de marché, la segmentation démographique (pour identifier par exemple des caractéristiques communes entre populations), la classification de documents en fonction de leur contenu, etc.
- **La classification.** Généralement associée à l'apprentissage supervisé ou non-supervisé, elle consiste à analyser de nouvelles données et à les affecter, en fonction de leurs caractéristiques ou attributs, à telle ou telle classe prédéfinie ou non. Les techniques de classification sont par exemple utilisées lors d'opérations de "mailing" pour cibler la bonne population et éviter ainsi un nombre trop important de non-réponse. De la même manière, cette démarche peut permettre de déterminer, pour une banque, si un prêt peut être accordé en fonction de la classe d'appartenance d'un client.

## 1.2 Vers l'extraction de comportements : les motifs séquentiels

Même si toutes ces approches permettent d'extraire de la connaissance de grandes bases de données, elles ne sont pas (ou mal) adaptées à l'extraction de comportements des données. En 1995, la problématique de la recherche de règles d'association est étendue pour détecter des comportements typiques dans le temps et le concept de motifs séquentiels est introduit [AS95a]. La recherche de tels motifs consiste à extraire des ensembles d'objets couramment associés sur une période de temps spécifiée. Il est alors possible d'extraire des relations temporelles comme par exemple "36% des clients achètent une télévision, achètent un lecteur de DVD dans les deux ans qui suivent et un Home-Cinema 6 mois après" ou "30% des abonnés d'une vidéothèque qui ont emprunté *Marius*, empruntent *Fanny* un mois plus tard, puis *César* quelques semaines après".

De manière intuitive, l'extraction de motifs séquentiels consiste à rechercher, dans une base de données de transactions, les comportements les plus typiques. Cette notion est très proche de celle de la recherche de règles d'association dans de grandes bases de données mais possède une particularité essentielle : la nécessité de prendre en compte la temporalité des transactions. Depuis la définition de la problématique, les chercheurs de la communauté fouille de données se sont de plus en plus intéressés à l'extraction de tels motifs. En effet, même si la problématique initiale était celle du "panier de la ménagère", il est clair que les motifs apportent une connaissance supplémentaire qui était jusqu'alors inexistante : on ne cherche plus à connaître les items corrélés entre eux mais on s'intéresse aux comportements existants. L'engouement des travaux de recherche a bien entendu été motivé par les nombreux domaines d'application pour lesquels les motifs sont particulièrement adaptés. En effet, en étendant le panier de la ménagère aux domaines dans lesquels une base de données est constituée de nombreuses transactions dans lesquelles il existe une relation d'ordre entre les éléments, nous sommes alors à même d'extraire des motifs. Ainsi, par exemple, en considérant les navigations des clients sur un site Web, nous pouvons connaître les comportements des différents clients. Si la base de données est constituée de texte, nous pouvons, via les motifs, extraire les tendances dans les textes par exemple. De la même manière en extrayant le comportement de capteurs, nous pouvons prévoir et anticiper une panne.

### 1.3 Des motifs séquentiels ! oui, mais ...

L'objectif principal des travaux réalisés sur les motifs séquentiels par la communauté fouille de données a été de se focaliser sur des algorithmes de plus en plus efficaces en temps de réponses. Bien entendu, nous avons partagé cet objectif et au sein du projet nous avons proposé l'algorithme PSP. Cependant, nos différentes recherches et l'expérience acquise dans le domaine nous ont montré que même si une approche efficace était indispensable, il existait de nombreux défis qu'il nous fallait relever.

Le premier de ces défis est lié au fait que, quelle que soit l'approche retenue, cette dernière est pénalisée par le fait qu'elle ne fonctionne que sur une base de données statique. De manière à illustrer ce point, reprenons l'exemple du site Web. Généralement, lorsque l'on souhaite analyser le comportement des utilisateurs sur un site Web, cette opération est réalisée en récupérant un fichier log contenant les données sur une période quelconque et en extrayant les motifs. L'avantage de cette approche est que l'on connaît le comportement des usagers finement sur la période considérée. L'inconvénient est que les connaissances acquises s'avèrent vite obsolètes (surtout dans le cas d'un site Web) dans la mesure où de nouvelles données vont rapidement arriver. Dans ce cas, que faire des nouvelles données par rapport à la connaissance préalablement acquise ? Une approche naïve serait d'intégrer ces données dans la base d'origine et de relancer le processus d'extraction. Bien entendu, cette approche est complètement inefficace et il convenait d'en trouver une autre. Notre objectif, lors de ces travaux, a en fait été de prendre en compte le plus possible la connaissance extraite pour la mettre à jour. Même s'il commençait à exister des travaux sur la maintenance ou la fouille de données incrémentale à l'époque (ces derniers ne considéraient que la recherche des règles d'association), nous avons proposé l'une des premières approches, appelée ISE, pour les motifs séquentiels. Il est important de noter que cet aspect est devenu depuis une problématique importante.

L'hypothèse initiale de la recherche de motifs séquentiels est de considérer que les

données sont booléennes : un client achète ou n'achète pas un produit. La seconde hypothèse est que, pour minimiser l'espace de recherche, un objet ne peut intervenir qu'une seule fois dans un ensemble d'achats. Étendre la recherche de motifs pour prendre en compte plusieurs occurrences d'un objet ne nécessite que peu de modifications dans les approches traditionnelles. Par contre, l'intérêt des motifs extraits est très discutable. En effet, dans ce cas, nous n'obtiendrons que des motifs de la forme : "les personnes qui ont acheté trois bouteilles de vin ont aussi acheté deux fromages". Or les nombreuses valeurs numériques potentiellement prises par ces quantités et la faible différence sémantique entre elles (acheter 2 ou 3 fromages est difficilement séparable strictement) rendent ces motifs difficiles à extraire et peu informatifs. Notre objectif, dans ce cadre, est de permettre d'assouplir ces notions en intégrant des degrés d'appartenance. Plus précisément, nous souhaitons intégrer une composante floue dans l'extraction des motifs séquentiels. Bien entendu, cette intégration aura des conséquences sur le calcul des séquences fréquentes. Pour cette raison, nous avons défini trois approches complémentaires "plus ou moins floues" : SPEEDYFUZZY, MINIFUZZY et TOTALLYFUZZY.

Même si les motifs offrent une connaissance nouvelle, il faut reconnaître que pour certains domaines d'application ces derniers sont difficilement utilisables dans un contexte d'aide à la décision. Considérons par exemple le motif suivant extrait d'un grand magasin : "47% des clients achètent du champagne en janvier puis des châtaignes en octobre". Il est clair que cette connaissance n'est pas utile dans la mesure où il n'existe pas de corrélation entre ces deux événements espacés d'une année. De manière à affiner les connaissances acquises, R. Srikant et R. Agrawal [SA96b] ont proposé, via l'algorithme GSP, de prendre en compte différentes contraintes temporelles. Ces dernières permettent entre autre de s'intéresser à des comportements à court ou à long terme. Lors de nos travaux nous avons montré que l'approche GSP souffrait de nombreuses opérations de "backtracking" effectuées lors de l'application des contraintes de temps. Pour pallier ce problème, nous avons tout d'abord proposé l'algorithme GTC [MPT04] dont l'originalité était de prétraiter les contraintes de temps. L'un des avantages de GTC est d'être suffisamment générique pour pouvoir être utilisé par les principaux algorithmes de recherche de motifs de type Apriori. Cependant même si cette approche est efficace, elle impose au décideur de spécifier des contraintes strictes (e.g. il faut qu'il y ait un intervalle de temps de cinq jours entre deux achats). Pour faciliter ces spécifications, nous avons proposé l'algorithme GETC qui tire profit des avantages de GTC mais intègre en plus une composante floue qui permet de relâcher les contraintes et d'offrir ainsi "plus de flexibilité" dans le processus d'extraction.

Les données manipulées étant de plus en plus complexes, nous nous sommes intéressés à l'utilisation des motifs séquentiels pour traiter des données semi structurées et plus particulièrement des données textuelles. Dans ce cadre, le défi est de montrer que les motifs apportent une connaissance supplémentaire par rapport aux approches traditionnelles. Les principaux travaux sur la fouille de texte [IS07] concernent la classification supervisée ou non. Le problème principal de ces approches est qu'elles considèrent souvent des sacs de mots et ne tiennent pas compte de l'ordre d'apparition des mots au sein du texte. En utilisant les motifs, notre objectif était de renforcer les connaissances extraites en se focalisant sur cet ordre. Nous avons ainsi proposé l'approche SPAC qui extrait des séquences à partir de données textuelles. Les expérimentations que nous avons menées ont montré que cette approche était, si ce n'est toujours meilleure, au moins égale aux approches traditionnelles. Par contre, elle possède un avantage indéniable pour le décideur : nous obtenons des règles de classification interprétables et compréhensibles.

L'originalité des motifs est d'extraire des comportements sur une dimension d'analyse : les achats des clients. Nous disposons pourtant souvent d'autres informations qui pourraient enrichir la connaissance apprise. Par exemple, pour chaque client, nous pouvons savoir dans quelle ville il a acheté des produits ou bien quelle est la catégorie socio professionnelle du client. Dans ce cas, les données sont beaucoup plus complexes que précédemment car nous devons intégrer les connaissances déjà acquises sur le client et surtout offrir une extraction sur plusieurs dimensions d'analyse. Le défi que nous avons alors à considérer est la manière de gérer ce nouvel espace de recherche? En effet, nous verrons au cours de ce mémoire que l'espace de recherche pour les motifs séquentiels est très grand du fait des différentes permutations que l'on peut faire sur les objets manipulés. En prenant en compte de nouvelles dimensions nous étendons donc également l'espace de recherche associé. Pour résoudre cette problématique nous avons proposé différentes approches : *M<sup>2</sup>SP* et *HYPE* qui extraient des motifs et qui offrent la possibilité d'intégrer la hiérarchie sur les dimensions.

Notre objectif dans ce mémoire est de présenter les principaux travaux que nous avons menés ces dernières années dans le domaine de l'extraction des motifs séquentiels. Nous avons volontairement décidé de ne pas présenter tous nos travaux de recherche mais plutôt de nous focaliser sur les grandes étapes. Par exemple, nous ne discuterons pas de nos recherches sur la détection de séquences cachées ou dans les systèmes pair à pair. Même si nous avons également travaillé sur des données arborescentes (i.e. dans le cadre des thèses de Pierre-Alain Laur et de Federico Del Razo Lopez) et que ces derniers partagent des points communs avec les motifs, la représentation du comportement associé reste cependant éloignée de celle des motifs séquentiels, cœur du travail que nous avons choisi de présenter.

## 1.4 Organisation du mémoire

Les différents chapitres du mémoire reprennent nos principales contributions. Pour chacune d'entre elles, nous précisons les co-encadrements et donnerons une description succincte des algorithmes que nous avons définis. Par souci d'homogénéité, nous ne décrirons pas dans le mémoire les analyses en complexité, les expérimentations ainsi que certaines preuves sachant que tous ces éléments se retrouvent dans les publications liées à ces travaux. Enfin, chaque chapitre se termine par une discussion.

Le mémoire est organisé de la manière suivante :

Dans le chapitre 2, nous revenons sur les concepts fondamentaux des motifs séquentiels et présentons les principaux travaux de ces dernières années. L'objectif de ce chapitre est bien entendu de montrer les différentes évolutions mais également de montrer que la communauté s'est particulièrement intéressée à définir des algorithmes de plus en plus efficaces en temps de réponses. A l'issue de ce chapitre, nous revenons sur les motivations qui nous ont poussées à étendre les motifs séquentiels.

Le chapitre 3 présente les travaux que nous avons menés dans le cadre des motifs séquentiels incrémentaux. Etant donné qu'il existait peu de travaux sur ce domaine lorsque nous avons abordé cette problématique, nous revenons dans la discussion sur les travaux récents et regardons s'ils peuvent facilement être adaptés à notre approche.

De manière à offrir plus de souplesse dans le processus d'extraction de motifs, nous proposons dans le chapitre 4 d'intégrer une composante floue.

L'objectif du chapitre 5 est de présenter comment les contraintes de temps peuvent être prises en compte lors de l'extraction de motifs. Ce chapitre étend une proposition initiale que nous avons faite en intégrant une composante floue qui permet d'intégrer de manière moins stricte les contraintes de temps.

Le chapitre 6 présente les travaux que nous avons menés dans le cas de données textuelles. Il décrit un nouveau classifieur basé sur des motifs séquentiels.

Lorsque l'on ne se contente plus d'une seule dimension d'analyse, il est possible d'extraire des motifs multi-dimensionnels. L'extraction de ce type de motif est décrit dans le chapitre 7.

Enfin, dans le chapitre 8, nous concluons ce mémoire en revenant sur les conditions de la recherche et en décrivant les principaux résultats que nous avons obtenus ces dernières années. Nous proposons également différentes perspectives de recherche que nous souhaitons mener ces prochaines années.

## Chapitre 2

# Problématique et définitions

Dans un premier temps, le problème de l'extraction de motifs séquentiels peut sembler proche de celui de l'extraction de règles d'association. Ce rapprochement s'avère cependant très fragile en raison d'un élément clé qui est propre à l'extraction de motifs séquentiels : la temporalité. Cette notion permet à la fois de distinguer à l'intérieur des enregistrements un ordre d'apparition mais aussi de regrouper certains éléments. En effet si les règles d'association s'appliquent à des données de type ensemble d'items, i.e. des itemsets, et permettent l'extraction de règles intra-transaction, la recherche de motifs séquentiels s'applique à des données de type liste d'ensemble d'items (et permet donc l'extraction de règles inter-transactions). Même si dans la définition initiale, la notion de motifs a été introduite pour prendre en compte la temporalité entre événements, elle est bien entendu généralisable à tout domaine où il existe une relation d'ordre entre les éléments.

Nous proposons dans ce chapitre de décrire la problématique de l'extraction de motifs séquentiels qui est à la base des travaux que nous avons menés récemment. En présentant également les différentes approches qui existent à l'heure actuelle nous souhaitons décrire les principales orientations retenues ces dernières années.

Le chapitre est organisé de la manière suivante. Dans la section 2.1 nous proposons les définitions associées à la recherche de motifs séquentiels et présentons brièvement la problématique étudiée. Nous présentons dans la section 2.2, les principales méthodes d'extraction de motifs en nous focalisant sur les différentes stratégies de parcours de l'espace de recherche et sur les nouvelles tendances qui s'intéressent aux motifs clos. Enfin nous concluons ce chapitre par une discussion.

### 2.1 Définitions et problématique

Initialement introduite dans [AS95a], la notion de séquence est définie de la manière suivante.

**Définition 1 (Séquence de données)** *Une transaction constitue, pour un client  $C$ , l'ensemble des items achetés par  $C$  à une même date. Dans une base de données client, une transaction s'écrit sous la forme d'un ensemble :  $id\text{-client}$ ,  $id\text{-date}$ ,  $itemset$ . un  $itemset$  est un ensemble non vide d'items évalué à vrai noté  $(i_1 i_2 \dots i_k)$ . Une séquence est une liste ordonnée, non vide, d'itemsets notée  $\langle s_1 s_2 \dots s_n \rangle$  où  $s_j$  est un  $itemset$ . une séquence de données est une séquences représentant les achats d'un client. soit  $T_1, T_2, \dots, T_n$  les transactions d'un client, ordonnées par dates d'achat croissantes et*

soit  $\text{itemset}(T_i)$  l'ensemble des items correspondants à  $T_i$ , alors la séquence de données de ce client est  $\langle \text{itemset}(T_1) \text{ itemset}(T_2) \dots \text{itemset}(T_n) \rangle$

**Exemple 1** Soit  $C$  un client et  $S = \langle (10) (20 30) (40) \rangle$ , la séquence de données représentant les achats de ce client.  $S$  peut être interprétée par "C a acheté l'item 10, puis en même temps les items 20 et 30 et enfin l'item 40".

**Définition 2 (Inclusion)** Une séquence  $S' = \langle s'_1 s'_2 \dots s'_n \rangle$  est une sous-séquence de  $S = \langle s_1 s_2 \dots s_m \rangle$ , notée  $S' \preceq S$ , si  $\exists i_1 < i_2 < \dots < i_j \dots < i_n$  tels que  $s'_1 \subseteq s_{i_1}$ ,  $s'_2 \subseteq s_{i_2}, \dots, s'_n \subseteq s_{i_n}$ . Si  $S \not\preceq S'$  et  $S' \not\preceq S$ , les séquences sont dites incomparables et sont notées  $S \not\prec S'$ . De plus, une séquence est dite régulière si chaque itemset  $it_j$  contient le même unique item  $i$ .

**Exemple 2** La séquence  $S' = \langle (30) (50 60) (80) \rangle$  est incluse dans la séquence  $S = \langle (10) (30 80) (70) (50 60 90) (80) \rangle$  (i.e.  $S' \preceq S$ ) car  $(30) \subseteq (30 80)$ ,  $(50 60) \subseteq (50 60 90)$  et  $(80) \subseteq (80)$ . En revanche  $\langle (30) (60) \rangle \not\preceq \langle (30 60) \rangle$  (et vice versa).

Un client supporte une séquence  $s$  si  $s$  est incluse dans la séquence de données de ce client. Le support d'une séquence  $s$  est calculé comme étant le pourcentage des clients qui supportent  $s$ .

**Définition 3 (Support)** Soit  $C_{trans}$  la liste ordonnée des transactions pour un client  $C$  (i.e. la séquence maximale supportée par  $C$ ). Le support d'une séquence  $S$  dans une base transactionnelle  $\mathcal{D}$ , noté  $\text{Support}(S, \mathcal{D})$ , est défini tel que :  $\text{Support}(S, \mathcal{D}) = |\{C \in \mathcal{D} | S \preceq C_{trans}\}|$ .

**Remarque 1** Une séquence de données n'est prise en compte qu'une seule fois pour calculer le support d'une séquence fréquente, i.e. le client peut présenter plusieurs fois le même comportement, le processus de recherche de séquences considère qu'il produit ce comportement sans tenir compte du nombre de ses apparitions dans la séquence de données du client.

La propriété suivante considère le cas des sous ensembles par rapport aux calculs du support et de l'inclusion.

**Propriété 1 antimonotonie [AS95a, PHW02]**

Soit  $S'$  et  $S$  deux séquences. Si  $S' \subseteq S$  alors  $\text{support}(S') \geq \text{support}(S)$ .

Avec ces définitions, nous pouvons maintenant décrire formellement le problème d'extraction des motifs séquentiels et sa solution.

**Définition 4 (Extraction des motifs séquentiels fréquents)** Soit  $\mathcal{D}$  une base de données contenant des transactions regroupées par client où chaque transaction  $T$  consiste en : un identifiant de client, noté  $C_{id}$  ; une estampille temporelle, notée  $time$  et un ensemble d'items (appelé itemset) noté  $it$ . Soit  $\mathcal{SP}$  la séquence maximale théorique pouvant être générée à partir des clients dans  $\mathcal{D}$ . La solution au problème d'extraction des motifs séquentiels fréquents est définie telle que :

$$\text{FreqSeqSet}(S, \mathcal{D}, \sigma) = \{S \preceq \mathcal{SP} | \text{Support}(S, \mathcal{D}) \geq \sigma\}$$

Où  $\sigma$  est un seuil de support minimal défini par l'utilisateur,  $0 \leq \sigma \leq |\mathcal{C}|$  et  $\mathcal{C}$  est l'ensemble des clients dans  $\mathcal{D}$ .

La propriété suivante est une conséquence de la propriété 1.

Client	Date	Items
$C_1$	01/04/2007	20,60
$C_1$	02/04/2007	20
$C_1$	04/04/2007	30
$C_1$	18/04/2007	80,90
$C_2$	11/04/2007	10
$C_2$	12/04/2007	30
$C_2$	29/04/2007	40,60,70
$C_3$	05/04/2007	30,50,70
$C_3$	12/04/2007	10,20
$C_4$	06/04/2007	20,30
$C_4$	07/04/2007	40,70
$C_4$	08/04/2007	90

**Fig. 2.1:** Une base de données exemple

**Propriété 2** Soit  $S'$  une séquence non fréquente. Quelle que soit  $S$  telle que  $S' \subseteq S$ ,  $S$  est une séquence non fréquente.

En effet, d'après cette propriété,  $\text{support}(B) \leq \text{support}(A) \leq \sigma$ , donc B n'est pas fréquent.

**Exemple 3** Considérons la base de données  $D$  illustrée par la figure 2.1. Avec un support minimum de 50% (i.e. pour qu'une séquence soit retenue, il faut que deux clients dans la base de données supportent cette séquence), les séquences fréquentes maximales sont alors les suivantes :  $\langle (10) \rangle$ ,  $\langle (60) \rangle$ ,  $\langle (20) (90) \rangle$ ,  $\langle (30) (90) \rangle$  et  $\langle (30) (40,70) \rangle$ . La première fait partie des achats de  $C_2$  et  $C_3$ , alors que la dernière apparaît dans les séquences de données des clients  $C_2$  et  $C_4$ .

Dans la section suivante, nous présentons les principales approches de recherche de motifs.

## 2.2 Le point

Comme nous le disions en introduction, la problématique de la recherche de motifs semble proche de celle des règles d'association. Cependant, le fait que nous prenons en compte la temporalité des itemsets engendre des combinaisons supplémentaires qu'il convient d'examiner. De manière plus concrète, dans le cas de la recherche d'itemset, la taille de l'espace de recherche correspond à  $2^I$  où  $I$  correspond au nombre d'items différents. Si nous considérons une séquence  $\langle s_1 s_2 \dots s_m \rangle$  et que  $n_i = |s_i|$  représente la cardinalité d'un itemset alors l'espace de recherche, i.e. l'ensemble de toutes les séquences potentielles, est  $2^{n_1+n_2+\dots+n_m}$ .

### 2.2.1 Méthodes basées sur un parcours en largeur d'abord

La méthode GSP (*Generalized Sequential Patterns*) [SA96b] a été l'une des premières propositions pour résoudre la problématique des motifs séquentiels (ce travail fait suite à [AS95a]). Les auteurs, en définissant la problématique de l'extraction de motifs séquentiels, ont également proposé un algorithme reprenant les principes d'Apriori, conçu pour l'extraction de règles d'association. Les difficultés relatives à la prise en compte de la temporalité ont rapidement conduit à la mise en place d'une méthode



de génération de candidats adaptée à ce contexte. Celle-ci maintient cependant les principes d'une recherche "en largeur d'abord" puisque les candidats sont générés en fonction de leur longueur et non de leur préfixe.

### Algorithme pionnier : GSP et sa structure

Dans [AS95a] nous trouvons un résumé des techniques mises en œuvre depuis le début du projet Quest d'IBM. Ce projet est à l'origine de l'algorithme GSP [SA96b], extension de Apriori, lui-même destiné à reprendre l'algorithme AIS présenté dans [AIS93a].

GSP est un algorithme basé sur la méthode générer-élaguer mise en place depuis Apriori et destinée à effectuer un nombre de passes raisonnable sur la base de données. La technique généralement utilisée par les algorithmes de recherche de séquences est basée sur une création de candidats, suivie du test de ces candidats pour confirmer leur fréquence dans la base. Bénéficiant de propriétés relatives aux séquences et à leur fréquence d'apparition, ces techniques sont tout de même contraintes "d'essayer" des séquences avant de les déterminer fréquentes (ou non). Le principe de génération des candidats tient compte de la propriété d'anti-monotonie du support. Les candidats de tailles  $k$  sont générés par auto-jointure des séquences fréquentes de taille  $k-1$  en considérant des S-extensions (ajout d'une sous séquence) et des I-extensions (ajout dans le dernier itemset).

Pour évaluer le support de chaque candidat en fonction d'une séquence de données, GSP utilise une structure d'arbre de hachage destinée à organiser les candidats. Les candidats sont stockés en fonction de leur préfixe. Pour ajouter un candidat dans l'arbre des séquences candidates, GSP parcourt ce candidat et effectue la descente correspondante dans l'arbre. Pour trouver quelles séquences candidates sont incluses dans une séquence de données, GSP parcourt l'arbre en appliquant une fonction de hachage sur chaque item de la séquence de données. Quand une feuille est atteinte, elle contient des candidats potentiels pour la séquence de données.

Depuis la définition de la problématique, de nombreuses approches ont été proposées pour améliorer les temps d'extractions des motifs. Celles-ci sont basées principalement sur de nouvelles structures de données ou sur l'hypothèse que la base peut être maintenue en mémoire centrale.

### PSP

Les auteurs de [MCP98] estiment que l'arbre de hachage utilisé dans [AS95a, SA96b] présente un défaut qu'il est facile de constater. En effet lors de la recherche des feuilles susceptibles de contenir des candidats inclus dans la séquence analysée, la structure utilisée ne tient pas compte des changements de date entre les items de la séquence qui servent à la navigation. Par exemple, avec la séquence  $\langle (10\ 30) (20\ 40) \rangle$ , l'algorithme va atteindre la feuille du sommet 30 (fils de 10), alors que cette feuille peut contenir deux types de candidats :

- ceux qui commencent par  $\langle (10) (30) \dots$  d'un côté
- et ceux qui commencent par  $\langle (10\ 30) \dots$  de l'autre.

Le but est alors de mettre en place une structure d'arbre de préfixes, pour gérer les candidats. L'algorithme PSP (*Prefix Tree for Sequential Pattern*), destiné à exploiter cette structure, est basé sur la méthode générer-élaguer. Le principe de base de cette structure consiste à factoriser les séquences candidates en fonction de leur préfixe. Cette factorisation, inspirée de celle mise en place dans [AS95a], pousse plus loin l'exploitation des préfixes communs que présentent les candidats. En effet les auteurs proposent de prendre en compte les changements d'itemsets dans cette factorisation.

L'arbre de préfixes ainsi proposé ne stocke plus les candidats dans les feuilles mais permet de retrouver les candidats de la façon suivante : tout chemin de la racine à une feuille représente un candidat et tout candidat est représenté par un chemin de la racine à une feuille. De plus, pour prendre en compte le changement d'itemset, l'arbre est doté de deux types de branches. Le premier type, entre deux items, signifie que les items sont dans le même itemset alors que le second signifie qu'il y a un changement d'itemset entre ces deux items.

### SPADE

Dans [Zak01], les auteurs proposent l'approche SPADE. L'originalité de cet algorithme est de considérer une représentation verticale de la base de données. Dans ce cas, la base est transformée de manière à représenter pour chaque item de la base, et pour chaque séquence, son numéro d'itemsets correspondant dans la séquence. Ensuite, à l'aide d'une stratégie à la Apriori, i.e. génération de candidats-élagage, les motifs sont extraits sans accès à la base. Deux types de génération de candidats sont considérées : les jointures temporelles et les jointures naturelles. Les premières correspondent à des S-Extension : recherche uniquement les sous-séquences qui peuvent étendre une séquence. Les jointures naturelles sont des I-extension et dans ce cas, les séquences recherchées correspondent à celles qui possèdent même numéro d'itemsets dans les séquences.

### SPAM

La méthode SPAM [AFGY02] considère une représentation de la base de données sous la forme de vecteurs de bitmaps. L'idée générale est d'utiliser un arbre représentant en fait l'espace de recherche dans lequel sont élaguées les branches non fréquentes. A chaque étape des candidats sont générés via des opérateurs logiques entre les vecteurs. Ainsi, la S-extension nécessite tout d'abord de transformer la séquence à étendre et à appliquer un opérateur AND entre la séquence transformée et la séquence à ajouter. La I-extension se résume à appliquer uniquement un AND entre les deux séquences.

## 2.2.2 Méthodes basées sur une projection de la base

Plus récemment, de nouvelles propositions, considérant également que la base de données peut tenir en mémoire, se sont intéressées à projeter la base de données. C'est le principe adopté par [HPMa<sup>+</sup>00] avec FREESPAN et amélioré par [PHMa<sup>+</sup>01] avec l'algorithme PREFIXSPAN. PREFIXSPAN implémente de plus un principe de réécriture de la base de données en fonction des préfixes des motifs séquentiels fréquents découverts (ou d'une indexation en fonction de la mémoire disponible).

### PREFIXSPAN

Dans [HPMa<sup>+</sup>00], les auteurs proposent l'algorithme FREESPAN (*Frequent pattern projected Sequential pattern mining*). L'idée générale est de proposer des projections récursives de la base de données en fonction des items fréquents. La base est alors projetée en plusieurs bases plus petites et les séquences fréquentes grandissent avec le nombre de projections. Les temps de réponses sont alors améliorés car chaque base projetée est plus petite et facile à traiter. Ce travail est le point de départ d'autres études sur la projection de bases de données en recherche de motifs séquentiels. FREESPAN présente tout de même un défaut selon ses auteurs : une sous-séquence peut être générée par n'importe quelle combinaison dans une séquence, donc FREESPAN doit

conserver la totalité de la séquence dans la base d'origine sans réduire sa taille.

La méthode PREFIXSPAN, présentée dans [PHMa<sup>+</sup>01], se base sur une étude du nombre de candidats qu'un algorithme de recherche de motifs séquentiels peut avoir à produire afin de déterminer les séquences fréquentes. L'objectif des auteurs est alors de réduire le nombre de candidats générés. Pour parvenir à cet objectif, PREFIXSPAN propose (à l'instar de PSP avec les candidats) d'analyser les préfixes communs que présentent les séquences de données de la base à traiter. À partir de cette analyse, l'algorithme construit des bases de données intermédiaires qui sont des projections de la base d'origine déduites à partir des préfixes identifiés. Ensuite, dans chaque base obtenue, PREFIXSPAN cherche à faire croître la taille des motifs séquentiels découverts en appliquant la même méthode de manière récursive.

Deux sortes de projections sont alors mises en place pour réaliser cette méthode : la projection dite "niveau par niveau" et la "bi-projection". Au final, les auteurs proposent une méthode d'indexation permettant de considérer plusieurs bases virtuelles à partir d'une seule, dans le cas où les bases générées ne pourraient être maintenues en mémoire en raison de leurs tailles.

### 2.2.3 Recherche des motifs séquentiels fermés

L'extraction de motifs séquentiels devient problématique selon la longueur des motifs séquentiels extraits. Les auteurs de [YHA03] illustrent ce problème avec l'exemple d'une base de données ne contenant qu'un seul motif :  $\langle (a_1) (a_2) \dots (a_{100}) \rangle$ . Dans ce cas, il faudra générer  $2^{100} - 1$  sous-séquences fréquentes avec un support minimum de 1. Ces sous-séquences seront redondantes car elles auront toutes le même support que  $\langle (a_1) (a_2) \dots (a_{100}) \rangle$ . Dans [YHA03], les auteurs définissent donc la problématique de la recherche des motifs séquentiels fermés (*closed sequential patterns*), inspirée de la recherche d'itemsets fermés. Ils proposent CLOSPAN, le premier algorithme capable de résoudre ce problème et optimisé pour cela. Dans [WH04], l'approche BIDE utilise une nouvelle manière d'étendre les séquences et optimise l'espace de recherche en analysant à l'avance les motifs à étendre.

**Définition 5** Soit  $\sigma$ , le support minimum et  $FS$  l'ensemble des motifs séquentiels fréquents correspondants. L'ensemble des motifs séquentiels fermés  $CS$  est défini comme :

$$CS = \{s/s \in FS \text{ et } \nexists s' \text{ telle que } s \subset s' \text{ et } support(s') = support(s)\}.$$

#### CLOSPAN

CLOSPAN [YHA03] est une méthode basée sur le principe depth-first et implémente l'algorithme PREFIXSPAN. En fait, il s'agit d'une optimisation de ce dernier, destinée à élaguer l'espace de recherche en évitant de parcourir certaines branches dans le processus de divisions récursives (en détectant par avance les motifs séquentiels non fermés). Le principe de CLOSPAN repose sur deux éléments essentiels : l'ordre lexicographique des séquences et la détection de liens systématiques entre deux items (i.e. " $\beta$  apparaît toujours avant  $\gamma$  dans la base de données").

#### BIDE

Etant donné que CLOSPAN conserve l'historique des séquences candidates, il ne s'avère pas efficace dans le cas de bases contenant de trop nombreuses séquences

fermées. Pour pallier ce problème, une nouvelle approche, BIDE (BI-Directional Extension) est proposée dans [WH04]. L'idée générale est d'étendre les séquences dans les deux directions, i.e. en avant (forward extension) et en arrière (backward extension). En effet, considérons une séquence  $S = i_1 i_2 \dots i_n$ , celle-ci peut être étendue de trois manières possibles : ajout d'un item après  $i_n$ , ajout d'un item entre  $i_1 i_2 \dots i_n$ , ajout d'un item avant  $i_1$ . La première correspond à une extension en avant et les deux dernières à une extension en arrière. Ainsi, les auteurs montrent que pour une séquence  $S$ , s'il n'y a pas d'extension avant ni d'extension arrière alors  $S$  est une séquence fermée. Comme dans CLOSPAN, une base projetée est constituée. Pour une séquence  $S$ , son ensemble d'items extensibles en avant, i.e. les items qui peuvent être ajoutés à la fin de  $S$ , est constitué par les items locaux dont le support est égal à celui de la séquence. Ces items locaux sont simplement trouvés en parcourant la base projetée pour ce préfixe et en comptant le nombre d'items. Pour effectuer rapidement cette opération, la projection utilisée est une pseudo projection comme dans [PHW02]. De manière à définir les extensions possibles en arrière, il faut dans un premier temps rechercher, pour les items d'une séquence, quelles sont les extensions en arrière possibles. Pour cela, il est nécessaire de remonter dans la séquence pour examiner avec quel item il est possible de l'étendre [WH04].

## 2.3 Discussion

Depuis la définition de la problématique de la recherche de motifs séquentiels de nombreux travaux se sont intéressés non seulement à la définition d'algorithmes pour extraire ces motifs mais surtout à la recherche d'approches de plus en plus efficaces. En effet, il suffit de considérer les dernières évolutions pour s'en convaincre. Cette volonté est souvent liée aux différents domaines d'applications pour lesquels les motifs sont particulièrement adaptés. Par exemple, si nous considérons un site Web, en cherchant à extraire les motifs nous souhaitons mieux appréhender le comportement des utilisateurs. Cependant, si le processus d'extraction des motifs est trop long, les connaissances extraites ne sont plus forcément représentatives et peuvent même s'avérer obsolètes. Ainsi, même si la recherche de solutions efficaces est indispensable, il existe de nombreux problèmes qui n'ont malheureusement pas ou peu été abordés.

Considérons à nouveau le cas du site Web, il est évident que pendant que nous extrayons des motifs séquentiels concernant le comportement des utilisateurs, de nouveaux usagers viennent se connecter sur le site. La conséquence immédiate est que ce que nous avons appris concerne uniquement une période passée de la vie du site. Que se passe-t'il, comme c'est souvent le cas, si le comportement des nouveaux utilisateurs est totalement différent de celui des précédents? Quel est dans ce cas l'intérêt de conserver une connaissance qui ne correspond plus à une réalité. Bien entendu, nous pourrions imaginer d'effacer la connaissance apprise précédemment et d'appliquer à nouveau nos algorithmes sur les nouvelles données. Cette solution est irréaliste pour deux raisons : 1) ne tenir compte que des nouvelles données n'offre qu'une connaissance trop sommaire sur le comportement des internautes et n'est pas du tout représentative de ce qui se passe réellement dans la vie du site ; 2) si l'on tente de suivre le comportement à 'long terme' d'un utilisateur nous ne disposons plus des informations permettant de les extraire. La seule solution consiste donc à tirer profit des connaissances acquises précédemment. Dans ce cas, notre objectif est d'utiliser cette connaissance au mieux en évitant de ré-exécuter totalement le processus : on assemble la base de données initiale et on ajoute la dernière partie. En d'autres termes, nous souhaitons proposer une approche incrémentale.

Un autre problème inhérent aux motifs séquentiels est qu'un motif est soit fréquent

soit non fréquent. Cette séparation est bien entendu trop stricte pour de nombreux domaines d'applications. N'est-il pas plus intéressant de dire qu'un motif est un peu fréquent ou très peu fréquent ? Les différents projets de transfert de technologie, nous ont confirmé que souvent l'utilisateur ne voulait pas une réponse aussi stricte et qu'il fallait donc proposer d'"assouplir" les motifs. De la même manière, les données traitées dans les motifs sont booléennes : un client achète un produit ou n'achète pas un produit. Même s'il est facile d'étendre les approches traditionnelles à la prise en compte de duplicats dans les itemsets (bien entendu au prix d'un accroissement de l'espace de recherche), nous nous retrouvons confrontés à une notion de support trop stricte. Une solution à ce type de problème est de proposer des degrés d'appartenance et donc d'introduire des supports "flous".

L'avantage des motifs est que nous sommes à même d'extraire des séquences qui respectent un ordre. Par contre, la définition initiale des motifs n'offre pas de possibilité de contraindre les motifs extraits pour qu'ils rentrent dans une fenêtre temporelle ou qu'il existe au moins un délai  $t$  entre deux itemsets. La première proposition de ce type a été introduite dans GSP mais là aussi il est parfois difficile à l'utilisateur de spécifier de manière stricte ses contraintes. Un utilisateur sera plus intéressé de savoir que quatre ou sept jours après avoir acheté du chocolat, les clients ont acheté du café. Pour obtenir ce type de résultat une solution naïve serait de rechercher tous les motifs et ensuite d'appliquer une étape de post traitement. Cette approche souffre cependant de nombreuses lacunes (nous y reviendrons dans la partie perspectives du chapitre de conclusion). Nous souhaitons donc pousser les contraintes temporelles au cœur de l'algorithme de fouille tout en garantissant une certaine souplesse dans prise en compte des contraintes.

En revenant sur la relation d'ordre des séquences, une question se pose : traditionnellement les séquences considèrent une estampille temporelle mais que se passe-t'il si l'on considère l'ordre des phrases dans les documents ? en d'autres termes, est-ce que les motifs peuvent s'appliquer facilement à des données textuelles ? Intuitivement la réponse à cette question est oui mais ce qui est surtout intéressant de connaître c'est le gain que cela apporte par rapport aux approches de classification traditionnelle.

Les motifs considèrent une seule dimension. En effet, nous souhaitons connaître pour un client quels sont les achats qu'il a effectué. Même si la connaissance extraite est importante, il est intéressant de voir si nous ne pourrions pas extraire des motifs qui intègrent les connaissances supplémentaires que l'on peut avoir sur le client. Dans ce cas, nous cherchons à rechercher des motifs multi-dimensionnels.

Au cours des chapitres suivants, nous revenons sur ces problèmes et présentons nos solutions pour y répondre.

## Chapitre 3

# Extraction et incrémentalité des motifs extraits

Ces propositions ont été réalisées lors de l'encadrement de

**Doctorant :** Florent Massegia  
**Co-encadrant :** Pascal Poncelet (Professeur  
EMA, LIGI2P)

Ce chapitre adresse les problématiques

*Représentation des données :* Données classiques  
*Représentation des comportements :* Motifs séquentiels  
*Extraction de motifs :* ISE

### 3.1 Introduction

Etant donné que les bases de données évoluent, le problème de la maintenance des motifs séquentiels sur une longue période de temps devient essentiel puisqu'un grand nombre de nouveaux enregistrements peuvent être ajoutés à la base. Pour refléter le nouvel état courant de la base pour lequel d'anciens fréquents peuvent disparaître alors que de nouveaux peuvent apparaître, il est nécessaire de définir des algorithmes assurant la maintenance de la connaissance extraite. L'objectif de cette composante est de tirer profit de la connaissance acquise précédemment (en l'occurrence les fréquents) pour obtenir les nouveaux motifs séquentiels. En plus d'être adaptés à la problématique, ces algorithmes se doivent d'être efficaces car l'approche proposée nécessite d'être plus rapide que l'approche naïve qui consisterait à recommencer entièrement le processus d'extraction lors des mises à jour de la base. Les bénéfices, que l'on peut tirer d'un raisonnement incrémental sur la fouille de données, sont largement exploités pour les règles d'association [CHNW96, CLK97, AP95, SS98, TBAR97, RMR96, RMR97]. La problématique des motifs séquentiels se doit également de s'adapter au problème des mises à jour et de l'incrémentalité.

La problématique de la fouille incrémentale peut être définie plus formellement de la manière suivante. Soit  $DB$  la base de données d'origine et  $\sigma$  le support minimal. Soit  $db$  la base de données ajoutée contenant de nouvelles transactions et de nouveaux clients. Soit  $U = DB \cup db$  la base de données mise à jour contenant toutes les séquences de  $db$  et de  $DB$ . Soit  $L^{DB}$  l'ensemble des séquences fréquentes de  $DB$ . Le problème de la fouille incrémentale de motifs séquentiels consiste à rechercher toutes les séquences fréquentes en considérant la même valeur de support minimal.

Ce chapitre est organisé de la façon suivante. La section 3.2 présente les principaux travaux existants. Nous présentons notre approche, ISE, dans la section 3.3. Nous concluons cette partie par une discussion.

### 3.2 Le point

Peu de travaux concernent la prise en compte des mises à jour de la base de données dans le cadre des motifs séquentiels. Dans [PZOD99], les auteurs proposent un algorithme de recherche incrémentale, basé sur l'approche SPADE [Zak01], qui met à jour les motifs séquentiels d'une base de données lorsque de nouvelles transactions ou de nouveaux clients sont ajoutés. L'algorithme est basé sur un treillis de séquences constitué de toutes les séquences fréquentes et de toutes les séquences de la bordure négative de la base de d'origine. Cette bordure négative correspond à toutes les séquences qui ne sont pas fréquentes mais dont toutes les sous-séquences sont fréquentes. En outre, le support de toutes les séquences et sous-séquences est conservé dans le treillis. L'idée principale de l'algorithme est, lors d'une mise à jour, de parcourir la base incrément une première fois pour rajouter les informations dans le treillis. Ces nouvelles données sont alors combinées avec les séquences fréquentes et la bordure négative afin de déterminer quelle partie de la base d'origine doit être parcourue à nouveau. Même si cette approche est efficace (elle ne nécessite que peu de parcours de la base), la maintenance de la bordure négative est très difficile à gérer en mémoire et pour cela, les auteurs précisent que leur approche n'est utilisable que pour de petites bases.

Cust-Id	Itemsets		
C1	10 20	20	50 70
C2	10 20	30	40
C3	10 20	40	30
C4	60	90	

(DB)

Itemsets			
50 60 70	80 100		
50 60	80 90		

(db)

**Fig. 3.1:** Une base de données d'origine (DB) et une base de données incrément avec de nouvelles transactions (db)

### 3.3 Vers une approche incrémentale : ISE

#### 3.3.1 Principe

L'algorithme ISE (*Incremental Sequence Extraction*) résout le problème de la recherche incrémentale de séquences en utilisant les informations trouvées lors d'une extraction précédente. Considérons que  $k$  soit la longueur de la plus grande séquence trouvée lors d'une recherche précédente. Nous décomposons le problème de la manière suivante :

1. Rechercher toutes les nouvelles séquences de taille  $j \leq (k + 1)$ . L'objectif de cette phase est de rechercher parmi les séquences de DB si celles qui étaient fréquentes le reste en considérant l'incrément mais également de rechercher celles qui n'étaient pas fréquentes précédemment et qui le deviennent en ajoutant des données. Bien entendu, lors de cette phase, nous recherchons également les séquences qui seront fréquentes sur l'incrément. A l'issue de cette phase, nous obtenons donc toutes les séquences de DB qui deviennent fréquentes, i.e. le nombre d'apparition de ces séquences en prenant en compte DB et db est tel qu'il est supérieur au support, celles qui restent fréquentes avec l'incrément, les séquences fréquentes contenues dans l'incrément et enfin les extensions des fréquentes de DB auxquelles on ajoute un item de db.
2. Rechercher toutes les séquences fréquentes de taille  $j > (k + 1)$ .

Le second problème peut être résolu facilement en utilisant un algorithme comme PSP ou GSP dans la mesure où nous disposons, à la fin de la première phase, de toutes les séquences fréquentes de taille  $(k + 1)$ . Dans la suite de ce paragraphe, nous nous intéressons au premier problème.

Pour découvrir les séquences fréquentes de taille  $j \leq (k + 1)$ , l'algorithme ISE fonctionne de manière itérative.

#### *Première étape*

Dans la première passe sur db, nous comptons le support des items et nous obtenons ainsi l'ensemble  $1\text{-candExt}$  contenant les items qui interviennent au moins une fois dans db. En comparant ces items avec ceux de DB, nous obtenons l'ensemble  $L_1^{db}$  qui contient les items de db qui sont fréquents dans U. A la fin de cette phase, nous supprimons de  $L^{DB}$ , les séquences fréquentes qui ne vérifient plus le support.

**Exemple 4** Considérons la base de données incrément de la figure 3.1. En parcourant db, nous obtenons le support de chaque item :  $\{((\{50\}), 2), ((\{60\}), 2), ((\{70\}), 1), ((\{80\}), 2), ((\{90\}), 1), ((\{100\}), 1)\}$ . Considérons maintenant que nous avons obtenu les items suivants lors d'une extraction précédente sur DB.

item	10	20	30	40	50	60	70	90
support	3	3	2	2	1	1	1	1



En combinant ces items avec ceux de  $db$ , nous obtenons l'ensemble des 1-séquences fréquentes sur  $U$  et qui sont contenues dans  $db$  :  $L_1^{db} = \{\langle(50)\rangle, \langle(60)\rangle, \langle(70)\rangle, \langle(80)\rangle, \langle(90)\rangle\}$ .

Lors de la seconde étape, deux opérations sont effectuées en parallèle de manière à vérifier lors du parcours de la base différents candidats. La première concerne la génération des candidats de taille 2 situés dans l'incrément, i.e. ceux obtenus lors de la première étape. La seconde consiste à rechercher tous les séquences fréquentes qui précèdent un item de l'incrément.

*Seconde étape - a - Génération des 2-candidats dans l'incrément*

Les 1-séquences fréquentes de  $db$  sont utilisées pour générer de nouvelles 2-séquences candidates. Lors d'un nouveau parcours sur  $db$ , nous obtenons l'ensemble  $2-candExt$  composé des 2-séquences incluses au moins une fois dans  $db$ . Un parcours sur  $U$  avec les éléments de  $2-candExt$  permet de trouver les 2-séquences fréquentes qui sont insérées dans  $2-freqExt$ .

**Exemple 5** *Considérons l'ensemble  $L_1^{db}$  de l'exemple précédent. A partir de cet ensemble, nous pouvons générer les séquences suivantes  $\langle(50\ 60)\rangle, \langle(50\ (60))\rangle, \langle(50\ 70)\rangle, \langle(50\ (70))\rangle, \dots, \langle(80\ (90))\rangle$ . Pour découvrir  $2-candExt$  dans la base de données mise à jour, nous examinons si un item intervient au moins une fois. Par exemple, puisque le candidat  $\langle(50\ (60))\rangle$  n'apparaît pas dans  $db$ , il n'est pas considéré lors du parcours de  $U$ . A la fin du parcours de  $U$  avec les candidats restants, nous obtenons l'ensemble suivant des 2-séquences fréquentes  $2-freqExt = \{\langle(50\ 60)\rangle, \langle(50\ (80))\rangle, \langle(50\ 70)\rangle, \langle(60\ (80))\rangle\}$ .*

*Seconde étape - b - Recherche des itemsets fréquents précédents*

Une opération supplémentaire, pour rechercher dans  $DB$  les sous-séquences fréquentes de  $L^{DB}$  précédant les items de  $db$ , est réalisée sur les items découverts fréquents dans  $db$ . Pour rechercher efficacement ces sous-séquences fréquentes nous utilisons un tableau qui a autant d'éléments que le nombre d'items fréquents dans  $db$ . En parcourant les séquences de données de  $U$ , pour chaque sous-séquence, nous examinons si elle est incluse. Dans ce cas, le support de chaque sous séquence précédant l'item est incrémenté.

Lors de la passe pour déterminer  $2-freqExt$ , nous obtenons aussi l'ensemble de toutes les sous-séquences précédant les items de  $db$ . A partir de cet ensemble, en ajoutant les items de  $db$  aux sous-séquences fréquentes, nous obtenons un nouvel ensemble  $freqSeed$  contenant des nouvelles séquences fréquentes dont la taille est inférieure à  $k + 1$ .

**Exemple 6** *Considérons l'item 50 dans  $L_1^{db}$ . Pour le client  $C_1$ , 50 est précédé par les sous-séquences fréquentes suivantes :  $\langle(10)\rangle, \langle(20)\rangle$  et  $\langle(10\ 20)\rangle$ . Si nous considérons maintenant le client  $C_2$  avec la transaction mise à jour, nous obtenons l'ensemble de sous-séquences fréquentes précédant 50 suivant :  $\langle(10)\rangle, \langle(20)\rangle, \langle(30)\rangle, \langle(40)\rangle, \langle(10\ 20)\rangle, \langle(10\ (30))\rangle, \langle(10\ (40))\rangle, \langle(20\ (30))\rangle, \langle(20\ (40))\rangle, \langle(10\ 20\ (30))\rangle$  et  $\langle(10\ 20\ (40))\rangle$ . Ce principe est répété jusqu'à ce que toutes les transactions soient examinées. La figure 3.2 illustre les sous-séquences fréquentes ainsi que leurs supports sur  $U$ . L'ensemble  $freqSeed$  est obtenu en ajoutant à chaque item de  $L_1^{db}$  sa sous-séquence fréquente associée. Par exemple, en considérant l'item 70, les sous-séquences suivantes sont insérées dans  $freqSeed$  :  $\langle(10\ (70))\rangle, \langle(20\ (70))\rangle$  et  $\langle(10\ 20\ (70))\rangle$ .*

A la fin de la première passe sur  $U$ , nous disposons donc des 2-séquences fréquentes (dans  $2-freqExt$ ) et d'un ensemble de séquences fréquentes dont la taille est inférieure ou

Items	Sous-séquences Fréquentes
<b>50</b>	$\langle(10)\rangle_3 \langle(20)\rangle_3 \langle(30)\rangle_2 \langle(40)\rangle_2$ $\langle(10) (30)\rangle_2 \langle(10) (40)\rangle_2 \langle(20) (30)\rangle_2 \langle(20) (40)\rangle_2 \langle(10 20)\rangle_3$ $\langle(10 20) (30)\rangle_2 \langle(10 20) (40)\rangle_2$
<b>60</b>	$\langle(10)\rangle_2 \langle(20)\rangle_2 \langle(30)\rangle_2 \langle(40)\rangle_2$ $\langle(10) (30)\rangle_2 \langle(10) (40)\rangle_2 \langle(20) (30)\rangle_2 \langle(20) (40)\rangle_2 \langle(10 20)\rangle_2$ $\langle(10 20) (30)\rangle_2 \langle(10 20) (40)\rangle_2$
<b>70</b>	$\langle(10)\rangle_2 \langle(20)\rangle_2$ $\langle(10 20)\rangle_2$
<b>80</b>	$\langle(10)\rangle_2 \langle(20)\rangle_2 \langle(30)\rangle_2 \langle(40)\rangle_2$ $\langle(10) (30)\rangle_2 \langle(10) (40)\rangle_2 \langle(20) (30)\rangle_2 \langle(20) (40)\rangle_2 \langle(10 20)\rangle_2$ $\langle(10 20) (30)\rangle_2 \langle(10 20) (40)\rangle_2$
<b>90</b>	-

**Fig. 3.2:** Sous-séquences fréquentes intervenant avant les items de  $db$

égale à  $k+1$  (dans  $freqSeed$ ). Dans les itérations suivantes, nous sommes donc amenés à rechercher les séquences fréquentes qui ne sont pas encore dans ces deux ensembles.

#### Passes suivantes

Examinons maintenant les passes suivantes en considérant que nous sommes à la  $j^{ieme}$  passe avec  $j \leq k + 1$ . Nous commençons par générer de nouveaux candidats à partir des ensembles obtenus précédemment. L'idée principale est de retrouver parmi les séquences de  $freqSeed$  et de  $j-freqExt$ , deux séquences ( $s \in freqSeed, s' \in j-freqExt$ ) tels qu'un item  $i \in L_1^{db}$  soit le dernier item de  $s$  et le premier item de  $s'$ . Dès que la condition est vérifiée pour un couple  $(s, s')$ , une nouvelle séquence candidate est créée en supprimant le dernier item de  $s$  et en lui ajoutant  $s'$ . De manière complémentaire, nous générons à partir de  $j-freqExt$ , de nouvelles  $(j + 1)$ -séquences candidates en utilisant la même génération que GSP. Le support de tous les candidats est obtenu en parcourant la base  $U$  et nous obtenons respectivement  $freqInc$  et  $(j + 1)-freqExt$  qui sont utilisés pour générer de nouveaux candidats. Le processus s'arrête lorsque toutes les séquences fréquentes ont été découvertes ou que  $j = k + 1$ .

A la fin de cette phase, nous obtenons à partir de  $L^{DB}$  et des séquences maximales de  $freqSeed \cup freqInc \cup freqExt$ , l'ensemble  $L^{U_{k+1}}$  contenant toutes les séquences ayant une taille inférieure ou égale à  $k + 1$ .

**Exemple 7** Reprenons notre exemple, nous savons que  $k = 3$ , i.e. la plus grande taille des séquences fréquentes dans  $L^{DB}$ . Nous pouvons donc générer à partir de  $freqExt$  la nouvelle séquence candidate  $\langle(50 60) (80)\rangle$  puisque sa taille est inférieure à  $k$ .

Examinons à présent comment sont générées les nouvelles séquences à partir de  $freqSeed$  et  $2-freqExt$ . En considérant la séquence  $s = \langle(20) (40) (50)\rangle$  de  $freqSeed$  et  $s' = \langle(50 60)\rangle$  de  $2-freqExt$ , la nouvelle séquence candidate  $\langle(20) (40) (50 60)\rangle$  est obtenue en supprimant 50 de  $s$  et en  $s'$  à la séquence restante.

Les séquences maximales fréquentes telles que  $j \leq (k + 1)$  sont précisées dans la figure 3.3.

Une fois que toutes les séquences de taille  $j \leq (k + 1)$  sont découvertes, nous recherchons les nouvelles  $j$ -séquences fréquentes dans  $U$  avec  $j > k + 1$ . Pour cela, nous récupérons des trois ensembles précédents ( $freqSeed, freqExt$  et  $freqInc$ ) les  $(k+1)$ -séquences fréquentes. De nouvelles  $(k+2)$ -séquences candidates sont alors générées en

<i>freqInc</i>	<i>freqSeed</i>	<i>freqExt</i>
$\langle\langle(10) (50 60) (80)\rangle\rangle$	$\langle\langle(10 20) (30) (50)\rangle\rangle$	$\langle\langle(60) (90)\rangle\rangle$
$\langle\langle(20) (50 60) (80)\rangle\rangle$	$\langle\langle(10 20) (40) (50)\rangle\rangle$	
$\langle\langle(30) (50 60) (80)\rangle\rangle$	$\langle\langle(10 20) (30) (60)\rangle\rangle$	
$\langle\langle(40) (50 60) (80)\rangle\rangle$	$\langle\langle(10 20) (40) (60)\rangle\rangle$	
$\langle\langle(20) (30) (50 60)\rangle\rangle$	$\langle\langle(10 20) (30) (80)\rangle\rangle$	
$\langle\langle(20) (40) (50 60)\rangle\rangle$	$\langle\langle(10 20) (40) (80)\rangle\rangle$	
$\langle\langle(10 20) (50 60)\rangle\rangle$		
$\langle\langle(10) (30) (50 60)\rangle\rangle$		
$\langle\langle(10) (40) (50 60)\rangle\rangle$		
$\langle\langle(10) (30) (50 60)\rangle\rangle$		
$\langle\langle(10) (30) (50) (80)\rangle\rangle$		
$\langle\langle(10) (40) (50) (80)\rangle\rangle$		
$\langle\langle(20) (30) (50) (80)\rangle\rangle$		
$\langle\langle(20) (40) (50) (80)\rangle\rangle$		
$\langle\langle(10 20) (50) (80)\rangle\rangle$		
$\langle\langle(10 20) (50 70)\rangle\rangle$		

**Fig. 3.3:** Séquences fréquentes maximales telles que  $j \leq (k + 1)$

utilisant une approche similaire à GSP et le processus continue jusqu'à ce qu'il n'y ait plus de candidats à générer. En éliminant les séquences non maximales, i.e. les séquences incluses, nous obtenons  $L^U$ , l'ensemble de toutes les séquences fréquentes dans la nouvelle base de données mise à jour.

**Exemple 8** *A partir des  $(k+1)$ -séquences fréquentes découvertes dans l'exemple précédent, nous pouvons générer les séquences candidates suivantes :  $\langle\langle(10 20) (30) (50 60)\rangle\rangle$ ,  $\langle\langle(10 20) (40) (50 60)\rangle\rangle$ ,  $\langle\langle(20) (30) (50 60) (80)\rangle\rangle$  et  $\langle\langle(20) (40) (50 60) (80)\rangle\rangle$ . Comme elles sont fréquentes sur  $U$ , elles sont utilisées pour générer de nouveaux candidats à l'étape suivante. Nous obtenons donc à la fin du processus, i.e. dès qu'il n'est plus possible de générer des candidats, les deux séquences fréquentes suivantes :  $\langle\langle(10 20) (30) (50 60) (80)\rangle\rangle$  et  $\langle\langle(10 20) (40) (50 60) (80)\rangle\rangle$ .*

### 3.3.2 L'algorithme ISE

L'algorithme ISE, dont le principe repose sur les explications données jusqu'ici, est décrit figure 3.4.

Nous avons prouvé la validité de l'algorithme ISE (toutes les séquences fréquentes sont découvertes) [MPT03].

## 3.4 Discussion

L'approche ISE est basée sur la découverte de nouvelles séquences fréquentes en utilisant celles qui ont été extraites précédemment. Les expériences que nous avons menées aussi bien avec des données réelles (e.g. données d'usage du Web [MPT00]) que sur des jeux de données synthétiques (e.g. [MPT03]) ont montré que ce type d'approche était beaucoup plus efficace que de recommencer tout le processus. Nous avons même pu constater, lors de nos expérimentations, que l'approche était plus efficace pour extraire des motifs que GSP. L'idée dans ce cas consiste à initier le processus avec un sous ensemble de la base de données sur lequel GSP est appliqué et de découper le reste de la base de données en différents "incrémentes" qui seront traités via ISE.

**Algorithm ISE**

**Input :**  $DB$  la base de données d'origine,  $L^{DB}$  l'ensemble des séquences fréquentes sur  $DB$ , l'incrément  $db$  et  $\sigma$  le support minimum.

**Output :**  $L^U$  l'ensemble des séquences fréquentes sur  $U = DB \cup db$

**Method :**

```

//première phase sur db
 $L_1^{db} \leftarrow \emptyset$ 
foreach  $i \in db$  do
  if ( $support_{DB \cup db}(i) \geq \sigma$ ) then  $L_1^{db} \leftarrow L_1^{db} \cup \{i\}$ ;
enddo
 $2\text{-candExt} \leftarrow L_1^{db} \times L_1^{db}$ ;
Eliminer de  $2\text{-candExt}$  les séquences  $s/s \notin db$ ;
générer l'ensemble des sous-séquences de  $L^{DB}$ ;
passe sur  $U$  : valider les  $2\text{-candExt}$  et construire  $freqSeed$ ;
 $2\text{-freqExt} \leftarrow$  frequent sequences from  $2\text{-candExt}$ ;

// Phases suivantes
j=2;
While ( $j\text{-freqExt} \neq \emptyset$ ) do
   $candInc \leftarrow$  générer les candidats depuis  $freqSeed$  et  $j\text{-freqExt}$ ;
  j++;
   $j\text{-candExt} \leftarrow$  générer les candidats depuis  $j\text{-freqExt}$ ;
  Filtrer les séquences de  $j\text{-candExt}$   $s/s \notin db$ ;
  if ( $j\text{-candExt} \neq \emptyset$  OR  $candInc \neq \emptyset$ ) then
    Valider  $j\text{-candExt}$  et  $candInc$  sur  $U$ ;
  endif
  Mettre à jour  $freqInc$  et  $j\text{-freqExt}$ ;
enddo
 $L^U \leftarrow L^{DB} \cup \{séquences\ maximales\ de\ freqSeed \cup freqInc \cup freqExt\}$ ;
end Algorithm ISE

```

**Fig. 3.4:** L'algorithme ISE

La difficulté principale dans le cas de la recherche incrémentale est que toute mise à jour de la base de données peut avoir des conséquences sur le support de la base. Ainsi, par exemple, une séquence  $s$  qui était fréquente à un moment donné peut devenir non fréquente lorsque l'on ajoute un incrément. La conséquence immédiate est alors de supprimer  $s$  de la connaissance acquise. Considérons à présent, qu'à partir d'un certain moment, la séquence  $s$  apparaisse à nouveau. Cette séquence même si elle est fréquente dans la globalité des données peut ne pas pouvoir réapparaître car elle ne sera pas assez fréquente via les incréments. Pour pallier ce problème, IncSpan [CYH04] propose de considérer deux types de support : celui de fréquent, i.e.  $\sigma$ , et un support de semi fréquent (appelé  $\mu$ ). Le principe général est alors de conserver dans l'ensemble des semi fréquents les séquences dont le nombre d'occurrences est compris entre  $\mu$  et  $\sigma$  et de ne supprimer que les séquences infréquentes (i.e.  $\leq \mu$ ). Ce principe peut bien entendu être facilement adapté à ISE en stockant les séquences supprimées dans un ensemble de semi fréquent.

Quelque soit l'approche retenue pour extraire des motifs de manière incrémentale, nous avons pu constater qu'il est difficile de savoir à partir de quelle taille d'incrément il fallait exécuter ISE. Répondre à cette question est difficile car de nombreux paramètres interviennent (taille des séquences, items communs, nouveaux items, nouveaux clients, ...) et il serait intéressant d'avoir des mesures qui permettent d'optimiser les exécutions. Une piste de recherche pourrait être d'utiliser des techniques d'échantillonnage pour estimer la différence entre l'ancienne et la nouvelle base (e.g. une telle approche avait été proposée par [LCK98] pour les règles d'association). Nous pensons toutefois que ce problème peut, à l'heure actuelle, être généralisé à celui des flots de données. En effet, dans le cas des flots, nous avons une base de données qui est constamment mise à jour. Nous reviendrons sur cet aspect lors des perspectives de recherche que nous proposons dans le chapitre de conclusion.

## Chapitre 4

# Les motifs séquentiels flous

Ces propositions ont été réalisées lors de l'encadrement de

**Doctorant :** Céline Fiot  
**Co-encadrant :** Anne Laurent (Maître de Conférences,  
UMII, LIRMM)

Ce chapitre adresse les problématiques

<i>Représentation des données :</i>	Données classiques
<i>Représentation des comportements :</i>	Motifs séquentiels flous
<i>Extraction de motifs :</i>	SPEEDYFUZZY, MINIFUZZY et TOTALLY-FUZZY

## 4.1 Introduction

La plupart des bases de données issues du monde réel sont constituées de données numériques et historisées (e.g. données de capteurs, données démographiques, ...). Dans le cadre de la fouille de grandes bases de données, peu de travaux ont été réalisés pour traiter cette problématique et la majorité des propositions rentrent dans le contexte des règles d'association [FWS<sup>+</sup>98, KFW98, SA96a]. Par exemple, dans [SA96a], les auteurs traitent les données quantitatives pour la recherche de règles d'association grâce à un découpage des attributs en intervalles discrets. Toutefois, ce découpage trop "strict" peut dissimuler des associations fréquentes en raison des bornes trop restrictives des différents intervalles. De manière à permettre des coupures moins brutales entre les intervalles, les travaux de [KFW98] proposent une extension des règles d'association, basée sur la théorie des ensembles flous, pour permettre de raisonner ainsi sur des attributs quantitatifs.

A l'heure actuelle, il existe peu de travaux qui prennent en compte les données numériques lors de l'extraction de motifs séquentiels tout en se basant sur la théorie des ensembles flous. Soit  $\mathcal{D}$  une base de données transactionnelle où chaque transaction  $t$  est un  $n$ -uplet de  $\mathcal{D}$ . Soit l'ensemble  $\mathcal{I}$  des attributs  $i$  apparaissant dans  $\mathcal{D}$ . On note  $t[i]$  la valeur de l'attribut  $i$  pour la transaction  $t$ . A chaque attribut  $i$ , on associe plusieurs sous-ensembles flous, qui définissent une partition floue. Soit l'ensemble  $\mathcal{F}_i = \{F_i^1, F_i^2, \dots, F_i^{l_i}\}$  de sous-ensembles flous associés à l'attribut  $i$ . On note  $\mu_{F_i^{\lambda_i}}(t[i])$  la fonction d'appartenance de l'attribut  $i$  de la transaction  $t$  au sous-ensemble flou  $F_i^{\lambda_i}$ . On considère que ce découpage ainsi que les fonctions d'appartenance aux sous-ensembles flous sont fournis par un expert du domaine. Le problème de la fouille de motifs séquentiels flous consiste à rechercher toutes les séquences fréquentes d'items flous selon le mécanisme de comptage adopté et en considérant la même valeur de support minimal.

Dans ce chapitre, nous présentons une approche complète et efficace d'extraction de motifs séquentiels flous qui permet de considérer les données numériques. Cette approche est fondée sur la définition d'intervalles et plus précisément sous forme d'intervalles flous. Nous définissons trois approches SPEEDYFUZZY, MINIFUZZY et TOTALLYFUZZY qui diffèrent dans leur définition du support. Ceci permet à l'utilisateur final de choisir entre rapidité d'obtention des résultats et précision des motifs fréquents obtenus.

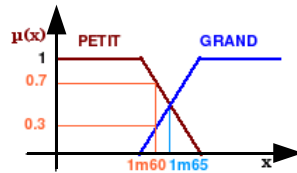
Ce chapitre est structuré de la façon suivante. La section 4.2 propose un rappel sur la théorie des sous-ensembles flous. Nous revenons sur les quelques travaux existants dans la section 4.3. La section 4.4 présente les trois algorithmes (SPEEDYFUZZY, MINIFUZZY et TOTALLYFUZZY) en détaillant les différentes définitions du support et leur implication dans le parcours des données. Nous concluons ce chapitre par une discussion.

## 4.2 Théorie des sous-ensembles flous : quelques rappels

La théorie des sous-ensembles flous, introduite par [Zad65] autorise l'appartenance partielle à une classe et la gradualité de passage d'une situation à une autre. Cette théorie constitue une généralisation de la théorie ensembliste classique, des situations

intermédiaires entre le tout et le rien étant admises. Un objet peut alors appartenir à la fois à un ensemble et à son complément.

**Exemple 1** On considère par exemple l'univers  $X$  des tailles possibles d'un individu. Un sous-ensemble flou  $A$  (e.g. PETIT ou GRAND) est défini par une fonction d'appartenance  $\mu_A$  qui décrit le degré avec lequel chaque élément  $x \in X$  appartient à  $A$ , ce degré étant compris entre 0 et 1. Par exemple, la figure 4.1 illustre une représentation possible de ces sous-ensembles flous. Ainsi, un individu de 1m65 pourra à la fois être grand et petit avec un degré de 0.7 pour le sous-ensemble flou GRAND et 0.3 pour le sous-ensemble flou PETIT.



**Fig. 4.1:** Représentation des sous-ensembles flous GRAND et PETIT relatifs à la taille d'un individu

Les opérateurs en logique floue sont une généralisation des opérateurs classiques. On considère notamment la négation, l'intersection et l'union. L'opérateur  $\top$  ou *t-norme* (norme triangulaire) est l'opérateur binaire d'intersection :  $\mu_{A \cap B}(x) = \top(\mu_A(x), \mu_B(x))$ . L'opérateur  $\perp$  ou *t-conorme* (conorme triangulaire) est l'opérateur d'union :  $\mu_{A \cup B}(x) = \perp(\mu_A(x), \mu_B(x))$ . Nous noterons  $\overline{\top}$  (resp.  $\underline{\perp}$ ) l'opérateur  $\top$  (resp.  $\perp$ ) généralisé au cas n-aire.

Il existe plusieurs opérateurs de t-norme et de t-conorme (min/max, produit/somme probabiliste, ...) ayant diverses propriétés. L'opérateur *min* étant idempotent nous avons choisi d'utiliser le couple *min* et *max*, respectivement utilisés pour la t-norme et la t-conorme.

Il existe de nombreuses possibilités pour représenter les opérateurs  $\top$  et  $\perp$ . Le tableau proposé figure 4.2 récapitule plusieurs fonctions fréquemment utilisées comme opérateurs  $\top$  et  $\perp$ .

Appellation	$\top$	$\perp$	NON
Zadeh [Zad65]	$\min(x, y)$	$\max(x, y)$	$1 - x$
Probabiliste [DP80]	$xy$	$x + y - xy$	$1 - x$
Lukasiewicz [Luk67]	$\max(x + y - 1, 0)$	$\min(x + y, 1)$	$1 - x$
Hamacher [Ham76] ( $\beta > 0$ )	$\frac{xy}{\beta + (1-\beta)(x+y-xy)}$	$\frac{x+y-xy-(1-\beta)xy}{1-(1-\beta)xy}$	$1 - x$
Weber [Web83]	$\begin{cases} x & \text{si } y = 1 \\ y & \text{si } x = 1 \\ 0 & \text{sinon} \end{cases}$	$\begin{cases} x & \text{si } y = 0 \\ y & \text{si } x = 0 \\ 1 & \text{sinon} \end{cases}$	$1 - x$

**Fig. 4.2:** Principales t-normes et t-conormes duales



Nous savons que la cardinalité d'un ensemble non-flou est le nombre d'éléments qui appartiennent à cet ensemble. Dans le cas d'un ensemble flou, le problème de comptabiliser le nombre d'éléments qu'il contient revient en fait à définir quand un élément appartient ou non à un ensemble flou. Dans le contexte des motifs séquentiels différentes méthodes peuvent être envisagées :

- Comptabiliser tous les éléments pour lesquels le degré d'appartenance est non nul et que quelle que soit la quantité.
- Considérer les achats pour lesquels le degré d'appartenance dépasse un certain seuil et quelle que soit la quantité, tous les éléments sont équivalents. Ce comptage est appelé *comptage seuillé*.
- Considérer que tous les achats n'ont pas la même importance, selon leur degré d'appartenance. Il s'agit alors d'un  $\Sigma$ -comptage, i.e. en sommant les degrés d'appartenance de chaque élément.
- Combiner les deux comptages précédents, en considérant que tous les éléments n'ont pas la même importance mais qu'ils ne sont assez significatifs pour être comptabilisés si leur degré d'appartenance ne dépasse pas un certain seuil. On réalise alors un  $\Sigma$ -comptage seuillé.

Bien entendu, l'utilisation de chacun de ces comptages est fonction des objectifs du comptage et de sa signification. Nous verrons par la suite que dans notre contexte, cela aura de l'influence sur la définition du support.

La figure 4.3 récapitule l'ensemble des notations qui seront utilisées dans la suite de ce chapitre.

On utilisera les notations  $\overline{\top}$  et  $\perp$  respectivement pour la T-norme et la T-conorme n-aires, extension des opérateurs binaires de T-norme et T-conorme  $\top$  et  $\perp$ . On définit également un opérateur n-aire d'agrégation  $\odot$ , qui doit être commutatif (il ne faut pas que l'ordre des calculs influe sur le résultat) et monotone (si l'on associe dans une séquence  $S_1$  un itemset A avec un itemset B, et dans  $S_2$  A avec C, tels que  $B \subseteq C$ , on souhaite que l'agrégation dans les séquences  $S_1$  et  $S_2$  rende compte de cet ordre).

Intuitivement, pour obtenir des motifs séquentiels flous, il s'agit de partitionner les quantités de chaque item ou produit acheté en plusieurs sous-ensembles flous puis d'utiliser ces sous-ensembles flous pour la recherche de séquences fréquentes.

### 4.3 Le point

La première proposition d'une approche de recherche de motifs séquentiels flous a été réalisée par [HLW01]. Leur proposition est basée sur un découpage en intervalles flous. Cependant, pour minimiser le nombre d'items flous manipulés, ils ne conservent, pour chaque item, que le sous-ensemble flou de cardinal le plus élevé pour toute la base (par  $\Sigma$ -comptage).

[CTCH01, HCTS03] ont adopté quant à eux une approche très théorique du problème sans algorithme ou implémentation. Leur proposition présente un formalisme et des notations ambigus pour le calcul du support d'un itemset flou et donc d'une séquence floue. Il est notamment difficile d'identifier les différences dans le calcul du support des séquences  $\langle(10) (20)\rangle$  et  $\langle(10 20)\rangle$ . Or ce point est fondamental dans un contexte de recherche de motifs séquentiels puisque les dates associées aux items interviennent lors de l'extraction des fréquents.

Ensemble des transactions	$\mathcal{T}$
Une transaction	$t$
Nombre total de transactions	$ \mathcal{T}  = \Theta$
Ensemble des clients	$\mathcal{C}$
Un client	$c$
Nombre total de clients	$ \mathcal{C}  = \Gamma$
Ensemble des transactions du client $c$	$\mathcal{T}_c$
Nombre de transactions du client $c$	$ \mathcal{T}_c  = \theta_c$
Ensemble des items	$\mathcal{I}$
Un item	$i$
Nombre total d'items	$ \mathcal{I}  = I$
Sous-ensembles flous associés à l'item $i$	$\mathcal{F}_i = \{F_i^1, \dots, F_i^{l_i}\}$
Nombre de sous-ensembles flous associés à $i$	$ \mathcal{F}_i  = l_i$
Indice utilisé pour le parcours des éléments de $\mathcal{F}_i$	$\lambda_i = 1 \dots l_i$
Fonction d'appartenance d'un item $i$ de $t$ à un sous-ensemble flou $F_i^{\lambda_i}$	$\mu_i^{\lambda_i}(t[i]) = \mu_{F_i^{\lambda_i}}(t[i])$
Fonction d'appartenance seuillée d'un item $i$ de $t$ à un sous-ensemble flou $F_i^{\lambda_i}$	$\alpha_i^{\lambda_i}(t[i]) = \alpha_{F_i^{\lambda_i}}(t[i])$
Un item flou	$(i, F_i^\lambda)$
Un itemset flou	$(X, A) = ([x_1, a_1] \dots [x_p, a_p])$ ou $(X, A) = (\{x_1, \dots, x_p\}, \{a_1, \dots, a_p\})$ avec $x_j \in \mathcal{I}$ et $a_j \in \mathcal{F}_{x_j}$
Une séquence floue	$S$ , composée d'itemsets flous $s$
Une $g$ - $k$ -séquence floue	$S = \langle s_1 \dots s_g \rangle$ où $S$ contient $k$ items flous
Support	$Supp$
Support seuillé	$TSupp$
Support flou	$FSupp$
Confiance floue	$FConf$
Seuil d'appartenance minimale	$\omega$

Fig. 4.3: Notations

## 4.4 Vers des motifs séquentiels flous

Pour pallier les lacunes de ces approches de recherche de motifs séquentiels flous, nous proposons tout d'abord une définition des concepts associés : item, itemset,  $g$ - $k$ -séquence et support flous [FDLT04]. Nous en proposons ensuite plusieurs mises en œuvre possibles en fonction, en particulier, de la méthode de comptage adoptée (C.f. section 4.2) et proposons trois algorithmes, associés chacun à une définition précise du support, permettant ainsi trois niveaux de "fuzzification" lors de la recherche de motifs séquentiels flous.

### 4.4.1 Principe

Les notions d'item et d'itemset sont redéfinies par rapport aux motifs séquentiels classiques. Un **item flou** est l'association d'un item et d'un sous-ensemble flou correspondant noté  $[x, a]$  avec  $x$  un item et  $a$  un sous-ensemble flou. Par exemple,  $[bonbons, beaucoup]$  est un item flou où  $beaucoup$  est un sous-ensemble flou défini par une fonction d'appartenance. Un **itemset flou** est alors défini comme un ensemble d'items

floos. On note  $(X, A)$  un itemset flou,  $X$  étant un ensemble d'items et  $A$  un ensemble de sous-ensembles floos associés. Par exemple,  $(X, A) = ([bonbons, beaucoup][soda, peu])$  est un itemset flou. Enfin, une  $g$ - $k$ -séquence  $S = \langle s_1 \cdots s_g \rangle$  est définie comme une séquence composée de  $g$  itemsets floos  $s = (X, A)$  regroupant au total  $k$  items floos. Par exemple, la séquence  $\langle ([bonbons, beaucoup][jeuxvideos, peu])([soda, beaucoup]) \rangle$  regroupe 3 items floos au sein de 2 itemsets. Il s'agit d'une 2-3-séquence floue.

Nous utiliserons dans la suite de cet article les notations suivantes :  $\mathcal{C}$  représente l'ensemble des clients et  $\mathcal{T}_c$  est l'ensemble des transactions d'un client  $c$ .  $\mathcal{I}$  représente l'ensemble des attributs et  $t[i]$  la valeur de l'attribut  $i$  pour la transaction  $t$ . Chaque attribut  $i$  est partitionné en sous-ensembles floos.

Pour illustrer les différentes définitions, nous utilisons la base d'achats décrite figure 4.4 (une case vide indique que le produit n'a pas été acheté). Nous considérons le degré minimal  $\omega = 0.49$  et le support minimal  $minSupp = 0.55$ . Pour les opérateurs  $\overline{\quad}$  et  $\underline{\quad}$ , nous adoptons respectivement  $min$  et  $max$ . L'opérateur d'agrégation  $\odot$  correspond à la moyenne.

Dans un premier temps, il s'agit de convertir la base des quantités en base de degrés d'appartenance. Chaque attribut est donc partitionné en sous-ensembles floos selon la figure 4.5 qui représente les fonctions d'appartenance pour chaque sous-ensemble. La construction des partitions est réalisée de manière automatique en séparant les univers des quantités en intervalles en regroupant la même proportion de clients et en "fuzzifiant" ensuite ces intervalles pour garantir une meilleure généralisation.

A partir des fonctions d'appartenance ci-dessus, on obtient les degrés d'appartenance de chaque transaction pour chacun des sous-ensembles floos. La figure 4.6 décrit ces valeurs pour l'ensemble des transactions du client 1.

Clients	Date	Items				
		bonbons	dentifrice	soda	ballon	jeu video
C1	d1	2				
	d2	1	3	1		
	d3	4		1		
	d4			1	5	
	d5			2		2
C2	d1	2		2	1	
	d2			1		
	d3		4			
	d4	3				
C3	d1		1			3
	d2	3				
	d3				4	5
	d4			2		
	d5		2			
C4	d1					
	d2					
	d3	2				4
	d4			3		
	d5		2			
	d6			2		

Fig. 4.4: Transactions

Le **support d'un itemset flou** se calcule comme le pourcentage de clients supportant cet itemset flou par rapport au nombre total de clients dans la base :

$$FSupp_{(X,A)} = \frac{\sum_{c \in \mathcal{C}} [S(c, (X, A))]}{|\mathcal{C}|}$$

où le degré support  $S(c, (X, A))$  indique si le client  $c$  supporte l'itemset flou  $(X, A)$ .

Nous avons vu précédemment, section 4.2, que la cardinalité d'un sous-ensemble flou dépend de la technique de comptage utilisée. Nous transposons trois de ces techniques dans le contexte des motifs séquentiels floos et proposons trois définitions du

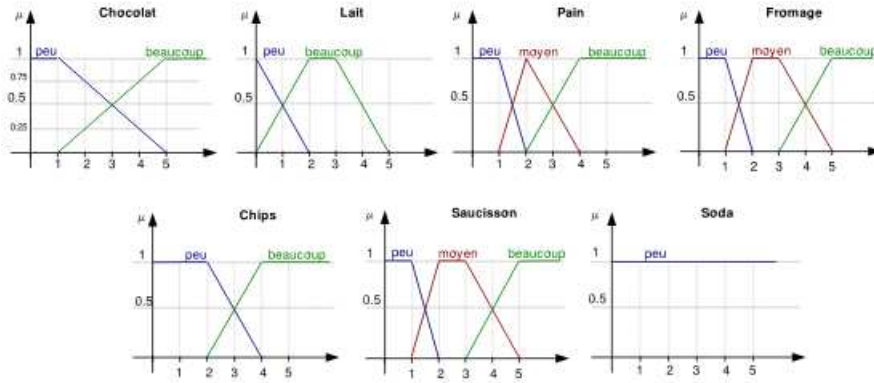


Fig. 4.5: Fonctions d'appartenance pour les différents sous-ensembles d'attribut

	Date	Items													
		bonbons		p	dentifrice		soda		p	ballon		jeu video			
		peu	bcp			my	bcp	peu		bcp		my	bcp	p	my
C1	d1	0.75	0.25												
	d2	1			0.5	0.5	0.5	0.5							
	d3	0.25	0.75				0.5	0.5							
	d4						0.5	0.5							
	d5							1			1				

Fig. 4.6: Degrés d'appartenance pour le Client 1

support :

- SPEEDYFUZZY se base sur le comptage “supporte / ne supporte pas” (première méthode de comptage). Pour le support d'un itemset flou, cela consiste à comptabiliser chaque client ayant réalisé au moins une fois la séquence d'achats. Quel que soit le degré d'appartenance de l'achat du client pour l'item flou, si celui-ci est non nul, chaque client aura le même poids :

$$S_{SpeedyFuzzy}(c, (X, A)) = 1 \text{ si } \forall [x, a] \in (X, A), \mu_a(t[x]) > 0 \text{ et } 0 \text{ sinon.}$$

- MINIFUZZY repose sur un comptage seuillé (deuxième méthode de comptage). Dans cette méthode, le nombre de clients supportant un itemset flou n'est incrémenté que si chaque item de la séquence candidate vérifie le seuil pour le degré d'appartenance dans la séquence d'achats du client :

$$S_{SpeedyFuzzy}(c, (X, A)) = 1 \text{ si } \forall [x, a] \in (X, A), \mu_a(t[x]) > \omega \text{ et } 0 \text{ sinon.}$$

- TOTALLYFUZZY réalise un  $\Sigma$ -comptage seuillé (quatrième méthode de comptage). Il s'agit de prendre en compte l'importance de chaque itemset flou parmi les transactions d'un client dans le calcul du support. Pour cela, on définit  $\alpha$  qui représente la fonction d'appartenance seuillée :

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

Nous obtenons alors :  $S_{TotallyFuzzy}(c, (X, A)) = \underline{\perp}_{j=1}^{\theta_c} \overline{\top}_{[x,a] \in (X,A)} [\alpha_a(t_j[x])]$ , où  $\overline{\top}$  et  $\underline{\perp}$  sont les opérateurs de t-norme et t-conorme généralisés.

Le  $\Sigma$ -comptage (troisième méthode) est en fait un  $\Sigma$ -comptage seuillé, avec un seuil  $\omega$  à zéro, il n'est donc pas nécessaire d'en faire un cas particulier.

#### 4.4.2 Les algorithmes SPEEDYFUZZY, MINIFUZZY et TOTALLYFUZZY

La démarche adoptée pour SPEEDYFUZZY et MINIFUZZY est similaire. Pour chaque client, il s'agit de parcourir l'ensemble de ses transactions pour trouver la séquence candidate. Pour chaque itemset de la séquence, il est nécessaire de vérifier si le degré d'appartenance est : non nul pour SPEEDYFUZZY ou supérieur au seuil  $\omega$  pour MINIFUZZY. Dès que la séquence candidate est validée (l'ensemble ordonné des itemsets est supporté par le client), le parcours dans les transactions du client est stoppé et le support de la séquence est incrémenté.

L'algorithme SPEEDYFUZZY (resp. MINIFUZZY) se base sur la fonction de calcul du support *CalcSpeedySupp* (resp. *CalcMiniSupp*) et la fonction *FindSpeedy-Seq* (resp. *FindMiniSeq*) qui recherche une séquence candidate parmi toutes les transactions d'un client.

L'algorithme TOTALLYFUZZY quant à lui est plus complexe car il repose sur un  $\Sigma$ -comptage seuillé. Ainsi pour chaque client et chaque séquence, il faut considérer le meilleur degré d'appartenance des itemsets parmi les transactions de ce client. Ce degré est calculé comme l'agrégation des supports des itemsets de la séquence. Il faut également respecter l'ordre des itemsets flous retenus. Ceci impose un parcours exhaustif des transactions. Néanmoins, à partir de la démarche proposée dans [FDLT04], nous proposons une mise en œuvre efficace d'un tel parcours par l'intermédiaire de la notion de chemin. Un chemin correspond à une instantiation possible des itemsets de la séquence candidate dans les transactions du client. Plusieurs chemins peuvent être initiés pour un client et il s'agit de conserver celui de plus fort degré, s'il est complet, pour le calcul du support.

##### Illustration

Toujours à partir des transactions du client 1, figure 4.6, nous allons illustrer la démarche adoptée par TOTALLYFUZZY pour calculer le support de la séquence candidate  $g-S = \langle ([bonbons, peu])([soda, beaucoup]) \rangle$ , avec un seuil  $\omega$  à 0.2. Un chemin est un triplet contenant la séquence déjà trouvée, l'item suivant recherché ainsi que les degrés d'appartenance des différents itemsets. Pour initier le processus, il y a création d'un chemin  $ch1$  qui contient  $(\emptyset, ([bonbons, peu]), 0)$  qui correspond à la séquence déjà trouvée (*seq*), l'item recherché (*rechCour*) et le degré d'appartenance (*degCour*). Pour la transaction  $d1$ ,  $rechCour = d1[1]$ , donc le chemin  $ch1$  est mis à jour par  $(\langle ([bonbons, peu]) \rangle, ([soda, bcp]), 0.75)$ .

Ensuite, pour la transaction  $d2$ , un nouveau chemin est créé  $ch2 \leftarrow (\langle ([bonbons, peu]) \rangle, ([soda, bcp]), 1)$  car  $d2$  contient l'item  $[bonbons, peu]$  qui est le premier item de la séquence candidate.  $ch1$  est mis à jour car  $d2$  contient  $[soda, bcp]$ .  $ch1$  est clos car il contient tous les éléments de la séquence  $g-S$ . La transaction  $d3$  est ensuite examinée. Elle contient  $ch2.rechCour$ , le chemin  $ch2$  est donc modifié en  $(\langle ([bonbons, peu]) \rangle, ([soda, bcp]), \emptyset, 0.75)$ , ce chemin est clos. Néanmoins, le parcours est optimisé et ne conserve, pour deux chemins au même stade de parcours, que celui de meilleur degré d'appartenance. Ainsi, le chemin  $ch1$  est éliminé de la liste des chemins pour le client 1 car ayant un degré d'appartenance inférieur à  $ch2$ . Le parcours se poursuit ensuite comme indiqué figure 4.7.

##### Les fonctions utilisées

L'algorithme TOTALLYFUZZY utilise la fonction *FindTotallySeq* qui réalise un parcours ordonné des transactions d'un client. Lorsque le premier itemset de la séquence est trouvé, un chemin est créé avec le support de l'itemset.

**Fonction FindTotallySeq**

**Données** :  $g$ -S,  $g$ - $k$ -séquence candidate,  $T$ , ensemble de transactions à parcourir

**Résultat** :  $m$ , le degré support de la meilleure représentation de  $g$ -S instanciée parmi les transactions de  $T$

**début**

```

  Chemins ← liste de parcours → (seq, rechCour, degCour)
  //- seq est la sous-séquence de  $g$ -S déjà trouvée, rechCour est l'itemset
  suivant dans  $g$ -S
  degCour est la liste des degrés d'appartenance des différents itemsets de seq -

  //-initialisation-
  Chemins ← Chemin( $\emptyset$ ,  $g$ S.first, 0)
  pour chaque transaction  $t \in T$  faire
  | pour chaque parcours  $ch \in Chemins$ , non mis à jour à la
  | transaction  $t$  faire
  | | si  $ch$  non clos alors
  | | | si  $ch.rechCour \in t$  alors
  | | | | //-  $t$  contient l'itemset si le degré de chaque item de l'itemset
  | | | | est supérieur au seuil  $\omega$ -
  | | | |  $ch.degCour \leftarrow ch.degCour - \overline{\top}_{[x,a] \in ch.rechCour} \alpha_a(t[x])$ 
  | | | | Update( $ch$ )
  | | |
  | | | pour  $j = 2$  à  $ch.rechCour - 1$  faire
  | | | | //- on recherche ici une amélioration possible du chemin courant-
  | | | | si  $(gS.get(j) \in t) \ \&\& \ (\overline{\top}_{[x,a] \in gS.get(j)} \alpha_a(t[x]) > ch.degCour[j])$ 
  | | | | alors
  | | | | | newRechCour ←  $gS.get(j)$ 
  | | | | | pour  $i = 1$  à  $j-1$  faire
  | | | | | | newSeq ← newSeq —  $gS.get(i)$ 
  | | | | | | newDegCour ← newDegCour —  $ch.degCour[i]$ 
  | | | | | newDegCour ← newDegCour —  $\overline{\top}_{[x,a] \in gS.get(j)} \alpha_a(t[x])$ 
  | | | | | Chemins ← Chemins  $\cup$  update((newSeq, newRechCour,
  | | | | | newDegCour))
  | | |
  | | | si  $(gS.first \in t) \ \&\& \ (non(PremierPassage))$  alors
  | | | | //- un nouveau chemin est créé si on rencontre le premier itemset de
  | | | | la séquence-
  | | | |  $ch \leftarrow Chemin(\emptyset, gS.first, \overline{\top}_{[x,a] \in gS.first} \alpha_a(t[x]))$ 
  | | | | Chemins ← Chemins  $\cup ch$ 
  | | | | Update( $ch$ )
  | |
  | Chemins.Optimize()
  | //- élimination des moins bons chemins-

  pour chaque parcours  $ch \in Chemins$  faire
  | si  $ch$  non clos alors
  | | Cut( $ch$ ); //- élimination des chemins qui ne couvrent pas toute la
  | | séquence-

   $ch \leftarrow Chemins.first$ 
  //- Chemins ne contient plus que le meilleur parcours complet-
   $m \leftarrow \odot(ch.degCour)$ 
  //- on agrège pour trouver puis renvoyer le degré support-
  retourner  $m$ 

```

**fin****Algorithme 1:** FindTotallySeq : (Cherche la séquence candidate)

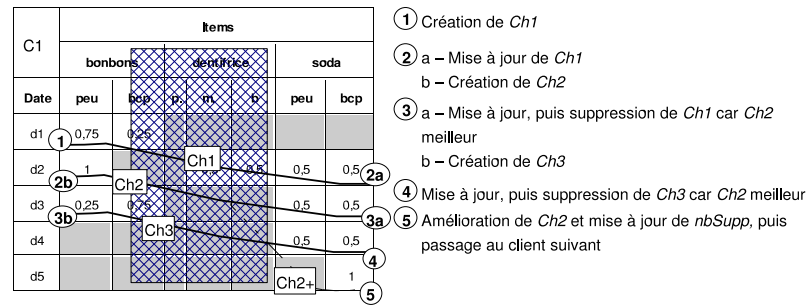


Fig. 4.7: Exemple de parcours pour le client 1

Les transactions suivantes sont examinées pour chercher soit la suite de la séquence, soit une nouvelle fois le début de la séquence, soit une amélioration des chemins trouvés. Tous les *chemins* possibles sont ainsi complétés au fur et à mesure de l'examen des transactions du client, le degré support du meilleur chemin pour toute la séquence est ensuite retourné.

La fonction *Update* permet de mettre à jour l'état d'avancement de chaque chemin. La fonction *Optimize*, non présentée ici, permet d'éliminer au fur et à mesure les chemins si un chemin identique mais de meilleur degré d'appartenance a été trouvé pour le client. La fonction *CalcTotallySupp* calcule le support d'une séquence candidate en agrégeant, pour chaque client, la valeur du chemin optimal pour la séquence.

#### Fonction CalcTotallySupport

**Données :**  $gS$ ,  $g$ - $k$ -séquence candidate

**Résultat :**  $FSupp$  support flou de la séquence  $gS$

début

$FSupp, nbSupp, m \leftarrow 0$

**pour chaque** *client*  $c \in \mathcal{C}$  **faire**

$m \leftarrow \text{FindTotallySeq}(gS, \mathcal{T}_c)$

$FSupp \leftarrow FSupp + m$

$nbSupp += m$

$FSupp \leftarrow nbSupp / |\mathcal{C}|$

**retourner**  $FSupp$

fin

**Algorithme 2:** CalcTotallySupport : (Calcule le support par un  $\Sigma$ -comptage seuillé)

## 4.5 Discussion

Dans ce chapitre, nous avons proposé trois algorithmes permettant une extraction efficace de motifs séquentiels basée sur la logique floue. Via notre proposition, il devient possible d'extraire des motifs de la forme :

$$\langle\langle [chocolat, peu][pain, peu]([lait, moyen]) \rangle\rangle$$

```

Fonction Update
Données : ch, parcours à mettre à jour ;
début
  | ch.seq ← ch.seq ∪ ch.rechCour
  | si ch.rechCour.next ≠ ∅ alors
  |   | ch.rechCour ← ch.rechCour.next
  |   |
  |   | sinon
  |   |   | //- une fois toute la séquence
  |   |   |   | trouvée, on clot le chemin-
  |   |   |   | Clore(ch)
  |   |
  | fin

```

**Algorithme 3:** Update : (*Met à jour un chemin*)

signifiant que si un client achète un peu de chocolat et un peu de pain le même jour, ce même client achètera du lait (en quantité moyenne) quelque temps plus tard. Les différentes propositions ont été implémentées en utilisant un parcours inspiré de l'algorithme PSP. Pour valider la faisabilité et la robustesse de l'approche les expérimentations ont été menées aussi bien sur des données synthétiques que réelles.

Les motifs séquentiels flous ont, en effet, été récemment considérés dans la littérature. Ils sont essentiels pour la manipulation de données numériques stockées de manière historisée, e.g. les données démographiques ou les données de capteurs. L'extraction de motifs séquentiels sur de telles bases est en effet très intéressant pour la détection d'événements et la recherche de tendances mais les algorithmes classiques existant ne permettent pas la gestion de données numériques. Nous avons ainsi défini de manière claire et complète les différents concepts associés aux motifs séquentiels flous. Il faut noter que ces derniers étaient présentés de manière incomplète dans les autres travaux portant sur cette problématique. Via nos propositions, le décideur peut alors choisir entre les trois niveaux de "fuzzification" que nous avons introduits. Ce choix permet l'extraction de fréquents qui sont plus ou moins précis et donc obtenus plus ou moins rapidement. Nous avons vu précédemment que l'un des objectifs des approches d'extraction de motifs était de se focaliser sur l'efficacité du processus. En offrant d'assouplir les motifs extraits nous pénalisons notre approche : il est plus long d'obtenir des motifs flous que des motifs stricts. Toutefois, cette pénalité est nécessaire pour offrir au décideur des connaissances qui sont en adéquation par rapport aux données manipulées.

Revenons sur la remarque énoncée en introduction sur les données principalement booléennes (absence/présence). Si nous considérons les différents domaines d'application que nous avons étudiés ces dernières années, nous avons pu constater que c'était rarement le cas. En effet, dans la plupart des projets de transfert de technologie que nous avons réalisés, nous avons pu constater que les données étaient très souvent numériques. Bien entendu, il est facile de transformer ces dernières en données booléennes mais la conséquence immédiate, dans ce cas, est de restreindre la connaissance acquise. Avec notre approche, nous évitons également d'obtenir des connaissances trop restrictives. En effet, savoir que "les personnes qui ont acheté trois bouteilles de vin ont aussi acheté deux fromages" est sans doute une connaissance intéressante. Elle est, par contre, trop restrictive. D'une part car les probabilités de trouver une quantité précise fréquente est réduite et d'autre part pour le décideur qui a souvent besoin d'avoir une idée approchée des connaissances.





## Chapitre 5

# Les contraintes de temps

Ces propositions ont été réalisées lors de l'encadrement de

**Doctorant :** Céline Fiot  
**Co-encadrant :** Anne Laurent (Maître de Conférences,  
UMII, LIRMM)

Ce chapitre adresse les problématiques

<i>Représentation des données :</i>	Données classiques
<i>Représentation des comportements :</i>	Motifs séquentiels avec contraintes temporelles étendues
<i>Extraction de motifs :</i>	GETC

## 5.1 Introduction

Motivée par de nombreux domaines d'applications (e.g. détection de fraudes, détection de défaillances, analyse de comportements, supervision de procédés, etc.), la recherche de connaissances prenant en compte des contraintes temporelles commence à intéresser un nombre croissant de travaux de recherche. Par exemple, certaines techniques d'apprentissage permettent de gérer et de raisonner sur de telles connaissances (e.g. [All90] définit notamment des opérations sur des règles associées à des intervalles de temps). Des techniques d'extraction de connaissances cherchent quant à elles à extraire des épisodes récurrents à partir d'une longue séquence [MTV97, RPT05] ou de bases de séquences [AIS93b, AS95b, MCP98]. La recherche de telles informations devient d'autant plus intéressante qu'elle permet de prendre en compte un certain nombre de contraintes entre les événements comme par exemple la durée minimale ou maximale séparant deux événements, [SA96c, Zak00, MPT04, MR04], ou encore des contraintes d'expressions régulières ou de répétitions, [GRS02, CMB02, LRBE03, ALB03].

Ce chapitre est organisé de la façon suivant. Nous présentons, section 5.2, les différents travaux menés autour des contraintes temporelles lors de l'extraction de motifs séquentiels. Puis, section 5.3, nous définissons les contraintes de temps étendues ainsi que la précision temporelle d'une séquence. La section 5.4 présente l'algorithme GETC et une illustration est proposée section 5.4.3. Nous concluons ce chapitre par une discussion.

## 5.2 Le point

La recherche de motifs séquentiels généralisés a été introduite dans [SA96c]. Cette technique de fouille de données permet d'obtenir des séquences fréquentes respectant des contraintes spécifiées par l'utilisateur, à partir d'une base de données de séquences. Différents algorithmes ont été proposés afin de gérer ces contraintes. Certains les introduisent directement dans le processus d'extraction, c'est le cas notamment de l'algorithme GSP [SA96c] tandis que d'autres proposent un pré-traitement introduisant les contraintes dans les séquences qui sont ensuite analysées par n'importe quel outil d'extraction de motifs séquentiels, ce mode de fonctionnement est celui de GTC (*Graph for Time Constraint*), proposé dans [MPT99].

Toutefois, si ces méthodes sont efficaces et robustes, notamment l'approche par graphe de séquences, elles nécessitent de l'utilisateur qu'il connaisse précisément les valeurs des contraintes à spécifier, sous peine d'obtenir des connaissances erronées ou inutiles. Pourtant dans certains cas, ces valeurs ne sont pas connues avec certitude. Ainsi, les contraintes temporelles telles qu'elles sont spécifiées permettent de mettre en évidence de nouveaux motifs séquentiels, mais elles sont encore trop rigides et il peut être nécessaire de faire plusieurs tentatives avec différentes combinaisons de ces paramètres avant d'obtenir des résultats satisfaisants. Des travaux ont été proposés afin de déterminer de manière automatique la fenêtre optimale d'observation pour la recherche d'épisodes dans une séquence [MR04], mais ils sont difficilement adaptables à l'extraction de motifs séquentiels et dans ce domaine, aucun travail à notre connaissance ne propose une détermination automatique des contraintes de temps optimales.

Par ailleurs, pour certaines applications il pourrait également être intéressant d'assouplir les contraintes spécifiées par les experts du domaine afin d'affiner leurs connaissances. C'est notamment le cas lors de la prévision de pannes. Prenons l'exemple d'une décharge de batterie annonçant l'usure de celle-ci et donc une intervention pour son renouvellement. Si l'expert indique qu'une décharge accélérée précède de 5 jours l'usure

finale, on pourra, grâce aux contraintes de temps étendues, élargir cette fenêtre de 5 jours et par exemple détecter que cette décharge peut en fait avoir lieu 7 jours avant. Ce motif sera accompagné d'une certaine probabilité, indiquant à quel point l'extension de la fenêtre temporelle donne un motif valide. Ainsi, selon la probabilité obtenue, le renouvellement pourra alors être planifié avec 1 ou 2 jours d'avance sur la prévision initiale. La connaissance de l'expert est utilisée comme un point de départ et les résultats obtenus la complètent.

Enfin, le nombre de motifs séquentiels extraits, selon les contraintes de temps utilisées, peut rapidement devenir trop important pour que leur analyse soit efficace. Une mesure permettant l'exploitation des motifs séquentiels généralisés serait donc d'une grande utilité.

C'est pourquoi nous proposons une méthode qui permet à partir de contraintes de temps spécifiées par l'utilisateur ainsi que d'un degré de respect de ces valeurs, d'extraire des motifs séquentiels satisfaisant des contraintes étendues, accompagnés du degré de respect des contraintes initiales.

### 5.3 Extension des contraintes de temps

Les contraintes temporelles pour les motifs séquentiels généralisés sont au nombre de trois [SA96b] : *windowSize*, *maxgap* et *mingap*. L'inconvénient de telles contraintes est qu'elles sont spécifiées par l'utilisateur et nécessitent donc une bonne connaissance a priori des données et des durées à spécifier. En effet, des contraintes qui ne correspondraient pas aboutiraient à des connaissances erronées ou incomplètes. Notre proposition d'extension des contraintes de temps pour les motifs séquentiels est fondée sur une analogie avec la théorie des sous-ensembles flous (C.f. Chapitre 4). Ainsi, on ne souhaite plus simplement qu'une séquence respecte ou non les contraintes spécifiées mais permettre à l'utilisateur de relâcher ces contraintes. Afin de répondre au mieux aux besoins de l'utilisateur, nous lui donnons la possibilité de spécifier la valeur  $\rho_x$  de respect minimum de la contrainte  $\mathcal{X}$ , de valeur initiale *init<sub>x</sub>*. Dans le cas des motifs séquentiels généralisés, une indication utile peut être celle donnée par les durées des séquences clients correspondant aux contraintes de temps. Or, l'extension de ces contraintes nous permet de définir une mesure du respect des contraintes initialement spécifiées. Nous offrons ainsi à l'utilisateur une flexibilité dans la spécification de ses contraintes ainsi qu'un outil d'analyse des motifs extraits.

Chacune des contraintes de temps peut être vue comme un sous-ensemble flou dont la fonction d'appartenance nous donnera, pour chaque valeur attribuée au paramètre de contrainte, la précision avec laquelle on respecte la contrainte initiale spécifiée par l'utilisateur. Afin de répondre au mieux aux besoins de l'utilisateur, nous lui donnons la possibilité de spécifier la valeur  $\rho_x$  de respect minimum de la contrainte  $\mathcal{X}$ , de valeur initiale *init<sub>x</sub>*. Le degré de respect des contraintes reposant sur la fonction d'appartenance de chacune des contraintes, les coefficients spécifiés sont compris dans l'intervalle  $[0,1]$ . Il est également possible de fixer une contrainte avec certitude :

- Si  $\rho_x = 1$ , l'utilisateur ne souhaite pas faire varier la valeur de la contrainte, qui restera donc fixée à sa valeur spécifiée et toutes les séquences générées auront une précision de 1.
- Si  $\rho_x = 0$ , l'utilisateur souhaite parcourir l'ensemble des valeurs possibles pour la contrainte, le poids d'une séquence dépendra alors de la valeur de la contrainte qui permet de la générer.
- Sinon,  $\rho_x \in ]0, 1[$ , la valeur  $x$  de la contrainte  $\mathcal{X}$  va varier entre sa valeur fixée *init<sub>x</sub>* et une valeur limite  $x_\rho$  pour laquelle  $\rho(x) = \rho_x$ .

Commençons par déterminer les valeurs limites utiles des contraintes de temps, qui correspondent au parcours de la totalité de l'espace de recherche, c'est-à-dire lorsque  $\rho_x = 0$ .

Ces valeurs sont calculées à partir des contraintes de temps strictes et correspondent aux valeurs limites autorisées par ces définitions. Les contraintes *maxgap* et *windowSize* correspondent à l'écart maximal qui sépare deux itemsets, la valeur maximale qu'elles pourront prendre pour un client correspond donc à la durée qui sépare la première transaction de ce client de la dernière. Pour toute la base, cette valeur correspondra donc à l'écart maximum pour tous les clients entre les dates minimales et maximales des transactions, c'est-à-dire  $M = \max_{c \in \mathcal{C}} (D_{c_{max}} - D_{c_{min}})$ .

La contrainte *mingap* correspond à l'écart minimal qui sépare deux itemsets. Il s'agit donc de définir la valeur limite de cette contrainte, en tenant compte de l'inégalité stricte qu'elle impose. L'écart minimum qui sépare deux transactions d'un client  $c$  est donné par  $\min_{t \in \mathcal{T}_c} (D_{t+1} - D_t)$ , et pour l'ensemble de la base, cet écart correspond à la valeur minimale pour l'ensemble des clients, soit  $\min_{c \in \mathcal{C}} (\min_{t \in \mathcal{T}_c} (D_{t+1} - D_t))$ . Dans le cas limite, la contrainte sur *mingap* s'exprime par l'inégalité :

$$\begin{aligned} \min_{c \in \mathcal{C}} (\min_{t \in \mathcal{T}_c} (D_{t+1} - D_t)) &> \textit{mingap} \\ \min_{c \in \mathcal{C}} (\min_{t \in \mathcal{T}_c} (D_{t+1} - D_t)) &\geq \textit{mingap} + 1 \\ \min_{c \in \mathcal{C}} (\min_{t \in \mathcal{T}_c} (D_{t+1} - D_t)) - 1 &\geq \textit{mingap} \end{aligned}$$

La valeur limite de *mingap* est donc  $\min_{c \in \mathcal{C}} (\min_{t \in \mathcal{T}_c} (D_{t+1} - D_t)) - 1$ . Or dans le cas où un client n'aurait qu'une transaction, cette valeur serait -1, ce qui reviendrait à dire que l'on pourrait avoir un écart nul entre deux itemsets consécutifs. Autrement dit, on pourrait transformer un itemset (10 20 30) en une séquence  $\langle (10) (20) (30) \rangle$ . On impose donc dans ce cas que la valeur de *mingap* est nulle. La valeur limite de *mingap* s'exprime alors par  $m = \max(\min_{c \in \mathcal{C}} (\min_{t \in \mathcal{T}_c} (D_{t+1} - D_t)) - 1, 0)$ .

Dans la suite de cette section, nous utiliserons les notations suivantes, résumées Figure 5.1, afin de distinguer les trois contraintes de temps :

- Soit *init\_ws*, *init\_mg* et *init\_MG* les valeurs initiales spécifiées pour les contraintes de temps *windowSize*, *mingap* et *maxgap* et  $\rho_{ws}$ ,  $\rho_{mg}$  et  $\rho_{MG}$  les niveaux minimum de précision associés à chacune d'elles. Ces coefficients vont permettre de définir les limites de variation des contraintes de temps correspondant aux besoins de l'utilisateur.
- On notera *ws*, *mg* et *MG* les valeurs variables des contraintes et  $\rho(ws)$  (resp.  $\rho(mg)$  et  $\rho(MG)$ ) les précisions des contraintes selon la valeur de *ws* (resp. *mg* et *MG*).

Nous allons maintenant présenter l'extension de chacune des trois contraintes.

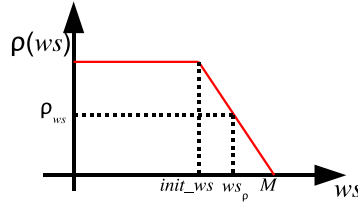
**Extension de *windowSize*** La valeur *ws* de la contrainte *windowSize* peut varier entre sa valeur spécifiée *init\_ws* et sa valeur maximum utile  $M$ , comme décrit Figure 5.2. L'utilisateur peut alors choisir de contraindre cette valeur à respecter une précision minimale  $\rho_{ws}$ . Cette précision implique une valeur limite  $ws_\rho$  de *ws*.

L'extension de la contrainte s'exprime sous la forme du sous-ensemble flou décrit par la fonction d'appartenance (5.1) qui donne la précision correspondant à une valeur donnée de *ws* :

$$\rho(ws) = \begin{cases} 1 & \text{si } ws \leq \textit{init\_ws} \\ \frac{1}{\textit{init\_ws} - M} ws - \frac{M}{\textit{init\_ws} - M} & \text{si } \textit{init\_ws} < ws \leq ws_\rho \\ 0 & \text{sinon} \end{cases} \quad (5.1)$$

$M$	valeur maximale possible de <i>windowSize</i> et <i>maxgap</i> $M = \max_{c \in C} (D_{c_{max}} - D_{c_{min}})$
$init\_ws$	valeur initiale du paramètre <i>windowSize</i> , fixé par l'utilisateur
$ws$	variable correspondant au paramètre <i>windowSize</i> , peut varier entre $init\_ws$ et $M$
$\rho_{ws}$	précision minimale souhaitée par l'utilisateur <i>windowSize</i>
$\rho(ws)$	précision de <i>windowSize</i> pour la valeur $ws$
$init\_MG$	valeur initiale du paramètre <i>maxgap</i> , fixé par l'utilisateur
$MG$	variable correspondant au paramètre <i>maxgap</i> , peut varier entre $init\_MG$ et $M$
$\rho_{MG}$	précision minimale souhaitée par l'utilisateur <i>maxgap</i>
$\rho(MG)$	précision de <i>maxgap</i> pour la valeur $MG$
$m$	valeur minimale possible de <i>mingap</i> $m = \max_{c \in C} (\min_{t \in T_c} (D_{t+1} - D_t)) - 1, 0)$
$init\_mg$	valeur initiale du paramètre <i>mingap</i> , fixé par l'utilisateur
$mg$	variable correspondant au paramètre <i>mingap</i> , peut varier entre $m$ et $init\_mg$
$\rho_{mg}$	précision minimale souhaitée par l'utilisateur pour <i>mingap</i>
$\rho(mg)$	précision de <i>mingap</i> pour la valeur $mg$

Fig. 5.1: Notations

Fig. 5.2: Précision de *windowSize* selon la valeur de la contrainte étendue.

La valeur limite  $ws_\rho$  correspond à la valeur de *windowSize* pour laquelle la précision des séquences générées vaut  $\rho_{ws}$ . Elle est donnée par l'équation :

$$ws_\rho = \lfloor (init\_ws - M)\rho_{ws} + M \rfloor \quad (5.2)$$

**Extension de *maxgap*** La valeur  $MG$  de la contrainte *maxgap* peut varier entre sa valeur spécifiée  $init\_MG$  et sa valeur maximum utile  $M$ . L'extension de la contrainte s'exprime sous la forme du sous-ensemble flou décrit par la fonction d'appartenance (5.3) qui donne :

$$\rho(MG) = \begin{cases} 1 & \text{si } MG \leq init\_MG \\ \frac{1}{init\_MG - M} MG - \frac{M}{init\_MG - M} & \text{si } init\_MG < MG \leq MG_\rho \\ 0 & \text{sinon} \end{cases} \quad (5.3)$$

La valeur limite  $MG_\rho$  correspond à la valeur de *maxgap* pour laquelle le poids des séquences générées vaut  $\rho_{MG}$ . Elle est donnée par l'équation :

$$MG_\rho = \lfloor (init\_MG - M)\rho_{MG} + M \rfloor \quad (5.4)$$

**Extension de *mingap*** La valeur  $mg$  de la contrainte *mingap* peut varier entre sa valeur minimum utile  $m$  et sa valeur spécifiée  $init\_mg$ . L'extension de la contrainte

s'exprime sous la forme du sous-ensemble flou décrit par la fonction d'appartenance (5.5) qui donne :

$$\rho(mg) = \begin{cases} 1 & \text{si } mg \geq \text{init\_mg} \\ \frac{1}{\text{init\_mg} - m} mg - \frac{m}{\text{init\_mg} - m} & \text{si } \text{init\_mg} > mg \geq mg_\rho \\ 0 & \text{sinon} \end{cases} \quad (5.5)$$

La valeur limite  $mg_\rho$  correspond à la valeur de  $mingap$  pour laquelle le poids des séquences générées vaut  $\rho_{mg}$ . Elle est donnée par l'équation :

$$mg_\rho = \lceil (\text{init\_mg} - m)\rho_{mg} + m \rceil \quad (5.6)$$

On remarque que les définitions des sous-ensembles flous des trois contraintes correspondent à une variation linéaire de la fonction d'appartenance entre la valeur initiale d'une contrainte et sa valeur limite, mais cette fonction pourrait également être définie par palier ou en proportion du nombre de clients de la base respectant chacune des valeurs de la contrainte.

Nous allons maintenant définir le degré de respect des contraintes d'une séquence en considérant les trois contraintes simultanément. Pour chacune des séquences fréquentes trouvées à la fin du processus d'extraction, il s'agit de combiner les valeurs des contraintes de temps qui permettent de les générer afin de déterminer la précision avec laquelle chacune d'elles respecte les valeurs initiales.

On définit la **précision temporelle** d'une séquence  $s$  pour un client  $c$  comme le niveau de respect simultané des trois contraintes de temps calculé à l'aide d'une t-norme ( $\top$ ). Pour chaque client, on cherche, parmi toutes les séquences d'achats  $\varsigma_c$ , l'occurrence de  $s$  qui respecte au mieux les contraintes de temps, en utilisant une t-conorme ( $\perp$ ).

La précision temporelle d'une séquence  $s = \langle s_1 \cdots s_n \rangle$  pour le client  $c$  est donnée par :

$$\varrho(s, c) = \perp_{s \in \varsigma_c} \left( \begin{array}{l} \top_{i \in [1, n]} \left( \rho_{ws}(\text{date}(s_{u_i}) - \text{date}(s_{l_i})) \right), \\ \top_{i \in [2, n]} \left( \rho_{mg}(\text{date}(s_{l_i}) - \text{date}(s_{u_{i-1}})), \right. \\ \left. \rho_{MG}(\text{date}(s_{u_i}) - \text{date}(s_{l_{i-1}})) \right) \end{array} \right) \quad (5.4)$$

Pour l'ensemble de la base, la précision temporelle d'une séquence  $s$  est donnée par l'agrégation par la moyenne des précisions de chacun des clients :

$$\Upsilon(s) = \frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} \varrho(s, c) \quad (5.5)$$

## 5.4 Vers des contraintes de temps étendues : GETC

Notre proposition d'implémentation des contraintes de temps repose sur les graphes de séquences pour les contraintes de temps (GTC) proposées dans [MPT99]. Il s'agit de transformer une séquence d'un client en un graphe de séquences respectant les contraintes de temps. Les graphes de séquences des différents clients sont ensuite utilisés pour déterminer les séquences fréquentes par un algorithme d'extraction de motifs séquentiels.

L'efficacité de cette approche ayant été démontrée dans [MPT99, MPT04], nous avons choisi de nous en inspirer pour développer notre solution. Nous proposons donc un algorithme permettant de construire un graphe de séquences pour les contraintes de temps étendues qui nous permettra également, dans un deuxième temps, de calculer la précision des motifs séquentiels généralisés extraits.

### 5.4.1 Principe

Nous utilisons GETC comme prétraitement pour la prise en compte des contraintes de temps étendues. Une fois les séquences des clients transformées en graphes de séquences respectant les contraintes de temps étendues, nous utilisons PSP pour l'extraction des motifs séquentiels. En utilisant ainsi le graphe de séquences obtenu par GETC, la vérification des contraintes de temps est rendue inutile pendant le parcours des candidats, seule l'inclusion devant être vérifiée. Cette méthode est similaire à celle proposée dans [MPT04] qui permet d'optimiser l'extraction de motifs séquentiels généralisés par un parcours indépendant et sans retour arrière de l'arbre des séquences candidates pour la vérification des contraintes de temps.

Une fois les motifs séquentiels extraits, les graphes de séquences sont valués puis reparcourus une dernière fois, afin de calculer la précision temporelle de chacun des motifs séquentiels généralisés extraits.

### 5.4.2 L'algorithme GETC

A partir d'une séquence d'entrée  $d$ , l'algorithme GETC construit son graphe de séquences  $G_d(S, A)$ . Tout d'abord, GETC commence par créer les sommets correspondants aux itemsets de la séquence. Chaque sommet  $x$  du graphe de séquences est caractérisé par sa date de début  $x.begin()$  et par sa date de fin  $x.end()$ . On peut également accéder à sa liste de prédécesseurs par  $x.prev()$  et à la liste de ses successeurs par  $x.succ()$ , ainsi qu'à l'itemset  $x.itemset()$ . La fonction *addWindowSize* ajoute ensuite à l'ensemble des sommets, l'ensemble des combinaisons d'itemsets permises selon les différentes valeurs de la contrainte *windowSize*.

L'étape suivante consiste en l'ajout des arcs respectant les contraintes *mingap* et *maxgap*. Ainsi, pour chaque sommet, on cherche le premier niveau accessible pour la contrainte *mingap* (ie.  $l.begin() - x.end() > mg_\rho$ ) et pour chaque sommet  $z$  de ce niveau, on construit les arcs  $(x, z)$ , pour chaque sommet  $z$  tel que  $z.end() - x.begin() \leq MG_\rho$ . La fonction *addEdge* permet d'éviter les inclusions de chemins, grâce à la construction d'arcs temporaires dans des cas d'inclusions possibles. Dans le cas où pour un sommet  $x$ , on ne peut atteindre le niveau  $l$  à cause du non respect de la contrainte *mingap*, on utilise la fonction *propagate* pour "propager ce saut".

L'avant-dernière étape utilise la fonction *pruneMarked* qui élimine les sommets de sous-séquences incluses. Enfin, *convertEdges* transforme les arcs temporaires indispensables en arcs définitifs et supprime les arcs de sous-séquences incluses.

L'algorithme *addWindowSize* (C.f. Algorithme 5) parcourt chaque sommet  $x$  et détermine pour chacun d'entre eux quels sommets  $y$  (différents de  $x$ ) peuvent être "fusionnés" avec  $x$  (si  $y.date() - x.date() \leq ws$ ). Chaque sommet  $i$  correspond alors à un itemset  $i.itemset()$  qui possède une date de début  $i.begin()$  et une date de fin  $i.end()$ . Les sommets correspondant au même itemset et de même date de début ou de fin ne sont pas tous construits selon les lemmes 1 et 2 présentés section 5.4.2. Les sommets ainsi construits sont regroupés en niveau par date de fin des itemsets,  $l.end()$ . Cela permet par la suite de tester le respect des contraintes pour un niveau et non plus pour chaque sommet.

L'algorithme *addEdge* (C.f. Algorithme 6) permet de construire les arcs entre des sommets qui respectent les contraintes de temps *mingap* et *maxgap*. On créera un arc définitif si les sommets ne sont pas déjà liés par une séquence ou une inclusion de leurs successeurs ou prédécesseurs. Dans ce cas, l'arc construit sera temporaire et ne deviendra définitif que si la séquence qu'il forme est maximale. C'est également lors



```

Fonction GETC
Données :  $d$ , une séquence de données ;
Résultat :  $G_d(S, A)$ , le graphe de séquences associé à  $d$ ,
            $S$  l'ensemble des sommets de  $G_d$ ,  $A$  l'ensemble des arêtes ;

début
  S  $\leftarrow$  buildVertices( $d$ )
  addWindowSize(S)
  tant que  $x \neq S.first()$  faire
     $l \leftarrow x.level().prec()$  ;  $mg \leftarrow init\_mg$ 
    tant que  $x.begin() - l.end() \leq mg$  faire
       $contmg \leftarrow FALSE$ 
      si  $x.begin() > l.end()$  alors
        tant que  $mg \geq mg_\rho$  faire
          si  $constming(x,l)$  alors
             $contmg \leftarrow TRUE$ 
             $mg \leftarrow mg_\rho - 1$ 
          sinon
             $mg - -$ 
        si  $contmg == FALSE$  alors
           $propagate(x,l)$  ;  $l \leftarrow l.prec()$ 
      pour chaque  $w \in l$  faire
         $included \leftarrow TRUE$ 
         $MG \leftarrow init\_MG$ 
        tant que  $MG \leq MG_\rho$  faire
          si  $constMaxG(x,w)$  alors
             $addEdge(w,x)$ 
             $MG \leftarrow MG_\rho + 1$ 
          sinon
             $MG++$ 
         $x \leftarrow S.next(x)$ 
     $pruneMarked(G_d(S, A))$ 
     $convertEdges(G_d(S, A))$ 
  retourner  $G_d(S, A)$ 
fin

```

Algorithme 4: GETC : (Fonction de construction du graphe de séquences)

```

Fonction addWindowSize
Données :  $S$ , ensemble  $S$  des sommets à traiter (ordonnés en ordre croissant,
          d'abord par date de début, puis par date de fin);

début
  copyS  $\leftarrow S$ ;  $x \leftarrow S.first()$ 
  tant que  $x \neq S.last()$  faire
    xnext  $\leftarrow S.next(x)$ 
     $y \leftarrow S.next(x)$ 
    ws  $\leftarrow init\_ws$ 
    % tant que  $ws \leq ws_p$  faire
      tant que  $constWS(x,y)$  faire
         $i \leftarrow group(x,y)$ 
        copyS.addOrReplace( $i$ )
         $x \leftarrow i$ 
         $y \leftarrow S.next(y)$ 
      ws ++
     $x \leftarrow xnext$ 
  S  $\leftarrow copyS$ 
fin

```

**Algorithme 5:** (Ajoute les sommets combinaisons)

de l'exécution de cet algorithme que les sommets inclus sont marqués pour pouvoir ensuite être supprimés s'ils sont inutiles.

Pour chacun des sommets  $y$  d'un niveau inaccessible par  $x$ , la fonction *propagate* ajoute, si nécessaire et si on respecte les contraintes *mingap* et *maxgap*, un arc entre chacun des successeurs de  $x$  et ce sommet  $y$ . Comme dans *addEdge*, on construit des arcs temporaires ou définitifs selon que la séquence construite peut être incluse ou n'a aucune chance de l'être.

L'algorithme *convertEdges* permet de transformer les arcs temporaires indispensables en arcs définitifs et de supprimer les arcs de sous-séquences incluses. Pour chaque arc temporaire entre  $x$  et  $y$ , si  $y$  est inclus dans un successeur  $z$  de  $x$  et si les successeurs de  $y$  sont également tous des successeurs de  $z$ , alors il existe une sous-séquence incluse, l'arc est inutile, il est donc supprimé. Dans les autres cas, l'arc est indispensable pour obtenir toutes les séquences maximales, il est donc converti en arc définitif.

Les fonctions *propagate* et *convertEdges* sont détaillées dans [FLT05].

### Des graphes de séquences complets

GETC étant utilisé comme prétraitement pour la prise en compte de contraintes temporelles en vue de l'extraction de motifs séquentiels, il doit générer toutes les séquences issues d'une séquence de données. Par ailleurs, afin d'améliorer le temps d'extraction, il est nécessaire que GETC n'extrait que les séquences les plus longues. Nous avons monté ici que GETC extrait exactement toutes les séquences maximales supportées par la séquence d'entrée.

**Théorème 1** *Le graphe de séquences généré par GETC ne contient pas de séquences incluses.*

**Preuve 1** *Supposons qu'il existe, dans le graphe de séquences, deux séquences  $s_1$  et  $s_2$  telles que  $s_1 \subset s_2$ . Cela signifie que le graphe de séquences contient un sommet  $y$  tel que l'un des*

```

Fonction addEdge
Données : deux sommets  $r$  et  $s$ ,  $A$  l'ensemble des arcs du graphe;
début
  si  $r.succ() == \emptyset$  alors
    si  $s.prev() == \emptyset$  alors
       $A \leftarrow A \cup \{(r, s)\}$ 
      unmark( $r$ )
      unmark( $s$ )
    sinon
      pour chaque  $p \in s.prev()$  faire
        si  $(r \subset p)$   $\&\&$ 
           $(r.succ() \subset p.succ())$  alors
             $\llcorner$   $included \leftarrow TRUE$ 
        si  $included == TRUE$  alors
          arcTmp( $r, s$ )
          mark( $r$ )
           $A \leftarrow A \cup \{(r, s)\}$ 
          unmark( $r$ )
          unmark( $s$ )
      sinon
        pour chaque  $t \in r.succ()$  faire
          si  $(s \subset t)$   $\&\&$ 
             $(s.succ() \subset t.succ())$  alors
               $\llcorner$   $included \leftarrow TRUE$ 
          si  $(s.prev() \neq \emptyset)$   $\&\&$ 
             $(included == FALSE)$  alors
              pour chaque  $p \in s.prev()$  faire
                si  $(r \subset p)$   $\&\&$ 
                   $(r.succ() \subset p.succ())$  alors
                     $\llcorner$   $included \leftarrow TRUE$ 
              si  $included == TRUE$  alors
                arcTmp( $r, s$ )
                mark( $r$ )
              sinon
                 $A \leftarrow A \cup \{(r, s)\}$ 
                unmark( $r$ )
                unmark( $s$ )
    fin
  fin

```

Algorithme 6: (Ajoute les arcs)

prédécesseurs  $x$  de  $y$  est accessible par  $t$ , également prédécesseur de  $y$ , i.e. il existe un chemin  $(t, \dots, y)$  de longueur supérieure ou égale à 2 et un arc  $(t, y)$ .

Par construction, ce type de cas ne se présente que lors de l'exécution de `propagate`. L'existence d'un tel chemin implique une intersection entre les sommets qui précèdent  $y$  et ceux qui suivent  $t$ . Or si une telle situation se présente, `propagate` ne crée pas de nouvel arc entre  $t$  et  $y$ . Ainsi `GETC` ne construit bien que le chemin maximal et élimine bien les chemins inclus.

**Théorème 2** *L'algorithme `GETC` construit exactement toutes les solutions de la plus grande taille possible pour les séquences respectant `mingap` et `maxgap`.*

**Preuve 2** *L'algorithme `GETC` parcourt tous les sommets du graphe, ce qui implique que si un chemin respectant `mingap` et `maxgap` contient le sommet  $x$ , alors ce chemin fait partie du graphe après l'exécution de `GETC` (même s'il est restreint au seul sommet  $x$ ). Tous les sommets font partie d'un chemin, l'inclusion de chemin est impossible (Théorème 1) et si deux chemins  $(x, \dots, y)$  et  $(y', \dots, z)$  peuvent être fusionnés (ie si  $y'.date() - y.date() > mg_\rho$  et  $z.date() - x.date() \leq MG_\rho$ ), ils le seront, car l'algorithme, en explorant le sommet  $y$ , va construire l'arc  $(y, y')$ .*

*L'algorithme `GETC` construit donc exactement toutes les solutions de la plus grande taille possible pour les séquences respectant `mingap` et `maxgap`.*

**Théorème 3** *L'algorithme `addWindowSize` construit exactement tous les sommets susceptibles de contribuer à la construction de toutes les solutions de la plus grande taille possible pour les séquences respectant `mingap` et `maxgap`.*

**Preuve 3** *D'après les lemmes 1 et 2, démontrés dans [FLT05], `addWindowSize` construit bien exactement tous les sommets susceptibles de contribuer à la construction de toutes les solutions de la plus grande taille possible pour les séquences respectant `mingap` et `maxgap`, en ne permettant pas d'inclusion inutile ni de redondance de séquence.*

**Lemme 1** *Pour deux sommets représentant le même itemset, et ayant la même date de fin, seul celui qui a la date de début la plus tardive est nécessaire pour le problème de la recherche de chemin de longueur maximale.*

**Lemme 2** *Pour deux sommets représentant le même itemset, et ayant la même date de début, seul celui qui a la date de fin la plus ancienne est nécessaire pour le problème de la recherche de chemin de longueur maximale.*

**Théorème 4** *L'algorithme `GETC` construit exactement toutes les solutions de la plus grande taille possible pour les séquences respectant les contraintes étendues `windowSize`, `mingap` et `maxgap`.*

**Preuve 4** *D'après le théorème 2, `GETC` construit exactement toutes les solutions de la plus grande taille possible pour les séquences respectant `mingap` et `maxgap`. D'après le théorème 3, le traitement de la contrainte `windowSize` permet de générer tous les sommets nécessaires, il reste à vérifier que ce traitement ne permet pas non plus d'inclusion.*

*Supposons que le graphe de séquences calculé par `GETC` contient deux séquences  $s_1$  et  $s_2$  tels que  $s_1 \subset s_2$ . Cela signifie que le graphe de séquences contient un sous-graphe tel que l'un des sommets  $y$  inclus dans un autre sommet  $z$  et tel que  $y.next() \subseteq z.next()$ .*

*Or les algorithmes `addEdge` et `propagate` marquent un tel sommet  $y$  lors de la construction des chemins et l'algorithme `pruneMarked` supprime les sommets marqués. Par construction, une telle inclusion est donc impossible.*

`GETC` construit bien exactement toutes les séquences maximales supportées par la séquence d'entrée, il peut donc être utilisé comme prétraitement pour la prise en compte des contraintes de temps étendues en vue de l'extraction de motifs séquentiels.

Date	1	3	4	5	6	8	9	10	12	17	18
Client 1	1	-	2 3	3 4	4	4	-	5	6	7	8
Client 2	2 3	4	-	-	5	6	-	-	-	-	-
Client 3	1 2	-	3	3 4	4	-	5 6	-	-	-	-

Fig. 5.3: Base exemple

### Calcul de la précision temporelle

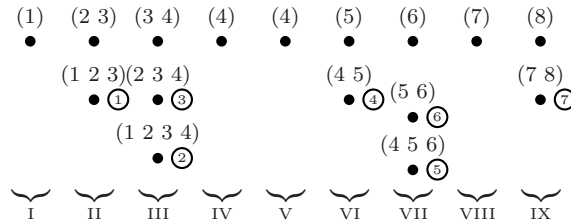
Une fois le graphe de séquences construit, on sait quelles sont les séquences autorisées par les contraintes de temps et celles qui sont interdites. Cependant, certaines séquences respectent les contraintes fortes de l'utilisateur alors que d'autres ont été construites en appliquant les contraintes étendues. Elles ne sont donc pas équivalentes. On va donc calculer le niveau de précision temporelle de chacun des chemins (séquences maximales) et l'affecter à chacune des sous-séquences qui le composent.

Afin de déterminer le respect des contraintes de temps par le chemin, on value chaque arc  $(x,y)$  par  $\top(\mu_{mg}(y.begin()-x.end()), \mu_{MG}(y.end()-x.begin()))$  selon les valeurs de  $mg$  et  $MG$  qui permettent de le construire. Chaque sommet est valué sur le même principe, par  $\mu_{ws}$ . Ces valuations sont réalisées par la fonction *valueGraph*, détaillé dans [FLT05]. La précision temporelle d'une séquence est alors donnée par la formule (5.5), section 5.3. Ce calcul nécessite une passe supplémentaire, après l'extraction des motifs séquentiels, pour retourner, en plus du support de chaque motif, sa précision temporelle.

### 5.4.3 Illustration

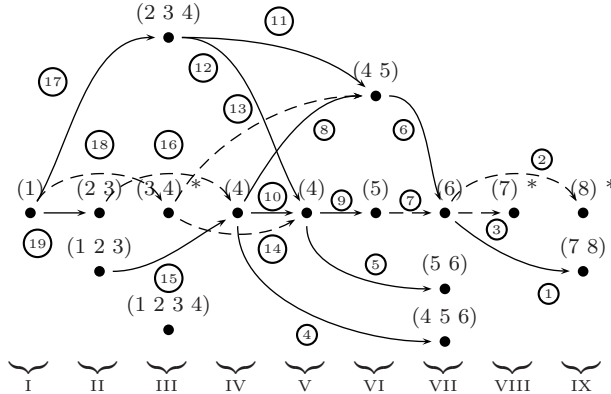
Considérons la base exemple décrite Figure 5.3 et les paramètres suivants pour les contraintes de temps : pour *windowSize*, *init\_ws*=2 et  $\rho_{ws} = 0.87$ , donc  $ws_{\rho}$ =4; pour *maxgap*, *init\_MG*=4 et  $\rho_{MG} = 0.84$  donc  $MG_{\rho}$ =6; pour *mingap*, *init\_mg*=2 et  $\rho_{mg} = 0.5$ , donc  $mg_{\rho}$ =1. D'après les données de la figure 5.3,  $M = 17$  et  $m = 0$ .

Nous présentons ici la construction du graphe de séquences pour la séquence de données du client 1. La première étape consiste en la création de l'ensemble des sommets initiaux, correspondant aux itemsets des transactions présentés dans la base de la figure 5.3. Ensuite, la contrainte *addWindowSize* est prise en compte afin de créer les sommets combinaisons, grâce à la fonction *addWindowSize* (C.f. Figure 5.4).

Fig. 5.4: Graphe de séquences après prise en compte de *windowSize* étendue; ordre de construction des sommets par *addWindowSize*.

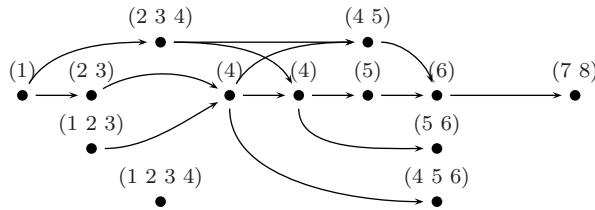
L'étape suivante est celle de la construction des arcs respectant les contraintes *mingap* et *maxgap*, grâce à l'algorithme principal, ainsi qu'aux fonctions *propagate* et

*addEdge* (C.f. Figure 5.5).



**Fig. 5.5:** Graphe de séquences après la prise en compte de *mingap* et *maxgap* étendus; ordre de construction des arcs par GETC.

La fonction *pruneMarked* supprime ensuite les sommets inclus, puis les arcs temporaires sont convertis en arcs définitifs ou supprimés par la fonction *convertEdges*. Le graphe de séquences obtenu est présenté dans la figure 5.6.



**Fig. 5.6:** Graphe de séquences final

Les séquences maximales incluses dans la séquence du client 1 sont :

- $\langle(1\ 2\ 3)(4)(4\ 5\ 6)\rangle$  -  $\langle(1\ 2\ 3)(4)(4)(5)(6)(7\ 8)\rangle$  -  $\langle(1)(2\ 3)(4)(4)(5)(6)(7\ 8)\rangle$
- $\langle(1\ 2\ 3)(4)(4\ 5)(6)\rangle$  -  $\langle(1)(2\ 3)(4)(4\ 5\ 6)\rangle$  -  $\langle(1)(2\ 3\ 4)(4)(5\ 6)\rangle$
- $\langle(1\ 2\ 3)(4)(4)(5\ 6)\rangle$  -  $\langle(1)(2\ 3)(4)(4\ 5)(6)\rangle$  -  $\langle(1)(2\ 3\ 4)(4)(5)(6)(7\ 8)\rangle$
- $\langle(1\ 2\ 3\ 4)\rangle$  -  $\langle(1)(2\ 3)(4)(4)(5\ 6)\rangle$  -  $\langle(1)(2\ 3\ 4)(4\ 5)(6)\rangle$

A partir de la base de données de la figure 5.3 et des contraintes de temps spécifiées précédemment, on construit les trois graphes de séquences correspondants. Puis on procède à l'extraction des motifs séquentiels. Les motifs séquentiels généralisés extraits avec *minSupp* = 70%, sont les séquences :  $\langle(2\ 3\ 4)\rangle$ ,  $\langle(2\ 3)(4)(5\ 6)\rangle$ ,  $\langle(2)(4\ 5)\rangle$ ,  $\langle(3\ 4)(5)\rangle$ ,  $\langle(3\ 4)(6)\rangle$  et  $\langle(3)(4\ 5)\rangle$ . Tous ont un support de 100%. Afin de pouvoir distinguer leur pertinence par rapport aux besoins de l'utilisateur, nous calculons la précision temporelle de chacun. Pour cela, chacun des graphes de séquences est valué comme précisé dans la section 5.4.2. Les sommets construits avec *ws*=0,1,2 ont un degré de 1, avec *ws*=3, un degré de 0.93 et avec *ws*=4, un degré de 0.87. De même les arcs construits avec *mg*=1 ont un degré pour *mingap* de 0.5 et de 1 pour *mg*=2. En ce qui concerne *maxgap*, le degré est de 1 pour  $MG \leq 4$ , de 0.92 pour  $MG=5$  et 0.84 pour  $MG=6$ .

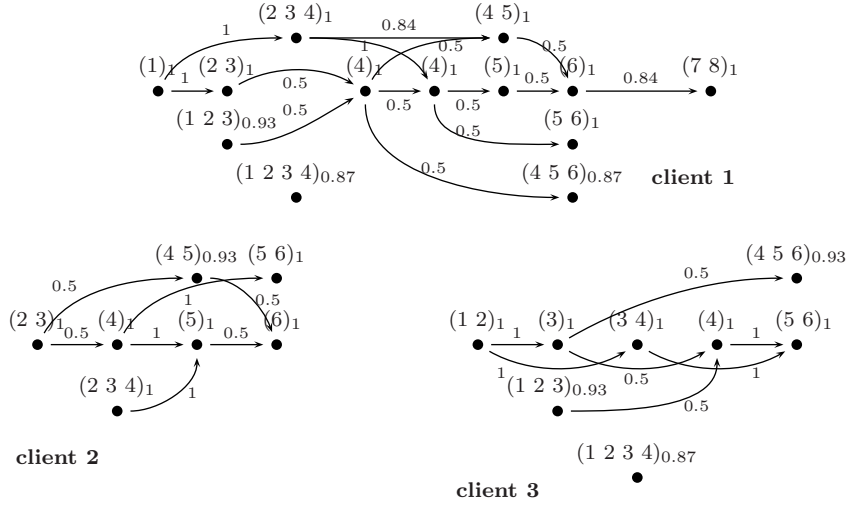


Fig. 5.7: Graphes de séquences valués pour les clients 1, 2 et 3

On utilise ces évaluations pour calculer le degré de respect des contraintes de temps par les motifs extraits. Les résultats sont présentés dans la figure 5.8).

Motifs séquentiels	$\varrho_{C11}$	$\varrho_{C12}$	$\varrho_{C13}$	$\Upsilon$
$\langle\langle 2\ 3\ 4 \rangle\rangle$	1	1	0.87	0.96
$\langle\langle 2\ 3 \rangle\langle 4 \rangle\langle 5\ 6 \rangle\rangle$	0.5	0.5	0.5	0.5
$\langle\langle 2 \rangle\langle 4\ 5 \rangle\rangle$	0.84	0.5	1	0.78
$\langle\langle 3\ 4 \rangle\langle 5 \rangle\rangle$	0.84	1	1	0.95
$\langle\langle 3\ 4 \rangle\langle 6 \rangle\rangle$	0.5	0.5	1	0.67
$\langle\langle 3 \rangle\langle 4\ 5 \rangle\rangle$	0.84	0.5	0.5	0.61

Fig. 5.8: Calcul de la précision pour les motifs séquentiels extraits

Une fois les motifs obtenus avec leur précision temporelle, on peut analyser plus précisément les contraintes qui ont permis de les générer. Plus la précision est proche de 1, plus les valeurs initiales spécifiées par l'utilisateur correspondent aux dates dans la base de données. A l'inverse, une précision faible indique que les contraintes sont peu appropriées à ce jeu de données.

## 5.5 Discussion

L'approche GETC est basée sur un pré-traitement des séquences lors de leur vérification sur la base de données. Les expériences que nous avons menées aussi bien sur des données réelles que synthétiques ont montré que ce type d'approche est beaucoup plus efficace que de traiter les contraintes temporelles dans l'arbre des candidats. En effet, dans ce cas, il est indispensable d'effectuer des backtracking dans l'arbre et cela pénalise fortement le processus d'extraction. L'une des originalités de GETC est donc d'introduire les contraintes temporelles au plus tôt dans l'algorithme de fouille. En proposant une telle approche, nous évitons d'effectuer des post traitements sur les résultats obtenus comme le font de nombreuses approches de prise en compte de contrainte. Revenons cependant sur ces post traitements. Même si nous souhaitons les effectuer, le problème est la manière de les spécifier. En effet, considérons que nous ayons extrait la séquence fréquente suivante  $\langle\langle 10 \rangle\langle 20 \rangle\langle 30 \rangle\rangle$ . Si nous souhaitons par exemple appliquer un *mingap* de 5 entre chaque itemset, le problème est que nous ne

connaissions pas les dates des différents itemsets. Ainsi, pour pouvoir effectuer un tel post traitement, il est indispensable de rechercher toutes les séquences de la base de la forme  $\langle(10) \dots(20) \dots(30)\rangle$  afin de vérifier pour chacune d'elle si oui ou non elle respecte la contrainte. Il est évident qu'une telle approche est complètement inefficace. Via GETC, nous obtenons toutes les séquences fréquentes et qui vérifient les contraintes directement. Nous répondons également, au moins pour les contraintes temporelles, au nouveau challenge de la communauté fouille de données : comment pousser les contraintes dans l'algorithme? Dans notre cadre, nous nous sommes intéressés aux contraintes temporelles, il est intéressant d'examiner maintenant si d'autres types de contraintes peuvent être traités de la même manière. Nous revenons sur ce problème dans les perspectives du mémoire en conclusion.

Par ailleurs, GETC a l'originalité d'offrir plus de souplesse dans la prise en compte des contraintes de temps. Nous avons vu que si nous spécifions  $\rho_x = 1$  cela revient à prendre en compte les contraintes dans leur globalité sans aucune souplesse. Dans ce cas, nous sommes tout à fait similaires à l'approche que nous avons proposée dans nos travaux précédents, i.e. GTC. Par contre, dès que la valeur de  $\rho$  varie, nous offrons à l'utilisateur de nouveaux résultats qui sont tout à fait intéressants. Cet intérêt peut se mesurer dans la difficulté qu'il peut y avoir de spécifier des contraintes. En effet, le décideur n'a souvent que peu d'informations sur la base qu'il est en train de traiter. Dire que la contrainte de mingap vaut 5 semble simple mais pourtant les conséquences sont importantes. Je ne veux que les séquences pour lesquelles il existe un intervalle de 5 jours entre les itemsets. Bien entendu, via GETC ou GTC, nous serons à même de les lui fournir mais quel est leur utilité si par exemple l'intervalle entre deux itemsets dans la base varie entre 4 et 6 jours. En offrant la possibilité d'assouplir la prise en compte de la contrainte nous aidons non seulement à mieux appréhender les comportements contenus sur la base mais nous permettons d'offrir une première aide au décideur qui pourra, s'il le souhaite, affiner la connaissance extraite en spécifiant des valeurs strictes aux paramètres temporels.





## Chapitre 6

# Données textuelles

Ces propositions ont été réalisées lors de l'encadrement de

**Doctorant :** Simon Jaillet  
**Co-encadrant :** Jacques Chauché (Professeur, UMII,  
LIRMM)

Ce chapitre adresse les problématiques

<i>Représentation des données :</i>	Documents textuels
<i>Représentation des comportements :</i>	Motifs séquentiels et Catégoriseur associé
<i>Extraction de motifs :</i>	SPAC

## 6.1 Introduction

Depuis de nombreuses années, les techniques de fouilles de données ont montré qu'elles étaient tout à fait adaptées à des données textuelles, i.e. *Text Mining*. Plusieurs tâches de fouille de textes existent telles que la recherche d'information, l'extraction d'information ... [IS07]. Dans ce mémoire, nous nous intéressons plus particulièrement aux tâches de classification de textes. Dans ce cadre, l'objectif est de classer de façon automatique les documents dans des catégories préalablement définies soit par un expert, il s'agit alors de classification supervisée ou catégorisation, soit de façon automatique, il s'agit alors de classification non supervisée ou encore clustering [SM83, IT95]. Même si les premiers travaux sur la classification automatique de textes datent des années 60 [Mar61], l'explosion du nombre de documents électroniques disponibles a mis en évidence le besoin de nouvelles méthodes efficaces pour traiter de gros volumes de données [Seb02, Yan99, JAM03]. Les résultats obtenus sont utiles pour d'autres tâches de fouille de textes.

Dans ce chapitre, nous nous intéressons plus particulièrement à la catégorisation de documents c'est-à-dire aux approches de classifications supervisées. La plupart des approches de catégorisation existantes reposent sur des méthodes considérées comme des approches statistiques [Seb02] telles que les SVM (*Support Vector Machines*) [SV95, Joa98]. Néanmoins, même si leurs performances, en terme de classification, sont intéressantes, aucune de ces méthodes ne fournit un résultat compréhensible de la connaissance extraite et surtout ré-utilisable. Pour résoudre ce problème, une approche de classification basée sur les règles d'association a été initialement proposée par Bing Liu (CBA) [LHM98]. Cependant, les performances comme classifieur n'étaient pas très satisfaisantes et de nouvelles extensions ont été proposées [WZH00, LHP01, JWB<sup>+</sup>03, BG03].

Basées sur des itemsets, ces méthodes considèrent les textes comme des *sacs de mots* où aucun ordre n'est pris en compte lors du processus de classification. Dans ce chapitre, nous nous posons la question de l'utilisation des motifs séquentiels obtenus à partir de bases de documents de type textuel et en particulier dans un objectif de classification automatique. Nous souhaitons ainsi répondre aux questions suivantes : Quelles sont les avantages offerts en adoptant les motifs séquentiels dans un objectif de catégorisation ? Quelles sont les limites inhérentes à une telle approche ?

Ce chapitre est organisé de la façon suivante. La section 6.2 présente les principales techniques de recherche de motifs par catégorisation. Nous présentons dans la section 6.3, une nouvelle approche pour la catégorisation de documents appelée SPAC (*Sequential PATterns for Categorization*). Au cours de la section 6.4, nous revenons sur le double problème de représentations des données qui sont d'une part celle des documents et d'autre part celles des catégories qui sont associées. L'idée dans ce cas est de pouvoir comparer la "sémantique" des textes et des catégories (déduite des ou de la méthode(s) d'apprentissage utilisée(s)). Nous concluons cette partie par une discussion.

## 6.2 Le point

La fouille de textes a été très étudiée [LAS97, AMS97, AJS00, Seb02]. Notre objectif dans cette section est de se focaliser sur la classification de textes et les motifs fréquents.

## Classification basée sur les règles d'association : la méthode CBA

Dans [LHM98], la méthode CBA basée sur les règles d'association est proposée. CBA est composée de deux modules, un générateur de règles (CBA-RG) basé sur l'algorithme Apriori et un constructeur de classifieur basé sur les règles précédemment obtenues (CBA-CB).

### Le générateur de règles CBA-RG

Il s'agit de trouver toutes les paires  $\rho = \langle \text{conset}, C_i \rangle$ , avec *conset* une liste d'items et  $C_i$  une catégorie, dont le support est supérieur au support minimum. Chaque paire  $\rho$  correspond à une règle  $\text{conset} \rightarrow C_i$  dont le support et la confiance sont définis par :

$$\text{supp}(\rho) = \frac{\#\text{textes de } C_i \text{ supportant conset}}{\#\text{textes de la base}}$$

$$\text{conf}(\rho) = \frac{\#\text{textes de } C_i \text{ supportant conset}}{\#\text{textes de la base supportant conset}}$$

Les paires dont le support est supérieur au support minimum sont des paires fréquentes. Si deux paires ont le même ensemble d'items, alors la paire ayant la confiance la plus élevée sera choisie comme règle. L'ensemble des règles d'association pour les catégories (CARs) est constitué de toutes les règles dont le support et la confiance sont supérieurs au support minimum et à la confiance minimum spécifiés par l'utilisateur.

Les motifs fréquents sont extraits en utilisant une unique valeur pour le support minimum quelle que soit la catégorie. Or, toutes les catégories ne contiennent pas le même nombre de documents. Un support minimum élevé ne permettra pas de trouver de motifs fréquents pour les petites catégories et à l'opposé, un support trop élevé va conduire à la génération d'un nombre trop important de règles pour les catégories contenant de nombreux textes. C'est pourquoi d'autres travaux ont proposé d'utiliser des valeurs de support minimum adaptées à chaque catégorie (msCBA) [JWB<sup>+</sup>03, LMW00]. Les règles sont alors extraites en adoptant une stratégie de supports minimums multiples définis en adéquation avec la fréquence de distribution de chaque catégorie et un support minimum initial donné par l'utilisateur :

$$\text{minSup}_{C_i} = \text{minSup}_{\text{initial}} * \text{frequenceDistribution}(C_i)$$

### Le constructeur de classifieur CBA-CB

Soit  $R$  l'ensemble des règles CARs obtenues lors de l'étape précédente et  $T_{\text{Train}}$  le jeu d'entraînement, le catégoriseur est construit à partir de la liste des règles  $r_i \in R$  ordonnées suivant leur confiance. Chaque règle est ensuite testée sur  $T_{\text{Train}}$ . Si la règle n'améliore pas le taux d'apprentissage du classifieur, alors la règle est éliminée de la liste des règles pour la catégorie examinée. L'algorithme 7 détaille le processus de construction du catégoriseur.

Le catégoriseur obtenu est une liste du type :

$$\langle (r_1, r_2, \dots, r_k), C_i \rangle$$

où  $C_i$  est la catégorie cible et  $r_j$  une des règles associées.

Une fois le catégoriseur obtenu, pour tout nouveau texte à classer, les règles de classification sont évaluées sur le document tant qu'aucune règle n'est supportée. La catégorie affectée est alors la catégorie cible de la règle de classification ayant été validée pour le texte.

```

begin
   $R = \text{tri}(R)$ ;
  foreach règle  $r \in R$  do
     $temp \leftarrow \emptyset$ ;
    foreach texte  $d \in D$  do
      if  $d$  valide  $r$  then
         $temp \leftarrow temp \cup d.id$  et marquer  $r$  si  $d$  a été bien classé;
      if  $r$  est marqué then
        ajouter  $r$  dans  $C$ ;
         $D \leftarrow D \setminus temp$ ;
        choisir une catégorie pour  $C$ ;
        calculer le taux d'erreur de  $C$ ;
    trouver la 1ère règle  $p \in C$  minimisant le taux d'erreur et effacer toutes les
    règles après  $p$ ;
    retourner  $C$ ;
end

```

**Algorithme 7:** Construction du catégoriseur CBA-CB

## Améliorations et autres approches

Dans [JWB<sup>+</sup>03], les auteurs proposent de remplacer la mesure de confiance par l'intensité d'implication comme critère de tri des règles lors de la construction du classifieur. Dans [ea01], le classifieur est amélioré en lui adjoignant des arbres de décision afin d'améliorer le taux d'apprentissage. Dans [AMS97], les règles d'association sont utilisées pour permettre une classification partielle, c'est-à-dire que le système ne permet pas de classer pour toutes les catégories. En particulier, cette approche est intéressante dans le cas de valeurs manquantes. [BG03] propose également une méthode de classification basée sur les règles d'association mais contrairement à CBA-CB qui ne tient compte d'une seule règle pour prendre la décision d'affectation d'un texte à une catégorie, les auteurs proposent de considérer plusieurs règles puis d'adopter la catégorie majoritaire. Cette méthode intègre une étape d'élagage de règles basée sur le  $\chi^2$ , comme dans [LHP01]. De plus, un paramètre *maxrules* définit le nombre maximal de règles à vérifier lors de l'étape de classification de nouveaux textes. Pour améliorer les performances, les règles sont classées par niveau, le système étudiera les règles du niveau supérieur si et seulement si aucune règle de classification n'a pu être mise en œuvre au niveau inférieur. Le système a été amélioré dans [BCG04] en considérant une stratégie de supports minimums multiples.

Nous pouvons également citer d'autres travaux connexes. [Had03] propose une méthode basée sur les règles d'association dans le cadre de l'extraction de syntagmes nominaux. Dans ses travaux, l'auteur montre que les règles d'association sont très intéressantes pour rendre compte de la structure linguistique des textes. Cependant, les règles obtenues ne sont pas utilisées pour catégoriser les textes. [Mas03] propose une méthode de catégorisation pour l'analyse de comportements d'utilisateur de sites web (web usage mining) à l'aide de motifs séquentiels et de réseaux de neurones. Les motifs séquentiels sont utilisés pour diviser itérativement la base de données (logs) en sous-logs (eux-mêmes redivisés) représentant chacun un comportement différent. Les réseaux de neurones permettent de composer les groupes selon les motifs séquentiels trouvés. Cette méthode, bien qu'intéressante, n'est pas basée sur les motifs séquentiels

eux-mêmes pour l'étape de classification et n'est pas dédiée aux textes. L'utilisation des réseaux de neurones rend la méthode inapplicable face à de très gros volumes de données. De plus, la méthode est dédiée à l'analyse de comportements peu fréquents (pour ne pas découvrir de connaissance déjà connue) et n'est donc pas adaptée pour la classification d'un ensemble de données et la découverte de tendances. Enfin, il ne s'agit pas de classification supervisée mais plutôt de classification non-supervisée (hiérarchique, par divisions successives de la base).

En ce qui concerne les motifs séquentiels appliqués aux textes, nous pouvons citer [LAS97, WCF<sup>+</sup>00]. [WCF<sup>+</sup>00] propose une approche intégrant deux méthodes. La première est basée sur la visualisation des occurrences des mots afin de détecter des motifs séquentiels. La seconde adopte un algorithme de recherche de motifs séquentiels. Dans [LAS97], les auteurs montrent l'intérêt d'utiliser les motifs séquentiels sur de grandes bases de documents, en particulier pour mettre en évidence les différentes tendances au cours du temps.

Comme nous avons pu le constater, la fouille de textes basée sur les motifs fréquents correspond soit à des travaux sur la classification à l'aide de règles d'association soit à d'autres problématiques résolues en adoptant les motifs séquentiels. Aucune approche à notre connaissance n'utilise les motifs séquentiels comme outil de classification de textes. Dans la section suivante, nous proposons une approche originale permettant d'intégrer une notion d'ordre au sein des textes tout en permettant le traitement de gros volumes de données.

## 6.3 Vers une catégorisation par motif séquentiels : SPAC

Dans cette section nous présentons SPAC (*Sequential Patterns for Categorization*), une approche de catégorisation originale basée sur les motifs séquentiels. La méthode se décompose en deux phases :

1. L'extraction des motifs séquentiels à partir de la base de documents. La granularité considérée est celle de la phrase, i.e. chaque document est considéré comme une suite ordonnée de phrases, elles mêmes constituées d'un ensemble de mots non ordonnés.
2. La construction d'un classifieur basé sur les motifs séquentiels préalablement obtenus.

### 6.3.1 Première étape - Des textes aux motifs séquentiels

Chaque texte du jeu d'entraînement est transformé afin d'appliquer un algorithme de recherche de motifs séquentiels. Nous proposons de prendre en compte l'ordre des phrases au sein du texte mais les phrases quant à elles sont considérées comme des "sacs de mots". En effet, nous faisons l'hypothèse que l'ordre des mots dans la phrase a une importance limitée mais que celui des phrases dans le texte a un impact lors du processus de classification. Nous considérons donc chaque texte comme un client et chaque phrase comme une transaction estampillée par sa position au sein du texte. L'ensemble des mots représente l'ensemble des items et correspond à un itemset. Les mots d'une même phrase sont ainsi assimilés aux achats effectués par un client à une même date en adéquation avec la problématique du "panier de la ménagère". La figure 6.1 résume les règles de correspondances entre les textes et les concepts associés motifs séquentiels.

Base de données d'achats		Base de données textuelle
client	↔	texte
item	↔	mot
itemset/transaction	↔	phrases (ensemble de mots)
date	↔	position de la phrase dans le texte

**Fig. 6.1:** Correspondance entre les textes et les motifs séquentiels

Nous réalisons un pré-traitement linguistique de type stemmatisation (radicalisation) et une suppression des mots non informatifs (stop-list) des textes traités. L'étape de stemmatisation supprime tous les suffixes des mots afin d'obtenir des stemmes qui sont plus génériques. L'étape de suppression des mots non informatifs élimine les mots comme "le, des, une" pouvant générer du bruit lors de la phase d'apprentissage. Toujours pour atténuer le bruit, une politique de suppression des mots non discriminant, basée sur une mesure d'entropie, a été mise en œuvre. Ceci nous permet d'effectuer une recherche de motifs avec un plus faible support sans générer un trop grand nombre de candidats.

L'élimination par entropie est réalisée sur la base d'un seuil. Pour chaque mot  $w$ ,  $H(w)$  définit l'entropie de ce mot pour l'ensemble des catégories  $C_i$  :

$$H(w) = - \sum_{C_i} p(w).p(C_i|w).log(p(C_i|w)) - ((1 - p(w)).p(C_i|\bar{w}).log(p(C_i|\bar{w})))$$

Le seuil d'élimination a été déterminé de manière empirique. Les meilleurs résultats ont été obtenus en éliminant 5 à 10% des mots. Ces résultats concordent avec ceux définis par la loi de Zipf [SYY75].

Notation	Signification
$\mathcal{C} = \{C_1, \dots, C_n\}$	l'ensemble des $n$ catégories
$C_i \in \mathcal{C}$	une catégorie de $\mathcal{C}$
$minSup_{C_i}$	le support minimum de la catégorie $C_i$ , défini par l'utilisateur
$T$	l'ensemble des textes
$T^{C_i} \subseteq T$	les textes appartenant à $C_i$
$T_{Train} = \{(C_i, T^{C_i})\}$	le jeu d'apprentissage constitué d'un ensemble de textes associés à leur catégorie.
$SEQ$	accesseur contenant l'ensemble des séquences ordonnées par catégorie $C_i$ , client $c$ et date $t$
$SP$	un tableau de motifs séquentiels
$RuleSP$	un tableau de tuple $(sp_j, C_i, conf_{i,j})$ correspondant à la séquence $sp_j$ , la catégorie $C_i$ et la confiance $conf_{i,j}$ de la règle $sp_j \rightarrow C_i$

**Fig. 6.2:** Notations

SPAC extrait l'ensemble des motifs séquentiels selon une politique de supports minimums multiples identique à celle de msCBA. Cela permet de définir un support pour chacune des catégories  $C_i$ . Lors d'une recherche de motifs séquentiels avec un support de 10%, la recherche ne s'effectuera plus sur toute la base mais catégorie par

catégorie. Ainsi, les motifs séquentiels d'une catégorie contenant peu de textes (dont le nombre est inférieur à 10% de la base) ne seront pas ignorés. Contrairement à msCBA où le support minimum est défini selon une formule (C.f. section 6.2), dans SPAC, les supports minimums de chaque catégorie sont définis par l'utilisateur. Ceci permet d'affiner l'étape de classification en intégrant dans le processus la connaissance des experts pour chaque catégorie. Les expérimentations réalisées dans [BCG04] indiquent l'importance d'un support spécifique à chaque classe et au sein d'un même jeu de données, les supports optimaux entre les différentes classes varient nettement. Nous divisons le jeu d'entraînement en  $n$  sous-ensembles correspondant aux textes des  $n$  catégories. Ensuite, un algorithme de recherche de motifs séquentiels est appliqué sur chacun des sous-ensembles selon le support minimum spécifié. Les motifs séquentiels fréquents sont donc obtenus pour chaque catégorie et leur support conservé. Le support d'un motif fréquent correspond au nombre de textes qui le supporte (ou qui le contient).

**Définition 6** Soit  $\langle s_1 \dots s_p \rangle$  une séquence. Le support de  $\langle s_1 \dots s_p \rangle$  est défini par :

$$\text{supp}(\langle s_1 \dots s_p \rangle) = \frac{\#\text{textes supportant } \langle s_1 \dots s_p \rangle}{\#\text{textes de la base}}$$

L'algorithme 8 définit la phase d'extraction de motifs séquentiels. La fonction  $SPMining()$  appelle l'algorithme SPAM [AGYF02] pour rechercher les séquences fréquentes.

```

Data :  $T_{Train}$  : jeu d'entraînement
          $\{minSup_{C_i}\}$  : l'ensemble des supports minimums pour chacune des
         catégories  $C_i$ 

Result : SP : un ensemble de motifs séquentiels

begin
  SEQ  $\leftarrow \emptyset$ ; customer  $\leftarrow 0$ ; timestamp  $\leftarrow 0$ ;
  foreach Catégorie  $C_i \in \mathcal{C}$  do
    foreach Texte  $T_j \in T^{C_i}$  do
      foreach Phrase  $S_k \in T_j$  do
         $V_s = \text{TFIDF}(\text{Stemme}(S_k))$ ; // Génère un vecteur de type TF-
        IDF à partir de la phrase  $S_k$ 
        for ( $s = 0$ ;  $s < |V_s|$ ;  $s++$ ) do
          if  $V_s[s] > 0$  then
             $SEQ[C_i][customer][timestamp].\text{additem}(s)$ ;
          timestamp++;
        timestamp  $\leftarrow 0$ ; customer++;
      customer  $\leftarrow 0$ ;
    foreach Catégorie  $C_i \in \mathcal{C}$  do
       $SP[C_i] = \text{SPMining}(SEQ[C_i], minSup_{C_i})$ ;
  end

```

**Algorithme 8:** SPaC – RG : génération des règles

Par exemple, les motifs fréquents suivants ont été extraits de la catégorie “Achats-Logistique” d'un jeu de données français :

```

< (cacao) (ivoir) (abidjan)>
< (ble soja) (mai)>
< (soj)(blé lespin victor)(maï soj )(maï )(grain soj)(soj tourteau)>

```



Le premier motif indique que pour les textes de la catégorie, il apparaît régulièrement une phrase contenant le mot *cacao* suivie d'une phrase contenant le mot *ivoire* et enfin une phrase contenant *Abidjan*. Le second motif séquentiel signifie qu'un certain nombre de textes (au moins le support minimal) contiennent les mots *ble* et *soja* au sein d'une même phrase suivie d'une phrase contenant *maï*. Le troisième motif séquentiel peut être interprété par exemple : le mot *maï* apparaît dans deux phrases successives et est suivi par une phrase contenant le mot *grain*.

Nous pouvons constater que l'utilisation de motifs séquentiels permet de prendre en compte certaines occurrences multiples de mots (contrairement aux règles d'associations).

### 6.3.2 Deuxième étape - Des motifs séquentiels aux catégories

L'objectif de cette seconde étape est de générer un catégoriseur à partir des motifs séquentiels extraits lors de l'étape précédente. Cette construction est basée sur une notion de confiance et se définit comme suit : Pour chaque motif séquentiel  $\langle s_1 \dots s_p \rangle$  extrait pour une catégorie  $C_i$ , la règle  $\gamma$  est définie de la façon suivante :

$$\gamma : \langle s_1 \dots s_p \rangle \rightarrow C_i$$

Cette règle signifie : si un texte contient  $s_1$  suivi de  $s_2 \dots$  et de  $s_p$ , alors le texte valide son appartenance à  $C_i$ . La confiance de cette validation est déterminée par la confiance de la règle définie par :

$$\text{conf}(\gamma) = \frac{\#\text{textes de } C_i \text{ supportant } \langle s_1 \dots s_p \rangle}{\#\text{textes de la base supportant } \langle s_1 \dots s_p \rangle}$$

Plus la confiance d'une règle est grande, plus le motif séquentiel est discriminant pour la catégorie qui lui est associée. Les règles sont ensuite ordonnées selon leur confiance et selon la taille de leur séquence (second critère).

Pour chaque nouveau texte à classer, la politique de catégorisation est la suivante : une fois la liste des règles ordonnée, on parcourt cette liste de façon décroissante en appliquant les motifs séquentiels de chacune des règles au texte à catégoriser. Une fois les  $K$  premières règles valides trouvées, le texte est affecté à la catégorie majoritaire défini sur ces  $K$  règles. Cette méthode correspond à la méthode de "vote majoritaire" adoptée dans [BG03]. Si deux ou plusieurs catégories obtiennent le même score, alors un choix aléatoire est effectué pour déterminer la catégorie d'appartenance du texte. Il se peut qu'il n'existe pas  $K$  règles valides. Dans ce cas particulier, le vote majoritaire s'effectue normalement sur les  $n$  règles valides (avec  $n < K$ ). Et si finalement il n'existe aucune règle valide pour le texte en question, alors ce dernier n'est pas catégorisé.

L'étape de catégorisation de SPAC est décrite par l'algorithme (SPAC-C) suivant :

```

Data :  $T_{Test}$  : A Test Set
        KFS (le paramètre  $K$ ),
        SP (le tableau des motifs séquentiels générés par SPAC-RG)

begin
   $nb \leftarrow 1$  ;
  foreach Catégorie  $C_i \in C$  do
    foreach  $sp_j \in SP[C_i]$  do
       $RuleSP[nb] \leftarrow (sp_j, C_i, conf(sp_j \rightarrow C_i))$  ;
       $nb++$  ;
    end
    Trier  $RuleSP$  selon la confiance des règles (et par la taille de la séquence en second
    critère);
     $nfs \leftarrow 0$ ;  $classable \leftarrow 0$ ;
    foreach Texte  $T_k \in T_{Test}$  do
      foreach Règle  $(sp_j \rightarrow C_i) \in RuleSP$  do
        if  $T_k$  supporte  $SP_j$  then
           $T_k.score[C_i]++$ ;  $classable \leftarrow 1$ ;  $nfs++$  ;
          if  $nfs \geq KFS$  then break
        end
      if  $classable$  then
        Affecter  $T_k$  à la catégorie ayant obtenu le meilleur score;
       $classable \leftarrow 0$ ;  $nfs \leftarrow 0$  ;
    end
  end

```

## 6.4 La classification de documents : le MCT

Dans cette section, nous revenons sur le problème de la représentation des documents et définissons formellement un modèle pour permettre de comparer des classificateurs. Nous illustrons la représentation des catégories que nous avons utilisées dans nos travaux avec les approches Rocchio et SVM.

Les documents numériques disponibles sont en nombre perpétuellement croissant. L'intérêt de disposer de méthodes, de techniques efficaces de classification n'est plus à démontrer et de nombreux travaux de recherche se focalisent sur cet aspect [Seb02, YL99].

De façon très globale, le processus de catégorisation de document peut être décomposé selon :

- le modèle de représentation des documents et des catégories,
- la mesure de similitude et le système d'élection qui permettent de déterminer l'appartenance, ou non, d'un document à une catégorie.

Pour réaliser un processus de catégorisation, la première étape consiste donc à formaliser les textes afin qu'ils soient utilisables aussi bien pour les algorithmes d'apprentissage, que lors de l'étape de catégorisation. Cette étape est bien entendu cruciale car c'est elle qui permettra ou non aux méthodes d'apprentissage de produire une bonne généralisation à partir du jeu d'apprentissage.

Formellement, un processus de catégorisation se définit comme une fonction :

$$\check{\Phi} : D \times C \rightarrow \{Vrai, Faux\}$$

avec  $D$  l'ensemble des documents et  $C$  l'ensemble des catégories

L'objectif d'un processus de catégorisation est donc d'approximer la fonction précédente par une fonction  $\Phi$  dans le but de maximiser une fonction d'évaluation.

Nous définissons donc le modèle de catégorisation textuel suivant afin de représenter les différentes étapes du processus de catégorisation.

Le modèle de catégorisation textuel général  $MCT_{Gen}$  se définit par le tuple :

$$MCT_{Gen} = (V_T, T, R_T, rep_T, V_C, C, R_C, rep_C, sim_{TC}, CVS)$$

avec :

{	$V_T$	un vocabulaire qui est un ensemble fini de dimension $ V_T $ .
	$T$	un ensemble de segments textuels tels que : $\forall t \in T, t \in V_T^*$ .
	$R_T$	une représentation mathématique (espace métrique, ensemble ordonné, etc...).
{	$rep_T(t) \rightarrow R_T$	une fonction permettant de générer une représentation $r \in R_T$ à partir d'un segment textuel $t \in T$ .
	$V_C$	un ensemble de segments textuels finis de dimension $ V_C $ .
	$C$	un ensemble de classes tel que : $C \subseteq P(V_C)$ .
{	$R_C$	une représentation mathématique (espace métrique, ensemble ordonné, etc...).
	$rep_C(c) \rightarrow R_C$	une fonction permettant de générer une représentation $r \in R_C$ à partir d'une classe $c \in C$ .
	$sim_{TC}(r_t, r_c) \rightarrow \mathbb{R}^+$	une relation entre $r_t \in R_T, r_c \in R_C$ .
{	$CVS(t, c) \rightarrow [0, 1]$	une politique de catégorisation avec $t \in T, c \in C$ .

Nous définissons aussi  $\{T_{App}, T_{Test}\}$  une partition de  $T$  définissant respectivement le jeu d'apprentissage et le jeu de test.  $T_{App}$  est utilisé pour construire  $rep_C(c)$ ,  $T_{Test}$  sert seulement lors de l'évaluation. L'objectif de ce modèle est de formaliser, et de différencier, chacune des étapes du processus de catégorisation qui sont : la formalisation des textes et des classes <sup>1</sup> ainsi que la définition d'une mesure de similitude et d'une politique de catégorisation.

Dans une problématique de catégorisation, l'intérêt de  $rep_T$  (formalisation des textes) réside dans sa capacité à pouvoir "extraire" l'information du texte nécessaire à une bonne catégorisation. Quant à l'intérêt de  $rep_C$  (formalisation des catégories), il réside dans sa capacité à pouvoir modéliser la notion de classe, c'est-à-dire extraire d'un ensemble de textes l'information qui leur est commune.

Dans la section 6.4.1 associée à la représentation des documents, nous ne définissons que la partie "haute" du MCT, c'est-à-dire la partie servant à représenter les textes, tandis que dans la section concernant les méthodes de catégorisation 6.4.2, ce n'est que la partie "basse" du MCT qui sera développée. Un processus de catégorisation complet est donc défini par une partie "haute" ainsi qu'une partie "basse" du MCT.

### 6.4.1 Représentation des documents

La représentation textuelle la plus utilisée est issue de [Sal71, SM83] dont l'implémentation la plus connue est SMART. Dans ce formalisme vectoriel, chaque dimension de l'espace correspond à un élément textuel, nommé terme d'indexation, préalablement extrait du jeu d'apprentissage. La construction du vecteur d'un texte est déterminée par des propriétés statistiques de chacun des termes d'indexation du texte en question.

<sup>1</sup>Il est très courant que l'espace de représentation permettant de formaliser les textes et les classes soit identique ( $R_T = R_C$ ).

Dans [JTCP03], les auteurs proposent une nouvelle méthode de représentation des documents. Au lieu de définir un espace vectoriel dont chaque dimension représente un terme d'indexation, souvent assimilé à un stem (radical), l'ensemble des termes est projeté sur un ensemble fini de concepts extrait d'un thésaurus. L'intérêt d'une telle méthode est de réduire les effets polysémiques du vocabulaire. En effet, deux synonymes partageront un ensemble de mêmes concepts. Cette représentation permet donc une factorisation des termes par regroupement de leur champ sémantique.

Pour permettre une telle représentation des documents, il est nécessaire de pouvoir projeter n'importe quelles lexies du dictionnaire sur l'espace généré par l'ensemble des concepts prédéfinis. Comme espace de concepts, nous utilisons le thésaurus Larousse composé de 873 concepts hiérarchisés en 4 niveaux. Par exemple, le mot "mélodie", défini par les concepts 741,781 et 784 (phrase, musique et chant) du thésaurus, sera représenté par un vecteur de dimension 873 dont toutes les composantes sont nulles sauf celles associées aux concepts 741, 781 et 784 qui seront identiques. Le thésaurus Larousse sera donc défini comme un ensemble de couples de  $L \times \mathbb{R}^{873}$  avec  $L$  correspondant à l'ensemble des lemmes du thésaurus.

Bien que se basant aussi sur le formalisme vectoriel pour représenter les documents, cette représentation reste fondamentalement différente de la représentation saltonnienne [SM83]. Les dimensions de l'espace vectoriel ne sont pas associées ici à des termes d'indexation mais à des concepts comme dans [Cha90].

Cependant, l'inconvénient majeur de cette représentation reste que les noms propres du document ne sont pas pris en compte. En effet, les noms propres, étant sémantiquement vides par définition, ne possèdent pas de représentation au sein du thésaurus.

Dans le cas de SPAC, les vecteurs conceptuels des textes ont été générés grâce au lemmatiseur défini dans [Sch94]. Même si ce type d'analyseur reste limité pour ce qui est de l'analyse syntaxique, il offre néanmoins l'avantage de fonctionner dans toutes les langues.

### Représentation statistique des documents (*TF-IDF*)

La majorité des approches de catégorisation sont axées sur une représentation vectorielle des textes de type *TF-IDF* qui est très utilisée en recherche d'information [Seb02]. En effet, *TF* (*Term Frequency*) par *IDF* (*Inverse Document Frequency*) correspond à la fréquence d'un terme multipliée par l'inverse de sa fréquence en document.

L'étape de représentation textuelle des documents peut-être représentée par le  $MCT_{TF-IDF}$  partiel suivant :

$$\left\{ \begin{array}{ll} V_T & \text{l'ensemble des termes d'indexation de dimension } |V_T| . \\ T & \text{un ensemble de textes tels que : } \forall t \in T, t \in V_T^* . \\ R & \text{l'espace vectoriel } \mathbb{R}^{|V_T|} . \\ rep_T(t) \rightarrow R & \text{une fonction permettant de générer un vecteur } \\ & \vec{r} \in R \text{ à partir d'un segment textuel } t \in T . \\ C & \text{un ensemble de classes tel que : } C \subseteq \mathcal{P}(T_{App}) . \end{array} \right.$$

Avant de définir  $rep_T$ , nous introduisons les fonctions suivantes :

$$\left\{ \begin{array}{l} \text{STEMMER}(t) \rightarrow \text{LISTE-STEMMES} \\ \quad \text{qui produit une liste de stemmes à partir d'un texte } t \in T . \\ \text{VECTEUR(LISTE-STEMMES)} \rightarrow \mathbb{R}^{|V_T|} \\ \quad \text{qui génère un vecteur de type } TF-IDF \text{ à partir d'une liste de stemmes.} \end{array} \right.$$

La fonction de représentation des documents devient :

### Représentation conceptuelle des documents

```

Data      : Un texte  $t \in T$ 
Result   : Une représentation  $\vec{r}_t \in R$ 
begin
  LISTE-STEMMES = STEMMER(t);
   $\vec{r}_t$  = VECTEUR(LISTE-STEMMES);
  return  $\vec{r}_t$ ;
end

```

**Algorithme 10:**  $rept_{TF-IDF}$

L'étape de représentation textuelle des documents peut-être représentée par le  $MCT_{Concept}$  partiel suivant :

$$\left\{ \begin{array}{ll} V_T & \text{l'ensemble des termes d'indexation de dimension } |V_T| . \\ T & \text{un ensemble de textes tels que : } \forall t \in T, t \in V_T^* . \\ R & \text{l'espace vectoriel } \mathbb{R}^{|V_T|} . \\ rept(t) \rightarrow R & \text{une fonction permettant de générer un vecteur} \\ & \vec{r} \in R \text{ à partir d'un segment textuel } t \in T . \\ C & \text{un ensemble de classes tel que : } C \subseteq \mathcal{P}(T_{App}) . \end{array} \right.$$

Mais avant de présenter la fonction  $rept$ , nous définissons les trois fonctions suivantes :

$$\left\{ \begin{array}{l} \text{TREE-TAGGER}(t) \rightarrow \text{LISTE-LEMME} \\ \quad \text{qui produit une liste de lemmes à partir d'un texte } t \in T . \\ \text{VECTEUR(LEMME)} \rightarrow \mathbb{R}^{873} \\ \quad \text{qui génère un vecteur à partir d'une liste de lemmes.} \\ \text{THESAURUS}(l) \rightarrow \mathbb{R}^{873} \\ \quad \text{qui associe à chaque lemme } l \in L \text{ du thésaurus un vecteur } \in \mathbb{R}^{873} . \end{array} \right.$$

Après avoir extrait l'ensemble des lemmes d'un texte, une association, grâce à la fonction *THESAURUS*, est donc réalisée entre les lemmes et le vecteur qui leur est associé au sein du thésaurus. Ensuite, le vecteur conceptuel de chaque texte est calculé en fonction de la moyenne normalisée des lemmes qu'il contient :

$$\vec{r}_t = \frac{r_{l1} + r_{l2} + \dots + r_{ln}}{\|r_{l1} + r_{l2} + \dots + r_{ln}\|}$$

C'est la fonction *VECTEUR* de l'algorithme 11 qui associe un vecteur conceptuel à un texte donné en entrée.

La fonction de représentation des documents sur l'espace conceptuel est décrite dans l'algorithme 11.

```

Data      : Un texte  $t \in T$ 
Result   : Une représentation  $\vec{r}_t \in R$ 
begin
  LISTE-LEMME = TREE-TAGGER(t);
   $\vec{r}_t$  = VECTEUR(LISTE-LEMME);
  return  $\vec{r}_t$ ;
end

```

**Algorithme 11:**  $rept_{Concept}$

## 6.4.2 Représentation des catégorisations et du catégoriseur

Nous illustrons la représentation des catégories à partir de deux catégoriseurs reconnus : Rocchio [Roc71] et les machines à vecteur de support (SVM) [Bur98] que nous allons décrire selon le MCT général défini.

## Rocchio

Rocchio [Roc71] est l'une des méthodes les plus anciennes en catégorisation. Nous la définissons ici dans sa version basique. Les catégories sont représentées dans un espace vectoriel similaire aux documents. En effet, le vecteur d'une catégorie est défini comme la moyenne des vecteurs des textes qu'elle contient ( $rep_C$ ).

Une fois les textes et les catégories représentés dans un même espace, la similitude entre un texte et une classe est définie par la distance euclidienne ( $sim$ ). Par conséquent, la politique de catégorisation se résume à associer à chaque texte la catégorie dont la distance euclidienne est la plus proche ( $CVS$ ).

Sans détailler les algorithmes  $rep_{C_{Rocchio}}$ ,  $sim_{Rocchio}$  et  $CVS_{Rocchio}$  triviaux, nous représenterons la méthode Rocchio grâce au  $MCT_{Rocchio}$  partiel suivant :

$$\left\{ \begin{array}{ll} C & \text{un ensemble de classes tel que : } C \subseteq P(T_{App}). \\ rep_{C_{Rocchio}}(c) \rightarrow R & \text{la fonction permettant de générer un vecteur} \\ & r_c \in R_C \text{ à partir d'une classe } c \in C. \\ sim_{Rocchio}(r_t, r_c) \rightarrow \mathbb{R} & \text{la distance euclidienne entre } r_t \text{ et } r_c \text{ avec} \\ & r_t \in R_T, r_c \in R_C. \\ CVS_{Rocchio}(t, c) \rightarrow [0, 1] & \text{la politique de catégorisation définie avec } t \in T, c \in C. \end{array} \right.$$

## Machines à vecteur de support

Les machines à vecteur de support (SVM) sont à l'origine de nouvelles méthodes de catégorisation [Joa98] bien que les premières publications sur le sujet datent des années 60 [VC64]. Le principe des SVM consiste en une stratégie de minimisation structurelle du risque [Vap95]. Le lecteur peut se référer à [Bur98] pour une présentation générale de la méthode. En ce qui concerne son application à la problématique de catégorisation de documents, l'approche par SVM permet de définir, par apprentissage, une surface de séparation entre des exemples positifs et négatifs minimisant le risque d'erreur et maximisant la marge entre deux classes. La figure 6.3 montre une telle séparation dans le cas d'une séparation linéaire par un hyperplan. Il est intéressant de remarquer qu'en réduisant le jeu d'entraînement uniquement aux vecteurs de support, l'algorithme calculerait le même hyperplan que pour le jeu d'entraînement complet. La marge se présente alors comme la plus courte distance entre un vecteur de support et "son" hyperplan.

De manière formelle, un hyperplan peut être défini par :

$$\vec{w} \cdot \vec{x} - b = 0$$

Avec  $\vec{x}$  un point arbitraire,  $\vec{w}$  un vecteur et  $b$  le biais.

Soit  $D = \{(\vec{x}_i, y_i)\}$  notre jeu d'entraînement et  $y_i \in \{\pm 1\}$  définissant l'état, positif ou négatif, de l'exemple. Trouver l'hyperplan maximisant la marge séparatrice ( $\frac{2}{\|\vec{w}\|}$ ) revient à résoudre le problème suivant :

$$\left\{ \begin{array}{ll} \text{minimiser} & \frac{1}{2} \|\vec{w}\|^2 \\ \text{sous les contraintes} & \forall i, y_i(\vec{w} \cdot \vec{x}_i + b) - 1 \geq 0 \end{array} \right.$$

Grâce à une extension de cet algorithme, il est aussi possible de résoudre des problèmes qui ne sont pas linéairement séparables, mais l'amélioration obtenue pour la catégorisation de documents reste minime [Joa98]. Pour la construction vectorielle des textes, ce sont en général les stemmes (radicaux) qui sont utilisés comme termes d'indexation.

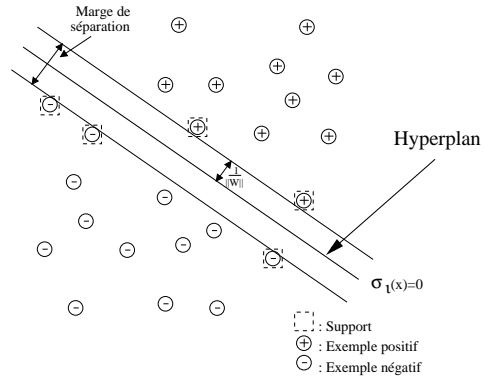


Fig. 6.3: Représentation de l'hyperplan optimal

On représentera la catégorisation par SVM linéaire grâce au  $MCT_{SVM}$  partiel suivant :

$$\left\{ \begin{array}{ll} C & \text{un ensemble de classes tel que : } C \subseteq P(T_{App}). \\ R_C & \text{un ensemble d'hyperplans .} \\ rep_{C_{SVM}}(c) \rightarrow R_C & \text{la fonction permettant de générer un hyperplan} \\ & r_c \in R_C \text{ à partir d'une classe } c \in C. \\ sim_{SVM}(r_t, r_c) \rightarrow \mathbb{R} & \text{la position du point } r_t \text{ par rapport à l'hyperplan } r_c \text{ avec} \\ & r_t \in R, r_c \in R_C. \\ CVS_{SVM}(t, c) \rightarrow [0, 1] & \text{la politique de catégorisation définie avec } t \in T, c \in C. \end{array} \right.$$

Avec pour  $rep_{C_{SVM}}$ ,  $sim_{SVM}$  les algorithmes suivants :

```

Data      : Une classe  $c \in C$ 
Result   : Une représentation  $r_c \in R_C$ 
begin
  //  $\bar{c}$  est défini comme le complémentaire de  $c$  sur  $C$ 
   $\bar{c} = C - c$ ;
   $r_c$  = l'hyperplan maximisant la marge entre  $\bar{c}$  et  $c$  et minimisant l'erreur;
  return  $r_c$ ;
end

```

Nous ne détaillerons pas  $CVS_{SVM}$  qui est trivialement basé sur le résultat de la fonction  $sim_{SVM}$ .

## 6.5 Discussion

Dans ce chapitre, nous avons proposé un système de catégorisation basé sur les motifs séquentiels. L'extraction des motifs séquentiels est réalisée pour chaque catégorie à partir d'une représentation textuelle des textes de type  $TF-IDF$  et d'une transformation en terme de triplet  $\langle \text{client, date, item} \rangle$ . Dans cette approche, chaque nouveau texte à classer est affecté à une catégorie en fonction des différents motifs séquentiels qu'il supporte grâce à une politique de "vote majoritaire". Nous avons également défini un modèle de classification de document, le MCT, et nous nous sommes attachés à modéliser plusieurs représentations des documents ainsi que différentes approches de catégorisation.

Lors de nos expérimentations, nous avons constaté que SPAC obtenait des résultats meilleurs que msCBA et atteignait ceux de SVM pour des bases difficiles. Toutefois,

```

Data   : Deux représentations  $r_1 \in R, r_2 = (\vec{w}, b) \in R_C$ 
Result : Un booléen  $\in [0, 1]$ 
begin
   $\vec{w}$  = la normale de  $r_2$ ;
   $b$  = la constante de  $r_2$ ;
  //Calcule la position de  $r_1$  par rapport à l'hyperplan  $r_2$ 
  if ( $r_1 \cdot \vec{w} + b \geq 1$ ) then
     $\perp$  return 1;
  else
     $\perp$  return 0;
end

```

**Algorithme 13:**  $sim_{SVM}$

SVM reste le plus efficace sur le corpus Reuters. Néanmoins, il est important de souligner que l'extraction de connaissances "compréhensibles" est un atout tout aussi important qu'une bonne catégorisation. Ces descriptions sont primordiales pour les experts démunis face aux grandes quantités de textes à analyser et traiter. Les motifs séquentiels peuvent être aisément analysés afin de mieux comprendre les forces et les faiblesses du catégoriseur construit et peuvent aussi être utilisés pour rechercher les tendances au sein des bases de textes. Dans notre problématique de catégorisation, les motifs séquentiels ont montré un potentiel très attractif. De plus, SPAC est efficace lorsque des catégoriseurs reconnus comme SVM montrent leurs limites. Pour résumer, un tel système possède trois qualités essentielles :

1. il s'appuie sur des règles compréhensibles et interprétables pour les utilisateurs finaux (contrairement à la grande majorité des systèmes de catégorisation comme les SVM, Rocchio, Naïve Bayes,...),
2. il permet de réaliser des analyses de tendance, comme proposées dans [LAS97], suite aux différentes évaluations constatées dans les catégories et enfin
3. les motifs séquentiels sont plus précis et informatifs que les règles d'association.

Ces travaux très prometteurs ouvrent de nombreuses perspectives. Tout d'abord, l'utilisation des motifs séquentiels généralisés [AS95b] incluant les contraintes de temps permettrait d'améliorer les performances. La mise en œuvre de différents niveaux de motifs comme proposés dans [BCG04] permet de conserver des règles de classification très spécifiques sans nuire aux performances générales du classifieur. Il s'agit d'utiliser un ensemble de règles compactes afin de diminuer le support minimum. Dans cet objectif, il serait intéressant d'étendre les travaux de [CB02] aux motifs séquentiels : utiliser les  $\delta$ -libres pour trouver des règles de classification dont la partie gauche est la plus courte possible. Précédemment, nous avons montré qu'il était important d'offrir une approche incrémentale. Il serait intéressant de proposer un catégoriseur incrémental ou en temps réel (nous reviendrons sur cet aspect dans les perspectives de ce mémoire) pour la catégorisation automatique de nouvelles où l'actualité est sans cesse mouvante. Les motifs séquentiels sont donc, à notre avis, un premier pas vers la classification temps réel de textes, où OLTCP (*On Line Text Classification Process*).





## Chapitre 7

# Données multidimensionnelles

Ces propositions ont été réalisées lors de l'encadrement de

**Doctorant :** Marc Plantevit  
**Co-encadrant :** Anne Laurent (Maître de Conférences,  
UMII, LIRMM)

Ce chapitre adresse les problématiques

*Représentation des données :* Données multidimensionnelles  
*Représentation des comportements :* Motifs multidimensionnels  
*Extraction de motifs :* M<sup>2</sup>SP, Hype

## 7.1 Introduction

Même si, comme nous l'avons vu précédemment, il existe de nombreux travaux permettant une extraction de motifs séquentiels efficace, ces propositions ne prennent en compte qu'une seule dimension d'analyse, nommée *produit* dans les approches de type "*panier de la ménagère*". Ainsi, même si cette dimension peut être transposée pour des applications de recherche de motifs séquentiels à d'autres domaines que le panier de la ménagère (i.e. l'étude des comportements d'internautes [TTM04], musique [HLC01], sécurité [LS98], séquences de protéines [WdIJ+02]), il n'en reste pas moins qu'il est impossible d'analyser plus d'une seule dimension à la fois. Ainsi, il n'existe pas à l'heure actuelle de méthode permettant de mettre en évidence des corrélations entre valeurs de différents attributs pour découvrir des règles de la forme :

$$\langle \{(surf, NY), (housse, NY)\}, \{(combi, SF)\} \rangle$$

indiquant qu'un nombre suffisant (au sens du support) de personnes ont acheté leur planche de surf et la housse à New York puis qu'un nombre suffisant de personnes ont acheté une combinaison à San Francisco. Si la littérature recense des contributions liées aux motifs séquentiels multidimensionnels proposées par l'équipe de Jiawei Han [PHP<sup>+</sup>01], celles-ci ne permettent pas de combiner plusieurs attributs au sein des motifs extraits pour ce qui est de la partie séquentielle. En effet, les multiples attributs n'apparaissent que pour restreindre le cadre dans lequel se trouve la séquence fréquente.

Ce chapitre est organisé de la façon suivante. La section 7.2 décrit les approches de la littérature ayant traité le problème de l'extraction de motifs séquentiels dans un contexte multidimensionnel (i.e. plusieurs dimensions d'analyse). Les algorithmes associés sont décrits dans la section 7.3.2. Nous concluons par une discussion.

## 7.2 Le point

Combiner plusieurs dimensions d'analyse permet d'extraire des connaissances qui décrivent mieux les données. Il n'y a plus seulement des corrélations entre items partageant le même itemset et entre itemsets au sein de la séquence. L'extraction de séquences dans un contexte multidimensionnel permet de mettre en évidence des corrélations entre les dimensions instanciées d'un item. Dans [PHP<sup>+</sup>01] les auteurs sont les premiers à définir des motifs séquentiels multidimensionnels. Ainsi, les achats ne sont plus décrits en fonction des seuls date et identifiant du client, mais en fonction d'un ensemble de dimensions telles que *Type de consommateur*, *Ville*, *Age*. Cette approche permet d'extraire des séquences d'items sur la dimension *produits* et de les caractériser à l'aide des informations fréquentes sur les clients (*Patterns*) qui tendent à supporter les séquences. Des connaissances de la forme :  $(chicago, business, \langle \{a, b\}, \{c\} \rangle)$  peuvent donc être extraites. Cette méthode ne permet pas d'avoir des séquences où plusieurs patterns sont présents puisque le *pattern* décrit les clients qui supportent la totalité de la séquence. Une séquence est donc identifiée par un seul pattern. Elle ne permet donc pas d'extraire des connaissances de la forme :  $\langle \{(business, *, *, a)(*, chicago, *, b)\}, \{(*, *, young, c)\} \rangle$  alliant différents patterns multidimensionnels. Dans [YC05], les auteurs proposent d'extraire des séquences au sein de séquence de données multidimensionnelles organisées en différents niveaux de hiérarchie. Néanmoins, les séquences de données ne sont pas réellement multidimensionnelles dans la mesure où les différentes dimensions entretiennent un lien hiérarchique très strict (un jour comporte des sessions qui sont elles-mêmes composées de pages visitées).

### 7.3. VERS DES MOTIFS SÉQUENTIELS MULTIDIMENSIONNELS : $M^2SP$ , $HYPE75$

D (Date)	B (Bloc <sub>ID</sub> )	Pl (Lieu)	P (Produit)
1	1	Allemagne	Bière
1	1	Allemagne	Cacahuètes
2	1	Allemagne	Aspirine
3	1	Allemagne	Chocolat
4	1	Allemagne	Smecta
1	2	France	Coca
2	2	France	Vin
2	2	France	Cacahuètes
3	2	France	Aspirine
1	3	UK	Whisky
1	3	UK	Cacahuètes
2	3	UK	Aspirine
1	4	LA	Chocolat
2	4	LA	Smecta
3	4	NY	Whisky
4	4	NY	Coca

**Fig. 7.1:** Base de données exemple *DB*

Nous pouvons encore citer les travaux de [dAFGL04] qui proposent une approche basée sur la logique temporelle du premier ordre pour l'extraction de motifs séquentiels multidimensionnels, [Lee05] propose également une nouvelle méthode de génération des séquences multidimensionnelles présentes dans des bases de transactions. Cependant cette méthode de génération se réduit uniquement à des séquences d'items sans tenir compte de la possibilité d'avoir des séquences d'itemsets.

## 7.3 Vers des motifs séquentiels multidimensionnels : $M^2SP$ , $HYPE$

Cette section détaille et étend les concepts (motifs séquentiels multidimensionnels, item h-généralisés, etc.) et introduit les algorithmes proposés ( $M^2SP$ ,  $HYPE$ ) [PCL<sup>+</sup>05b], [PCL<sup>+</sup>05a] et [PLT06].

Pour illustrer les différents concepts et définitions, nous proposons la base exemple (C.f. figure 7.1) qui décrit les achats de produit réalisés dans différentes villes du monde.

### 7.3.1 Principes

#### Données manipulées

Nous étendons les concepts présentés précédemment (client - date - items) en considérant non plus des attributs simples pour décrire les données, mais des ensembles d'attributs. Nous supposons qu'il existe au moins une dimension (e.g. temporelle) dont le domaine est totalement ordonné.

**Définition 7 (Partition des dimensions)** *Pour tout ensemble de transactions  $DB$  défini sur un ensemble de  $n$  dimensions  $D$ , on considère une partition de  $D$  en trois sous-ensembles notés respectivement :*

- $D_R$  pour l'ensemble des dimensions de référence (client dans contexte classique) qui permettent de déterminer si une séquence est fréquente.
- $D_T$  pour l'ensemble des dimensions (date dans contexte classique) permettant d'introduire une relation d'ordre.
- $D_A = \{D_1, \dots, D_m \text{ où } D_i \subset \text{Dom}(D_i)\}$  pour l'ensemble des dimensions d'analyse (produits dans contexte classique) d'où sont extraites les corrélations.

Il en découle que chaque  $n$ -uplet  $c = (d_1, \dots, d_n)$  peut s'écrire sous la forme d'un triplet  $c = (r, a, t)$  où  $r$  (respectivement  $a$  et  $t$ ) sont les restrictions de  $c$  sur  $D_R$  (respectivement  $D_A$  et  $D_T$ ).

**Définition 8 (Bloc)** Etant donnée une base  $DB$ , l'ensemble des  $n$ -uplets qui ont la même restriction  $r$  sur  $D_R$  constitue un bloc.

Chaque bloc  $B$  est identifié par un  $n$ -uplet  $r$ . Nous notons  $B_{DB, D_R}$ , l'ensemble des blocs identifiés sur  $D_R$  constituant la base  $DB$ .

$D$	$B$	$Pl$	$P$
1	1	Allemagne	Bi.
1	1	Allemagne	Ca.
2	1	Allemagne	A.
3	1	Allemagne	Ch.
4	1	Allemagne	S.

**Fig. 7.2:** bloc (1)

$D$	$B$	$Pl$	$P$
1	3	UK	W.
1	3	UK	Ca.
2	3	UK	A.

**Fig. 7.4:** bloc (3)

$D$	$B$	$Pl$	$P$
1	2	France	Co.
2	2	France	V.
2	2	France	Ca.
3	2	France	A.

**Fig. 7.3:** bloc (2)

$D$	$B$	$Pl$	$P$
1	4	LA	Ch.
2	4	LA	S.
3	4	NY	W.
4	4	NY	Co.

**Fig. 7.5:** bloc (4)

**Fig. 7.6:** Partition de  $DB$  (figure 7.1) en fonction de  $D_R = \{B\}$

Cette définition des blocs est nécessaire pour définir le support d'une séquence multidimensionnelle. Son application dans notre base exemple est simple puisque  $|D_R| = 1$ , les différents blocs obtenus sont décrits figure 7.6.

**Définition 9 (Item multidimensionnel)** Un item multidimensionnel  $e = (d_1, \dots, d_m)$  est un  $m$ -uplet défini sur les dimensions d'analyse  $D_A$  tel que  $d_i \in \text{dom}(D_i)$ .

Etant donné  $D_A = \{Pl, P\}$ ,  $(LA, \text{Chocolat})$ ,  $(France, \text{Aspirine})$  et  $(UK, \text{Cacahuètes})$  sont des items multidimensionnels.

D'après la définition précédente, un item ne peut être trouvé que s'il existe une combinaison de valeurs de domaines de  $D_A$  se retrouvant fréquemment dans les données de  $DB$ . Or il peut arriver qu'aucune combinaison ne soit pas fréquente. C'est pour cette raison que nous introduisons une valeur *joker* symbolisée par '\*'. Cette valeur signifie que l'on ne tient pas compte de la valeur sur la dimension d'analyse. On appelle de tels items des items  $\alpha$ -étoilés.

**Définition 10 (Item multidimensionnel  $\alpha$ -étoilé)** Soit  $e_{[d_i/\delta]}$  la substitution dans  $e$  de  $d_i$  par  $\delta$ ,  $e$  est un item  $\alpha$ -étoilé si les conditions suivantes sont vérifiées :

### 7.3. VERS DES MOTIFS SÉQUENTIELS MULTIDIMENSIONNELS : $M^2SP$ , HYPE77

- (i)  $\forall i \in [1, m], d_i \in \text{Dom}(D_i) \cup \{*\}$ ,
- (ii)  $\exists i \in [1, m]$  tel que  $d_i \neq *$ ,
- (iii)  $\forall d_i = *, \nexists \delta \in \text{Dom}(D_i)$  tel que  $e_{[d_i/\delta]}$  est fréquent.

Étant donné  $D_A = \{Pl, P\}, (*, \text{Chocolat}), (\text{France}, *)$  sont des items multidimensionnels  $\alpha$ -étoilés.

**Définition 11 (Itemset multidimensionnel)** *Un itemset multidimensionnel  $i = \{e_1, \dots, e_k\}$  est un ensemble non vide d'items multidimensionnels.*

$\{(*, \text{Vin}), (*, \text{Cacahuètes})\}$  est un itemset multidimensionnel.

Il est important de remarquer que tous les items d'un même itemset sont deux à deux distincts par définition (i.e. un itemset est un ensemble).

**Définition 12 (Séquence multidimensionnelle)** *Une séquence multidimensionnelle  $s = \langle i_1, \dots, i_j \rangle$  est une liste ordonnée par rapport à  $D_t$  et non vide d'itemsets multidimensionnels.*

$\{\{(*, \text{Vin}), (*, \text{Cacahuètes})\}, \{(*, \text{Aspirine})\}\}$  est une séquence multidimensionnelle  $\alpha$ -étoilée.

Calculer le support d'une séquence multidimensionnelle  $\alpha$ -étoilée revient à compter le nombre de blocs définis par les dimensions de référence  $D_R$  qui supportent la séquence. Un bloc supporte une séquence multidimensionnelle  $\alpha$ -étoilée s'il est possible de trouver un ensemble de n-uplets qui la satisfasse. Pour chaque itemset de la séquence, nous devons exhiber une date du domaine de  $D_t$  telle que tous les items multidimensionnels  $\alpha$ -étoilés de l'itemset sont supportés par des n-uplets relatifs à cette date. Tous les itemsets doivent être retrouvés à différentes dates appartenant au domaine de  $D_t$  tels que l'ordre des itemsets respecte la séquentialité.

**Définition 13** *Un bloc  $B$  supporte une séquence  $\alpha$ -étoilée  $\varsigma = \langle i_{s1}, \dots, i_{sl} \rangle$  si  $\forall j \in [1, l], \exists \delta_j \in \text{Dom}(D_t), \forall e = (d_{i_1}, \dots, d_{i_m}) \in i_j, \exists t = (f, r, (x_{i_1}, \dots, x_{i_m}), \delta_j) \in B$  avec  $d_i = x_i$  or  $d_i = *$  et  $\delta_1 < \delta_2 < \dots < \delta_l$ .*

**Définition 14 (Support d'une séquence)** *Soient  $D_R$  l'ensemble des dimensions de référence et  $DB$  l'ensemble des transactions partitionné en un ensemble de blocs  $B_{T, D_R}$ . Le support d'une séquence  $\varsigma$  est :*

$$\text{support}(\varsigma) = \frac{|\{B \in B_{DB, D_R} \text{ t.q. } B \text{ supporte } \varsigma\}|}{|B_{DB, D_R}|}$$

**Exemple 9** *Par rapport à notre base de données exemple  $DB$ , considérons  $D_R = \{B_{id}\}, D_A = \{\text{Lieu}, \text{Produit}\}$  et  $D_T = \{\text{Date}\}$ ,  $\text{support} = 2$ , et  $\varsigma = \{\{(*, \text{cacahuètes})\}, \{(*, \text{aspirine})\}\}$ . Pour que la séquence soit fréquente, au moins deux blocs de la partition de  $DB$  doivent supporter la séquence.*

**1. bloc (1)** (Figure 7.2). *A la date 1, nous avons bien le premier itemset  $\{(*, \text{Cacahuètes})\}$  de  $\varsigma$  grâce au n-uplet  $(1, 1, \text{Allemagne}, \text{Cacahuètes})$ . A une date postérieure (2), le dernier itemset  $\{(*, \text{Aspirine})\}$  est présent. La séquence  $\varsigma$  est supportée par ce bloc.*

**2. bloc (2)** (Figure 7.3). *Nous retrouvons bien le premier itemset de la séquence à la date 2 alors qu'à la date 3 le second itemset est présent. Nous retrouvons bien la séquence  $\varsigma$  dans ce bloc.*

**3. bloc (3)** (Figure 7.4). *Ce bloc supporte également la séquence  $\varsigma$ .*

**4. bloc (4)** (Figure 7.5). Ce bloc ne supporte pas la séquence  $\varsigma$  puisque la dimension *Produit* ne contient aucune instance de *Cacahuètes* et *Aspirine*.

Le support de  $\varsigma$  est donc égal à 3. La séquence est fréquente.

Nous avons posé les définitions fondamentales des motifs séquentiels multidimensionnels. Les algorithmes permettant la mise en œuvre de l'extraction de motifs séquentiels multidimensionnels  $\alpha$ -étoilés ou non sont décrits dans la section 7.3.2.

Il est cependant très difficile d'extraire des connaissances de qualité en fonction du support. Si le support minimal choisi est trop élevé, le nombre de règles découvertes est faible mais si le support est trop bas, le nombre de règles obtenues est très important et rend difficile l'analyse de celles-ci. L'utilisateur est alors confronté au problème suivant : comment baisser le support minimal sans générer la découverte de règles non pertinentes ? Ou comment augmenter le support minimal sans perdre les règles utiles ? Est-il alors nécessaire de faire un compromis entre qualité des connaissances extraites et support ?

L'utilisation des hiérarchies dans l'extraction de connaissances représente un excellent moyen de résoudre ce dilemme. Elle permet de découvrir des règles au sein de plusieurs niveaux de hiérarchies. Ainsi, même si un support élevé est utilisé, les connaissances importantes dont le support est faible dans les données sources peuvent être *incluses* dans des connaissances plus générales qui, elles, seront comptabilisées comme fréquentes.

Dans le contexte dans lequel nous nous situons, nous considérons qu'il existe des relations hiérarchiques sur chaque dimension d'analyse<sup>1</sup>. Nous considérons que ces relations hiérarchiques sont matérialisées sous la forme de *taxonomie*.

### Taxonomies et hiérarchies

Une taxonomie est un arbre orienté dans lequel les arcs sont des relations de type *is-a*. La relation de *généralisation/spécialisation* s'effectue ainsi de la racine vers les feuilles. Chaque dimension d'analyse possède donc une taxonomie qui permet de représenter les relations hiérarchiques entre les éléments de son domaine.

Soit  $T_{DA} = \{T_1, \dots, T_m\}$  l'ensemble des taxonomies associées aux dimensions d'analyse où :

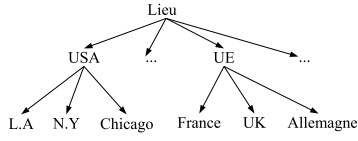
- $T_i$  est la taxonomie représentant les relations hiérarchiques entre les éléments de la dimension d'analyse  $D_i$ .
- $T_i$  est un arbre orienté.
- $\forall$  nœud  $n_i \in T_i$ ,  $label(n_i) \in Dom(D_i)$ .

On note  $\hat{x}$  un ancêtre de  $x$  dans la taxonomie et  $\check{x}$  un de ses descendants. Par exemple,  $Boisson = \widehat{Coca}$  signifie que *Boisson* est un ancêtre de *Coca* dans la relation *Généralisation/Spécialisation*. Plus précisément, *Boisson* est une instance plus générale que *Coca*.

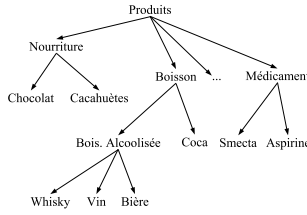
Les deux taxonomies associées à la base exemple (Figure 7.1) décrivant les relations hiérarchiques entre les éléments de la dimension *Produits* (resp. *Lieu*) sont représentées dans la figure 7.8 (resp. Figure 7.7).

Chaque dimension d'analyse  $D_i$  d'une transaction  $b$  de  $DB$  ne peut être instanciée qu'avec une valeur  $d_i$  dont le nœud associé à l'étiquette  $d_i$  dans la taxonomie  $T_i$  est une *feuille*. Plus formellement,  $\forall d_i \in \pi_{D_i}(B), \forall$  nœud  $n_i$  tq  $label(n_i) = d_i \nexists$  nœud  $n'$  tq  $n' =$

<sup>1</sup>Dans le pire des cas, la hiérarchie minimale se représente par un arbre de profondeur 1 où la racine est étiquetée par \* (gestion des valeurs jokers dans  $M^2SP$ ).



**Fig. 7.7:** Taxonomie sur la dimension *Lieu*



**Fig. 7.8:** Taxonomie sur la dimension *Produit*

$\tilde{n}_i$  ( $n_i$  feuille).

Par exemple, la base de transactions  $DB$  ne peut pas contenir la valeur *Boisson* s'il existe des instances plus spécifiques dans la taxonomie comme *Coca* ou *vin*.

### Item, Itemset, Séquence multidimensionnels h-généralisés

Dans cette section, nous étendons les définitions précédentes et définissons les concepts fondamentaux d'items, d'itemsets et de séquences multidimensionnels h-généralisés.

**Définition 15 (Item multidimensionnel h-généralisé)** *Un item multidimensionnel h-généralisé*  $e = (d_1, \dots, d_m)$  *est un*  $m$ -*uplet défini sur les dimensions d'analyse*  $D_A$  *telles que*  $d_i \in \{\text{label}(T_i)\}$ .

Contrairement aux transactions de  $DB$ , un item multidimensionnel h-généralisé peut être défini avec n'importe quelle valeur  $d_i$  dont le nœud associé dans la taxonomie n'est pas nécessairement une feuille.

$(USA, Boisson)$  et  $(France, Boisson Alcool.)$  sont, par exemple, des items multidimensionnels h-généralisés.

Comme les items multidimensionnels h-généralisés sont instanciés sur différents niveaux de hiérarchies, il est possible que deux items soient comparables, c'est-à-dire qu'un item soit plus *spécifique* ou *général* qu'un autre.

Par abus de langage et afin de ne pas alourdir les notations, nous utilisons directement la notion d'*ancêtre* sur l'item et la transaction sans nous situer dans la taxonomie correspondante.

**Définition 16 (Inclusion hiérarchique d'items)** *Soient deux items multidimensionnels h-généralisés*  $e = (d_1, \dots, d_m)$  *et*  $e' = (d'_1, \dots, d'_m)$ , *on dit que :*

- $e$  *est plus général que*  $e'$  ( $e >_h e'$ ) *si*  $\forall d_i, d_i = \hat{d}'_i$  *ou*  $d_i = d'_i$
- $e$  *est plus spécifique que*  $e'$  ( $e <_h e'$ ) *si*  $\forall d_i, d_i = \check{d}'_i$  *ou*  $d_i = d'_i$
- $e$  *et*  $e'$  *sont incomparables s'il n'existe pas de relation entre eux* ( $e \not>_h e'$  *et*  $e' \not<_h e$ )

On a par exemple les relations hiérarchiques suivantes entre items :

- $(USA, Boisson) >_h (USA, Coca)$ .
- $(France, Vin) <_h (UE, Boisson Alcool.)$ .
- $(France, Vin)$  *et*  $(USA, Coca)$  *sont incomparables.*

**Définition 17** *Une transaction*  $b$  *supporte un item*  $e$  *si*  $\Pi_{D_A}(b) <_h e$ .



La transaction  $(1, 1, France, Vin)$  supporte l'item  $(UE, Boisson Alcool.)$ .

**Définition 18 (Itemset multidimensionnel h-généralisé)** *Un itemset multidimensionnel h-généralisé  $i = \{e_1, \dots, e_k\}$  est un ensemble non vide d'items multidimensionnels h-généralisés où tous les items sont incomparables entre eux.*

Deux items comparables ne peuvent pas être présents dans le même itemset. Nous adoptons un point de vue ensembliste et préférons ainsi représenter l'information la plus précise possible au sein d'un itemset.

Ainsi,  $\{(France, Vin), (USA, Coca)\}$  est un itemset multidimensionnel h-généralisé alors que  $\{(France, Vin), (UE, Boisson Alcool.)\}$  n'est pas un itemset multidimensionnel h-généralisé car  $(France, vin) <_h (UE, Boisson Alcool.)$ .

La notion de séquence multidimensionnelle h-généralisée découle de la notion d'itemset.

**Définition 19 (Séquence multidimensionnelle h-généralisée)** *Une séquence multidimensionnelle h-généralisée  $s = \langle i_1, \dots, i_j \rangle$  est une liste ordonnée non vide d'itemsets multidimensionnels h-généralisés.*

$\langle \{(UE, Boisson Alcool.), (USA, Coca)\}, \{(UE, Aspirine)\} \rangle$  est une séquence multidimensionnelle h-généralisée.

**Définition 20 (Inclusion de séquences)** *Une séquence multidimensionnelle h-généralisée  $\varsigma = \langle a_1, \dots, a_l \rangle$  est une sous-séquence de la séquence  $\varsigma' = \langle b_1, \dots, b_{l'} \rangle$  s'il existe des entiers  $1 \leq j_1 \leq j_2 \leq \dots \leq j_l \leq l'$  tel que  $a_1 \subseteq b_{j_1}, a_2 \subseteq b_{j_2}, \dots, a_l \subseteq b_{j_l}$ .*

**Remarque 2** *L'inclusion des itemsets multidimensionnels doit respecter l'inclusion hiérarchique des items multidimensionnels h-généralisés.*

Cette remarque est importante. En effet, l'inclusion hiérarchique joue un rôle important dans l'inclusion de séquences h-généralisées.

- La séquence  $\langle \{(France, Vin)\}, \{(Allemagne, Bière)\} \rangle$  est une sous-séquence de la séquence  $\langle \{(France, Vin), (USA, Coca)\}, \{(Allemagne, Bière)\} \rangle$ .
- La séquence  $\langle \{(France, Vin)\}, \{(Allemagne, Bière)\} \rangle$  est une sous-séquence de la séquence  $\langle \{(France, Boisson Alcool.), (USA, Boisson)\}, \{(UE, Boisson Alcool.)\} \rangle$ .
- La séquence  $\langle \{(UE, Vin)\}, \{(Allemagne, Bière)\} \rangle$  n'est pas une sous-séquence de la séquence  $\langle \{(France, Vin), (USA, Coca)\}, \{(Allemagne, Bière)\} \rangle$  car  $(UE, Vin) \not\subseteq_h (France, Vin)$ , l'inclusion hiérarchique n'étant pas respectée.

### Support d'une séquence multidimensionnelle h-généralisée

Le calcul du support d'une séquence se définit comme précédemment, il faut compter le nombre de blocs qui supportent la séquence.

**Définition 21 (Support d'une séquence)** *Soient  $D_R$  l'ensemble des dimensions de référence et  $DB$  l'ensemble des transactions partitionné en un ensemble de blocs  $B_{T, D_R}$ . Le support d'une séquence  $\varsigma$  est :*

$$support(\varsigma) = \frac{|\{B \in B_{DB, D_R} \text{ t.q. } B \text{ supporte } \varsigma\}|}{|B_{DB, D_R}|}$$

**Exemple 10** Par rapport à notre base de données exemple  $DB$ , considérons  $D_R = \{B_{id}\}$ ,  $D_A = \{\text{Lieu}, \text{Produit}\}$  et  $D_T = \{\text{Date}\}$ ,  $\text{support} = 2$ , et  $\varsigma = \{(UE, \text{Boisson Alcool.}), (UE, \text{Cacahuètes})\} \{(UE, \text{Aspirine})\}$ . Pour que la séquence soit fréquente, au moins deux blocs de la partition de  $DB$  doivent supporter la séquence.

**1. bloc (1)** (Figure 7.2). Si l'on se réfère aux taxonomies relatives aux dimensions d'analyse (Figures 7.7 et 7.8), Allemagne est une instance plus spécifique de UE et bière est une instance de Boisson Alcool.. Ainsi à la date 1, nous avons bien le premier itemset  $\{(UE, B.A), (UE, \text{Cacahuètes})\}$  de  $\varsigma$ . A une date postérieure (2), le dernier itemset  $\{(UE, \text{Aspirine})\}$  est présent. La séquence  $\varsigma$  est supportée par ce bloc.

**2. bloc (2)** (Figure 7.3). France est une instance de UE et Vin est une instance de Boisson Alcool.. Nous retrouvons bien la séquence  $\varsigma$  dans ce bloc.

**3. bloc (3)** (Figure 7.4). UK est une instance de UE et whisky est une instance de Boisson Alcool.. Ce bloc supporte la séquence  $\varsigma$ .

**4. bloc (4)** (Figure 7.5). Ce bloc ne supporte pas la séquence  $\varsigma$  puisque la dimension Lieu ne contient aucune instance de UE.

Le support de  $\varsigma$  est donc égal à 3. La séquence est fréquente.

### 7.3.2 Les algorithmes M<sup>2</sup>SP et HYPE

Dans cette section, nous décrivons les algorithmes relatifs à M<sup>2</sup>SP et HYPE. Ces deux approches ont un comportement général similaire.

M<sup>2</sup>SP et HYPE se comportent de la même façon. Les algorithmes 14 (M<sup>2</sup>SP) et 15 (HYPE) décrivent le comportement général de ces approches. La principale différence réside dans l'extraction des items fréquents. M<sup>2</sup>SP extrait des items multidimensionnels  $\alpha$ -étoilés alors que HYPE extrait des items multidimensionnels h-généralisés. Ces deux méthodes se basent sur le paradigme APriori, la génération des séquences candidates est donc similaire.

**Données :**  $DB, D_R, D_A, \sigma_{min}$   
**Résultat :** Motifs séquentiels  $\alpha$ -étoilé

**début**

	Extraction des items $\alpha$ -étoilés maximalement spécifiques ;
	$k \leftarrow 1$ ;
	<b>tant que</b> $L_k \neq \emptyset$ <b>faire</b>
	générer les $k + 1$ séquences $\alpha$ -étoilés candidates;
	extraire l'ensemble $L_{k+1}$ des $k + 1$ séquences $\alpha$ -étoilés fréquentes;
	$k \leftarrow k + 1$ ;
	<b>retourner</b> $\bigcup_{i=0}^k L_i$ ;

**fin**

**Algorithme 14:** M<sup>2</sup>SP

Quelque soit l'approche choisie (M<sup>2</sup>SP ou HYPE), le processus d'extraction de motifs séquentiels multidimensionnels se divise en deux phases :

- La génération des items candidats
- La génération des séquences candidates.

Nous décrivons précisément ces deux phases ainsi que le calcul du support d'une séquence multidimensionnelle.

**Données :**  $DB, D_R, D_A, T_{D_A}, \sigma_{min}$   
**Résultat :** Motifs séquentiels h-généralisés  
**début**  
  *Extraction des items h-généralisés maximalement spécifiques ;*  
   $k \leftarrow 1$ ;  
  **tant que**  $L_k \neq \emptyset$  **faire**  
    *générer les  $k + 1$  séquences h-généralisées candidates;*  
    *extraire l'ensemble  $L_{k+1}$  des  $k + 1$  séquences h-généralisées fréquentes;*  
     $k \leftarrow k + 1$ ;  
  **retourner**  $\bigcup_{i=0}^k L_i$ ;  
**fin**

Algorithme 15: HYPE

### Génération des items candidats

Les items multidimensionnels  $\alpha$ -étoilés (*resp.* h-généralisés) fréquents sont la base de l'extraction de motifs séquentiels multidimensionnels  $\alpha$ -étoilés (*resp.* h-généralisés). Ils représentent les fréquents de taille 1 puisqu'ils correspondent à des séquences composées d'un seul item contenu dans un seul itemset.

Il est donc nécessaire de définir une méthode qui limite à la fois le nombre d'items candidats générés et le nombre de passes sur la base. Afin de limiter le nombre d'items candidats aux seuls items dont la probabilité d'être fréquents est non nulle, nous adoptons une méthode de génération par niveau.

Tout d'abord, nous considérons les items multidimensionnels pour lesquels une seule dimension d'analyse est spécifiée, les autres dimensions n'étant pas encore spécifiées. Les items multidimensionnels fréquents sont alors *joint*s entre eux pour obtenir l'ensemble des items candidats pour lesquels deux dimensions d'analyse sont spécifiées. Seuls les fréquents sont retenus. Cette procédure est réitérée tant que l'ensemble des items candidats est non vide pour l'extraction des items multidimensionnels  $\alpha$ -étoilés (au plus  $m - 1$  fois) alors qu'elle est réitérée exactement  $m - 1$  fois pour l'obtention des items multidimensionnels h-généralisés où les  $m$  dimensions d'analyse sont instanciées. Parmi ces items, seuls les plus spécifiques seront retenus.

L'opération de *jointure* entre deux items fréquents suppose que les items soient  $\bowtie$ -compatibles, c'est-à-dire qu'ils partagent un nombre suffisant de valeurs de dimensions d'analyse (cf. définition 22). Pour être  $\bowtie$ -compatibles, deux items multidimensionnels définis sur  $n$  dimensions doivent partager  $n - 2$  valeurs de dimension. Par exemple,  $(a, *, c)$  et  $(*, b, c)$  sont deux items définis sur 3 dimensions d'analyse et partagent  $3 - 2 = 1$  valeur sur la dimension  $C$ . Ils sont donc  $\bowtie$ -compatibles. En revanche, les items  $(a_1, b_1, *)$  et  $(a_2, b_2, *)$  ne sont pas  $\bowtie$ -compatibles.

**Définition 22 ( $\bowtie$ -Compatibilité)** Soient deux items multidimensionnels  $e_1 = (d_1, \dots, d_n)$  et  $e_2 = (d'_1, \dots, d'_n)$  où  $d_i$  et  $d'_i \in \text{dom}(D_i) \cup \{*\}$ . On dit que  $e_1$  et  $e_2$  sont  $\bowtie$ -compatibles si

- $e_1$  et  $e_2$  sont distincts
- $\exists \Delta = \{D_{i_1}, \dots, D_{i_{n-2}}\} \subset \{D_1, \dots, D_n\}$  t.q.  $d_{i_1} = d'_{i_1} \neq *$  et  $d_{i_2} = d'_{i_2} \neq * \dots$  et  $d_{i_{n-2}} = d'_{i_{n-2}} \neq *$
- Pour  $\{D_{i_{n-1}}, D_{i_n}\} = \{D_1, \dots, D_n\} \setminus \Delta$ , on a  $d_{i_{n-1}} = *$  et  $d'_{i_{n-1}} \neq *$  et  $d_{i_n} \neq *$  et  $d'_{i_n} = *$

L'opération de jointure mise en œuvre pour générer les items multidimensionnels  $\alpha$ -étoilés ou h-généralisés potentiellement fréquents se définit de la façon suivante :

**Définition 23 (Jointure)** Soient 2 items multidimensionnels  $\bowtie$ -compatibles  $e_1 = (d_1, \dots, d_n)$  et  $e_2 = (d'_1, \dots, d'_n)$ . On définit  $e_1 \bowtie e_2 = (v_1, \dots, v_n)$  avec :

- $v_i = d_i$  si  $d_i = d'_i$
- $v_i = d_i$  si  $d'_i = *$
- $v_i = d'_i$  si  $d_i = *$

La génération des items multidimensionnels s'effectue donc à l'aide d'un treillis. Néanmoins le nombre de candidats générés reste important, on peut imaginer utiliser la recherche d'items multidimensionnels dérivables pour limiter le calcul du support à un nombre réduit d'items (recherche équivalente à la recherche d'itemsets dérivables).

### Génération des séquences fréquentes

Les items multidimensionnels  $\alpha$ -étoilés (*resp.* h-généralisés) sont donc des séquences multidimensionnelles  $\alpha$ -étoilées (*resp.* h-généralisées) de taille 1. Ils sont donc des 1-fréquents.

Pour extraire les séquences fréquentes, nous adoptons la philosophie *Générer/Elaguer*. En effet, nous conservons la propriété d'antimonotonie du support dans le contexte multidimensionnel (Tout sous-ensemble d'un ensemble fréquent est fréquent, tout sur ensemble d'un ensemble non fréquent est non fréquent).

Une fois les 1-fréquents extraits (items multidimensionnels  $\alpha$ -étoilés ou h-généralisés les plus spécifiques), les  $k + 1$ -candidats ( $k \geq 1$ ) sont générés et testés afin de savoir s'ils sont fréquents. Cette opération est itérée tant que des  $k + 1$ -candidats fréquents sont extraits.

Pour stocker les séquences candidates, nous utilisons une structure d'*arbre préfixé* ([MCP98]) afin d'éviter toute redondance.

### Calcul du support d'une séquence

Les opérations de calculs de support de séquences multidimensionnelles sont sensiblement identiques pour M<sup>2</sup>SP et HYPE. L'unique différence dans HYPE est le parcours de la taxonomie adéquate afin d'utiliser les relations *ancêtres/descendants*.

Les dimensions de référence permettent d'identifier tous les blocs de l'ensemble des données susceptibles de supporter une séquence  $\varsigma$ . L'énumération de tous les blocs définis par les dimensions de référence  $D_R$  est indispensable pour calculer le support d'une séquence et définir ainsi si la séquence est fréquente ou non.

L'algorithme 16 vérifie pour chaque bloc de  $DB$  si la séquence est supportée ou non. Si la séquence est supportée, alors le support est incrémenté. L'algorithme retourne ensuite le ratio des blocs supportant  $\varsigma$ .

L'algorithme 17 permet de vérifier si le bloc  $B$  supporte la séquence  $\varsigma$ . Pour cela, cet algorithme cherche à instancier la séquence itemset par itemset en conjuguant *récurtivité* et *ancrage*. L'ancrage correspond à une n-uplet du bloc  $B$  à partir duquel la séquence pourra être instanciée. Cet n-uplet correspond donc à une date à laquelle le premier item du premier itemset de la séquence est trouvé. À partir de cet n-uplet, seuls les n-uplets pertinents sont retenus, c'est-à-dire ceux qui partagent la même date. On ne retient donc que les n-uplets partageant la même date. Si le sous-bloc résultant de l'ancrage supporte l'itemset alors on appelle la fonction sur les autres itemsets de  $\varsigma$ . Cet appel est effectué en réduisant l'espace de recherche aux seuls n-uplets dont la date est supérieure à la date de l'ancrage précédent, puisque l'on passe à l'itemset suivant, donc à une date ultérieure. Si l'ancrage échoue, on continue la recherche du premier itemset en tentant d'autres ancrages. L'appel récursif s'arrête dès que la séquence placée en paramètre d'entrée est vide. Une telle propriété signifie en effet que tous les itemsets de la séquence ont été trouvés. On retourne donc la valeur *vrai*. La valeur

*faux* est retournée si aucun ancrage n'a réussi et si tout le bloc a été parcouru sans succès.

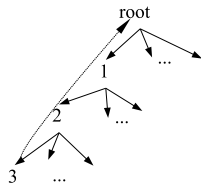
**Fonction compterSupport** Données :  $\varsigma, DB, D_R$   
**Résultat** : le support de la séquence  $\varsigma$   
**début**  
     Entier *support*  $\leftarrow 0$ ;  
     Booleen *seqSupportée*;  
      $\mathcal{B}_{DB, D_R} \leftarrow \{\text{bloc de } DB \text{ identifiés sur } D_R\}$ ;  
     **pour chaque**  $B \in \mathcal{B}_{DB, D_R}$  **faire**  
         *seqSupportée*  $\leftarrow \text{supportBloc}(\varsigma, B)$  ;  
         **si** *seqSupportée* **alors**  
             *support*  $\leftarrow \text{support} + 1$ ;  
     **retourner**  $\left( \frac{\text{support}}{|\mathcal{B}_{DB, D_R}|} \right)$   
**fin**

**Algorithme 16:** Calcul du support d'une séquence (compterSupport)

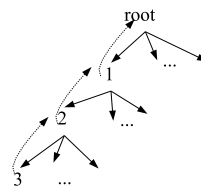
### Pourquoi les hiérarchies permettent une gestion plus fine de la valeur joker ?

La prise en compte des hiérarchies peut être vue comme un moyen plus fin de gérer les valeurs jokers. En effet, dans l'approche  $M^2SP$ , la racine d'une taxonomie représente la valeur joker '\*' sur la dimension associée. Ainsi, si aucune instantiation n'est possible, aucune étiquette feuille ne peut donc convenir, alors l'instanciation par la valeur joker '\*' permet de remonter directement à la racine de la taxonomie (figure 7.9).

La prise en compte des hiérarchies, permet d'extraire des connaissances plus fines. En effet, les taxonomies proposent plusieurs alternatives par rapport à l'approche  $M^2SP$  quand il est impossible d'instancier une dimension. En effet, on ne passe pas directement de la feuille à la racine, on essaie d'instancier par l'ancêtre le plus spécifique de la feuille (figure 7.10).



**Fig. 7.9:** Gestion de la valeur joker (\*)



**Fig. 7.10:** Gestion des hiérarchies

**Exemple 11 (Comparaison avec  $M^2SP$ )** Pour un support fixé à 2, la prise en compte des hiérarchies permet d'extraire des connaissances qui ne peuvent pas être extraites par  $M^2SP$ .

$M^2SP$

- $(*, \text{Chocolat}), (*, \text{Cacahuètes}), (*, \text{Smecta}), (*, \text{Coca}), (*, \text{Aspirine}), (*, \text{Whisky})$
- $\{\{(*, \text{Chocolat})\}\{(*, \text{Smecta})\}\}, \{\{(*, \text{Cacahuètes})\}\{(*, \text{Aspirine})\}\}$

**Fonction supportBloc****Données** :  $\varsigma, B$ **Résultat** : Booléen**début**

```

    //initialisation-
    booleen ItemSetTrouvé ← faux
    sequence ←  $\varsigma$ 
    itemset ← sequence.first()
    item ← itemset.first()
    //condition d'arrêt de la recursivité -
    si  $\varsigma = \emptyset$  alors
        ⊥ retourner (vrai)
    //parcours du bloc -
    tant que tuple ← B.next ≠ ∅ faire
        si supporte(tuple, item) alors
            itemSuivant ← itemset.second()
            si itemSuivant = ∅ alors
                ⊥ itemsetTrouvé ← vrai
            //Recherche de tous les items de l'itemset-
        sinon
            // On ancre par rapport à l'item (date)-
            B' ←  $\sigma_{date=cell.date}(B)$ 
            tant que tuple' ← B'.next() ≠ ∅ ∧ itemsetTrouvé = faux
                faire
                    si supporte(cell', itemSuivant) alors
                        itemSuivant ← itemset.next()
                        si itemSuivant = ∅ alors
                            ⊥ itemsetTrouvé ← vrai
            si itemsetTrouvé = vrai alors
                // recherche des autres itemsets-
                ⊥ retourner (supportBloc(sequence.tail(),  $\sigma_{date>tuple.date}(B)$ ))
            sinon
                itemset ← sequence.first()
                //réduction de l'espace de recherche-
                C ←  $\sigma_{date>cell.date}(B)$ 
    // -  $\varsigma$  non supportée -
    retourner (faux)
fin

```

**Algorithme 17:** supportBloc : (Vérifie si une séquence est supportée par un bloc donné)

**Prise en compte des hiérarchies**

- $(Lieu, Chocolat), (UE, Cacahuètes), (Lieu, Smecta), (Lieu, Coca), (UE, Aspirine), (Lieu, Whisky), (UE, Boisson Alcool.),$
- $\{(Lieu, Chocolat)\}(Lieu, Smecta)\}$
- $\{(UE, Cacahuètes)\}(UE, Aspirine)\}$
- $\{(UE, Boisson Alcool.)\}(UE, Aspirine)\}$
- $\{(UE, Boisson Alcool.), (UE, Cacahuètes)\}(UE, Boisson Alcool.)\}$

*Le prise en compte des hiérarchies permet ainsi d'extraire des séquences plus complètes que l'approche  $M^2SP$ .*

**7.4 Discussion**

Dans ce chapitre, nous avons défini une nouvelle forme de motifs séquentiels, nommés motifs séquentiels multidimensionnels. Contrairement aux propositions présentes dans la littérature, nous intégrons au sein même de la séquence plusieurs dimensions d'analyse, ce qui permet la construction de motifs de la forme :

$$\langle \{(surf, NY), (housse, NY)\}, \{(combi, LA)\} \rangle$$

indiquant que les personnes ayant acheté leur planche de surf et la housse à New York ont acheté plus tard leur combinaison à Los Angeles. Nous avons également introduit les motifs séquentiels étoilés permettant la prise en compte de valeurs jokers sur les dimensions d'analyse et sur la mesure. Les algorithmes associés ont été validés par des expérimentations sur des jeux de données synthétiques et plus récemment, dans le cadre d'un projet de transfert de technologie, sur des jeux de données réelles. Toutes ces expérimentations ont montré l'intérêt de l'introduction des valeurs jokers sur les dimensions d'analyse et sur la mesure pour traiter le cas où aucun fréquent n'est trouvé.

Notre proposition est présentée dans le cadre spécifique OLAP. L'extraction de tels motifs est cependant réalisable sur toute base "classique" multi-attributs.

Revenons à présent sur les contraintes de temps présentées précédemment. Le travail que nous avons réalisé offre de nombreuses perspectives en ce qui concerne la gestion des contraintes de temps pour la définition de motifs séquentiels généralisés, ainsi que sur l'intégration d'approximation de la mesure. Cette dernière approche permettrait en effet de ne pas perdre totalement la connaissance de la valeur de mesure, ce qui est le cas actuellement, tout en conservant une chance de trouver des motifs fréquents face au grand nombre de valeurs possibles dans les bases de données issues du monde réel. Ainsi, nous pourrions construire des règles de la forme : les personnes ayant acheté un lecteur de DVD à la FNAC achètent par la suite environ trois DVD dans un supermarché. De plus, outre ses applications immédiates au contexte du panier de la ménagère, cette proposition peut être utilisée au sein de bases de données multidimensionnelles MOLAP afin de rechercher des enchaînements fréquents de blocs de cellules au sein d'une représentation cubique des données. Des recherches similaires ont déjà été réalisées dans le cadre des règles d'association, il s'agirait de les étendre aux motifs séquentiels.

Dans ce chapitre, nous avons également défini les motifs séquentiels multidimensionnels  $\alpha$ -étoilés qui sont étendus aux motifs séquentiels multidimensionnels  $h$ -généralisés. Ceci permet l'extraction de séquences multidimensionnelles définies sur plusieurs niveaux de hiérarchies. L'intérêt de l'extraction de motifs multidimensionnels est accru

avec la prise en compte des hiérarchies. Elles montrent ainsi la capacité de *HYPE* à subsumer les connaissances ainsi que sa robustesse d'extraction face à la diversité des données (densité, spécialisation, etc.). Bien entendu, d'autres propositions peuvent être effectuées pour la gestion des hiérarchies. Nous pouvons imaginer une gestion modulaire des hiérarchies où certaines dimensions n'auraient pas le même comportement que les autres afin de s'adapter aux besoins du décideur (interdiction de dépasser le niveau de hiérarchie  $\lambda$  sur la dimension  $\varepsilon$ , etc.).





## Chapitre 8

# Conclusions et Perspectives

Dans les chapitres précédents, nous avons vu pour différents travaux menés quelles étaient les conclusions à tirer. Dans ce chapitre, nous souhaitons préciser les conditions dans lesquelles nous avons été amenés à effectuer ces recherches. Pour cela, dans la section 8.1, nous effectuons une synthèse de nos travaux de recherche ces dernières années en nous focalisant plus particulièrement sur les travaux liés à la fouille de données. Nous présentons les différentes publications obtenus ainsi que les encadrements effectués respectivement dans les sections 8.2 et 8.3. Nous précisons dans la section 8.4 comment les travaux que nous avons effectués ont été transférés auprès de différentes entreprises. Enfin dans la section 8.5 nous précisons les nombreuses perspectives que nous souhaitons traiter dans les prochaines années.

### 8.1 Un bref historique

#### 8.1.1 Synthèse

Initiée en 1990, au sein de l'équipe Bases de Données au laboratoire Informatique, Signaux et Systèmes (I3S) de l'Université de Nice-Sophia Antipolis, mon activité de recherche s'est poursuivie au Laboratoire Informatique de Marseille (LIM) à partir de 1994. Depuis Septembre 1995, elle a lieu au Laboratoire d'Informatique, de Micro-Electronique et de Robotique de Montpellier (LIRMM) dans l'équipe Bases de Données - Systèmes d'Informations.

Après une participation à la définition de la partie structurelle du modèle IFO<sub>2</sub>, extension du modèle IFO proposé par S. Abiteboul et R. Hull, mon travail a consisté à définir la partie comportementale d'IFO<sub>2</sub>. Ce travail s'est intégré dans le cadre d'un projet EERP (External European Research Project) avec la société Digital Europe. Enfin, la dérivation des spécifications événementielles a été définie pour réaliser leur implantation de manière automatique et ainsi optimiser le travail du développeur d'applications. Le travail sur cette composante de dérivation a été poursuivi dans le cadre d'un projet d'ASP (Action de Soutien Programmée) du GDR Bases Données : "Modélisation du comportement et contrôle de l'évolution d'une application persistante" de septembre 1994 à septembre 1996. Depuis 1996, mes travaux ont porté sur la définition de vérifications comportementales, opérant en amont de la génération de code en utilisant le modèle conceptuel IFO<sub>2</sub>. J'ai également participé à un projet de recherche traitant, de façon complémentaire, les aspects d'évolution de schémas dans les Bases de Données Orientées Objet. Basée sur l'utilisation du mécanisme de vue, cette approche permet de prendre en compte les évolutions désirées par l'utilisateur sans entraîner une ré-organisation coûteuse des données.

Depuis 1998, je me suis intéressée aux problèmes liés à l'extraction de connaissances (data mining). Cette activité a débuté par le co-encadrement du DEA puis de la thèse de Florent Massegli (Bourse ministérielle) au sein du Laboratoire LIRMM sur la prise en compte de la recherche de motifs séquentiels. Elle s'est poursuivie sur l'aspect incrémental des approches de recherches de règles d'association ou de motifs séquentiels dans de grandes bases de données. Ces travaux ont fait l'objet du co-encadrement de la thèse de Pierre-Alain Laur (Bourse ministérielle).

Depuis septembre 2001, j'ai défini un projet de recherche en collaboration avec l'équipe TAL (Traitement algorithmique du langage naturel) du LIRMM sur le thème de la fouille de texte (Text Mining). Ces recherches ont fait l'objet du co-encadrement de la thèse de Simon Jaillet (Bourse ministérielle) au LIRMM dont le travail consistait à définir une nouvelle approche d'extraction de connaissances dans de grandes bases de documents basée sur les vecteurs conceptuels.

Depuis septembre 2002, dans la continuité des travaux menés lors de la thèse de Pierre-Alain Laur sur les données semi-structurées, je me suis investie dans le domaine de la médiation de bases de données à large échelle du projet Ingénierie des Données et des Connaissances. Plus particulièrement, dans le cadre du co-encadrement de la thèse de Federico del Razo Lopez (Bourse SFERE - Programme mexicain), mes travaux de recherche portent sur les aspects intégration de schémas basés sur des méthodes de recherche de sous-structures fréquentes.

Depuis septembre 2003, avec l'intégration d'Anne Laurent, j'ai initié des recherches sur les méthodes approximatives et le traitement des données multidimensionnelles dans le contexte des motifs séquentiels. Les travaux sur la fouille de données approximative basée sur la théorie des sous-ensembles flous sont menés dans le cadre de l'encadrement du DEA et de la thèse de Céline Fiot (Bourse BDI co-financée CNRS-Région). Ceux associés à la recherche de motifs au sein de données multidimensionnelles correspondent au DEA et à la thèse de Marc Plantevit (Bourse ministérielle).

Actuellement, je suis co-responsable du projet TATOO (ExTraction de connAissances dans les bases de données : moTifs séquentiels et OntoLOgies) avec Danièle Héryn et j'assure l'animation du groupe Fouille de données au sein du LIRMM composé de deux permanents (Anne Laurent et moi-même), d'un associé (Pascal Poncelet) et de 4 doctorants.

### 8.1.2 Plus précisément, sur la fouille de données

Concernant les aspects Fouille de données, j'ai mené des travaux sur la recherche de motifs séquentiels et sur la prise en compte des données semi-structurées. De plus, avec l'intégration d'Anne Laurent en 2003, nous avons initié des recherches sur les méthodes approximatives et le traitement des données multidimensionnelles. Ces travaux se déclinent, selon les doctorants, de la façon suivante :

Dans le cadre de la thèse de Florent Massegli, nous avons proposé une approche originale et incrémentale de recherche de motifs séquentiels, une prise en compte des contraintes temporelles lors du processus de fouille ainsi qu'une architecture autorisant une extraction de connaissance en temps réel. Un nouvel algorithme, appelé ISE, a été développé pour permettre d'optimiser la recherche de connaissances en ne calculant que le minimum d'information, i.e. les informations nécessaires pour que la connaissance extraite soit représentative de la nouvelle base de données. Les évaluations ont montré qu'avec ISE, dans certains cas, la recherche de motifs séquentiels pouvait être nettement optimisée en considérant les données d'origine comme étant décomposées en une base et son incrément. (Ouvrage Encyclopédie 2005, ACM Sigweb 1999, Congrès RIDE'02, WISE'01, PKDD'00, EGC'02, BDA'01, BDA'00, BDA'99)

Ces travaux se poursuivent :

1. dans une collaboration avec Florent Masseglia actuellement chercheur à l'Inria Nice - Sophia Antipolis. Ces travaux étendent les propositions précédentes par la prise en compte de contraintes temporelles, la recherche de motifs dans un contexte distribué, l'extraction de périodes dans lesquelles les séquences apparaissent fréquemment (Ouvrage Encyclopédie 2006, Revue DMKD 2007, KAIS 2003, DKE 2003, ISI 2006, congrès AINA'06, TIME'04, EGC'06, Ateliers TDM'05)
2. par la définition d'une nouvelle approche, nommée SPEED, permettant de chercher les motifs séquentiels dans les flots de données dans le cadre du DEA de Chedy Raissi. (Congrès IS'06, BDA'05)

Dans le cadre de la thèse de Pierre Alain Laur, nous nous sommes intéressés à la prise en charge des données semi-structurées et deux algorithmes (PSP<sub>TREE</sub>) et (PSP<sub>TREEGENERALISE</sub>) ont été définis. Le premier correspond à un algorithme basé sur une structure préfixée qui offre la possibilité de rechercher des structures typiques en conservant la topologie des structures. Pour le second, différentes contraintes ont été relâchées notamment sur l'utilisation de niveaux. L'approche globale proposée, appelée AUSMS-Web, permet l'analyse de structures mais est également adaptée à la prise en compte du comportement des usagers du web. De plus nous avons étendu la proposition en intégrant une composante incrémentale mais cette fois-ci basée sur la notion de bordure négative et avons proposé une nouvelle méthode d'analyse de tendances des usagers originale. (Revue ISI 2003, Congrès IICAF'03, DEXA'03, AIMSA'00)

Ces travaux se poursuivent actuellement avec la thèse de Federico Del Razo Lopez dans le cadre de la médiation à large échelle. Nous avons défini un nouvel algorithme RSF de recherche de structure arborescente basée sur une représentation optimisée des arbres. Cette représentation offre de très nombreuses propriétés permettant d'optimiser l'ensemble des étapes de la fouille de données : génération de candidats, élagage et validation des fréquents. De plus nous nous intéressons à la définition de différents types d'inclusion (induite, incrustée, floue) afin d'affiner la recherche. L'objectif final est d'utiliser les sous-structures obtenues afin de proposer une construction automatique de schéma médiateur. Les expérimentations réalisées sont très prometteuses. (Ouvrage Semantic Web 2006, Revue RNTI 2005, Congrès EUSFLAT'05, EGC'06, IDEAS'04, Ateliers EGC'05)

La thèse de Simon Jaillet s'est intéressée au traitement de données de type textuel. Nous avons défini un modèle de référence pour les catégoriseurs : le modèle de catégorisation textuelle général (MCT). À partir du MCT, nous avons évalué différentes méthodes de représentation de documents (vecteurs conceptuels et/ou statistiques) ainsi que différentes méthodes de classification. Une nouvelle approche de catégorisation basée sur les motifs séquentiels a été définie et a donné lieu à l'algorithme SPAC. Il permet une classification supervisée de grosses bases de documents à l'aide de règles de catégorisation basée sur des motifs séquentiels extraits. Cette approche est réellement efficace pour des jeux de données où les classifieurs classiques sont moins performants. (Revue IDA 2006, RNTI 2005, Congrès IPMU'04, ICCI'03, BDA'04, TALN'03, INFORSID'03, Ateliers TDM'04)

Dans le cadre de la thèse de Céline Fiot, nous nous intéressons à l'intégration d'une méthode approximative lors de la recherche de motifs séquentiels. Trois algorithmes ont été proposés (SPEEDYFUZZY, MINIFUZZY et TOTALLYFUZZY qui proposent différents niveaux d'approximation selon les souhaits de l'utilisateur final. Le challenge ici est de fournir des algorithmes passant à l'échelle tout en conservant de très bonnes propriétés de flexibilité face au traitement de données numériques. A terme, ces travaux nous permettront de gérer les données manquantes (i) en vue de leur complétion à l'aide des motifs séquentiels flous obtenus ou (ii) lors de la génération des motifs séquentiels. De nombreuses applications sont offertes grâce à ces travaux liées notamment au traitement de données numériques historisées (capteurs). (Congrès FUZZ-IEEE 06,

FLINS'06, EGC'06, EGC'05, LFA'04)

Dans le cadre de la thèse de Marc Plantevit, nous nous préoccupons de la recherche de motifs au sein de données multidimensionnelles. Une première proposition,  $M^2SP$ , a été réalisée. Il s'agit d'une généralisation des travaux existants permettant d'extraire des motifs dans lesquels plusieurs attributs apparaissent. Nous étudions de plus l'utilisation de caractères jokers afin de ne pas pénaliser la recherche quand certains attributs sont très disparates. Nos solutions sont de plus envisagées par rapport au traitement des hiérarchies, ce qui est très novateur par rapport à la littérature. (Congrès DOLAP'06, EDA'06, PKDD'05, BDA'05)

Ces différents travaux ont également donné lieu à des collaborations internationales (Italie, Malaisie, Pakistan, Indonésie, USA) qui ont débouchées en particulier sur l'organisation d'un challenge associé au congrès ECML/PKDD 2007, l'animation d'un workshop Mining Spatio-Temporal Data (MSTD) associé aux congrès ECML/PKDD 2005, le co-encadrement de thèses ainsi qu'un projet STIC-ASIA Expedo.

## 8.2 Publications

### Edition d'ouvrage et de revue

- Co-Editor (with G. Andrienko, FhG AIS, Germany, D. Malerba, University of Bari, Italy and M. May, FhG AIS, Germany) special issue "*Mining Spatio-Temporal Data*" of the international journal *JIIS Journal of Intelligent Information Systems*, Kluwer Academic Publishers, Volume 27, Number 2, September 2006
- Co-Editor (with P. Poncelet EMA-LGI2P Nîmes and F. Masegla, INRIA Sophia Antipolis) Book "*Data Mining Patterns : New Methods and Application*", Idea Group Inc. Publishers, to appear 2007.
- Co-Editor (with P. Poncelet EMA-LGI2P Nîmes and F. Masegla, INRIA Sophia Antipolis) Book "*Successes and New Directions in Data Mining*", Idea Group Inc. Publishers, to appear 2007.

### Publications dans des ouvrages

- F. Masegla, P. Poncelet and M. Teisseire. "*Peer to Peer Usage Analysis*", Chapter in "*Encyclopaedia of Multimedia Technology and Networking*", M. Pagani (ed.), 2006, 10 pages.
- A. Laurent, P. Poncelet and M. Teisseire. "*Fuzzy Data Mining for the Semantic Web : Building XML Mediator Schemas*", Chapter in "*Fuzzy Logic and the Semantic Web*" - Elsevier, 2006, pp. 249-264.
- F. Masegla, M. Teisseire and P. Poncelet. "*Sequential Pattern Mining : A Survey on Issues and Approaches*", Chapter in "*Encyclopedia of Data Warehousing and Mining*", J. Wang (ed.), Information Science April 2005, 10 pages.
- M. Teisseire, P. Poncelet and R. Cichetti. "*Events as Behavioral Modeling Drivers*", First chapter of "*Object-Oriented Modeling*", Papazoglou M. P., Spaccapietra S. and Tari Z. (Eds), MIT-Press, 2000, 25 pages.

### Publications dans des revues internationales avec comité de lecture

- C. Fiot, A. Laurent and M. Teisseire. "*Softening the Blow of Frequent Sequence Analysis : Soft Constraints and Temporal Accuracy*", *International Journal of Web Engineering and Technology*, to appear 2008.

- F. Del Razo, S. Sanchez, A. Laurent, P. Poncelet and M. Teisseire. "Data structures for efficient tree mining : from crisp to soft embedding constraints", In International Journal of Applied Mathematics and Computer Science (AMCS), AMCS Special Issue, Soft computing for information management on the Web. To appear in 2007.
- F. Massegli, P. Poncelet and M. Teisseire. "Web Usage Mining : Extracting Unexpected Periods from Web Logs", In Data Mining and Knowledge Discovery (DMKD) Journal, Springer Verlag. To appear in 2007.
- C. Fiot, A. Laurent and M. Teisseire. "From Crispness to Fuzziness : Three Algorithms for Soft Sequential Pattern Mining", In IEEE Transaction on Fuzzy Sets, Volume 15, Issue 6, pp. 1263-1277, Dec. 2007.
- C. Raïssi, P. Poncelet and M. Teisseire. "Towards a new approach for mining frequent itemsets on data stream", In JIIS Journal of Intelligent Information Systems, Kluwer Academic Publishers, Issue 28, Number 1, pp. 23-36, February 2007.
- S. Jaillet, A. Laurent and M. Teisseire. "Sequential patterns for Text Categorization", In Intelligent Data Analysis (IDA) Journal, Volume 10, Issue 3, pp. 199 - 214, 2006.
- F. Massegli, M. Teisseire and P. Poncelet. "HDM : A Client/Server/Engine Architecture for Real Time Web Usage Mining", In Knowledge and Information Systems (KAIS) Journal, Volume 5, Issue 4, pp. 439 - 465, November 2003.
- F. Massegli, P. Poncelet and M. Teisseire "Incremental Mining of Sequential Patterns in Large Databases". In Data and Knowledge (DKE) Journal, Volume 46, Issue 1, pp. 97-121, July 2003.

#### Publications dans des revues nationales avec comité de lecture

- V. Kapoor, P. Poncelet, F. Troussset et M. Teisseire. "Préservation de la vie privée : recherche de motifs séquentiels dans des bases de données distribuées". Revue Ingénierie des Systèmes d'Information (ISI), Numéro spécial "Journées Bases de Données Avancées". To appear 2007.
- F. Del Razo Lopez, A. Laurent et M. Teisseire. "Une représentation des arborescences pour la recherche de sous-structures fréquentes", Revue des Nouvelles Technologies de l'Information - Numéro spécial "Extraction des connaissances : Etat et perspectives". Novembre 2005, Vol E-5, pp. 299-308 (version étendue de l'article présenté à l'atelier Fouille de données complexes - EGC 2005).
- S. Jaillet, M. Teisseire et G. Dray. "Adéquation des modèles de représentation aux méthodes de catégorisation", Revue des Nouvelles Technologies de l'Information - Numéro spécial "Fouille de données complexes". Octobre 2005, Vol E-4, pp. 191-209.
- F. Massegli, M. Teisseire et P. Poncelet. "Recherche des motifs séquentiels" Revue Ingénierie des Systèmes d'Information (ISI), numéro spécial "Extraction de motifs dans les bases de données". Décembre 2004, Vol. 9, N° 3-4, pp. 183-210.
- P.A. Laur, M. Teisseire et P. Poncelet. "Données Semi Structurées : extraction, maintenance and analyse de tendances", Revue Ingénierie des Systèmes d'Information (ISI), numéro spécial "Bases de Données semi-structurées". Décembre 2003, Vol. 8, N° 5-6, pp. 49-78.
- R. Cicchetti, P. Poncelet et M. Teisseire. "Modélisation et vérifications comportementales", Revue ISI Ingénierie des Systèmes d'Information, AFCET - Editions HERMES, Août 1997, Vol. 5, N° 3, pp. 265-285.
- P. Poncelet, M. Teisseire, R. Cicchetti et L. Lakhal. "IFO2, une approche pour la conception de bases de données avancées", Revue Ingénierie des Systèmes d'Information (ISI), Vol. 1, N°4, pp. 467-510, Décembre 1993.

**Publication invitée dans une revue internationale**

- F. Masegla, P. Poncelet, and M. Teisseire. "Using Data Mining Techniques on Web Access Logs to Dynamically Improve Hypertext Structure". In ACM SigWeb Letters, pp. 13-19, Vol. 8, N. 3, October 1999.

**Publications dans des conférences internationales avec comité de lecture**

2008

- C. Fiot and G. A. P. Saptawati and A. Laurent and M. Teisseire. "Learning Bayesian Network Structure from Incomplete Data without any Assumption", In Proceedings of the 13th International Conference on Database System for Advance Applications (DASFAA'08), to appear, 2008.

2007

- M. Plantevit, S. Goutier, F. Guisnel, A. Laurent and M. Teisseire. "Mining Unexpected Multidimensional Rules", In Proceedings of the ACM DOLAP'07 Conference, Lisbon, Portugal, November 2007.
- C. Fiot, A. Laurent and M. Teisseire. "SPoID : Do not throw meaningful incomplete sequences away!", In Proceedings of the 5th Conference of the European Society for Fuzzy Logic and Technology (EUSFLAT'07), September 2007.
- C. Fiot, A. Laurent and M. Teisseire. "Approximate Sequential Patterns for Incomplete Sequence Database Mining", In Proceedings of the 2007 IEEE International Conference on Fuzzy Systems, Imperial College, London, UK, July 2007.
- C. Fiot, A. Laurent and M. Teisseire. "Extended Time Constraints for Sequence Mining", In Proceedings of the 14th IEEE International Symposium on Temporal Representation and Reasoning (TIME'07), June 2007.

2006

- M. Plantevit, A. Laurent and M. Teisseire. "HYPE : Mining Hierarchical Sequential Pattern", In Proceedings of the ACM Ninth International Workshop on Data Warehousing and OLAP (DOLAP 2006) (in conjunction with ACM CIKM 2006), Arlington, US, November 2006.
- V. Kapoor, P. Poncelet, F. Troussset and M. Teisseire. "Privacy Preserving Sequential Pattern Mining in Distributed Databases", In Proceedings of the Fifteenth Conference on Information and Knowledge Management (CIKM 2006), Arlington, US, November 2006.
- C. Raissi, P. Poncelet and M. Teisseire. "SPEED : Mining Maximal Sequential Patterns over Data Streams", In Proceedings of the 3rd IEEE Conference On Intelligent Systems (IS'06), Westminster, UK, September 4-6, 2006.
- C. Fiot, A. Laurent and M. Teisseire. "Web Access Log Mining With Soft Sequential Patterns", In the proceedings of the 7th International FLINS Conference on Applied Artificial Intelligence (FLINS'06) Special Session "Web intelligence", Genova, Italy, 29-31 Août 2006.
- C. Fiot, A. Laurent, M. Teisseire and B. Laurent. "Why Fuzzy Sequential Patterns can Help Data Summarization : an Application to the INPI Trademark Database", In Proceedings of the IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2006), Vancouver, Canada, July 16-21, 2006.
- S. Sanchez, A. Laurent, P. Poncelet and M. Teisseire. "FuzBT : a Binary Approach for Fuzzy Tree Mining", In Proceedings of the 11th International Conference of Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU 06), Paris, France, July 2006.



- F. Masseglia, P. Poncelet and M. Teisseire. "Peer-to-Peer Usage Mining : a Distributed Mining Approach", In Proceedings of the IEEE 20th International Conference on Advanced Information Networking and Applications (AINA 2006), Vienna, Austria, April 2006, pp. 993-998.

2005

- F. Masseglia, P. Poncelet, M. Teisseire and A. Marascu. "Web Usage Mining : Extracting Unexpected Periods from Web Logs", In Proceedings of the 2nd IEEE Workshop on Temporal Data Mining (TDM'05). Held in conjunction with ICDM'05, Houston, USA, November 27, 2005.
- M. Plantevit, Y.W. Choong, A. Laurent, D. Laurent and M. Teisseire. "M2SP : Mining Sequential Patterns Among Several Dimensions", In Proceedings of PKDD'05 : Principles and Practice of Knowledge Discovery in Databases, Porto, Portugal, October 2005, LNCS n° 3721, Springer Verlag, pp. 205-216.
- F. Del Razo Lopez, A. Laurent, P. Poncelet and M. Teisseire. "RSF - A New Tree Mining Approach with an Efficient Data Structure", In Proceedings of EUS-FLAT'05 : European Society for Fuzzy Logic and Technologie, 2005, September, Barcelona, Spain, pp. 1088-1093.

2004

- S. Jaillet, A. Laurent, M. Teisseire and J. Chauché. "Order and Mess in text categorization : Why using sequential patterns to classify", In Proceedings of Third Workshop on Mining Temporal and Sequential Data (TDM'2004), in conjunction with The Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2004), August 22, 2004, Seattle, WA, pp. 121-128.
- S. Jaillet, M. Teisseire, G. Dray and M. Plantié. "Comparing Concept-based and Statistical Representations for Textual Categorization", In Proceedings of the Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU 2004) , July 4-9 2004, Perugia, Italy, pp. 91-98.
- J. Tranier, R. Baraer, Z. Bellahsene and M. Teisseire. "Where's Charlie : Family based heuristics for Peer-to-Peer Schema Integration", In Proceedings of the 8th International Database Engineering and Applications Symposium (IDEAS'04) July, 7th - 9th 2004 Coimbra, Portugal, pp. 227-235.
- F. Masseglia, P. Poncelet and M. Teisseire. "Pre-Processing Time Constraints for Efficiently Mining Generalized Sequential Patterns", In Proceedings of the 11th International Symposium on Temporal Representation and Reasoning (TIME'04), IEEE , Tatihou Island, Normandie, France, 1-3 July 2004, pp. 87-95.

2003

- P.A. Laur, M. Teisseire and P. Poncelet. "Web Usage Mining : Extraction, Maintenance and Behaviour Trends", In Proceedings of the 1st Indian International Conference on Artificial Intelligence (IICAI'03), Hyderabad, India, December 2003.
- P.A. Laur, M. Teisseire and P. Poncelet. "AUSMS : An Environment for Frequent Sub-Structures Extraction in a Semi-Structured Object Collection", In Proceedings of the 14th International Conference on Database and Expert Systems Applications (DEXA'03), LNCS 2736, pp. 38-45, Prague, Czech Republic, September 03.
- S. Jaillet, M. Teisseire, J. Chauché and V. Prince. "Classification of Documents by Content", In Proceedings of the 2nd IEEE International Conference on Cognitive Informatics (ICCI 2003), August 2003, London, UK, pp. 214-222.



2002

- F. Massegli, M. Teisseire and P. Poncelet. "Real Time Web Usage Mining with a Distributed Navigation Analysis", In Proceedings of the 12th International Workshop on Research Issues on Data Engineering (RIDE'02), February 2002, San Jose, USA.

2001

- F. Massegli, M. Teisseire and P. Poncelet. "*Real Time Web Usage Mining : a Heuristic based Distributed Miner*", In Proceedings of the Web Information Systems Engineering (WISE'01), December 2001, Kyoto, Japan.

2000

- P.A. Laur, F. Massegli, P. Poncelet, and M. Teisseire. "*A General Architecture for Finding Structural Regularities on the Web*", In Proceedings of the 9th International Conference on Artificial Intelligence (AIMSA'00), Lecture Notes in Artificial Intelligence, Springer Verlag, Varna, Bulgaria, September 2000, pp. 179-188.
- F. Massegli, P. Poncelet, and M. Teisseire. "*Web Usage Mining : How to Efficiently Manage New Transactions and New Clients*", In Proceedings of the 4th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD'2000), Poster session, Lecture Notes in Artificial Intelligence, Springer Verlag, Lyon, France, September 2000.

Avant 2000

- B. Faure, M. Teisseire and R. Cicchetti. "*Activity Threads : A Unified Framework for Aiding Behavioural Modelling*", In Proceedings of the International Conference on Database and Expert Systems Applications (DEXA'97), Toulouse, France, September 1997, pp. 122-132.
- Z. Bellahsene, P. Poncelet and M. Teisseire "*Views for Information System Design without Reorganization*", In Proceedings of the 8th International Conference on Advanced Information Systems Engineering (CAiSE'96), Lecture Notes in Computer Science, Springer Verlag, Crete, Greece, June 1996, pp. 496-513.
- M. Teisseire. "*Behavioural Constraints : Why using Events instead of States*", In Proceedings of the 14th International Conference on Object-Oriented Entity Relationship (O-O ER'95), Lecture Notes in Computer Science, Springer Verlag, Gold Coast, Australia, December 1995, pp. 123-132.
- M. Teisseire. "*Event Schema Updating*", In Proceedings of the International Workshop on Database and Expert Systems Applications (DEXA'95), London, UK, September 1995, pp. 453-460.
- M. Teisseire, P. Poncelet and R. Cicchetti. "*Dynamic Modelling with Events*", In Proceedings of the 6th Advanced Information Systems Engineering (CaiSE'94), Utrecht, The Netherlands, June 1994, Lecture Notes in Computer Science 811 Springer, pp. 186-199.
- M. Teisseire, P. Poncelet and R. Cicchetti. "*IFO2 : a Uniform Approach for Information System Modelling*", In Proceedings of the Fifth International Workshop on the Deductive Approach to Information Systems and Databases. September 1994, Aiguablava, Costa Brava, Catalonia (DAISD 1994), pp. 33-53.
- M. Teisseire and R. Cicchetti. "*An Algebraic Language for Event-Driven Modelling*", In Proceedings of the 5th International Conference on Database and Expert Systems Applications (DEXA '94), Athens, Greece, September 1994, Lecture Notes in Computer Science 856, pp. 300-309.

- M. Teisseire, P. Poncelet and R. Cicchetti. "Towards Event-Driven Modelling for Database Design", In Proceedings of the 20th International Conference on Very Large Data Bases, (VLDB'94), September 12-15, 1994, Santiago de Chile, Chile, pp. 285-296.
- P. Poncelet, M. Teisseire, R. Cicchetti and L. Lakhal. "Towards a Formal Approach for Object Database Design", In Proceedings of the 19th International Conference on Very Large Data Bases (VLDB'93), August 1993, Dublin, Ireland, pp. 278-289.

### Publications dans des conférences nationales avec comité de lecture

2008

- M. Plantevit, A. Laurent et M. Teisseire. "Extraction de Motifs Séquentiels Multidimensionnels Clos sans Gestion d'Ensemble de Candidats", Actes des 8èmes journées d'Extraction et Gestion des Connaissances (EGC'08), Nice, janvier 2008.

2007

- L. Di-Jorio, C. Fiot, L. Abrouk, D. Hérin et M. Teisseire. "Enrichissement d'ontologie : quand les motifs séquentiels labellisent des relations", Actes des 23èmes journées de Bases de Données Avancées (BDA'07), Marseille, octobre 2007.
- C. Fiot, A. Laurent et M. Teisseire. "SPoID : Extraction de motifs séquentiels pour les bases de données incomplètes", Actes des 7èmes journées d'Extraction et Gestion des Connaissances (EGC'07), janvier 2007.
- M. Plantevit, A. Laurent et M. Teisseire. "Extraction de séquences multidimensionnelles convergentes et divergentes", Actes des 7èmes journées d'Extraction et Gestion des Connaissances (EGC'07), janvier 2007.
- M. Plantevit, A. Laurent et M. Teisseire. "Extraction d'outliers dans des cubes de données : une aide à la navigation", Actes des 3èmes journées EDA, Blois, France, Juin 2007.

2006

- V. Kapoor, P. Poncelet, F. Troussel and M. Teisseire. "Privacy Preserving Sequential Pattern Mining in Distributed Databases", Actes des 22ièmes Journées Bases de Données Avancées, Lille, France, Octobre 2006.
- L. Di Jorio, D. Jouve, D. Kraemer, A. Serra, C. Raïssi, A. Laurent, M. Teisseire et P. Poncelet. "Vpsp : extraction de motifs séquentiels dans Weka", Démonstration dans les 22ièmes Journées Bases de Données Avancées, Lille, France, Octobre 2006.
- M. Plantevit, A. Laurent et M. Teisseire. "Hype : Prise en compte des hiérarchies lors de l'extraction de motifs séquentiels multidimensionnels", Actes des 2èmes journées EDA, Versailles, France, Juin 2006.
- C. Fiot, A. Laurent et M. Teisseire. "Des motifs séquentiels généralisés aux contraintes de temps étendues", Actes des 6ièmes Journées Francophones "Extraction et Gestion des Connaissances" (EGC 2006), Lille, France, Janvier 2006.
- F. Maseglier, P. Poncelet, M. Teisseire and A. Marascu. "Web Usage Mining : extraction de périodes denses à partir de logs", Actes des 6ièmes Journées Francophones "Extraction et Gestion des Connaissances" (EGC 2006), Lille, France, Janvier 2006, pp. 403-408.

- F. Del Razo Lopez, A. Laurent, P. Poncelet et M. Teisseire. "Recherche de sous-structures fréquentes pour l'intégration de schéma XML", Actes des 6èmes Journées Francophones "Extraction et Gestion des Connaissances" (EGC 2006), Lille, France, Janvier 2006, pp. 487-498.
- F. Masseglia, P. Poncelet et M. Teisseire. "Fouille de Données dans les systèmes pair-à-pair pour améliorer la recherche de ressources", Actes des 6èmes Journées Francophones "Extraction et Gestion des Connaissances" (EGC 2006), Lille, France, Janvier 2006, pp. 469-474.

2005

- C. Raissi, P. Poncelet and M. Teisseire. "Need for SPEED : Mining Sequential Patterns in Data Streams", Actes des 21èmes Journées Bases de Données Avancées (BDA 2005), Saint Malo, France, October 2005.
- M. Plantevit, Y.W. Choong, A. Laurent, D. Laurent et M. Teisseire. "Motifs séquentiels multidimensionnels étoilés", Actes des 21èmes Journées Bases de Données Avancées (BDA 2005), Saint Malo, France, October 2005.
- C. Fiot, A. Laurent et M. Teisseire. "Motifs Séquentiels Flous : un peu, beaucoup, passionnément" Actes des Journées Extraction et gestion des connaissances (EGC'05) Paris, Janvier 2005, pp. 507-518.
- F. Del Razo Lopez, A. Laurent et M. Teisseire. "Représentation Efficace des Arborescences pour la Recherche des Sous - Structures Fréquentes", Actes de l'Atelier Fouille de Données Complexes, EGC'05 : Extraction et Gestion des Connaissances, Paris Janvier 2005, pp. 113-120.

2004

- C. Fiot, G. Dray, A. Laurent et M. Teisseire. "A la Recherche des Motifs Séquentiels Flous", Actes de LFA'04 : Rencontres Francophones sur la Logique Floue et ses Applications, Novembre 2004, Nantes, pp. 131-138.
- S. Jaillet, M. Teisseire, A. Laurent et J. Chauché. "Ordre et désordre dans la catégorisation de textes", Actes des 20èmes journées Bases de Données Avancées (BDA'04), Octobre 2004, Montpellier, pp. 555-573.

2003

- J. Chauche, V. Prince, S. Jaillet et M. Teisseire. "Classification Automatique de Textes à partir de leur Analyse Syntaxico-Sémantique", Actes de TALN'03 : 10ème Conférence Internationale sur le Traitement Automatique du Langage Naturel, Juin 2003, pp. 55-65.
- S. Jaillet, M. Teisseire, J. Chauché et V. Prince. "Classification Automatique de Documents", Actes de INFORSID'03 : Informatique des Organisations et Systèmes d'Information et de Décision, Juin 2003, pp. 87-102.

2002

- F. Masseglia, M. Teisseire et P. Poncelet. "HDM, un module de fouille de données distribué en temps réel", Actes des 2èmes Journées Extraction et Gestion des Connaissances (EGC'02), Montpellier, France, Janvier, 2002.

2001

- F. Masseglia, M. Teisseire et P. Poncelet. "Web Usage Mining Intersites : Analyse du Comportement des Utilisateurs à Impact Immédiat", Actes des 17èmes Journées Bases de Données Avancées (BDA'01), Agadir, Maroc, Octobre 2001.

2000

- F. Masseglia, P. Poncelet and M. Teisseire. "Incremental Mining of Sequential Patterns in Large Databases", Actes des 16ièmes Journées Bases de Données Avancées (BDA'00), Blois, France, Octobre 2000.

Avant 2000

- F. Masseglia, P. Poncelet et M. Teisseire. "Extraction efficace de motifs séquentiels : le prétraitement des données", Actes des 15ièmes Journées Bases de Données Avancées (BDA'99), pp. 341-360, Bordeaux, France, Octobre 1999.
- R. Cicchetti, P. Poncelet et M. Teisseire. "Une aide à la conception et au contrôle de règles actives", Actes des XIIIièmes Journées Bases de Données Avancées (BDA'96), Cassis, Août 1996, pp. 19-34.
- R. Cicchetti, P. Poncelet et M. Teisseire. "Modélisation et validation comportementales", Actes du congrès INFORSID'96, Bordeaux, Juin 1996, pp. 407-425.
- M. Teisseire, P. Poncelet, R. Cicchetti et L. Lakhal. "Conception de bases de données avancées : le projet IFO2" Actes du Congrès AFCET'93, Colloque "Bases de Données", pages 105-114, Versailles, France, Juin 1993.
- P. Poncelet, M. Teisseire et L. Lakhal. "IFO2, modèle et principe pour la conception de Bases de Données Avancées", Actes des 8ièmes Journées Bases de Données Avancées (BDA'92), Trégastel, France, pp. 320-338, Septembre 1992.

## 8.3 Encadrements

### 8.3.1 Encadrement de Thèses

Depuis 1998, je participe au co-encadrement de thèse :

#### Thèses en cours

- **Paola Salle.** *Sujet* : Gestion de la dynamique des ontologies : mises-à-jour et mesure d'adéquation aux données  
*Taux d'encadrement* : 60% (avec S. Bringay, MCF UMIII, D. Hérin, Prof UMII)  
*Date de début de thèse* : Octobre 2007  
*Date de soutenance prévue* : Octobre 2010  
*Financement* : bourse Régionale  
*Lieu* : LIRMM
- **Cécile Low Kam.** *Sujet* : Etude probabiliste et statistique des motifs séquentiels, et l'application aux grandes bases de données  
*Taux d'encadrement* : 20% (avec A. Mas, MCF I3M)  
*Date de début de thèse* : Octobre 2007  
*Date de soutenance prévue* : Octobre 2010  
*Financement* : bourse Régionale  
*Lieu* : LIRMM
- **Lisa Di Jorio.** *Sujet* : Extraction de connaissances dans les bases de données : motifs séquentiels et ontologies  
*Taux d'encadrement* : 50% (avec A. Laurent, MCF Polytech Montpellier)  
*Date de début de thèse* : Octobre 2007  
*Date de soutenance prévue* : Octobre 2010  
*Financement* : bourse Régionale

*Lieu* : LIRMM

- **Marc Plantevit.** *Sujet* : Fouille de données multidimensionnelles  
*Taux d'encadrement* : 40% (avec A. Laurent , MCF Polytech Montpellier)  
*Date de début de thèse* : Octobre 2005  
*Date de soutenance prévue* : Octobre 2008  
*Financement* : bourse MESR  
*Lieu* : LIRMM

Dans le cadre de la collaboration entre l'Université de Montpellier II et l'ITB de Bandung Indonésie, je participe au co-encadrement de thèse :

- **Putri SAPTAWATI.** *Sujet* : Prise en compte des valeurs manquantes ou incomplètes pour la construction de réseau bayésien.
- **Sri PURWANTI.** *Sujet* : Classification de textes et imprécision.

Ces deux doctorantes sont actuellement enseignantes au sein de l'ITB.

#### Thèses soutenues

- **Céline Fiot.** *Sujet* : Traitement des données manquantes à l'aide des motifs séquentiels  
*Taux d'encadrement* : 60% (avec A. Laurent , MCF Polytech Montpellier)  
*Date de début de thèse* : Octobre 2004  
*Date de soutenance* : Octobre 2007  
*Financement* : bourse BDI CNRS-Région  
*Lieu* : LIRMM  
*Situation actuelle* : Post-Doc INRIA
- **Federico Del Razo Lopez.** *Sujet* : Recherche de structures fréquentes dans des données semi-structurées  
*Taux d'encadrement* : 60% (avec A. Laurent, MCF Polytech Montpellier)  
*Date de début de thèse* : Octobre 2003  
*Date de soutenance* : Juillet 2007  
*Financement* : bourse Sfere (programme mexicain)  
*Lieu* : LIRMM  
*Situation actuelle* : Enseignant Chercheur au Mexique
- **Simon Jaillet.** *Sujet* : Catégorisation Automatique de Documents Textuels : D'une Représentation Basée sur les Concepts aux Motifs Séquentiels  
*Taux d'encadrement* : 80% (avec J. Chauché, PR Université Montpellier II)  
*Date de début de thèse* : Octobre 2001  
*Date de soutenance* : Mars 2005  
*Financement* : bourse MESR  
*Lieu* : LIRMM  
*Situation actuelle* : Ingénieur de recherche
- **Pierre-Alain Laur.** *Sujet* : Données semi-structurées : Découverte, Maintenance et analyse de tendances  
*Taux d'encadrement* : 50% (avec P. Poncelet, PR Ecole des mines d'Alès)  
*Date de début de thèse* : Octobre 2001

*Date de soutenance* : Août 2004

*Financement* : bourse MESR

*Lieu* : LIRMM

*Situation actuelle* : ATER Université de Martinique

- **Florent Masseglia**. *Sujet* : Algorithmes et applications pour l'extraction de motifs séquentiels dans le domaine de la fouille de données : de l'incrémental au temps réel

*Taux d'encadrement* : 50% (avec P. Poncelet (MCF IUT d'Aix en Provence)

*Date de début de thèse* : janvier 1999

*Date de soutenance* : janvier 2002

*Financement* : bourse MESR

*Lieu* : LIRMM

*Situation actuelle* : Chercheur - Inria Sophia Antipolis

### 8.3.2 Encadrement de Stages Recherche de Master R (ou ex DEA)

- **Delphine Jouve**. "Détection d'outliers dans les motifs multidimensionnels", juin 2007 (taux d'encadrement 20% avec M. Plantevit, Doctorant et A. Laurent, MCF au LIRMM)
- **Lisa Di Jorio**. "Enrichissement d'ontologie avec les motifs séquentiels", juin 2007 (taux d'encadrement 60% avec L. Abrouk, ATER et D. Hérin, PR au LIRMM)
- **Cécile Low Kam**. "Détection de motifs séquentiels inattendus à l'aide d'approche statistiques", juin 2007 (taux d'encadrement 20 % avec A. Mas, MCF à l'ISM)
- **Mesbahi Larbi**. "Méthodes de fouille de données pour la classification de texte", juin 2006 (taux d'encadrement 20% avec M. Roche, MCF et J. Chauché, PR au LIRMM)
- **Haouyuan Li**. "Mining sequential patterns with transversal hypergraph computation", juin 2006 (taux d'encadrement 100%).
- **Marc Plantevit**. "Motifs séquentiels multidimensionnels", juin 2005 (taux d'encadrement 50% avec A. Laurent, MCF au LIRMM)
- **Chedy Raïssi**. "Motifs séquentiels et data streams", juin 2005 (taux d'encadrement 20% avec P. Poncelet, PR au LIG2P)
- **Céline Fiot**. "Motifs séquentiels flous", juillet 2004 (taux d'encadrement 50% avec A. Laurent, MCF au LIRMM)
- **Renaud Baraër**. "Traitement des requêtes pour le système Xpeer", juillet 2003 (taux d'encadrement : 50% avec Z. Bellahsene, MCF au LIRMM)
- **Simon Jaillet**. "Le prétraitement des données pour la recherche de motifs séquentiels", juin 2001 (taux d'encadrement : 80% avec F. Masseglia, Post-doc au LIRMM)
- **Bennouas Toufik**. "Les chaînes de Markov comme outil de recherche des motifs séquentiels", septembre 2000 (taux d'encadrement : 100%)
- **Florent Masseglia**. "L'extraction de motifs séquentiels", juin 1998 (taux d'encadrement : 50% avec P. Poncelet, MCF IUT Aix-en-Provence, Université de la Méditerranée)
- **Valérie Monfort**. "Spécification d'un noyau de composants conceptuels adaptés au contexte des bases de données orientées objets", juin 1991 (taux d'encadrement : 70% avec A. Cavarero, PR Université de Nice-Sophia Antipolis)



### 8.3.3 Encadrement de Mémoires d'Ingénieur CNAM

Ces dernières années, j'ai également eu l'occasion d'encadrer trois mémoires d'ingénieur CNAM - Montpellier :

- S. Sanchez. "Plate forme d'extraction de connaissances pour les données arborescentes issues du Web", mars 2006
- M. Pecoraro. "Conception d'une borne interactive relative au guide de l'auditeur du CNAM", novembre 1997
- M. Dori. "Conception d'une base de connaissances en contexte intranet : application safari", mars 1997

## 8.4 Transferts de Technologie

- Responsable scientifique pour le LIRMM du projet EDF. L'objectif de ce projet est d'évaluer et d'étudier de nouveaux algorithmes d'extractions de motifs multidimensionnels et inattendus dans de grands jeux de données variés. Partenaire : EDF R&D. Période : 2006-2007. Montant du Projet : 50K euros.
- Responsable scientifique pour le LIRMM de l'ARC SéSur. L'objectif de l'Action de Recherche Coopérative INRIA "Sécurité et Surveillance dans les flots de données" est d'étudier de nouvelles approches adaptées à la fois au monitoring et à l'extraction de connaissances dans des flots de données. Partenaires : IRISA (Rennes), INRIA (Rocquencourt, Sophia Antipolis), LGI2P (Nîmes). Période : 2007-2008, Part du LIRMM : 25 K euros.
- Responsable de projets de transfert de technologie régional avec les sociétés Axiliance. Ce projet rentrant dans le cadre d'une AFT régionale consiste à détecter des attaques sur un site Web à l'aide de techniques de fouille de données. Partenaires : société Axiliance, LGI2P (Nîmes). Période : 2006-2007. Part du LIRMM : 20K euros.
- Responsable du projet KEOSIA. L'objectif de ce projet qui rentre dans le cadre du LRI (Languedoc Roussillon Incubation) consiste à proposer des solutions de fouille de données pour l'amélioration de l'accompagnement et de la gestion de la relation de patients atteints d'affections de longue durée. Partenaire : KEOSIA. Période : 2006. Montant du projet : 10K euros
- Responsable du projet AIRTIST. L'objectif de ce projet qui rentre dans le cadre du LRI (Languedoc Roussillon Incubation) consiste à proposer des solutions de fouille de données pour cibler des publicités de téléchargements de musique sur Internet. Partenaire : AIRTIST. Période : 2006. Montant du projet : 6K euros.
- Responsable du projet BPSolar. Ce projet rentrant dans le cadre d'un contrat de plan Etat Région consiste à détecter automatiquement des pannes sur des capteurs. Partenaire : BPSolar. Période : 2003. Montant du projet : 30K euros.
- Participation au projet de transfert de technologie régional avec la société SQLI sur la classification de messages à l'aide des motifs séquentiels. Partenaire : SQLI. Période : 2004. Montant du projet : 30K euros.
- Responsable scientifique pour le LIRMM d'un contrat RNTL "Contexte Bourse". L'objectif de ce projet est d'élaborer une plateforme d'extraction de connaissances pour des données boursières. Partenaires : les sociétés Thalès, Firstinvest, Elseware et le laboratoire PRISM. Période : 2002-2003. Part du LIRMM : 45K euros
- Participation au projet Albert Inc. Ce projet, rentrant dans le cadre d'un contrat de plan Etat-Région s'intéresse à l'analyse de requête en langage naturel basée sur des techniques de fouille de données. Partenaire : Albert Inc. Période : 2000.

Montant du projet : 20k euros.

- Participation au projet CIMM. L'objectif de ce projet était d'étudier l'application de nouvelles méthodes de conception orientée objet. Partenaire : CIMM. Période : 1996. Montant du projet : 3K euros.

## 8.5 Perspectives

Comme nous l'avons vu tout au long de ce mémoire, les travaux de recherche que nous avons menés ont été inspirés par le fait qu'il fallait trouver des réponses aux nouveaux défis qui se présentent aux chercheurs en fouille de données. Les perspectives que nous souhaitons mener rentrent, bien entendu, dans ce cadre dans la mesure où nous souhaitons toujours offrir à l'utilisateur final des connaissances les plus utiles possibles. Certaines perspectives sont à moyen terme dans la mesure où elles consistent à étendre ou à poursuivre des travaux que nous avons déjà commencé. Nous verrons ainsi, par la suite, que les problématiques de préservation de la vie privée ou de détection de motifs dans des flots sont fortement liées au développement soit de nouvelles contraintes soit d'évolution technologique. D'autres perspectives, à plus long terme nous intéressent. Même si nous avons également commencé à travailler sur certaines d'entre elles, il est clair qu'elles nécessitent de nombreux travaux de recherche futurs. Par exemple, en considérant que la notion de support n'est plus suffisante, nous rejoignons les nouvelles tendances de la communauté qui montrent qu'il devient indispensable d'intégrer les contraintes à l'intérieur des algorithmes de fouille pour obtenir des résultats pertinents rapidement. En s'intéressant à de nouveaux types de motifs (inattendus, aberrants, surprenants), nous poussons ainsi les limites des approches traditionnelles car nous recherchons des motifs peu fréquents dans un grand espace de recherche. L'une des limites du processus d'extraction est souvent de ne pas incorporer la connaissance que nous avons du domaine lors des différentes étapes. Notre objectif, dans ce cas, est d'étudier les modifications à apporter non seulement au processus mais également à certaines étapes importantes (notamment la fouille) pour intégrer le plus possible la connaissance à chacune des étapes.

### 8.5.1 De la préservation de la vie privée

Ces dernières années, l'utilisation croissante des systèmes multi-bases a entraîné le développement d'un grand nombre de bases de données transactionnelles distribuées. Dans un contexte d'aide à la décision, les grandes organisations souhaitent alors pouvoir extraire de la connaissance à partir de l'ensemble de ces bases. Par exemple, si nous considérons une chaîne de magasins avec différentes franchises, chacune des bases transactionnelles peut contenir des informations sur l'historique des achats d'un même ensemble de clients. Fouiller les données en considérant l'union de toutes les bases transactionnelles offre de nouvelles connaissances utiles pour le décideur. Toutefois, même si ces gros volumes de données doivent permettre d'améliorer la qualité de la décision, nous sommes confrontés à la difficulté d'identifier efficacement des connaissances à partir de ces sources de données multiples [XZ03, ZYO99]. En effet, à l'heure actuelle, les algorithmes de fouille de données considèrent que les données sont toutes stockées sur un même site centralisé.

Récemment, de nouvelles lois, comme HIPAA (Health Insurance Portability and Accountability Act) [oHHS96] qui instaure un régime de protection des renseignements personnels en matière de santé au Etats Unis (ces lois sont à l'heure actuelle adoptées par de nombreux pays : Australie, Chine, Japon, ...) ou les nouvelles directives européennes, imposent de nouvelles contraintes sur la confidentialité des données afin



de préserver la vie privée des personnes. Bien entendu ce problème de confidentialité peut être adapté à de nombreux domaines d'applications (analyse de transactions financières, analyses de comportements sur des sites de e-commerce, ...). Préserver la vie privée dans un contexte de fouille de données nécessite de n'offrir des connaissances que si celles-ci garantissent de ne pas divulguer d'information sensible sur les individus concernés. Pour garantir que les algorithmes de fouille de données ne violent pas la vie privée des individus, certaines approches ont considéré qu'elles disposaient d'une connaissance préalable sur ce qui était sensible ou non. Cependant ce type d'approche reste très subjectif et est très difficile à mettre en œuvre.

L'une de nos perspectives est de nous intéresser à l'extraction de motifs séquentiels dans des bases de données distribuées tout en préservant la vie privée sans connaissance a priori. Comme nous l'avons vu au cours de ce mémoire, non seulement les approches existantes ne prennent pas en compte la contrainte de confidentialité mais également elles ne sont pas adaptées à de multiples sources de données. Traditionnellement les protocoles de calcul distribué sécuritaire multipartie ont été utilisés pour calculer de manière sécurisée n'importe quelle fonction générique. Cependant, la complexité de tels protocoles fait qu'ils ne sont pas adaptés à des tâches de fouille de données comme l'extraction de séquences. Récemment nous avons proposé un nouvel algorithme, PRIPSEP (*PRIVacy Preserving SEquential Patterns*), pour extraire des motifs séquentiels dans des bases de données distribuées tout en respectant la contrainte de préservation de la vie privée. Ce dernier est basé sur une architecture sécurisée constituée de sites semi-honnêtes, i.e. ils suivent le protocole correctement mais sont libres d'utiliser l'information qu'ils ont collectée pendant l'exécution du protocole et ils ne collaborent pas entre eux [KV02]. Même si l'approche que nous avons proposée garantit d'extraire des motifs fréquents sans divulguer d'information sur les sources de données, elle souffre cependant de certaines lacunes qu'il convient de combler. Tout d'abord, la nécessité d'utiliser une architecture spécialisée impose de modifier fortement les différentes applications existantes notamment pour faciliter le transfert des données. Ensuite, l'une des hypothèses fortes de notre approche est de considérer que les sites sont semi-honnêtes : quid s'ils essaient de collaborer entre eux. Enfin, PRIPSEP est basée sur des opérations binaires de type XOR qui impose de nombreux transferts de données qui peuvent s'avérer coûteux.

### 8.5.2 Des données disponibles de plus en plus rapidement

Nous avons vu tout au long de ce mémoire que la plupart des travaux considèrent que les données sont stockées de manière statique et tirent avantage de cette situation (plusieurs parcours sur la base, stockage de la base en mémoire centrale). En proposant une approche incrémentale (C.f. Chapitre 3), nous avons montré qu'il était possible de considérer l'aspect dynamique des données. ISE optimise ainsi l'étape de fouille en tirant profit des connaissances préalablement acquises et est donc adapté à de nombreux domaines d'applications où les données sont régulièrement mises à jour. Toutefois, suite aux évolutions technologiques et à l'apparition de nouveaux domaines d'applications (adaptation en temps réel à des utilisateurs dans le cas de clickstreams, détection de fraude sur les réseaux, supervision de processus, ...), de nouveaux problèmes apparaissent car les données manipulées sont obtenues en temps réel, de manière continue et ordonnées (*data streams*). Nous nous trouvons alors confrontés à des flux très importants de données (paquets TCP/IP, transactions, clickstreams, capteurs physiques, ...) et l'accès rapide à l'intégralité des données devient impossible. Pour répondre à ces applications, de nouveaux travaux de recherche, appelés "*Data Stream Mining*", se sont intéressés à la définition d'algorithmes de fouille pour appréhender ces nou-

velles contraintes. Dans ce cadre, les principaux travaux concernent les algorithmes de classification, de clustering, ou la maintenance d'items (i.e. minimiser le stockage des items dans le flux et assurer la mise à jour du support). De manière plus générale, la prise en compte des flots de données pose, pour la fouille, le deux défis principaux :

1. Les opérations traditionnelles sont inapplicables sur un flot de données. Les flots produisent des données en continu, très rapidement et de façon illimitée. Il est impossible d'utiliser des algorithmes traditionnels qui ont besoin de faire plusieurs passes sur les données. En prenant comme exemple l'extraction d'items ou de séquences fréquents, les principaux verrous à l'adaptation de méthodes traditionnelles sont : i) la technique "générer-élaguer" est inadaptée car l'étape de génération fait appel à des opérateurs de jointure, connus pour être typiquement bloquant car leur calcul nécessite de disposer de l'ensemble des données [GHP<sup>+</sup>03] ii) Les données ne peuvent être observées qu'une seule fois et iii) l'utilisation de la mémoire est limitée même si de nouveaux éléments continuent à être produits [MM02].
2. Le traitement exhaustif et exact des flots est impossible. La distribution des données change inévitablement dans le temps et l'utilisateur final est souvent plus intéressé par les changements récents (pour lesquels il veut une précision élevée) que par les changements plus éloignés (où une précision plus faible est satisfaisante) [GHP<sup>+</sup>03].

Même si des travaux apparaissent autour de la problématique de la recherche d'itemsets, il n'existe, à notre connaissance, peu d'approches pour extraire les séquences fréquentes. Récemment nous avons proposé une approche adaptée aussi bien à l'extraction d'itemsets fréquents (FIDS [RPT07]) qu'à celle de séquences fréquentes (SPEED [RMR06]). L'originalité de l'approche est d'utiliser une nouvelle structure qui permet de maintenir les motifs tout en appliquant une stratégie d'élagage rapide. A n'importe quel instant, un utilisateur peut poser des requêtes afin d'extraire les motifs maximaux fréquents dans un intervalle de temps choisi. Même si nos propositions sont efficaces et permettent d'extraire des motifs, elles peuvent dans certains cas se retrouver confrontées au problème d'un volume de connaissance (i.e. de séquences fréquentes) à conserver trop important et les techniques d'élagages associées pour permettre de conserver en mémoire les résultats imposent d'éliminer certaines connaissances. L'une de nos perspectives est de répondre aux questions suivantes : est-il possible d'appliquer des techniques d'échantillonnage sur le flot ? et surtout sommes nous à même de garantir que les résultats d'une extraction de motifs sont réellement représentatifs du contenu du flot tout en garantissant une marge d'erreur ?

### 8.5.3 Des motifs fréquents ? oui, mais ...

Proposer les motifs les plus fréquents dans un objectif d'aide à la décision n'est pas nécessairement le plus riche en connaissance. C'est pourquoi il est fondamental de se préoccuper des comportements atypiques qui à eux-seuls peuvent constituer la véritable pépite de connaissance de la base de données considérée [CDF<sup>+</sup>01, YWY04, SZ05]. Plus généralement, les *outliers sont des observations tellement différentes des autres qu'elles en sont suspicieuses et ont dû être générées par d'autres mécanismes* ([Haw80]). Dans la littérature, le terme d'outlier vient en opposition au terme d'exception. Dans le premier cas, ce sont les données atypiques par rapport à l'ensemble de la base qui sont extraites. Dans le second cas, ce sont les motifs qui viennent contredire des règles, des croyances ou des connaissances obtenues préalablement soit à l'aide d'un expert soit à l'aide d'une étape de fouille de données. Nos échanges avec les données agrégées d'EDF nous ont également confronté à cette problématique. Il existe des

comportements fréquents mais l'objectif est de mettre en évidence les éléments qui dérogent à la règle générale.

Dans notre contexte de données séquentielles et de motifs séquentiels, nous avons ainsi un double objectif :

- identifier des séquences de données atypiques par rapport à l'ensemble des données sources. Il n'existe à l'heure actuelle aucune méthode pour des séquences d'item-set.
- identifier des motifs qui contredisent soit un système de croyance soit un ensemble de motifs séquentiels déjà connus. Il n'existe aucune proposition à notre connaissance dans le contexte des motifs séquentiels

#### 8.5.4 Le processus d'extraction revisité

La plupart des approches que nous avons définies jusqu'à présent s'inscrivent dans le processus d'extraction traditionnel. A partir d'un ensemble de données, nous appliquons des algorithmes d'extraction de motifs pour extraire de la connaissance. Même si l'opérateur humain a un rôle non négligeable dans ce processus (sélection des attributs et éventuellement des données, sélection de l'algorithme le plus approprié, ...), il faut reconnaître que l'humain en tant qu'expert n'intervient qu'en bout de chaîne pour évaluer si la connaissance extraite est d'une part juste par rapport au domaine et d'autre part utile pour une prise de décision. L'un des problèmes de ce processus est que d'une part il faut attendre longtemps pour obtenir de la connaissance (e.g., dans le cas des motifs séquentiels, il est nécessaire de parcourir tout l'espace de recherche si la fonction d'anti monotonie ne permet pas d'élaguer des candidats) et que d'autre part les connaissances extraites peuvent être en contradiction pour diverses raisons avec la connaissance du domaine. L'une de nos perspectives est justement d'intégrer la connaissance du domaine plus tôt dans le cœur du processus.

Considérons par exemple un expert, il est à même de savoir dès le départ quelles sont les contraintes qu'il souhaite avoir sur les résultats obtenus. Pour cela, il lui suffit, par exemple pour les motifs, de restreindre à l'avance les motifs qu'il souhaite obtenir. Une approche triviale consiste bien sûr à extraire tous les motifs et à ensuite appliquer une étape de post traitement pour donner à l'expert les résultats désirés. Cette approche est malheureusement peu efficace dans la mesure où il est indispensable d'attendre l'extraction de tous les motifs (et donc de générer des motifs inutiles) avant de pouvoir rechercher ceux qui sont intéressants. L'autre difficulté est qu'il est également difficile de savoir quel type de post traitement réaliser. Il devient donc indispensable d'intégrer ces contraintes au plus tôt dans les algorithmes de fouille. Au cours de nos travaux, nous avons déjà analysé certaines contraintes de type temporelles et avons prouvé qu'une telle approche était très efficace. Il serait important de poursuivre ces travaux pour intégrer d'autres types de contraintes spécifiées par l'utilisateur. Considérons, à présent une connaissance du domaine décrite sous la forme d'une ontologie. Par exemple, dans le cas d'un site Web, nous pouvons rattacher à une URL les concepts associés dans l'ontologie et ainsi non seulement nous pouvons obtenir une connaissance plus riche (il ne s'agit plus de pages Web mais de concepts associés à un ensemble de pages) mais nous pouvons élaguer des espaces de recherche qui sont en contradiction avec la connaissance du domaine.

# Bibliographie

- [AFGY02] Jay Ayres, Jason Flannick, Johannes Gehrke, and Tomi Yiu. Sequential pattern mining using a bitmap representation. In *KDD*, pages 429–435, 2002.
- [AGYF02] J. Ayres, J. Gehrke, T. Yiu, and J. Flannick. Sequential pattern mining using bitmaps. In *Proc. of the 8th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining.*, 2002.
- [AIS93a] R. Agrawal, T. Imielinski, and A. Swami. Mining Association Rules between Sets of Items in Large Databases. In *Proc. of the 1993 ACM SIGMOD Conf.*, pages 207–216, Washington DC, USA, May 1993.
- [AIS93b] R. Agrawal, T. Imielinski, and A. N. Swami. Mining Association Rules between Sets of Items in Large Databases. In *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, pages 207–216, 1993.
- [AJS00] R. Agrawal, R. J. Bayardo Jr., and R. Srikant. Athena : Mining-based interactive management of text databases. In *Extending Database Technology*, pages 365–379, 2000.
- [ALB03] H. Albert-Lorincz and J.-F. Boulicaut. Mining Frequent Sequential Patterns under Regular Expressions : a Highly Adaptative Strategy for Pushing Constraints. In *3rd SIAM Int. Conf. on Data Mining (SIAM DM'03)*, pages 316–320, 2003.
- [All90] J. F. Allen. Maintaining Knowledge about Temporal Intervals. *Readings in qualitative reasoning about physical systems*, pages 361–372, 1990.
- [AMS97] K. Ali, S. Manganaris, and R. Srikant. Partial Classification Using Association Rules. In *Knowledge Discovery and Data Mining*, pages 115–118, 1997.
- [AP95] R. Agrawal and G. Psaila. Active Data Mining. In *Proceedings of the 1st International Conference on Knowledge Discovery in Databases and Data Mining*, August 1995.
- [AS95a] R. Agrawal and R. Srikant. Mining sequential patterns. In Philip S. Yu and Arbee L. P. Chen, editors, *Proceedings of the Eleventh International Conference on Data Engineering, March 6-10, 1995, Taipei, Taiwan*, pages 3–14. IEEE Computer Society, 1995.
- [AS95b] R. Agrawal and R. Srikant. Mining Sequential Patterns. In *11th Int. Conf. on Data Engineering*, pages 3–14, 1995.
- [BCG04] E. Baralis, S. Chiusano, and P. Garza. On support thresholds in associative classification. In *Proc. of the 2004 ACM Symposium on Applied Computing (SAC)*, pages 553–558, 2004.

- [BG03] E. Baralis and P. Garza. Majority classification by means of association rules. In *7th European Conf. on Principles and Practice of Knowledge Discovery in Databases (PKDD)*, pages 35–46, 2003.
- [Bur98] C.J.C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2) :121–167, 1998.
- [CB02] B. Cremilleux and J.F. Boulicaut. Simplest rules characterizing classes generated by delta-free sets. In *Proceedings of the 22nd BCS SGAI International Conference on Knowledge Based Systems and Applied Artificial Intelligence ES 2002*, pages 33–46. Springer-Verlag, 2002.
- [CDF<sup>+</sup>01] E. Cohen, M. Datar, S. Fujiwara, A. Gionis, P. Indyk, R. Motwani, J.D. Ullman, and C. Yang. Finding interesting associations without support pruning. *IEEE Trans. Knowl. Data Eng.*, 13(1) :64–78, 2001.
- [Cha90] J. Chauché. Détermination sémantique en analyse structurelle : une expérience basée sur une définition de distance. *TA Information*, 31/1 :17–24, 1990.
- [CHNW96] D.W. Cheung, J. Han, V.T. Ng, and C.Y. Wong. Maintenance of Discovered Association Rules in Large Databases : An Incremental Update Technique. In *Proceedings of the 12th International Conference on Data Engineering (ICDE'96)*, New-Orleans, Louisiana, March 1996.
- [CLK97] D.W. Cheung, S.D. Lee, and B. Kao. A General Incremental Technique for Maintaining Discovered Association Rules. In *Proceedings of the Fifth International Conference on Database Systems for Advanced Applications (DASFA'97)*, Melbourne, Australia, April 1997.
- [CMB02] M. Capelle, C. Masson, and J.-F. Boulicaut. Mining Frequent Sequential Patterns under a Similarity Constraint. In *3rd Int. Conf. on Intelligent Data Engineering and Automated Learning (IDEAL'02)*, pages 1–6, 2002.
- [CTCH01] R.-S. Chen, G.-H. Tzeng, C.-C. Chen, and Y.-C. Hu. Discovery of Fuzzy Sequential Patterns for Fuzzy Partitions in Quantitative Attributes. In *ACS/IEEE International Conference on Computer Systems and Applications (AICCSA)*, pages 144–150, 2001.
- [CYH04] X. Cheng, X. Yan, and J. Han. Incspan : Incremental mining of sequential patterns in large database. In *Proc. of the Intl. Conf. on Knowledge Discovery and Data Mining (KDD 04)*, 2004.
- [dAFGL04] S. de Amo, D. A. Furtado, A. Giacometti, and D. Laurent. An apriori-based approach for first-order temporal pattern mining. In *XIX Simpósio Brasileiro de Bancos de Dados, 18-20 de Outubro, 2004, Brasília, Distrito Federal, Brasil, Anais/Proceedings*, pages 48–62. 2004.
- [DP80] D. Dubois and H. Prade. *Fuzzy Sets and Systems - Theory and Applications*. Academic press, 1980.
- [ea01] V. Kumar et al, editor. *Classification Using Association Rules : Weaknesses and Enhancements*, 2001.
- [FDLT04] C. Fiot, G. Dray, A. Laurent, and M. Teisseire. A la recherche des motifs séquentiels flous. In *12èmes rencontres francophones sur la Logique Floue et ses Applications*, 11 2004.
- [FLT05] C. Fiot, A. Laurent, and M. Teisseire. Contraintes de temps étendues pour les motifs séquentiels. Technical Report 5056, LIRMM, 2005.
- [FWS<sup>+</sup>98] A. Fu, M. Wong, S. Sze, W. Wong, , and W. Yu. Finding Fuzzy Sets for the Mining of Fuzzy Association Rules for Numerical Attributes. In

- the First International Symposium on Intelligent Data Engineering and Learning (IDEAL)*, pages 263–268, 1998.
- [GHP<sup>+</sup>03] G. Giannella, J. Han, J. Pei, X. Yan, and P. Yu. Mining frequent patterns in data streams at multiple time granularities. In *Next Generation Data Mining, MIT Press*, 2003.
- [GRS02] M. Garofalakis, R. Rastogi, and K. Shim. Mining Sequential Patterns with Regular Expression Constraints. *IEEE Transactions on Knowledge and Data Engineering*, 14(3) :530–552, 2002.
- [Had03] H. Haddad. Utilisation des syntagmes nominaux dans un système de recherche d’information. In *Actes des 19 èmes Journées Bases de Données Avancées (BDA’03)*, pages 129–145, 2003.
- [Ham76] H. Hamacher. On logical connectives of fuzzy statements. In *Proceedings of the 3rd European meeting on Cybernetics and Systems Research*, Vienna, 1976.
- [Haw80] D. Hawkins. *Identification of Outliers*. Chapman and Hall, London, 1980.
- [HCTS03] Y.-C. Hu, R.-S. Chen, G.-H. Tzeng, and J.-H. Shieh. A Fuzzy Data Mining Algorithm for Finding Sequential Patterns. *International Journal of Uncertainty Fuzziness Knowledge-Based Systems*, 11(2) :173–193, 2003.
- [HLC01] J.-L. Hsu, C.-C. Liu, and A. L. P. Chen. Discovering nontrivial repeating patterns in music data. *IEEE Transactions on Multimedia*, 3(3) :311–325, 2001.
- [HLW01] T.P. Hong, K.Y. Lin, and S.L. Wang. Mining Fuzzy Sequential Patterns from Multiple-Items Transactions. In *Proceedings of the Joint 9th IFSA World Congress and 20th NAFIPS International Conference*, pages 1317–1321, 2001.
- [HPMa<sup>+</sup>00] J. Han, J. Pei, B. Mortazavi-asl, Q. Chen, U. Dayal, and M. Hsu. Freespan : Frequent pattern-projected sequential pattern mining. In *Proceedings of the 6th International Conference on Knowledge Discovery and Data Mining (KDD 00)*, pages 355–359, Boston, USA, 2000.
- [IS07] F. Ibekwe-SanJuan. *Fouille de textes : méthodes, outils et applications*. 2007.
- [IT95] M. Iwayama and T. Tokunaga. Cluster-based text categorization : a comparison of category search strategies. In *Proc. of SIGIR-95, 18th ACM Int. Conf. on Research and Development in Information Retrieval*, pages 273–281. ACM Press, 1995.
- [JAM03] G. Hubert J. Augé, K. Englmeier and J. Mothe. Catégorisation automatique de textes basée sur des hiérarchies de concepts. In *BDA’03 Journées Bases de données avancées*, pages 69–87, 2003.
- [Joa98] T. Joachims. Text categorization with support vector machines : learning with many relevant features. In *Proc. of ECML-98, 10th European Conf. on Machine Learning*, pages 137–142, Chemnitz, DE, 1998. Springer Verlag, Heidelberg, DE.
- [JTCP03] S. Jailliet, M. Teisseire, J. Chauche, and V Prince. Classification automatique de documents : Le coefficient des deux écarts. In *INFORSID*, pages 87–102, Nancy, 2003.
- [JWB<sup>+</sup>03] D. Janssens, G. Wets, T. Brijs, K. Vanhoof, and G. Chen. Adapting the cba-algorithm by means of intensity of implication. In *Proc. of the First Int. Conf. on Fuzzy Information Processing Theories and Applications*, pages 397–403, 2003.



- [KFW98] C. M. Kuok, A. W.-C. Fu, and M. H. Wong. Mining Fuzzy Association Rules in Databases. *SIGMOD Record*, 27(1) :41–46, 1998.
- [KV02] M. Kantarcioglu and J. Vaidya. An architecture for privacy-preserving mining of client information. In *Proc. of the Workshop on Privacy, Security, and Data Mining in conjunction with the 2002 IEEE ICDM Conf*, 2002.
- [LAS97] B. Lent, R. Agrawal, and R. Srikant. Discovering Trends in Text Databases. In *3rd Int. Conf. on Knowledge Discovery and Data Mining*, pages 227–230. AAAI Press, 14–17 1997.
- [LCK98] S.D. Lee, S.W. Cheun, and B. Kao. Is Sampling Useful in Data Mining? A Case in the Maintenance of Discovered Association Rule. *Data Mining and Knowledge Discovery*, 2 :233–262, 1998.
- [Lee05] C.-H. Lee. An entropy-based approach for generating multi-dimensional sequential patterns. In *PKDD Proceedings*, pages 585–592. 2005.
- [LHM98] B. Liu, W. Hsu, and Y. Ma. Integrating classification and association rule mining. In *Knowledge Discovery and Data Mining*, pages 80–86, 1998.
- [LHP01] W. Li, J. Han, and J. Pei. CMAR : Accurate and Efficient Classification Based on Multiple Class-Association Rules. In *Proc. 2001 Int. Conf. on Data Mining (ICDM'01)*, 2001.
- [LMW00] B. Liu, Y. Ma, and C. Kian Wong. Improving an association rule based classifier. In *Principles of Data Mining and Knowledge Discovery*, pages 504–509, 2000.
- [LRBE03] M. Leleu, C. Rigotti, J.-F. Boulicaut, and G. Euvrard. Constraint-Based Mining of Sequential Patterns over Datasets with Consecutive Repetitions. In *7th Eur. Conf. on Principles and Practice of Knowledge Discovery in Databases (PKDD'03)*, pages 303–314, 2003.
- [LS98] W. Lee and S. Stolfo. Data mining approaches for intrusion detection. In *Proceedings of the 7th USENIX Security Symposium*, San Antonio, TX, 1998.
- [Luk67] J. Lukasiewicz. Many-valued Systems of Propositional Logic, 1967.
- [Mar61] M. Maron. Automatic indexing : An experimental inquiry. *Journal of the ACM (JACM)*, 8 :404–417, 1961.
- [Mas03] F. Masegla. Diviser pour découvrir : une méthode d'analyse du comportement de tous les utilisateurs d'un site web. In *Actes des 19 èmes Journées Bases de Données Avancées (BDA'03)*, pages 227–246, 2003.
- [MCP98] F. Masegla, F. Cathala, and P. Poncelet. The PSP approach for mining sequential patterns. In *Proceedings of the 2nd European Symposium on Principles of Data Mining and Knowledge Discovery (PKDD 98)*, pages 176–184, Nantes, France, 1998.
- [MM02] G. Manku and R. Motwani. Approximate frequency counts over data streams. In *Proc. of VLDB'02 Conference*, 2002.
- [MPT99] F. Masegla, P. Poncelet, and M. Teisseire. Extraction efficace de motifs séquentiels généralisés : le prétraitement des données. In *15 ème Journée Bases de Données Avancées (BDA '99)*, pages 341–360, 1999.
- [MPT00] F. Masegla, P. Poncelet, and M. Teisseire. Web usage mining : How to efficiently manage new transactions and new clients. In *Proceedings of the 4th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD'2000)*, Lyon, France, September 2000.

- [MPT03] F. Massegli, P. Poncelet, and M. Teisseire. Incremental mining of sequential patterns in large databases. *Data and Knowledge Engineering*, 46(1) :97–121, 2003.
- [MPT04] F. Massegli, P. Poncelet, and M. Teisseire. Pre-Processing Time Constraints for Efficiently Mining Generalized Sequential Patterns. In *11th Int. Symp. on Temporal Representation and Reasoning (TIME '04)*, pages 87–95, 2004.
- [MR04] N. Meger and C. Rigotti. Constraint-Based Mining of Episode Rules and Optimal Window Sizes. In *8th Eur. Conf. on Principles and Practice of Knowledge Discovery in Databases (PKDD'04)*, pages 313–324, 2004.
- [MTV97] H. Mannila, H. Toivonen, and A. Inkeri Verkamo. Discovery of Frequent Episodes in Event Sequences. *Data Mining and Knowledge Discovery*, 1(3) :259–289, 1997.
- [oHHS96] United States Dept. of Health & Human Services. Health insurance portability and accountability act of 1996. <http://www.hipaa.org/>, August 1996.
- [PCL<sup>+</sup>05a] M. Plantevit, Y. W. Choong, A. Laurent, D. Laurent, and M. Teisseire. M<sup>2</sup>SP : Mining sequential patterns among several dimensions. In *PKDD*, pages 205–216. 2005.
- [PCL<sup>+</sup>05b] M. Plantevit, Y. W. Choong, A. Laurent, D. Laurent, and M. Teisseire. Motifs séquentiels multidimensionnels étoilés. In Véronique Benzaken, editor, *BDA 2005, Actes des 21<sup>es</sup> journées de Bases de données avancées, Saint-Malo, 17-20 octobre 2005*, pages 163–182. 2005.
- [PHMa<sup>+</sup>01] J. Pei, J. Han, B. Mortazavi-asl, H. Pinto, Q. Chen, and U. Dayal. Prefixspan : Mining sequential patterns efficiently by prefix-projected pattern growth. In *Proceedings of 17th International Conference on Data Engineering (ICDE 01)*, pages 215–224, Heidelberg, Germany, 2001.
- [PHP<sup>+</sup>01] H. Pinto, J. Han, J. Pei, K. Wang, Q. Chen, and U. Dayal. Multi-dimensional sequential pattern mining. In *Proceedings of the 2001 ACM CIKM International Conference on Information and Knowledge Management, Atlanta, Georgia, USA, November 5-10, 2001*, pages 81–88. ACM, 2001.
- [PHW02] J. Pei, J. Han, and W. Wang. Mining sequential patterns with constraints in large databases. In *Proceedings of the 10th International Conference on Information and Knowledge Management (CIKM 02)*, pages 18–25, MCLean, USA, 2002.
- [PLT06] M. Plantevit, A. Laurent, and M. Teisseire. HYPE : Prise en compte des hiérarchies lors de l'extraction de motifs séquentiels multidimensionnels. In *EDA 2006, Actes de la deuxième journée francophone sur les Entrepôts de Données et l'Analyse en ligne, Versailles, 19 juin 2006*. Cépaduès, 2006.
- [PZOD99] S. Parthasarathy, M. Zaki, M. Orihara, and S. Dwarkadas. Incremental and interactive sequence mining. In *Proc. of the 8th Intl. Conf. on Information and Knowledge Management (CIKM 99)*, 1999.
- [RMR96] C. P. Rainsford, M. K. Mohania, and J. F. Roddick. Incremental Maintenance Techniques for Discovered Classification Rules. In *Proceedings of the International Symposium on Cooperative Database Systems for Advanced Applications*, pages 302–305, Kyoto, Japan, 1996.



- [RMR97] C. Rainsford, M. K. Mohania, and J.F. Roddick. A Temporal Windowing Approach to the Incremental Maintenance of Association Rules. In *Proceedings of the Eighth International Database Workshop, Data Mining, Data Warehousing and Client/Server Databases (IDW'97)*, pages 78–94, Hong Kong, Fong,, 1997.
- [RMR06] C. P. Rainsford, M. K. Mohania, and J. F. Roddick. SPEED : Mining Maximal Sequential Patterns over Data Streams. In *Proceedings of the 3rd IEEE International Conference on Intelligent Systems (IEEE IS 2006)*, London, UK, 2006.
- [Roc71] J. Rocchio. Relevence feedback in information retrieval. In *in the SMART Retrieval System : Experiments in Automatic Document Processing*, pages 313–323, 1971.
- [RPT05] C. Raïssi, P. Poncelet, and M. Teisseire. Need for SPEED : Mining Sequential Patterns in Data Streams. In *21 ème Journée Bases de Données Avancées (BDA '05)*, 2005.
- [RPT07] C. Raïssi, P. Poncelet, and M. Teisseire. Towards a New Approach for Mining Maximal Frequent Itemsets over Data Streams. *Journal of Intelligent Information Systems*, to appear, 2007.
- [SA96a] R. Srikant and R. Agrawal. Mining Quantitative Association Rules in Large Relational Tables. In *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*, pages 1–12, Montreal, Quebec, Canada, 4–6 1996.
- [SA96b] R. Srikant and R. Agrawal. Mining Sequential Patterns : Generalizations and Performance Improvements. In *Proceedings of the 5th International Conference on Extending Database Technology (EDBT'96)*, pages 3–17, Avignon, France, 9 1996.
- [SA96c] R. Srikant and R. Agrawal. Mining Sequential Patterns : Generalizations and Performance Improvements. In *5th Int. Conf. on Extending Database Technology (EDBT '96)*, pages 3–17, 1996.
- [Sal71] G. Salton. The smart retrieval system – experiments in automatic document processing, 1971.
- [Sch94] H. Schmid. Probabilistic part-of-speech tagging using decision trees. In *International Conference on New Methods in Language Processing*, Manchester, UK, 1994.
- [Seb02] F. Sebastiani. Machine learning in automated text categorisation. In *Proc. of ACM Computing Surveys*, volume 34, pages 1–47, 2002.
- [SM83] G. Salton and M. J. McGill. *Introduction to modern information retrieval*. 1983.
- [SS98] N.L. Sarda and N. V. Srinivas. An Adaptive Algorithm for Incremental Mining of Association Rules. In *Proceedings of the 9th International Workshop on Database and Expert Systems Applications*, Indian Institute of Technology Bombay, 1998.
- [SV95] N. SVladimir and V. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 1995.
- [SYY75] G. Salton, C. Yang, and C. Yu. A theory of term importance in automatic text analysis. *Journal of the American Society for Information Science*, 36 :33–44, 1975.

- [SZ05] Einoshin Suzuki and Jan M. Zytkow. Unified algorithm for undirected discovery of exception rules. *International Journal of Intelligent Systems*, 20(7) :673–691, 2005.
- [TBAR97] S. Thomas, S. Bodagala, K. Alsabti, and S. Ranka. An Efficient Algorithm for the Incremental Updation of Association Rules in Large Databases. In *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining (KDD '97)*, Newport Beach, California, August 1997.
- [TTM04] D. Tanasa, B. Trousse, and F. Maseglia. *Mesures de l'internet*, chapter Fouille de données appliquées au logs web : état de l'art sur le Web Usage Mining, pages 126–143. édition Les Canadiens en Europe, 2004.
- [Vap95] V. Vapnik. *The Nature Of Statistical Learnig Theory*. Springer, 1995.
- [VC64] V. Vapnik and A. Chervonenkis. A note on one class of perceptrons. *Automatic and Remote Control*, 25, 1964.
- [WCF<sup>+</sup>00] P.C. Wong, W. Cowley, H. Foote, E. Jurrus, and J. Thomas. Visualizing sequential patterns for text mining. In *INFOVIS*, pages 105–, 2000.
- [WdlIJ<sup>+</sup>02] J.-J. Wesselink, B. de la Iglesia, S. A. James, J. L. Dicks, I. N. Roberts, and V. J. Rayward-Smith. Determining a unique defining dna sequence for yeast species using hashing techniques. *Bioinformatics*, 18(7) :1004–1010, 2002.
- [Web83] S. Weber. A General Concept of Fuzzy Connectives. *Fuzzy Sets and Systems*, 11(2) :115–134, 1983.
- [WH04] J. Wang and J. Han. Bide : Efficient mining of frequent closed sequences. In *Proceedings of the International Conference on Data Engineering (ICEDE 04)*, Boston, M.A., 2004.
- [WZH00] K. Wang, S. Zhou, and Y. He. Growing decision trees on support-less association rules. In *Knowledge Discovery and Data Mining*, pages 265–269, 2000.
- [XZ03] X.Wu and S. Zhang. Synthesizing high-frequency rules from different data sources. *IEEE Trans. on Knowledge and Data Engineering*, 15(2) :353–367, 2003.
- [Yan99] Y. Yang. An evaluation of statistical approaches to text categorization. *Information Retrieval*, 1(1/2) :69–90, 1999.
- [YC05] C.-C. Yu and Y.-L. Chen. Mining sequential patterns from multidimensional sequence data. *IEEE Transactions on Knowledge and Data Engineering*, 17(1) :pp. 136–140, 2005.
- [YHA03] X. Yan, J. Han, and R. Afshar. Clospan : Mining closed sequential patterns in large databases. In *Proceedings of the SDM 03 Conference*, San Francisco, CA, 2003.
- [YL99] Y. Yang and X. Liu. A re-examination of text categorization methods. In *22nd Annual International SIGIR*, pages 42–49, Berkley, August 1999.
- [YWY04] J. Yang, W. Wang, and P. S. Yu. Mining surprising periodic patterns. *Data Mining Knowledge Discovery*, 9(2) :189–216, 2004.
- [Zad65] L.A. Zadeh. Fuzzy sets. *Information and Control*, 3(8) :338–353, 1965.
- [Zak00] M. J. Zaki. Sequence Mining in Categorical Domains : Incorporating Constraints. In *9th Int. Conf. on Information and Knowledge Management (CIKM '00)*, pages 422–429, 2000.

- [Zak01] M. J. Zaki. SPADE : An efficient algorithm for mining frequent sequences. *Machine Learning*, 42(1/2) :31–60, 2001.
- [ZYO99] N. Zhong, Y. Yao, , and S. Ohsuga. Peculiarity oriented multi-database mining. In *Proc. of PKDD 99*, pages 136–146, 1999.