Université de Tunis El Manar
-Faculté des Sciences de Tunis-

Université Montpellier 2
-Sciences Et techniques Du Languedoc-

Thèse

En cotutelle

Pour obtenir le grade de

Docteur de l'Université Montpellier II

Discipline	: Informatique
Spécialité Doctorale	: *Informatique*
École doctorale	: *Information, Structure, Systèmes*

et de

Docteur de l'Université Tunis El Manar

Discipline	: Informatique
Spécialité Doctorale	: *Informatique*
École doctorale	:*Mathématiques, Informatique, Sciences et Technologie des Matériaux*

présentée et soutenue publiquement le 09 Mai 2012 par

Sarra Ayouni

Etude et Extraction de Règles graduelles floues : Définition d'algorithmes efficaces

Jury

Maria Rifqi, Maitre de Conférences, Université Pierre et Marie Curie, LIP6, ... Rapporteur
Zied Elouedi, Maitre de Conférences, Institut Supérieur de Gestion de Tunis, ... Rapporteur
Henri Prade, Directeur de recherche CNRS, Université Paul Sébastien, IRIT ..... Examinateur
Habib Ounelli, Professeur, Faculté des Sciences de Tunis, URPAH, ... Examinateur
Pascal Poncelet, Professeur, Université Montpellier2, Lirmm, ... Directeur de thèse
Sadok Ben Yahia, Maitre de conférences, Faculté des Sciences de Tunis, URPAH, ... Directeur de thèse

Laboratoire Lirmm-CNRS, Montpellier

Laboratoire Lpah-FST

Remerciements

Je présente toute ma reconnaissance à Madame. Maria Rifqi, Maitre de conférences à l'université Pierre et Marie Curie, et à Monsieur Zied Elouedi, Maitre de conférences à l'Institut Supérieur de Gestion de Tunis, pour avoir accepté d'être rapporteurs de ce travail.

Je tiens également à remercier Monsieur Habib Ounelli, Professeur à la Faculté des Sciences de Tunis, d'avoir accepté de présider le jury de cette thèse.

Merci au Professeur Henri Prade, Directeur de recherche CNRS de l'Université Paul Sébastien, de m'avoir fait l'honneur de participer à ce jury de thèse.

Il m'est impossible d'exprimer toute ma gratitude à Madame Anne Laurent, Professeur à l'Université Montpellier 2, qui a su diriger ce travail de thèse avec beaucoup de patience et de rigueur scientifique. Je la remercie pour sa disponibilité, pour ses conseils avisés et pour son investissement constant.

Je remercie mes directeurs de thèse, Monsieur Sadok Ben Yahia, Maitre de conférences à la Faculté des Sciences de Tunis et Monsieur Pascal Poncelet, Professeur à l'Université Montpellier 2. Cette thèse ne serait pas ce qu'elle est sans l'aide de Sadok. Je le remercie pour son soutien et sa disponibilité. Je le remercie également pour sa compétence et ses recommandations toujours appropriées ainsi que pour ses qualités humaines. Je remercie également Pascal pour sa sympathie et pour m'avoir accepté dans son équipe.

Je remercie tous les membres du Laboratoire Lirmm qui m'ont toujours chaleureusement accueilli pendant ces années de thèse. Je remercie plus particulièrement Nicolas Serrurier qui s'est occupé de beaucoup d'aspects administratifs de cette thèse.

Je tiens à exprimer toute ma gratitude et mon amitié à Amel Ghouila, pour son aide précieuse et sa présence incontournable lors de mes séjours à Montpellier.

Je remercie toutes les personnes avec qui j'ai partagé mes études et notamment ces années de thèse : Leila, Tassadit, Ghada, Mohamed Ali,Tarek, Slim, Moez, Islem, Narjes, Karima, Ines, Lisa, Pattataporn, Malaquias.

L'aboutissement de cette thèse aurait été plus difficile sans le soutien bienveillant et chaleureux de ma famille. Je remercie mes parents, ma soeur et mes frères pour leur soutien qui ne m'a jamais fait défaut. Je remercie mon beau père et mon beau frère pour leur présence quotidienne.

Il y a cependant des personnes qui n'auront plus jamais l'occasion de me lire. Je voudrais à travers ce travail leur rendre hommage. J'ai le profond regret de citer : ma belle mère, ma tante et mon grand-père.

Je garde enfin mes remerciements amoureux pour mon époux Mohamed pour son soutien et son encouragement au quotidien durant ces années de thèse.

À mon trésor Mohamed Ali.

Table des matières

Table des figures

Liste des tableaux

Introduction Générale

Avec le développement des outils informatiques, nous assistons ces dernières années à un accroissement considérable de la quantité d'informations stockées dans de grandes bases de données scientifiques, économiques, financières, médicales, etc. Le besoin d'interpréter et d'analyser ces grandes masses de données est devenu crucial au vu de l'incapacité des outils existants à répondre aux nouveaux besoins des utilisateurs, en occurrence l'extraction d'une connaissance cachée dans ces gisements de données [Han et Kamber, 2000]. Ainsi, la mise au point de nouvelles techniques d'analyse est devenue un réel défi pour la communauté scientifique. Pour répondre à cette pénurie de connaissances sur les données, de nouvelles méthodes d'extraction de l'information ont vu le jour, regroupées sous le terme générique de *fouille de données* [Berry et Linoff., 2004].
La fouille de données est un domaine de recherche en plein essor visant à exploiter les grandes quantités de données collectées chaque jour dans divers domaines d'application de l'informatique. Ce domaine pluri-disciplinaire se situe au confluent de différents domaines, tels que les statistiques, les bases de données, l'algorithmique, les mathématiques, l'intelligence artificielle, etc [Salleb, 2003]. Les techniques de la fouille de données sont utilisées dans un processus appelé *Extraction de Connaissances dans les Bases de Données* (**ECD**) (ou *Knowledge Discovery in Databases*). Selon Frawley *et al.* [Frawley *et al.*, 1992], l'**ECD** désigne le processus interactif et itératif non trivial d'extraction de connaissances implicites, précédemment inconnues et potentiellement utiles à partir de données stockées dans les bases de données.

Dans ce chapitre, nous commençons par présenter le processus d'extraction de connaissances (**ECD**) qui constitue le cadre général dont lequel s'inscrit notre travail. Dans la section 0.2, nous passons en revue les différents types de motifs pouvant être extraits dans le cadre de l'extraction de règles d'association. Nous présentons dans la section 0.3 nos motivations et objectifs. Enfin, dans la section 0.4, nous détaillons l'organisation de ce mémoire.

1 Processus d'extraction de connaissance et fouille de données

Le processus d'extraction de connaissances dans les bases de données a été définie comme "*le processus non trivial d'extraction d'informations valides, nouvelles, potentiellement utiles, et compréhensibles à partir d'un ensemble de données*" [Fayyad *et al.*, 1996]. Ainsi, l'*ECD* est un processus itératif de recherche de modèles ou de règles implicites et valides pouvant éclairer les décisions et choix des utilisateurs. Ce domaine de recherche a commencé à être distingué en 1989, quand G. Piatetsky-Shapiro a organisé la première réunion de chercheurs et d'utilisateurs sur l'extraction automatique de connaissance dans les grandes bases de données [Bastide, 2000].

Ce processus semi-automatique est constitué de plusieurs étapes comme le montre la figure 1, allant de la sélection et de la préparation des données jusqu'à l'interprétation et l'évaluation des résultats en passant par la phase de la fouille de données. Dans le cadre de cette thèse, nous nous intéressons à cette phase de l'*ECD* (*i.e.*,la fouille de données).

L'idée sous-jacente de la fouille de données est d'extraire les connaissances cachées à partir d'un ensemble de données. La fouille de données regroupe un certain nombre de tâches, telles que la prédiction, le regroupement par similitude, la classification, la découverte d'associations, etc [Berry et Linoff., 2004]. L'un des plus importants problèmes de la fouille de données est la recherche de règles d'association.

Le problème d'extraction de règles d'association introduit par Agrawal et *al.* [Agrawal *et al.*, 1993], fut développé pour l'analyse de bases de données de transactions de ventes. Chaque transaction est constituée d'une liste d'articles achetés, afin d'identifier les groupes d'articles achetés le plus fréquemment ensemble [Pasquier, 2000]. La vocation de l'extraction de règles d'association est d'identifier des corrélations cachées, potentiellement utiles, entre les attributs d'une base de données.

Le problème d'extraction de règles d'association a été décomposé en deux sous-problèmes [Pasquier, 2000]. Le premier consiste à identifier l'ensemble de motifs fréquents (*i.e.*, groupes d'attributs qui apparaissent fréquemment) à partir d'une base de données. Le second sous-problème consiste à générer les règles d'association à partir de ces motifs selon des critères fixés au préalable par l'utilisateur. En effet, l'utilisateur intervient dans l'extraction des motifs et des règles d'association, en précisant généralement des contraintes sur les motifs ou règles à prendre en compte. La première contrainte est le support minimal qui signifie le pourcentage à partir duquel les découvertes sont significatives [Bastide, 2000]. Ainsi, un motif est fréquent si sa valeur de support est supérieure ou égale à un support minimal (*minSup*).
La deuxième contrainte, généralement utilisée pour les règles d'association, est la confiance

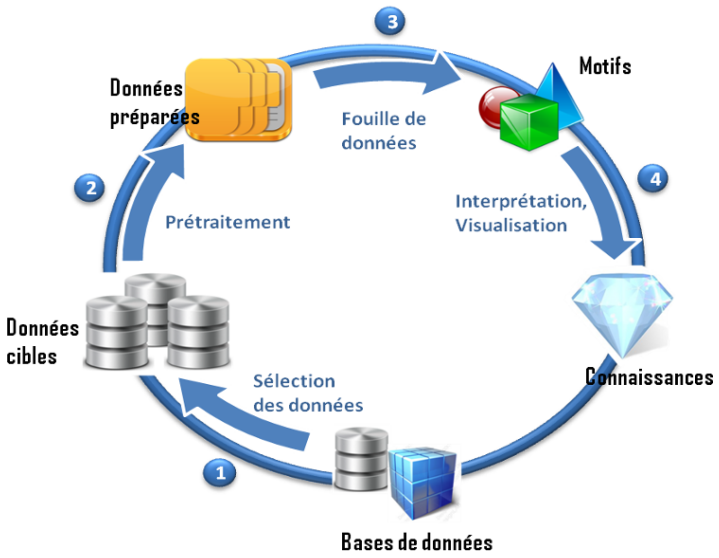

FIGURE 1 – Processus d'extraction de connaissances.

minimale (*minConf*), qui exprime le pourcentage de validité de la règle. Toutefois, d'autres contraintes peuvent être envisagées et qui reposent généralement sur l'expertise de l'utilisateur du domaine en question.

Depuis l'introduction du problème d'extraction de règles d'association dans les années 90 [Agrawal *et al.*, 1993], plusieurs approches furent développées pour extraire de manière efficace tous les motifs fréquents à partir d'une base de données. Deux principaux paradigmes de méthodes de parcours de l'espace de recherche et donc d'extraction de motifs fréquents sont à distinguer : les méthodes "*Générer-élaguer*" dont le parcours est en largeur et les méthodes "*Diviser-générer*" dont le parcours est en profondeur. Nous nous intéressons dans le cadre de cette thèse au premier paradigme (*i.e.*, *Générer-élaguer*), dont le principe est le suivant [Agrawal et Srikant, 1994] :

- Génération des motifs fréquents de taille 1,
- Génération des motifs candidats de taille $n+1$ par auto-jointure des motifs de taille n,
- Comptage de la fréquence des motifs candidats et élagage des non fréquents par rapport à *minSup*,
- Arrêt lorsqu'il n'y a plus de motifs candidats.

Toutes les méthodes d'extraction de motifs fréquents profitent d'une propriété liée à la contrainte de la fréquence minimale (*minSup*) afin de limiter l'espace de recherche. Cette propriété est la contrainte d'anti-monotonicité définie comme suit [Mannila et Toivonen, 1997].

Propriété 1 *Soit* $\mathbb{C} : \mathcal{P}(\mathcal{I}) \rightarrow \{vrai, faux\}$*, une contrainte (i.e., un prédicat de sélection) et* $\mathcal{P}(\mathcal{I})$ *l'ensemble des parties de* $\mathcal{I}$ *(i.e., l'ensemble de tous les sous-ensembles de* $\mathcal{I}$*) avec* $\mathcal{I}$ *est un ensemble d'attributs ou items.* $\mathbb{C}$ *est une contrainte anti-monotone si et seulement si :*

$$\forall (I, J) \in \mathcal{P}(\mathcal{I}) : \mathbb{C}(I) \wedge J \subset I \Rightarrow \mathbb{C}(J)$$

En particulier $\mathbb{C}$ est une propriété liée à la contrainte de la fréquence minimale (*minSup*). Ceci signifie que si nous avons deux motifs I_1 et I_2 avec $I_1 \subseteq I_2$, alors $Support(I_1) \geq Support(I_2)$. Cette propriété est importante dans les algorithmes d'extraction de motifs fréquents. En effet, si un motif de taille n n'est pas fréquent alors aucun de ces sur-motifs ne le sera. Ceci permet de limiter l'espace de recherche en ignorant les sur-motifs d'un motifs non fréquent.

Toutefois, deux problèmes majeurs relatifs à l'extraction de motifs fréquents ont donné lieu à de nombreuses pistes de recherche : le problème du temps d'extraction de ces motifs à partir des jeux de données (surtout lorsqu'ils sont denses ou avec une valeur de *minSup* basse) et le problème de leur nombre, tellement important que les experts peinent à les exploiter ce qui remet en cause la pertinence des règles d'association extraites à partir de ces motifs. Pour pallier ce problème, plusieurs travaux ont été proposés en introduisant la notion de représentation concise des motifs fréquents.

2 Types de motifs extraits

Dans la littérature, plusieurs approches et algorithmes ont été élaborés afin d'extraire les motifs et les règles d'association. Plusieurs types de motifs ont été définis selon le type de corrélation à extraire et selon la nature des données à partir desquelles l'extraction est faite. En effet, les motifs extraits à partir de données binaires diffèrent en la structure et en la technique d'extraction de ceux extraits à partir de données quantitatives. De même, les motifs décrivant une corrélation entre les attributs diffèrent de ceux décrivant une corrélation entre les variations des attributs et de ceux qui intègrent des contraintes temporelles (*i.e.*, les motifs séquentiels) [Masseglia, 2002]. Nous pouvons donc classer les motifs extraits selon deux axes à savoir : le type de la base de données à partir de laquelle l'extraction est faite et le type de corrélation décrite par le motif. Dans ce qui suit, nous allons énumérer les différents types de contextes d'extraction et les types de corrélations décrites par les motifs. Un contexte d'extraction est la base de données à partir de la quelle les motifs sont extraits. Ces contextes diffèrent par le type de données qu'ils décrivent.

– **Base de données binaires** :

  Un contexte d'extraction binaire est un contexte décrivant des données binaires, *i.e.,*

présence ou absence de l'attribut. Un exemple d'un tel contexte est illustré par la figure 2.

	Item1	Item2	Item3
C1	×		×
C2	×	×	×
C3		×	×
C4	×	×	×
C5	×		×

FIGURE 2 – Exemple d'un contexte avec données binaires

L'extraction de motifs ou itemsets fréquents à partir de bases de données binaires est une problématique, qui a suscité l'intérêt de la plupart des travaux de recherche sur les règles d'association. En effet, plusieurs travaux se sont concentrés sur l'amélioration des performances de ce problème. Toutefois, plusieurs de ces approches de recherche de motifs fréquents suivent le principe de l'algorithme pionnier *Apriori* [Agrawal et Srikant, 1994]. En effet, il s'agit d'une recherche nivelée : rechercher d'abord les plus petits motifs fréquents, puis ceux un peu plus grands, et ainsi de suite jusqu'à avoir trouvé les plus grands qui peuvent exister. Cet algorithme repose sur une propriété essentielle qui est la suivante : un motif fréquent n'a pas de sous-ensemble non fréquent (et réciproquement un motif non fréquent n'a pas de sur-ensemble fréquent).

– **Base de données quantitatives** :
La plupart des contextes réels ne se limitent pas à des données binaires mais ils contiennent aussi des données quantitatives (*e.g.* numériques, catégoriels). La figure 3 illustre un exemple de base de données quantitatives.

	Item1	Item2	Item3
C1	22	1200	0
C2	34	2100	1
C3	27	19000	0
C4	47	2300	2
C5	54	2000	2

FIGURE 3 – Exemple d'un contexte avec données quantitatives

Comme les algorithmes classiques d'extraction de motifs fréquents s'avéraient inadaptés pour traiter ce type de données, plusieurs approches [BenYahia et Jaoua, 2000, Chan et Au, 1997a, Chan et Au, 1997b, Chen *et al.*, 2000, Delgado *et al.*, 2003, Gyenesei, 2000, Jaoua *et al.*, 2000, Kuok *et al.*, 1998, Srikant et Agrawal, 1996] ont été proposées pour extraire des motifs "quantitatifs" ou "flous" à partir de telles bases de données.

– **Base de données temporelles** :
Dans ces bases, les données sont associées à un attribut temporel, comme pourrait l'illustrer la figure 4. Les motifs ainsi extraits sont des motifs périodiques, épisodes ou motifs séquentiels.

	Date1	Date2	Date3
C1	Item2 Item1 Item3	Item1	Item2 Item3
C2		Item3 Item1 Item2	Item1 Item2
C3	Item2 Item1	Item2	Item2 Item1 Item3

FIGURE 4 – Exemple d'un contexte avec données temporelles

La problématique d'extraction de motifs séquentiels [1] a été étudiée dans plusieurs travaux de recherches [R. Srikant ant R. Agrawal, , Ayres *et al.*, 2002, Fiot *et al.*, 2007, Fiot *et al.*, 2008a, Zhao et Bhowmick, 2003].

À part le type de contexte d'extraction, les motifs extraits diffèrent en la corrélation qu'ils pourraient décrire. En effet, les motifs fréquents *classiques* expriment une corrélation entre items (*i.e.*, items qui co-occurent ensemble). Cependant, d'autres types de corrélation peuvent exister telles que les co-variations des valeurs des attributs ou encore les corrélations temporelles des valeurs des attributs. Ces derniers correspondent à des motifs dits *séquentiels*, *i.e.*, ceux exprimant les co-variations des valeurs des attributs correspondent à des motifs dits *graduels*.

3 Objectifs et contributions

Le travail de cette thèse présente différentes contributions liées à deux principales problématiques. La première concerne l'extraction d'une représentation condensée de motifs flous à partir de données quantitatives et la deuxième concerne l'extraction de motifs graduels exprimant des corrélations de co-variations des valeurs des attributs. En effet, ce type de connaissance désigné par *motif graduel* a récemment émergé dans la communauté de la fouille de données [Berzal *et al.*, 2007, Di Jorio *et al.*, 2008, Hüllermeier, 2002].

Ainsi, dans cette perspective, nos contributions sont les suivantes :

1. Un motif séquentiel est de la forme $< \{item1, item2\}\{item1, item3\}\{item4\} > x\%$ où $\{item1, item2\}$ $\{item1, item3\}$ $\{item4\}$.

- **Formalisation des motifs flous clos** :

  Plusieurs approches ont été proposées afin d'extraire des motifs flous à partir de contextes quantitatifs. Néanmoins, toutes ces approches génèrent un nombre très élevé de motifs flous. Pour pallier ce problème, nous proposons d'extraire une représentation condensée des motifs flous basée sur la notion de cloture de la correspondance de Galois. Ainsi, nous proposons une nouvelle définition des opérateurs de la correspondance de Galois pour les motifs flous clos.

- **Formalisation des motifs graduels clos** :

  Bien que des algorithmes d'extraction de motifs graduels, de plus en plus efficaces [Di Jorio *et al.*, 2008, Di Jorio *et al.*, 2009b], soient proposés dans des travaux récents, il n'en reste pas moins que ces méthodes génèrent un nombre de motifs tellement important que les experts peinent à les exploiter. Nous proposons donc de minimiser ce nombre de motifs, qui pourrait être énorme dans certains contextes, sans avoir recours à une perte d'information. Nous proposons ainsi une représentation condensée des motifs graduels en introduisant des concepts théoriques nouveaux associés aux opérateurs de cloture sur de tels motifs. Cette proposition est validée par des expérimentations montrant le gain apporté par la prise en compte de cette représentation condensée.

- **Formalisation des motifs graduels flous** :

  La logique floue a montré son efficacité dans le traitement de données réelles. En effet, il est souvent le cas que l'information n'est pas contenue aux limites supérieures et inférieures des valeurs numériques que peut prendre un attribut (*min/max*), mais plutôt cachés entre ces deux extrémités. Par exemple, un motif intéressant pourrait être *"Plus l'âge d'un employé est proche de 46 ans, plus son salaire est élevé"*. De même, un tel motif graduel serait plus proche à la perception humaine et par suite beaucoup plus compréhensible. Nous proposons d'extraire des motifs graduels flous à partir de données numériques.

- **Construction de fonction d'appartenance** :

  L'extraction de motifs graduels flous nécessitent la fuzzification du contexte d'extraction. Plusieurs méthodes ont été proposées afin de fuzzifier des données numériques. Nous introduisons dans cette contribution deux méthodes de fuzzification et nous démontrons le gain de notre approche en nombre de motifs graduels par rapport à ceux extraits par les méthodes classiques.

- **Extraction de représentations condensées** :

  La principale faiblesse des algorithmes d'extraction de règles d'association est le problème

de l'utilité et de la pertinence des règles d'association extraites. En effet, dans la plupart des cas, les jeux de données réels conduisent à plusieurs milliers voire plusieurs millions de règles d'association dont la mesure de confiance est élevée, et parmi lesquelles se trouvent de nombreuses règles redondantes [Pasquier, 2000]. Afin d'éviter ce problème, nous proposons d'extraire un ensemble générateur de toutes les règles avec leurs valeurs de confiance et support. Ainsi, nous définissons des représentations condensées basées sur les motifs flous clos et graduels clos.

La validation de nos contributions est matérialisée par un ensemble d'expérimentations menées sur différents jeux de données.

4 Structure du document

Les résultats de nos travaux de recherche sont synthétisés dans ce mémoire, qui est composé d'une introduction générale, 6 chapitres et une conclusion générale.

Le premier chapitre introductif, présente le cadre général dans lequel s'inscrivent les travaux de cette thèse. Les trois chapitres, qui suivent, présentent un aspect bibliographique de notions que nous allons aborder tout le long de cette thèse. Ainsi, le deuxième chapitre présente les fondements mathématiques de la théorie de la logique floue et des ensembles flous. Il présente aussi les différentes méthodes dédiées à l'acquisition des fonctions d'appartenance des sous-ensembles flous.

Le troisième chapitre introduit les notions de motifs et de règles graduels. Une panoplie de propositions et d'approches d'extraction de motifs graduels est également examinée dans ce chapitre.

Le quatrième chapitre présente un tour d'horizon sur l'extraction de représentations condensées dans le cadre d'extraction des itemsets fréquents.

Les trois chapitres, qui suivent (i.e., cinquième, sixième et septième chapitres), présentent l'ensemble de nos contributions sur l'extraction de motifs et règles graduels et flous. Ainsi, le cinquième chapitre est divisé en deux sections. Dans la première section, nous proposons d'extraire des motifs flous clos en se basant sur la notion de cloture de la correspondance de Galois. Ainsi, nous définissons une nouvelle correspondance de Galois dans le cadre des motifs flous. Dans la deuxième section, nous introduisons la notion d'ensemble de séquences et les différentes opérations pouvant être effectuées sur ces ensembles. En se basant sur cette notion, nous défi-

nissons une nouvelle correspondance de Galois pour les motifs graduels clos. Nous démontrons la validité de nos opérateurs de la correspondance de Galois graduelle et la compacité des motifs graduels clos par rapport aux motifs graduels fréquents.

Le sixième chapitre sera dédié à l'extraction de motifs graduels flous. En effet, nous proposons deux approches permettant d'obtenir les modalités floues permettant d'extraire les motifs graduels flous. La première approche est basée sur la médiane des valeurs des attributs alors que la deuxième approche est basée sur les algorithmes génétiques.

Dans le septième chapitre, nous introduisons la notion de représentation condensée pour les règles graduelles/floues. En effet, nous définissons des représentations condensées pour les règles graduelles/floues *exactes*, *approximatives* et *transitives*. Nous proposons également un système d'inférence permettant de déduire, d'une manière informative, toutes les règles graduelles/floues redondantes.

Enfin, nous terminons le présent mémoire par une conclusion générale dans laquelle nous résumons l'ensemble de nos travaux et nous présentons quelques perspectives de recherche.

Chapitre 1

Notions de base de la logique floue

Sommaire

1.1 Introduction

Le raisonnement humain est basé sur des données imprécises ou incomplètes. En effet, il est aisé pour un être humain de déterminer si une personne est de *petite* ou de *grande* taille sans pour autant connaître sa taille exacte. Un ordinateur, lui est basé sur la logique classique traitant des données exactes. L'idée de la logique floue, introduite en 1965 par Zadeh [Zadeh, 1965], est de transmettre cette faculté du raisonnement humain et de faire accepter des données imprécises, à un ordinateur. Zadeh, disait qu'« *un contrôleur électromécanique doté d'un raisonnement humain serait plus performant qu'un contrôleur classique* » [Zadeh, 1996].
La notion de sous-ensemble flou a pour but de permettre des gradations dans l'appartenance

d'un élément à une classe [Zadeh, 1977], c'est-à-dire d'autoriser un élément à appartenir plus ou moins fortement à une classe. Cette notion permet l'utilisation de catégories aux limites mal définies (comme "vieux" ou "adulte"), de situations intermédiaires entre le tout et le rien ("presque vrai"), le passage progressif d'une propriété à une autre ("tiède" à "chaud" selon la température) et l'utilisation de valeurs approximatives ("environ 12 ans").

Étant donné un ensemble de référence U, appelé *Univers de discours*, nous pouvons indiquer les éléments de U appartenant à une certaine classe de U et ceux qui n'y appartiennent pas. Cette classe est un sous-ensemble classique de U. En revanche, si l'appartenance de certains éléments de U à une classe n'est pas absolue, nous pouvons indiquer avec quel degré chaque élément appartient à cette classe. Celle-ci est un sous-ensemble flou de U.

Dans ce chapitre, nous allons présenter les fondements mathématiques de la théorie des sous-ensembles flous et les méthodes et approches proposées pour l'acquisition des fonctions d'appartenance d'un ensemble flou.

1.2 Sous-ensembles flous

Dans cette sous-section, nous allons présenter les notions de base relatives à la théorie des sous-ensembles flous [Bouchon-Meunier, 1995, Dubois et Prade, , Dubois, 1991].

Définition 1 *Un sous-ensemble ordinaire (ou classique) A inclus dans U est défini par la fonction caractéristique $\mu_A : U \rightarrow \{0,1\}$. Un élément $x \in U$ est un élément de A si et seulement si : $\mu_A(x) = 1$. Un élément $x_1 \in U$ n'est pas un élément de A si et seulement si : $\mu_A(x_1) = 0$.*

Définition 2 *Un sous-ensemble flou $\tilde{A}$ inclus dans U est défini par la fonction d'appartenance $\mu_{\tilde{A}} : U \rightarrow [0,1]$, où $\mu_{\tilde{A}}(x)$ désigne le degré avec lequel un élément $x \in U$ est un élément de $\tilde{A}$. L'ensemble flou $\tilde{A}$ est dénoté par :*

$$\tilde{\mathrm{A}} = \{\ x_1^{\mu_{\tilde{A}}(x_1)}, x_2^{\mu_{\tilde{A}}(x_2)}, x_3^{\mu_{\tilde{A}}(x_3)}, ..., x_n^{\mu_{\tilde{A}}(x_n)}\}$$

Exemple 1 *Soit U= {a, b, c}. L'ensemble $\tilde{A} = \{a^{0,5}, b^{0,1}, c^{0,9}\}$ est un ensemble flou. Les degrés d'appartenance de a, b et c dans $\tilde{A}$ sont, respectivement, 0,5 ; 0,1 et 0,9.*

1.2.1 Opérations sur les sous-ensembles flous

Parmi les opérations fondamentales de la théorie des sous-ensembles flous, nous avons retenu l'inclusion, l'intersection et l'union.

Définition 3 *Un sous-ensemble flou $\tilde{A} \in U$ est* ***inclus*** *dans un autre sous-ensemble flou $\tilde{B} \in U$ ($\tilde{A} \subseteq \tilde{B}$) si et seulement si tout élément x de U qui appartient à $\tilde{A}$ appartient aussi à $\tilde{B}$ avec un degré au moins aussi grand, i.e.,*

$$\forall\ x \in U,\ \mu_{\tilde{A}}(x) \leq \mu_{\tilde{B}}(x)$$

Définition 4 ***L'intersection*** *de deux sous-ensembles flous $\tilde{A}$ et $\tilde{B}$ de U est un sous-ensemble flou constitué des éléments de U affectés du plus petit de leurs deux degrés d'appartenance, donnés par $\mu_{\tilde{A}}$ et $\mu_{\tilde{B}}$. C'est le sous-ensemble $\tilde{C} = \tilde{A} \cap \tilde{B}$ de U tel que :*

$$\forall\ x \in\ U,\ \mu_{\tilde{C}}(x) = min\ \{\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(x)\}$$

Définition 5 ***L'union*** *de deux sous-ensembles flous $\tilde{A}$ et $\tilde{B}$ de U est un sous-ensemble flou constitué des éléments de U affectés du plus grand de leurs deux degrés d'appartenance, donnés par $\mu_{\tilde{A}}$ et $\mu_{\tilde{B}}$. C'est le sous-ensemble $\tilde{C} = \tilde{A} \cup \tilde{B}$ de U tel que :*

$$\forall\ x \in\ U,\ \mu_{\tilde{C}}(x)\ =\ max\ \{\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(x)\}$$

Le choix des opérateurs *min* et *max* pour définir respectivement l'intersection et l'union de sous-ensembles flous est justifié par le fait qu'ils préservent presque toute la structure de la théorie des ensembles classiques. En effet, d'après les définitions données ci-dessus, nous pouvons retrouver les propriétés classiques de l'union et de l'intersection à savoir :

- Associativité et commutativité de $\cup$ et $\cap$,
- Distributivité dans les deux sens de $\cup$ et $\cap$,
- $\tilde{\mathrm{A}} \cup \emptyset = \tilde{\mathrm{A}}$, $\tilde{\mathrm{A}} \cup U = U$,
- $\tilde{\mathrm{A}} \cap U = \tilde{\mathrm{A}}$, $\tilde{\mathrm{A}} \cap \emptyset = \emptyset$.

Néanmoins, d'autres opérateurs sont envisageables si l'on est moins exigeant sur la préservation des propriétés classiques [2]. Ces opérateurs sont définis à l'aide d'une *norme triangulaire* et d'une *conorme triangulaire* définies comme suit [Klement *et al.*, 2002] :

Définition 6 *Une norme triangulaire " t-norme " est une fonction $\top$: [0, 1] $\times$ [0, 1] $\longmapsto$ [0, 1] vérifiant, pour tout x et y dans [0, 1], les propriétés suivantes [Klement* et al.*, 2002] :*

- $\top$ *est* ***commutative*** $\top(x, y) = \top(y, x)$,
- $\top$ *est* ***associative*** $\top(x, \top(y, z)) = \top(\top(x, y), z)$,
- $\top$ *est* ***croissante*** $\top(x, y) \leq \top(z, t)$ *si* $x \leq z$ *et* $y \leq t$,

2. Des travaux issus de la commande floue [Dubois et Prade, 1995, Dubois et Prade, 1996a] ont utilisé de tels opérateurs.

– $\top(x, 1) = x$.

D'une manière générale, l'opérateur d'intersection de deux sous-ensembles flous $\tilde{A}$ et $\tilde{B}$ peut être défini par une t-norme comme suit :

$$\mu_{\tilde{A}\cap_{\top}\tilde{B}}(x) = \top(\mu_{\tilde{A}(x)}, \mu_{\tilde{B}(x)})$$

Définition 7 *Une conorme triangulaire "t-conorme" est une fonction* $\perp$ *: [0,1]* $\times$ *[0,1]* $\longmapsto$ *[0, 1] vérifiant, pour tout x et y dans [0, 1], les propriétés suivantes [Bouchon-Meunier, 1995] :*

– $\perp$ *est* ***commutative*** $\perp(x, y) = \perp(y, x)$,

– $\perp$ *est* ***associative*** $\perp(x, \perp(y, z)) = \perp(\perp(x, y), z)$,

– $\perp$ *est* ***croissante*** $\perp(x, y) \leq \perp(z, t)$ *si* $x \leq z$ *et* $y \leq t$,

– $\perp(x, 0) = x$.

D'une manière générale, l'opérateur d'union de deux sous-ensembles flous $\tilde{A}$ et $\tilde{B}$ peut être défini par l'intermédiaire d'une t-conorme comme suit :

$$\forall\ x \in U,\ \mu_{\tilde{A}\cup_{\perp}\tilde{B}}(x) = \perp(\mu_{\tilde{A}(x)}, \mu_{\tilde{B}(x)})$$

Ainsi, les opérateurs les plus utilisés sont récapitulés par la Table 1.1 [Dubois et Prade, 1995, Dubois et Prade, 1996b] :

$\mathbf{x} \cap_{\top} \mathbf{y}$	$\mathbf{x} \cup_{\perp} \mathbf{y}$	**Nom**
min(x, y)	max(x, y)	Zadeh
max(x + y - 1, 0)	min (x + y, 1)	Lukasiewicz
x × y	x + y - xy	Probabiliste
$\begin{cases} x & \text{si x = 1} \\ y & \text{si y = 1} \\ 0 & \text{sinon} \end{cases}$	$\begin{cases} x & \text{si y = 0} \\ y & \text{si x = 0} \\ 1 & \text{sinon} \end{cases}$	Weber

TABLE 1.1 – Liste des opérateurs de t-norme et t-conorme duales

1.2.2 α - *coupures* d'un sous-ensemble flou

Une α-*coupure* d'un sous-ensemble flou $\tilde{A}$ est un ensemble ordinaire $\tilde{A}_{\alpha}$ contenant tous les éléments de l'univers de discours U, ayant un degré d'appartenance à l'ensemble $\tilde{A}$, supérieur ou égal à la valeur spécifiée de $\alpha \in [0\ ,\ 1]$.

Définition 8 *L'ensemble ordinaire* A_{α}*, appelé* α*-coupure, associé au sous-ensemble flou* $\tilde{A}$ *défini sur un univers de discours U, est défini par :*

$$A_\alpha = \{x \in U | \mu_{\tilde{A}}(x) \geq \alpha\}$$

Ainsi, si nous souhaitons se référer à des sous-ensembles ordinaires correspondant à un sous-ensemble flou donné, la façon la plus simple de réaliser cette approximation est de fixer une limite α inférieure aux degrés d'appartenance pris en considération.
Une α-coupure d'un ensemble flou nous permet donc, de déterminer quel sous-ensemble ordinaire est le plus proche à ce sous-ensemble flou. Cela, permet de retrouver les critères de décision de la théorie des ensembles classiques [Elloumi, 2002].

L'ensemble de toutes les α-coupures d'un sous-ensemble flou $\tilde{\mathrm{A}}$ forme une famille de sous-ensembles ordinaires de U emboîtés par rapport à la valeur α. En effet, si $\alpha \leq \alpha'$ alors $\mathrm{A}_{\alpha'} \subseteq A_\alpha$.

1.2.3 Produit cartésien de sous-ensembles flous

Soit $U_1, \ldots, U_r$ des ensembles de référence, et $U = U_1 \times \ldots \times U_r$ leur produit cartésien, dont les éléments sont des r-uplets ($\mathrm{x}_1, \ldots, \mathrm{x}_r$), avec $\mathrm{x}_1 \in U_1, \ldots, \mathrm{x}_r \in U_r$. À partir des sous-ensembles flous $\tilde{\mathrm{A}}_1, \ldots, \tilde{\mathrm{A}}_r$ respectivement définis sur $U_1, \ldots, U_r$, nous construisons un sous-ensemble flou $\tilde{\mathrm{A}} = \tilde{\mathrm{A}}_1 \times \ldots \times \tilde{\mathrm{A}}_r$ de U, considéré comme leur produit cartésien, ayant la fonction d'appartenance :

$$\forall\ \mathrm{x} \in U,\ \mu_{\tilde{A}}(x) = \min\{\mu_{\tilde{A}_1}(x), \ldots, \mu_{\tilde{A}_r}(x)\}$$

1.3 Relations floues

Étant donné deux ensembles de référence X et Y, une **relation floue** $\tilde{R}$ entre X et Y est définie comme un sous-ensemble flou défini sur le produit cartésien $X \times Y$.

En particulier, si nous avons un sous-ensemble ordinaire $\mathrm{A} \in X$ et un sous-ensemble flou $\tilde{\mathrm{B}} \in Y$, une **relation binaire floue** $\tilde{R}$ est un sous-ensemble flou défini sur le produit cartésien $\mathrm{A} \times \tilde{\mathrm{B}}$. Ainsi, la relation $\tilde{R} \subseteq A \times \tilde{B}$ est définie comme suit :

$$\tilde{\mathrm{R}} = \{(x,y)^{min(\mu_{A(x)}, \mu_{\tilde{B}(y)})} | x \in A, y \in \tilde{B}\}$$
$$\text{avec } \mu_{A(x)} = \begin{cases} 1 & \text{si } \mathrm{x} \in \mathrm{A}\,; \\ 0 & \text{sinon.} \end{cases}$$

Exemple 2 *Considérons l'ensemble classique $A = \{a\}$ et l'ensemble flou $\tilde{B} = \{p^{0,1}, q^{0,9}\}$. Ainsi, la relation $\tilde{R} \subseteq A \times \tilde{B}$ est définie comme suit :*

$\tilde{R} = \{(a,p)^{0,1}, (b,p)^{0}, (a,q)^{0,9}, (b,q)^{0}\}$.
L'élément $b \notin A$, *donc* $\mu_{A(b)} = 0$. *La relation résultante* $\tilde{R}$ *est une relation binaire floue.*

1.4 Implications floues

Avant de présenter la définition d'une implication floue, nous allons commencer au préalable par définir les notions de "*variable linguistique*" et de "*proposition floue*" [Dubois et Prade, 1996b].

Définition 9 *Une* ***variable linguistique*** *est représentée par un triplet* (V, U, T_V). V *étant une variable (e.g., âge, température, ...), définie sur un ensemble de référence* U *(e.g., l'ensemble des nombres entiers, des réels,...).* T_V *est un ensemble, fini ou infini, de sous-ensembles flous de* U *caractérisant* V.

Les variables linguistiques servent à modéliser les connaissances imprécises ou vagues sur une variable, dont la valeur précise est inconnue.

Exemple 3 *Considérons la taille comme une variable* V, *définie sur l'ensemble* U *des entiers positifs. Dans le cas des êtres humains, nous pouvons définir l'ensemble* T_V *comme illustré par la figure 1.1.*

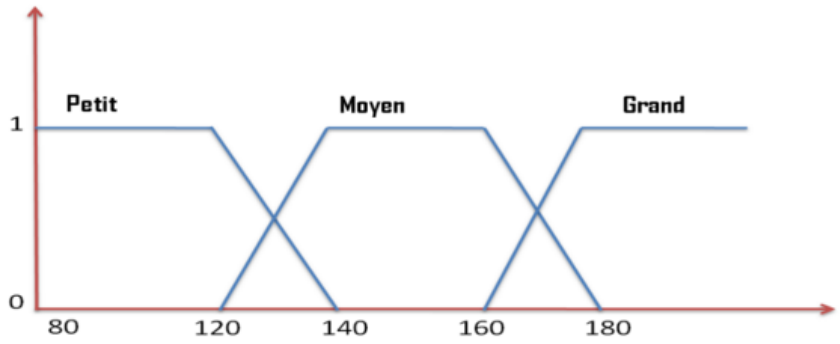

FIGURE 1.1 – Exemple d'une variable linguistique (V, U, T_V) utilisée pour décrire la taille d'un être humain.

Définition 10 *Une **proposition floue élémentaire** est définie à partir d'une variable linguistique (V, X, T_V) par la qualification "V est $\tilde{A}$", pour une caractérisation floue $\tilde{A}$ appartenant à T_V. La valeur de vérité d'une proposition floue élémentaire est définie par la fonction d'appartenance $\mu_{\tilde{A}}$ de $\tilde{A}$.*

Exemple 4 *"La taille est moyenne" constitue une proposition floue élémentaire.*

Définition 11 *Une **proposition floue composée** est obtenue par la composition de propositions floues élémentaires "V est $\tilde{A}$", "W est $\tilde{B}$", ... pour des variables linguistiques V , W, ... Cette composition est construite par conjonction, disjonction ou implication de propositions floues élémentaires.*

Définition 12 *Une **règle floue** est une proposition floue utilisant une implication floue. Une **implication floue** associée à toute règle floue du type "**si** V est $\tilde{A}$ **alors** W est $\tilde{B}$" est définie à partir des deux ensembles de référence (ou univers de discours) X et Y sur lesquels sont définis les deux variables linguistiques (V, X, T_V) et (W, Y, T_W).*

La valeur de vérité de la proposition floue obtenue par l'utilisation d'une implication floue entre les propositions floues "V est $\tilde{A}$" et "W est $\tilde{B}$" est définie par la fonction d'appartenance $\mu_{\tilde{R}}$ d'une relation floue entre X et Y. La fonction d'appartenance $\mu_{\tilde{R}}$, est définie, pour tout (x,y) de $X \times Y$, comme suit :

$$\mu_{\tilde{R}}\ (\mathrm{x},\ \mathrm{y}) = \Phi(\mu_{\tilde{A}}(\mathrm{x}),\ \mu_{\tilde{B}}(\mathrm{y}))$$

Φ est une fonction choisie de telle sorte que, dans le cas où $\tilde{A}$ et $\tilde{B}$ sont définis de façon précise et certaine (*i.e.*, $\tilde{A}$ et $\tilde{B}$ sont deux ensembles ordinaires), l'implication floue soit réduite à l'implication de la logique classique.
Il existe plusieurs implications floues dans la littérature [Bellman *et al.*, 1966, Dubois et Prade, 1991]. Le tableau 1.2 récapitule les implications floues les plus souvent utilisées [Dubois et Prade, 1991].

Dans la littérature différentes implications floues ont été proposées [Bellman *et al.*, 1966, Dubois et Prade, 1986]. Deux principales classes d'implications, les *R-implications* et les *S-implications*, ont été distinguées, elles sont définies comme suit :

Définition 13 *Pour a, b ∈ [0, 1], une R-implication est définie par [Bouchon-Meunier, 1995, Dubois et Prade, 1996b] :*

$$a \overset{R_{imp}}{\longrightarrow} b = sup\ \{x \in [0,1] | a \top x \leq b\}$$

Valeur de vérité $\mu_{\tilde{R}}$(x, y)	**Nom**	**Acronyme**
1 - $\mu_{\tilde{A}}$(x) + $\mu_{\tilde{A}}$(x) × $\mu_{\tilde{B}}$(y)	Reichenbach	I_R
max(1 - $\mu_{\tilde{A}}$(x),min($\mu_{\tilde{A}}$(x), $\mu_{\tilde{B}}$(y)))	Willmot	I_W
1 si $\mu_{\tilde{A}}$(x) ≤ $\mu_{\tilde{B}}$(y) et 0 sinon	Rescher-Gaines	I_{RG}
max(1 - $\mu_{\tilde{A}}$(x), $\mu_{\tilde{B}}$(y))	Kleene-Dienes	I_{KD}
1 si $\mu_{\tilde{A}}$(x) ≤ $\mu_{\tilde{B}}$(y) et $\mu_{\tilde{B}}$(y) sinon	Brouwer-Gödel	I_{BG}
min($\mu_{\tilde{B}}$(y)/$\mu_{\tilde{A}}$(x), 1) si $\mu_{\tilde{A}}$(x) ≠ 0 et 1 sinon	Goguen	I_G
min(1 - $\mu_{\tilde{A}}$(x) + $\mu_{\tilde{B}}$(y), 1)	Lukasiewicz	I_L

TABLE 1.2 – Principales implications floues

où $\top$ *désigne une t-norme triangulaire.*

Définition 14 *Pour a, b ∈ [0, 1], une S-implication est définie par [Bouchon-Meunier, 1995, Dubois et Prade, 1996b] :*

$$a \xrightarrow{R_{imp}} b = (1 - a) \perp b$$

où $\perp$ *désigne une t-conorme triangulaire.*

Une implication floue est une application qui vérifie au minimum les propriétés de l'implication booléenne pour les bornes de l'intervalle [0, 1] [Dubois et Prade, 1991].

1.5 Acquisition des modalités floues

Dans cette section, nous présentons les différentes approches proposées dans la littérature afin de construire les modalités flous d'un attribut (*i.e.*, fonctions d'appartenance d'un ensemble flou). Nous commençons tout d'abord par énumérer les différent types de modalités floues relatives à un ensemble flou.

1.5.1 Les types de fonctions d'appartenance

Il existe différents types ou formes de fonctions d'appartenance d'un ensemble flou. Les types les plus utilisés sont présentés ci-dessous.

Fonction d'appartenance triangulaire

Une fonction d'appartenance triangulaire, telle que illustrée par la figure 1.2, est caractérisée par trois paramètres x_1, x_2 et x_3 correspondant respectivement à la borne inférieure, la borne supérieure et une valeur modale. Ce type de fonctions est défini comme suit :

$$\mu_A(x) = \begin{cases} \frac{x-x_1}{x_2-x_1} & si\ x_1 \leq x < x_2 \\ \frac{x-x_3}{x_2-x_3} & si\ x_2 \leq x < x_3 \\ 0 & sinon \end{cases}$$

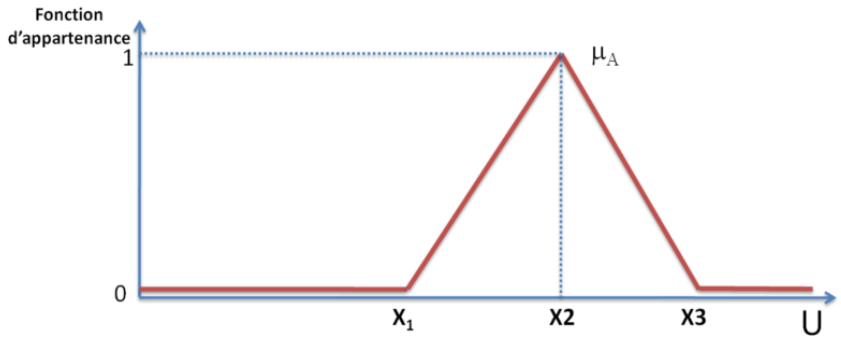

FIGURE 1.2 – Fonction d'appartenance triangulaire.

Fonction d'appartenance trapézoïdale

Une fonction d'appartenance trapézoïdale, telle que illustrée par la Figure 1.3, est définie par quatre paramètres x_1, x_2, x_3 et x_4. Les paramètres x_1 et x_4 représentent respectivement la limite inférieure et la limite supérieure du support. Les paramètres x_2 et x_3 sont respectivement la borne inférieure et la borne supérieure du noyau. Cette fonction est définie par l'expression suivante :

$$\mu_A(x) = \begin{cases} \frac{x-x_1}{x_2-x_1} & si\ x_1 \leq x < x_2 \\ 1 & si\ x_2 \leq x < x_3 \\ \frac{x-x_4}{x_3-x_4} & si\ x_3 \leq x < x_4 \\ 0 & sinon \end{cases}$$

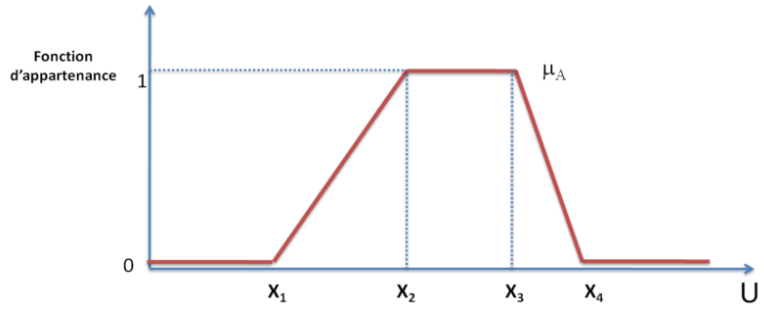

FIGURE 1.3 – Fonction d'appartenance trapézoïdale.

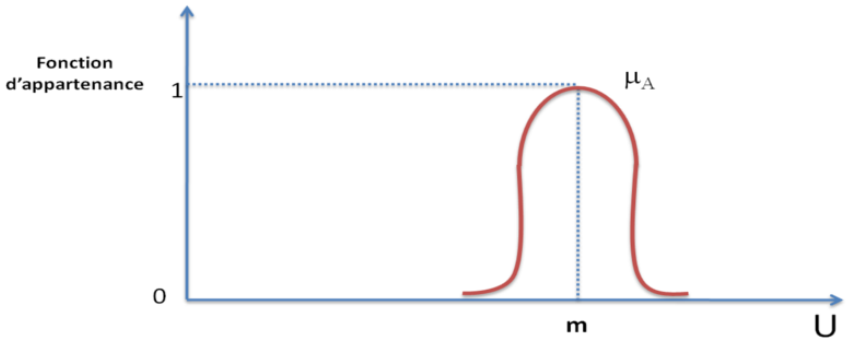

FIGURE 1.4 – Fonction d'appartenance gaussienne.

Fonction d'appartenance gaussienne

Une fonction d'appartenance gaussienne, telle que illustrée par la Figure 1.4, est caractérisée par sa valeur centrale m et son écart type σ. La fonction d'appartenance gaussienne est définie par :

$$\mu_A(x) = exp^{\left(-\frac{1}{2}\left(\frac{x-m}{\sigma}\right)^2\right)}$$

Nous avons présenté ci-dessus les fondements mathématiques des ensembles flous. Nous avons également présenté les types de fonctions d'appartenance les plus utilisées à savoir les fonctions triangulaires, trapézoïdales et gaussiennes. En outre, il existe d'autres formes telles que les fonctions en cloche, les fonctions en P, les fonctions sigmoïdes, etc.
Dans ce qui suit, nous présentons un panel d'approches et de méthodes générales permettant d'obtenir ces fonctions d'appartenance indépendamment de leurs formes.

1.5.2 Les approches de génération des fonctions d'appartenance

L'utilisation de la théorie des ensembles flous pour formaliser un problème, nous engage à manipuler autant de fonctions d'appartenance qu'il y a de caractérisations de concepts. Cependant, ces fonctions d'appartenance d'où proviennent-elles ? et comment les obtenir ?

Dans ce qui suit, nous allons essayer de répondre plus formellement à ces questions en illustrant les différentes méthodes proposées afin de construire ces fonctions d'appartenance. Dans la suite, ces différentes méthodes sont classées en quatres catégories [Aladenise et Bouchon-Meunier, 1997] : les méthodes automatiques, les méthodes statistiques, les méthodes psychométriques et les méthodes géométriques.

1. **Les approches automatiques** :
   Les méthodes automatiques d'acquisition de fonctions d'appartenance se déroulent en deux étapes. La première étape consiste à définir une première fonction d'appartenance mal ajustée voir aléatoire. Dans la deuxième étape, cette fonction est ajustée. Ces méthodes ont le mérite de se passer de l'expert. En effet, dans certains domaines il est difficile d'acquérir les connaissances des experts (*e.g.* ; biologie). Nous pouvons distinguer trois catégories de méthodes automatiques à savoir les méthodes basées sur la classification, celles basées sur les réseaux de neurones et celles utilisant les algorithmes génétiques.

   - **Les méthodes basées sur les réseaux de neurones** :
     Avec le développement des modèles des réseaux de neurones et leur application dans l'apprentissage, un intérêt considérable a été accordé à leur utilisation avec la théorie des ensembles flous [Takagi et Hayashi, 1991]. Le principe consiste à apprendre à classifier correctement des données à partir d'un jeu d'exemples déjà classifiés, c'est-à-dire l'apprentissage par l'expérience. Un réseau de neurones est constitué d'un graphe pondéré orienté dont les noeuds symbolisent les neurones. Ces neurones possèdent une fonction d'activation qui permet d'influencer les autres neurones du réseau. Les fonctions les plus souvent utilisées sont la fonction signe ou la fonction sigmoïde. Les connexions entre les neurones, que l'on nomme liens synaptiques, propagent l'activité des neurones avec une pondération caractéristique de la connexion. Nous appelons poids synaptique la pondération des liens synaptiques. Les neurones peuvent être organisés de différentes manières, c'est ce qui définit l'architecture et le modèle du réseau. L'architecture la plus courante est celle dite du perceptron multi-couches.

     Dans [Yamakawa et Furukawa, 1992], les auteurs présentent un algorithme d'apprentissage des paramètres de la fonction d'appartenance en utilisant un modèle neuro-flou.

- **Les méthodes basées sur la classification** :

  En utilisant la classification, les fonctions d'appartenance sont générées soit au cours du processus du clustering [Bezdek, 1981], soit sur la base de certains paramètres dérivés à partir des clusters obtenus [Fu *et al.*, 1998]. Dans toutes ces méthodes, les fonctions d'appartenance peuvent représenter le niveau d'adéquation de chaque élément au cluster auquel il appartient ou la distance entre chaque élément et le centre de gravité du cluster. Nous allons illustrer ci-dessous ces différentes méthodes.

  L'algorithme de classification floue FCM développé par [Bezdek, 1981] est basé sur l'optimisation d'un critère quadratique de classification où chaque classe est représentée par son centre de gravité. Le problème d'optimisation consiste à minimiser la somme des distances intra-clusters généralisée au cas floue et exprimée comme suit :

$$J_m(U,V,X) = \sum_{i=1}^{C}\sum_{k=1}^{n}(\mu_{ik})^m(d_{ik})^2 \tag{1.1}$$

  Où :

  – $X = \{x_1, ..., x_n\}$ est l'ensemble de données.
  – C est le nombre de clusters.
  – $U = \{\mu_{ik} = \mu_i(x_k), 1 \leq k \leq n, 1 \leq i \leq C\}$ est une matrice $n \times C$ représentant une C-partition floue de X telle que μ_{ik} est le degré d'appartenance de l'élément x_k à la classe i et $\forall k \sum_{i=1}^{C} \mu_{ik} = 1$.
  – $V = \{v_1, v_2, ..., v_n\}$ est l'ensemble de prototypes des classes.
  – d_{ik} est la distance entre l'élément x_k et le prototype v_i.
  – m est une métrique de la quantité floue dans la partition ($m \geq 1$).

  Cet algorithme procède de la manière suivante :

  (a) La matrice U est initialisée d'une manière aléatoire.

  (b) Les centroïdes v_i des classes sont calculés comme suit :

$$v_i = \frac{\sum_{k=1}^{n}(\mu_{ik})^m x_k}{\sum_{k=1}^{n}(\mu_{ik})^m} \tag{1.2}$$

  (c) La matrice d'appartenance U est ajustée selon la position des centroïdes.

  (d) Les étapes 2 et 3 sont répétées jusqu'à atteindre une solution stable.

  L'approche proposée par [Fu *et al.*, 1998] applique l'algorithme CLARANS [Raymond et Jiawei, 1994] pour créer une partition des données. la méthode CLARANS permet de calculer le centroïde de chaque cluster, qui est lui même représenté par une fonction triangulaire. Le centroïde, noté c_{ni}, représente le noyau de la focntion d'appartenance du cluster C_i. Pour une partition $C_k = C_1, \ldots, C_k$, les supports des

ensembles flous décrivant cette partition, sont déterminés comme suit :

- Le support du cluster C_1 est défini par $supp_1 = [min, cn_2[$, où min represente la valeur minimale du domaine de l'attribut.
- Le support du cluster C_i, $1 < i < k$, est défini par $supp_i =]cn_{i-1}, cn_{i+1}[$.
- Le support du k^{ime} cluster est défini par $supp_k =]cn_{k-1}, max]$, où max représente la valeur maximale de l'attribut correspondant.

Une autre approche de construction des fonctions d'appartenance est proposée dans [Derbel *et al.*, 2008]. Les auteurs proposent une approche automatique totalement indépendante de l'expert et qui tient compte également de l'aspect dynamique des données. En effet, toute opération d'insertion ou de suppression de données conduit à un réajustement des paramètres de la fonction d'appartenance déjà construite sans avoir besoin de reprendre les différentes étapes de construction des fonctions d'appartenance. L'approche proposée procède en trois étapes. La première étape consiste à générer une partition des données en se basant sur l'algorithme de clustering CLUSTERDB [Wilson et Sutcliffe, 2007]. Cette partition permet de définir le nombre des sous-ensembles flous à générer. La deuxième étape consiste à construire les noyaux des sous-ensembles flous. La dernière étape a pour but la dérivation des supports des sous-ensembles flous à partir des noyaux déterminés.

- **Les méthodes basées sur les algorithmes évolutionnaires** :

  Les algorithmes évolutionnaires sont des algorithmes d'optimisation, qui tentent de simuler le processus d'évolution des espèces, *i.e.*, la sélection naturelle, la génétique et les lois de survies énoncés par Darwin afin de résoudre des problèmes d'optimisation. Les algorithmes génétiques est l'une des plus importantes familles des algorithmes évolutionnaires [3]. Les algorithmes génétiques ont été associés à plusieurs domaines. En fouille de données, ils ont été utilisés pour découvrir les itemsets fréquents et/ou les règles d'association et leurs confiances [Alcalá-Fdez *et al.*, 2009, Kaya et Alhajj, 2006]. Pour le problème d'acquisition des fonctions d'appartenance, les algorithmes génétiques ont été utilisés afin de déterminer les paramètres de ces fonctions.

  Le principe général de ce paradigme est de maintenir une population d'individus décrits par des chromosomes, représentant ainsi les différentes solutions d'un problème donné. Pour le problème de construction des fonctions d'appartenance, ces

3. Nous revenons ultérieurement en détail sur les algorithmes génétiques.

chromosomes représentent les paramètres de ces fonctions. D'une manière générale, une population initiale d'individus décrits par des chromosomes est générée d'une manière aléatoire. Cette population va subir des opérations génétiques telles que la sélection, le croisement et la mutation et une nouvelle population est générée. Le processus de reproduction est répété jusqu'à obtenir la meilleure solution possible relative au problème posé, donc de la fonction d'appartenance la plus pertinente.

Botzheim et *al.* [Botzheim *et al.*, 2002] proposent une approche de génération automatique de fonctions d'appartenance dans le cadre de l'extraction des règles floues dans les systèmes flous. Cette approche utilise un algorithme génétique, pour la génération simultanée des fonctions d'appartenance trapézoïdales et des règles floues optimales. Une règle floue R_i est exprimée par :

$$\text{Si } (x_1 \text{ est } A_{i_1}) \text{ Et } (x_2 \text{ est } A_{i_2}) \text{ Et } \ldots \text{ Et } (x_n \text{ est } A_{i_n}) \text{ Alors } (y \text{ est } B_i)$$

Avec A_{i_j} et B_i sont des ensembles flous, x_j est la variable d'entrée et y est la variable de sortie du système flou. Initialement, les règles floues sont codées dans un individu de la population. Ce codage consiste à choisir les quatres paramètres de la fonction d'appartenance trapézoïdale pour chacun des ensembles flous A_{i_j} et B_i. L'initialisation du premier individu est faite d'une manière aléatoire. Ensuite, les autres individus sont générés et un ensemble d'opérations génétiques est appliqué :

(a) Générer N copies (clônes) en choisissant une partie de l'individu initial et modifier ses paramètres aléatoirement pour chaque individu généré.

(b) Évaluer les individus de la population en se basant sur une fonction de calcul d'erreur définie par :

$$e = \frac{1}{N} \sum_{hantillons} \frac{y - y'}{I_{max} - I_min}$$

où N est le nombre d'échantillons évalués, y' est la sortie désirée du système pour un échantillon d'entrée donné et y est la sortie du système flou pour la même entrée. I_{max} et I_{min} représentent respectivement la borne supérieure et la borne inférieure de l'interval de définition de la variable de sortie.

(c) Sélectionner l'individu ayant la plus petite valeur d'erreur et transférer la partie mutée aux autres individus.

Ces opérations sont répétées jusqu'à obtention de la meilleure base de règles.

Afin d'optimiser le nombre de règles, les auteurs [Botzheim *et al.*, 2002] proposent d'utiliser un ensemble d'opérateurs flous à savoir :

– Fusion : deux fonctions d'appartenance, relatives à une même variable, sont fusionnées en une seule dans les cas suivants :

(a) Elles sont proches l'une de l'autre : $|\frac{l_i}{l_j - 1}| < \gamma$ où l_i et l_j sont les longueurs des noyaux des fonctions d'appartenance de la même variable et γ est un seuil fixé par l'utilisateur.

(b) La différence entre la longueur de leurs noyaux est très petite : $|f| < \gamma$ où f est la mesure de distance entre les centres de l_i et l_j.

– Analyse sémantique : Si deux règles ont un même antécédent mais une conséquence différente, les fonctions d'appartenance de la variable de sortie sont fusionnées.

2. **Les approches statistiques et probabilistes** :
Plusieurs approches basées sur les statistiques et la probabilité ont été proposées :

- **Méthodes "Oui-Non"** :
Proposée initialement par Black [Black, 1937], cette méthode consiste à présenter un objet x à plusieurs individus et de prendre leurs avis concernant la compatibilité de cet élément avec un sous-ensemble flou A donné. La question posée n'autorise qu'une seule réponse binaire : "oui" ou "non". Le degré d'appartenance de x à l'ensemble flou A est égal à la proportion de la réponse "oui" dans l'ensemble des réponses :
$$m_A(x) = \frac{nbrT(\text{"}oui\text{"})}{nbrT(\text{"}oui\text{"})(\text{"}non\text{"})}$$
où $nbrT(\text{"}oui\text{"})$ représente le nombre total de réponses "oui" et $nbrT(\text{"}oui\text{"}, \text{"}non\text{"})$ représente le nombre total de réponses "oui" et "non".
Par exemple pour savoir si un homme de $1,72m$ est "grand", la question pourra être formulée ainsi : "Est-il vrai qu'un homme de 1,72m soit grand ?". Le degré d'appartenance de "1,72m" à l'ensemble flou "grand" sera égal à la proportion de réponses *"oui"* dans l'ensemble de toutes les réponses.

- **Méthode d'estimation d'ensembles** :
Dans cette méthode proposée par Wang [Wang, 1983], les membres de la population décrivent leurs notions de A par un sous-ensemble de X. La fonction d'appartenance de x à A, $\mu_A(x)$, prendra comme valeur la fréquence avec laquelle la description de A donnée par les membres p_i interrogés (notée $A(p_i)$) contenait x.
Prenons l'exemple de la Figure 1.5, $\mu_A(x) = \frac{2}{3}$ cela veut dire que deux des trois experts interrogés pensent que x possède la caractéristique A.

- **Méthode de Hisdal** :
Cette méthode proposée par Hisdal [Wang, 1988], consiste à demander à l'expert

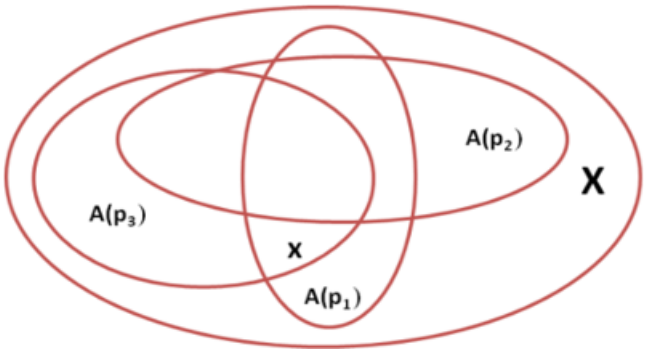

FIGURE 1.5 – Exemple d'estimation d'ensemble.

de répondre par "Oui" ou "Non" à la question "*Cet élément x de X appartient-il à A (sous-ensemble de X)* ?" plusieurs fois jusqu'à pouvoir obtenir une courbe à seuil non floue. Pour l'exemple des tailles, cette courbe est telle qu'elle est illustrée par la Figure 1.6.

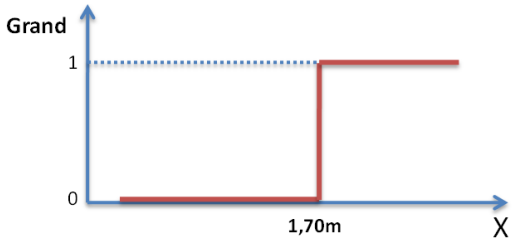

FIGURE 1.6 – Courbe à seuil non floue pour l'attribut taille.

L'idée de Hisdal est qu'il est probable que l'on se soit trompé sur le choix de l'élément frontière (1,70m sur l'exemple), mais que plus on s'éloigne de cet élément frontière, plus la probabilité que l'on se soit trompé est faible [Aladenise et Bouchon-Meunier, 1997]. Cette probabilité de l'erreur est matérialisée par une courbe d'erreur $E(u)$, qui est une distribution de probabilité gaussienne. Sur l'exemple des tailles, cette courbe $E(u)$, décrite par la Figure 1.7, est telle que l'intégrale de $E(u)$ soit égale à 1.

Finalement, la fonction d'appartenance $\mu_{grand}(x)$ est obtenue à partir de la convolution de la courbe à seuil avec la courbe de l'erreur estimée. La fonction d'appartenance $\mu_{grand}(x)$ est illustrée par la figure 1.8.

3. **Les approches psychométriques** :
   Ce sont les différentes méthodes permettant d'interroger un expert afin de lui faire nu-

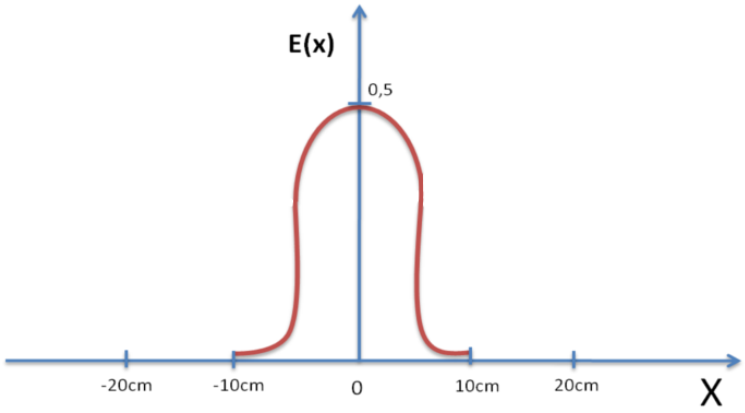

FIGURE 1.7 – Fonction d'erreur.

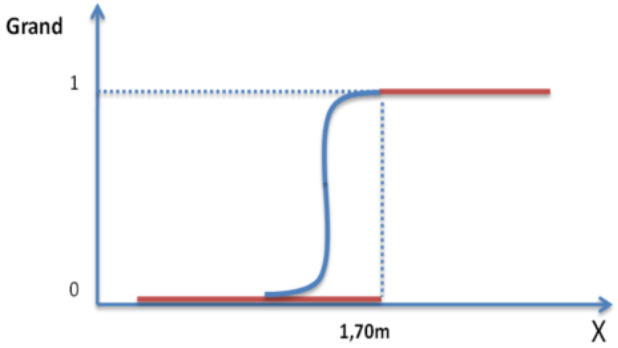

FIGURE 1.8 – Fonction à seuil arrondie.

mériser un concept linguistique. Il s'agit d'un exercice de psychométrie. Il existe plusieurs méthodes psychométriques pour l'obtention de fonctions d'appartenance.

- **Méthodes "Noyau-Support"** :

  Il s'agit d'interroger l'expert de manière à lui faire définir le noyau et le support de la fonction d'appartenance. Ainsi, l'expert est appelé à faire produire quatres paramètres (p_1, p_2, p_3, p_4). Ces paramètres permettent de construire une fonction de maximum 1 atteinte sur $[p_2, p_3]$ et de minimum 0 à l'extérieur de $[p_1, p_4]$ non croissante entre p_3 et p_4 et non décroissante entre p_1 et p_2. Un exemple de cette fonction est illustré par la figure 1.9.

- **Quantification structurelle** :

  Cette méthode proposée par Zhang [Zhang, 1993], consiste à partitionner l'ensemble

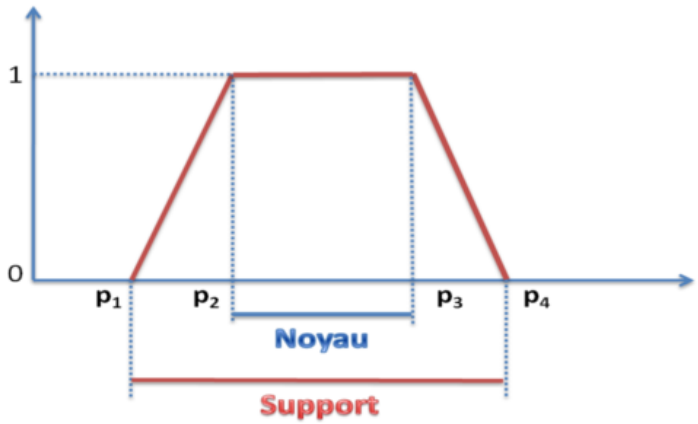

FIGURE 1.9 – Fonction d'appartenance avec quatre paramètres p_1, p_2, p_3 et p_4.

X en trois sous-ensembles X_0, X_1, X_f. X_1 est l'ensemble des éléments de X appartenant à A avec une certitude totale : $x|\mu_A(x) = 1$ (*i.e.*, le noyau de A). X_0 est l'ensemble des éléments de X n'appartenant pas à A avec une certitude totale : $x|\mu_A(x) = 0$ et X_f est l'ensemble de tous les éléments n'appartenant ni à X_1 ni à X_0 : $x|0 \leq \mu_A(x) \leq 1$. Nous avons alors :

$$\forall x \in X_f,\ \mu_A(x) = \frac{Card(y \in X_f | x >_A y)}{Card(X_f)} = \frac{nombre\ de\ y <_A x\ dans\ X_f}{nombre\ d'éléments\ de\ X_f}$$

où $x >_A y$ signifie que "$x \in A$" est plus vrai que "$y \in A$". La véracité de "grand" pour l'élément x dépend de la quantité de personnes qui soient moins grandes que x. L'idée de la quantification structurelle est qu'un sous-ensemble flou est défini par rapport aux autres sous-ensembles flous. Par exemple "bon" est "supérieur à moyen".

- **Grille répertoire** :

  Cette méthode de "*grille répertoire*" [Boose et Otto, 1985, Hart, 1992, Plaza *et al.*, 1986] consiste à interroger l'expert afin de lui faire préciser la signification de termes vagues à l'aide d'autres termes linguistiques eux-même vagues. Les résultats répertoriés dans une grille seront ensuite convertis en numériques.

  Pour chaque terme flou, l'expert fournit le terme antagoniste (*i.e.*, pour "grand" l'expert répond "petit"). Une échelle bipolaire est ainsi créée graduée de 1 à 5 par exemple (*i.e.*, elle peut être graduée de 1 à 3 ou de 1 à 9, etc) de telle sorte que les chiffres représentent pour l'expert :

  1 : très grand ; 2 : grand ; 3 :taille moyenne ; 4 :petit ; 5 :très petit.

4. **Méthode par interpolation** :

Cette méthode nécessite de connaître un nombre raisonnable de points appartenant à la fonction d'appartenance à construire sur un ensemble de définition X continu. À partir de ces points, il est possible de construire une fonction par interpolation tout en vérifiant certaines contraintes pour que la courbe obtenue corresponde à une fonction d'appartenance correcte. Dans [Chen et Otto, 1995], les auteurs proposent une méthode de construction de fonction d'appartenance $\mu : \mathbb{R} \to [0, 1]$ vérifiant les contraintes suivantes :

- $\mu(x) \in [0, 1]$ pour tout x ;
- μ est différentiable ;
- $\mu(x_i) = \mu_i$ pour tout ensemble fini de couples $(x_1, \mu_1), \ldots, (x_n, \mu_n)$;
- Si l'ensemble de couples connus $(x_1, \mu_1), \ldots, (x_n, \mu_n)$ est un ensemble convexe, alors μ est convexe.

Cette méthode permet de construire une fonction d'appartenance à partir de quelques points seulement.

1.6 Discussion

Dans ce chapitre, nous avons abordé les notions de base de la logique floue et des ensembles flous. Nous avons également présenté les méthodes et approches proposées pour d'acquisition des fonctions d'appartenance. En effet, plusieurs approches ont été proposées afin de modéliser des termes linguistiques par des sous-ensembles flous ce qui montre qu'aucune n'est universelle. Dans cette partie, nous avons présenté quelques méthodes de génération de fonction d'appartenance. Toute fois il existe bien d'autres méthodes d'obtention de ces fonctions d'appartenance propres aux systèmes auxquels elles ont été conçues.

Les méthodes automatiques permettent de définir des fonctions d'appartenance mal ajustées (aléatoire dans certains cas) et ajustement de ces fonctions de base par la suite (optimisation). Les autres méthodes aussi bien statistiques que psychométriques, permettent généralement de construire des fonctions d'appartenance à noyau vide ; les données de l'ensemble de référence X sont ordonnées par une structure d'ordre : $x_1 \geq x_2$ signifie que "$x_1 \in A$" est au moins aussi vrai que "$x_2 \in A$". D'un point de vue de l'effort à fournir pour l'obtention de ces fonctions d'appartenance, nous remarquons que dans les méthodes automatiques cet effort est minimal, puisque l'expert n'a pas à intervenir dans le processus d'acquisition. Les méthodes statistiques et probabilistes représentent un effort de recueil d'expertise qualitatif plutôt que quantitatif [Aladenise et Bouchon-Meunier, 1997]. Alors que les méthodes psychométriques s'appliquent à accorder une signification à la démarche proposée à l'expert.

Un autre regard, pour classer les méthodes d'acquisition des fonctions d'appartenance, a été proposé dans [Hachani, 2010]. En effet, l'auteur classe ces méthodes en deux catégories : celles manuelles et celles automatiques.

Les auteurs [Bilgiç et Turksen, 1995] proposent d'identifier la signification du flou pour pouvoir classer les méthodes d'acquisition des fonctions d'appartenance. Ainsi, ils identifient deux principales interprétations du flou : le flou est subjectif, par opposition au flou objectif et le flou individuel, par opposition au flou provenant d'un groupe de personnes (ou capteurs, etc.) ce qui est illustré dans la figure 1.10.

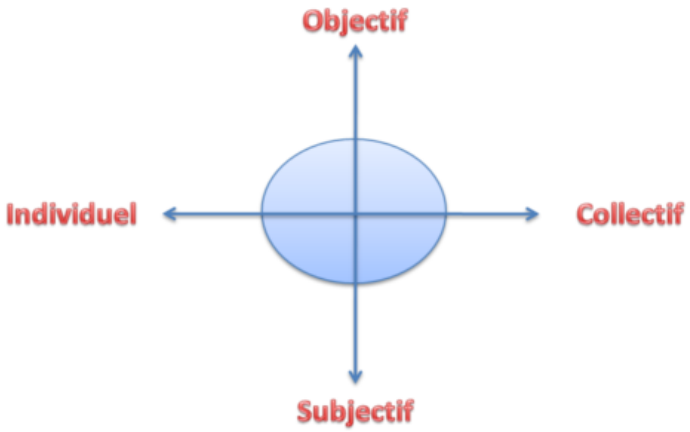

FIGURE 1.10 – Interprétation du flou.

Toutes ces classifications des méthodes d'acquisition des fonctions d'appartenance ont une correspondance étroite. Les méthodes automatiques produisent un flou objectif. Les méthodes statistiques manuelles correspondent à un flou collectif. Les méthodes probabilistes et psychométriques manuelles présentent un flou individuel. Cette correspondance est mise en évidence dans le tableau 1.3.

Approche	Méthode(s)		Manuelle/ (Semi)Automatique	Type de flou
Automatique	Réseaux de neurones		Automatique	Objectif
	Classification	Fcm [Bezdek, 1981]	Semi-automatique	Objectif
		Clarans [Raymond et Jiawei, 1994]	Semi-automatique	Objectif
		ClusterBd[Derbel *et al.*, 2008]	Automatique	Objectif
	Algorithmes évolutionnaires	Algos. Génétiques	Automatique	Objectif
Stats et probabilistes	"Oui-Non"		Manuelle	Subjectif Collectif
	Estimation d'ensemble		Manuelle	Subjectif Individuel
	Hisdal [Wang, 1988]		Manuelle	Subjectif Collectif
Psychométrique	Noyau-Support		Manuelle	Subjectif Individuel
	Quantification structurelle		Manuelle	Subjectif Individuel
	Grille répertoire		Manuelle	Subjectif Individuel
Géométrique	Méthode par interpolation		Manuelle	Objectif Individuel

Table 1.3 – Calssification des méthodes d'acquisition des fonctions d'appartenance

Chapitre 2

Gradualité : formalisation des motifs et règles graduels

Sommaire

2.1 Introduction

La notion de gradualité a été majoritairement utilisée dans les systèmes de recommandation et de contrôle afin de représenter les corrélations de variations entre le éléments numériques et d'associer des décisions à des situations [Dubois et Prade, 1992, Galichet *et al.*, 2004]. D'une manière générale, un motif graduel flou est de la forme "plus/moins A_1 est F_1, ..., plus/moins A_n est F_n", décrivant la variation d'un attribut A_i associé à une modalité floue F_i elle même

décrite par une fonction d'appartenance. Cependant, il n'existe pas de méthodologie standard pour construire les fonctions d'appartenance, notion de base de la théorie des ensembles flous. Toute la difficulté réside dans la représentation des termes linguistiques par un modèle numériques [Aladenise et Bouchon-Meunier, 1997]. En effet, s'il est facile de comprendre "très jeune", il est difficile de numériser "très" ou "jeune".

2.2 Motifs et règles graduels

La notion de gradualité a été abordée initialement dans la théorie de la logique floue (*i.e.*, gradualité de l'appartenance d'un élément à un ensemble). Cependant, cette gradualité ne permet pas de modéliser les co-variations entre ensembles. Ceci a poussé les chercheurs à définir des règles graduelles décrivant l'implication entre ensembles d'éléments [Di Jorio *et al.*, 2008, Hüllermeier, 2002].

2.2.1 Modélisation des règles graduelles

Dans [Dieng *et al.*, 1993], les règles graduelles ont été désignées par les termes "*topoi*" et "règles d'inférences graduelles". Les auteurs démontrent la présence de gradualité dans de nombreux contextes tels que l'extraction de connaissances ou encore le raisonnement automatique, et proposent deux solutions pour l'acquisition de ces règles. La première consiste à étudier le discours sémantique de l'expert. Les auteurs notent que des termes précis identifient une inférence graduelle, comme "d'autant plus", "augmente", "diminue", "plus", "au plus", "moins", "au moins", *etc.* L'ingénieur peut alors construire manuellement les règles graduelles à partir de discussions avec l'expert. La seconde consiste à construire automatiquement des arbres de décisions, puis s'appuyer sur ces arbres afin de définir manuellement les règles graduelles. Dans [Dieng *et al.*, 1993], aucune formalisation de la gradualité n'est fournie. En revanche, les auteurs étudient de manière remarquable l'impact de l'utilisation des règles graduelles à plusieurs niveaux des méthodes de modélisation des connaissances stratégiques et des connaissances du domaine (*KAD* et *KOD* [Breuker et de Greef, 1993]).

Dans [Bouchon-Meunier, 1990], l'auteur étudie l'impact des différents opérateurs d'implications floues sur les règles graduelles. Ainsi, étant donnée une règle graduelle de la forme R : $X \rightarrow Y$, elle lui correspond une valeur de vérité définie à partir d'une implication floue utilisant une t-norme (notée $\top$). Par ailleurs, l'auteur note que l'implication utilisée conditionne l'information graduelle véhiculée par la règle. Par exemple, pour la règle graduelle R_1 : "*moins le vent est fort, moins la mer est houleuse*", l'utilisation d'une implication employant la t-norme

de Lukasiewicz R^L affaiblit la liaison graduelle et transforme la règle en "*il est presque certain que moins le vent est fort, moins la mer est houleuse*". En revanche, l'utilisation de la t-norme de Rescher-Gaines R_{RG} vient renforcer la liaison graduelle de la règle R_1. Les auteurs étudient ainsi les cas de renforcement de la prémisse et de la conséquence de la règle, ainsi que la notion de certitude apportée par ces différentes implications floues.
Dans [Bouchon-Meunier, 1990], l'auteur a mis en évidence, sans les formaliser, les premières propriétés des règles graduelles et notamment celles de l'antonymie : il est possible de passer d'une règle à son contraire. Par exemple, la règle R_1 peut être transformée de manière évidente en "*plus le vent est fort, plus la mer est houleuse*". Cependant, l'auteurs ne propose pas de solution d'extraction automatique de telles règles, mais un système d'aide à leur formulation pour les experts.

Dans [Dubois et Prade, 1992], les auteurs proposent une première formalisation des règles d'inférence graduelles. Il s'agit de définir un contexte formel de règles graduelles floues de la forme "*plus x est A, alors plus y est B*" en vue de leur traitement automatique dans le cadre d'un système de raisonnement.

Depuis leur apparition, les règles graduelles ont été utilisées principalement dans de nombreux domaines tels que la modélisation des comportements de systèmes (*i.e.*, contrôleurs flous) [Dimitrov et Rykov, 2004, Dubois *et al.*, 1995], le domaine du traitement de l'imprécision [Galichet *et al.*, 2003, Galichet *et al.*, 2004, Isabela *et al.*, 2002], le domaine du traitement de la langue [Bosc *et al.*, 1997, Bosc *et al.*, 1999], ou encore en médecine [Agier *et al.*, 2007, Miyazaki *et al.*, 2001]. Récemment, les règles graduelles ont été utilisées afin de construire des règles de classification [Choong *et al.*, 2009, Dârlea, 2010].
À partir des années 2000, les recherches menées sur les règles graduelles s'orientent non plus sur comment les modéliser mais plutôt sur les méthodes de leur extraction automatique.

2.2.2 Formalisation des motifs graduels en fouille de données

Dans ce qui suit, nous allons présenter les différents algorithmes et formalisations proposés dans la littérature afin d'extraire automatiquement les dépendances ou règles graduelles. Nous commençons par définir quelques notions de base que nous allons utiliser dans la suite de ce mémoire.

Formellement, les règles graduelles sont extraites à partir de bases définies sur un schéma $(X_1, \ldots, X_m)$ de domaines $dom(X_i)$ numériques (ou au moins munies d'un ordre total). Une

base de données $\mathcal{D}$ est un ensemble de m-uplets de dom(X_1)×, ..., × dom(X_m). Dans le cas de données floues, les attributs X_i sont des variables linguistiques associées à des valeurs linguistiques définies par des ensembles flous ou modalités. Considèrons par exemple un attribut correspondant à la vitesse d'un véhicule. Dans le cas classique, il contient des valeurs numériques des vitesses mesurées. Notons que dans le cas flou, cet attribut peut être associé à trois variables linguistiques "*Faible*", "*Normale*", et "*Rapide*". Ces variables linguistiques ou modalités sont définies par des degrés d'appartenance (définis par une fonction d'appartenance), indiquant le degré avec lequel leurs vitesses appartiennent à chaque modalité.

- **Item graduel** :
  dans le cas classique, un item graduel est défini par couple $(i, *)$ associant un attribut i associé à un sens de variation $*$ croissant ou décroissant ($* \in \{\geq$ ou $\leq\}$).

  Exemple 1 *Soit A un attribut correspondant à la vitesse d'un véhicule,* $A^{\geq}$ *et* $A^{\leq}$ *sont des items graduels représentant respectivement* $(vitesse, plus)$ *et* $(vitesse, moins)$*. Ils représentent ainsi le fait que les valeurs de l'attribut augmentent (dans le cas de* $\geq$*) ou diminuent (dans le cas de* $\leq$*).*

- **Item graduel flou** :
  dans le cas des données floues, un item graduel flou est défini par un triplet $(i, m, *)$ associant un item i à une modalité floue m et un sens de variation $*$.

  Exemple 2 $(vitesse, rapide, *)$ *est un item graduel flou associé à une variation* $*$ *qui peut être une augmentation* $\leq$ *(i.e. ; "plus la vitesse est rapide") ou une diminution* $\geq$ *(i.e. ; "moins la vitesse est rapide").*

- **Motif graduel** :
  Un motif graduel ou itemset graduel est défini comme étant la combinaison de plusieurs items graduels, interprétés sémantiquement comme leurs conjonctions. Ainsi, le motif graduel $M = A^{\geq}B^{\leq}$ est interprété comme "Plus A et moins B", ce qui impose une contrainte de variation sur plusieurs items simultanément [Laurent *et al.*, 2009]. La taille d'un motif graduel est le nombre d'items le constituant.

- **Règle graduelle** :
  Une règle graduelle notée $M_1 \rightarrow M_2$, est définie par un couple de motifs graduels sur lequel est imposée une relation de causalité (*i.e.*, "*Plus la vitesse est rapide, plus le danger est grand*").

2.3 Approches et méthodes traitants des motifs et règles graduels

Plusieurs approches et sémantiques relatives à ce type de règles ont été proposées dans la littérature. Dans ce qui suit, nous passons en revue ces différentes approches.

2.3.1 Approche basée sur la régression :

Au meilleur de notre connaissance, [Hüllermeier, 2002] est la première proposition d'une méthode d'extraction automatique de règles graduelles. Nous distinguons dans cette proposition deux types de règles :

- Les règles de déviation (notées $A \rightarrow^d B$) expriment un écart significatif de la moyenne conditionnelle ;
- Les règles de tendance ou graduelles (notées $A \rightarrow^t B$) expriment la dépendance graduelle entre deux itemsets A et B.

Pour générer ces types de règles, il faut procéder ainsi :

1. *Construction d'un diagramme de contingence* :
   Pour évaluer les règles d'association classiques (*i.e* ; binaires) les mesures du support et de la confiance son employées. Ces mesures peuvent être obtenues à partir de la table de contingence comme illustré dans la table 2.1. En effet, le support de la règle d'association binaire $A \rightarrow B$ est donné par $\text{supp}(A \rightarrow B) = n_{11}/n$, sa confiance est $\text{conf}(A \rightarrow B) = n_{11}/n_{1\bullet}$. Pour traiter des données floues, l'auteur propose d'utiliser les diagrammes de contingence. Pour illustrer cette étape, nous allons prendre l'exemple de la base décrite dans le tableau 2.2. Cette base décrit trois personnes (p_1, p_2, p_3) associée chacune à trois attributs flous *"Age Jeune"*, *"Salaire Faible"* et *"Credit Elevé"* avec des degrés d'appartenance correspondants. Les diagrammes de contingence relatifs aux *"Plus l'âge est jeune, Plus le Salaire est faible"* et *"Plus l'âge est jeune, Plus le Crédit est Élevé "* sont respectivement illustrés par la figure 2.1 et 2.2.

	$B(y) = 0$	$B(y) = 1$	
$A(x) = 0$	n_{00}	n_{01}	$n_{0\bullet}$
$A(x) = 1$	n_{10}	n_{11}	$n_{1\bullet}$
	$n_{\bullet 0}$	$n_{\bullet 1}$	n

TABLE 2.1 – Table de contingence pour la règle classique $A \rightarrow B$

Personne	Age Jeune	Salaire Faible	Crédit Élevé
p_1	0,2	0,2	0,2
p_2	0,5	0,25	0,6
p_3	0,8	0,3	1

TABLE 2.2 – Base exemple 1.

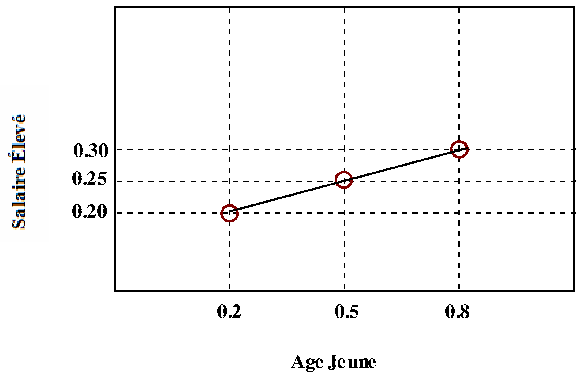

FIGURE 2.1 – Diagramme de contingence de la dépendance graduelle *"Plus l'Age est Jeune, plus le Salaire est Faible"* avec les paramètres de régression [0.167, 0.167] .

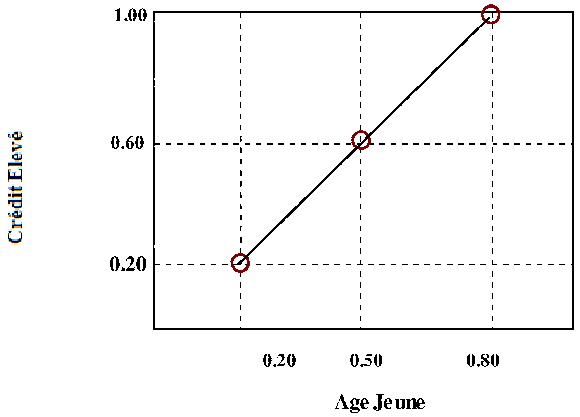

FIGURE 2.2 – Diagramme de contingence de la dépendance graduelle *"Plus l'Age est Jeune, plus le Crédit est Élevé"* avec les paramètres de régression [1.3, -0.67].

2. *Application d'une analyse de la régression linéaire :*
   Cette régression est appliquée sur le diagramme de contingence représentant l'ensemble des données. La validité de la règle est évaluée sur la base des coefficients (α et β) de la ligne de régression, qui se rapproche des points du diagramme de contingence et de la qualité de la régression Q telle que donnée par le coefficient de corrélation R^2.

3. *Génération de règles graduelles* :
   Les règles graduelles sont générées selon des seuils minimaux Q_{min} et α_{min} fixés par

l'utilisateur. En effet, une règle qui a une qualité de mesure Q inférieure à Q_{min} où la pente de la ligne de régression α est plus petite que α_{min}, ne sera pas générée. Dans le cas contraire, la règle $A \rightarrow B[\alpha, \beta]$ est générée. Notons que les règles générées contiennent un ou plusieurs attributs dans la partie prémisse et un seul dans la partie conclusion. Si la partie prémisse contient plusieurs attributs, alors l'auteur propose d'utiliser une conjonction logique modélisée à l'aide d'une t-norme [4].

2.3.2 Approche basée sur les dépendances graduelles :

L'extraction de règles graduelles en association avec la fouille de données a été proposée par [Berzal *et al.*, 2007]. Les auteurs utilisent l'algorithme APRIORI [Agrawal et Srikant, 1994] et donnent la première définition d'un item graduel.

Définition 15 *Un item graduel est un triplet de la forme [∗,X,A] avec :*

- $* \in \{\leq, \geq\}$
- X *un item*
- A *un ensemble flou défini sur l'item* X

Dans cette définition un item est associé à un des deux opérateurs de comparaison $\{\leq, \geq\}$. Le calcul du support d'un item graduel diffère de celui d'un item classique et il est basé sur la comparaison des degrés d'appartenance de l'item d'un objet de la base à un autre. Ainsi, la notion de support d'une règle graduelle est pour la première fois introduite. Dans le cas des itemsets classiques, la fréquence est la proportion d'objets de la base contenant l'itemset sur le nombre total d'objets de la base, alors que dans le cas de la gradualité, c'est la co-variation qui est mesurée. En réalité, cela revient à ordonner les valeurs de la base en fonction de l'itemset graduel, ce qui se traduit dans [Berzal *et al.*, 2007] par le calcul de tous les couples d'objets de la base. Ainsi, la fréquence est définie formellement par [Berzal *et al.*, 2007] de la manière suivante :

Définition 16 *Soit* $(*, X, A)$ *un item graduel et* $\mathcal{D}$ *une base de données, et* $GT^{\mathcal{D}}$ *tel que* $\forall o = (x, y)$, $o' = (x', y') \in \mathcal{D}$, $gt_{oo'} \in GT^{\mathcal{D}}$, *si* $A(x) * A(x')$ *Alors* $Freq([*, X, A]) = \frac{|\{gt_{oo'} \in GT^{\mathcal{D}} | A(x) * A(x')\}|}{|GT^{\mathcal{D}}|}$

Remarque 1 *D'après la définition 16, le support est le pourcentage de couples d'objets* $o = (x, y)$, $o' = (x', y')$ *dans* $\mathcal{D}$ *vérifiant* $A(x) * A(x')$. *Il décrit à quelle mesure un item peut apparaître dans une règle, avec un support dépassant un seuil minimal.*

4. Pour plus de détails sur les t-norme se référer à la section y du chapitre 3.

La propriété de complémentarité du support désignée par le terme "*antonymie*" par [Bouchon-Meunier, 1990] a été formalisée par [Berzal *et al.*, 2007] comme suit :

Propriété 2 *Soit c un opérateur de* $\{\leq, \geq\}$ *tel que* $c(\geq) = \leq$ *et* $c(\leq) = \geq$*, alors* $Supp([*, X, A]) = Supp[c(*), X, A]$

La propriété 2 permet de minimiser l'espace de recherche. En effet, la moitié des règles graduelles est générée alors que la deuxième moitié peut être automatiquement déduite à partir de la première. Afin d'extraire l'ensemble des itemsets graduels, [Berzal *et al.*, 2007] utilisent l'algorithme APRIORI. Le calcul du support impose la construction de la base des couples $GT^{\mathcal{D}}$ soit pour chaque itemset (chaque noeud de l'arbre des préfixes), soit dans sa totalité pour être simplement scannée à chaque passe. Cependant, l'utilisation de cette base rend la complexité de l'algorithme très élevée. Les auteurs proposent donc de partitioner les sous-ensembles flous en k partitions equi-depth [5], avec k fixé par l'utilisateur. Cela permet d'associer à chaque itemset un vecteur de longueur $k+1$, dont chaque indice contient le nombre d'objets de la base tel que $A(x) = j/k$. Ainsi, la complexité du calcul de la fréquence d'un itemset de longueur p passe de $\mathcal{O}(n^2)$ à $\mathcal{O}(n + k^p)$.

2.3.3 Approche basée sur la temporalité :

Dans [Fiot *et al.*, 2008a, Fiot *et al.*, 2008b] les auteurs proposent d'extraire des motifs séquentiels graduels. Rappelons qu'un motif séquentiel, contrairement aux règles d'associations, décrit la fréquence de certains comportements dans le temps. Ainsi, la gradualité peut être appliquée à deux niveaux :

- Au niveau des items, ce qui traduit une co-variation dans le même sens entre plusieurs items ;
- Au niveau de la liaison entre les itemsets, ce qui introduit les notions de "*puis rapidement*","*longtemps après*" ...

Les motifs séquentiels flous graduels obtenus sont de la forme "*Plus (moins)* X_{11} *est* A_{11} *et plus (moins)* X_{12} *est* A_{12} *précède une longue (courte) période de plus (moins)* X_{23} *est* A_{23} ... *précède une longue (courte) période de plus (moins)* X_{ij} *est* A_{ij}". Dans ce contexte, l'espace de recherche est infini. Ainsi, les auteurs dans [Fiot *et al.*, 2008b] procèdent en deux étapes :

- Construction d'une *base de variations* à partir de la base quantitative initiale. Cette base va contenir des degrés d'appartenance correspondant à des sous ensembles flous explicitant les variations entre toutes les dates. Par exemple, pour un objet ayant des valeurs renseignées

5. les partitions equi-depth contiennent le même nombre d'objets.

aux dates d_1, d_2 et d_3, la base contiendra les différences de valeur entre (d_1, d_2), (d_1, d_3) ainsi que (d_2, d_3). Cette base est appelée *base de tendance* et est construite à l'aide de l'algorithme TED [Fiot *et al.*, 2008b].

∗ Extraction de motifs séquentiels graduels, cette étape est elle même composée de deux sous-étapes :

1. Construction d'un graphe des séquences basé sur le modèle de [Masseglia *et al.*, 2004]. Chaque entrée de la base de tendance constitue un noeud, et un arc est présent si pour un même objet, une date est inférieure à une autre. Un tel graphe permet par la suite de traiter les séquences se chevauchant dans le temps.
2. Extraction des motifs séquentiels graduels à partir de ce graphe, en utilisant l'algorithme d'extraction de motifs séquentiels flous TotallyFuzzy [Fiot *et al.*, 2007].

2.3.4 Approche basée sur les ensembles de conflits :

Une autre définition du support d'un motif graduel a été proposée par [Di Jorio *et al.*, 2008]. En effet, étant donné un itemset graduel $s = (i_1^{*_1} \dots i_p^{*_p})$, son support est défini comme le nombre maximal d'objets $\{o_1, \dots, o_l\}$ pour lesquels il existerait une permutation π telle que $\forall\ j \in [1, l-1]$, $\forall\ k \in [1, p]$, nous avons $i_k(r_{\pi_j}) *_k i_k(r_{\pi_{j+1}})$. Plus formellement, le support d'un itemset graduel est défini par [Di Jorio *et al.*, 2008] comme suit :

Définition 17 *Soit $s = (i_1^{*_1} \dots i_n^{*_n})$ un itemset graduel et G_s l'ensemble des objets respectant s. Le support (ou la fréquence) de s est donnée par $Supp(s) = \frac{max(|G_s^i|)}{|\mathcal{O}|}$ où $G_s^i \subseteq G_s$ et $\mathcal{O}$ l'ensemble de touts les objets de la base de données.*

Les auteurs [Di Jorio *et al.*, 2008] proposent, pour calculer le support d'un itemset graduel, une première approche efficace en temps, mais non exhaustive. En effet, cette manière de calculer est basée sur une heuristique : le nombre d'items dans les règles est augmenté étape par étape en supprimant à chaque passe les plus grands ensembles d'objets de la base qui empêchent la gradualité appelé *ensemble de conflit.*

Exemple 5 *À partir de la base de données illustrée dans le tableau 2.3, nous pouvons extraire le motif ou l'itemset graduel $S_1 = A^{\geq}S^{\geq}C^{\leq}$. Il est obtenu par comparaison des valeurs prises de chaque attribut de la base (i.e., comparaison des valeurs d'un attribut d'un objet à un autre). Le support d'un motif graduel est donné par le nombre maximal d'objets vérifiant les variations de chaque item dans le motif.*

Personne	Age (A)	Salaire (S)	Crédit (C)
p_1	22	1200	4
p_2	28	1850	2
p_3	24	1200	3
p_4	35	2200	2
p_5	38	2000	0
p_6	44	3400	1
p_7	52	3400	5
p_8	41	5000	5

TABLE 2.3 – Base avec trois attributs quantitatifs

En effet, à partir du tableau 2.3, deux listes d'objets $l_1 = \{p_1, p3, p_2, p4, p_6\}$ *et* $l_2 = \{p_1, p_3, p_2, p_5\}$ *respectent les items graduels de* S_1*. Afin de choisir entre ces deux alternatives, les auteurs [Di Jorio* et al.*, 2008] proposent de prendre l'ensemble d'objets le plus représentatif (i.e ; contenant le plus d'objets) et le support de* S_1 *sera déduit à partir de la liste* l_1*. Ainsi, le support de* S_1 $Supp(S_1) = \frac{5}{8}$*. Ce qui signifie que* S_1 *est supporté par* 65% *des personnes.*

Le calcul du support d'un itemset graduel n'est pas aussi facile puisque nous devons choisir la meilleure solution (*i.e.*, liste d'objets la plus maximale) parmi plusieurs solutions possibles. Pour faire ce choix, les auteurs proposent une heuristique permettant de sélectionner la liste d'objets ayant un petit ensemble de conflit et qui peut ne pas être optimal pour un niveau donné, mais peut conduire à de meilleurs résultats au niveau suivant. Afin d'illustrer l'idée des ensembles de conflit, nous prenons l'itemset $A^{\geq}S^{\geq}$ et nous allons chercher à ordonner la base de la table 2.3 selon cet itemset. Ceci revient à considérer la base illustrée dans le tableau 2.4. La troisième colonne ($\mathcal{C}_i$) de ce dernier représente l'ensemble de conflit correspondant à chaque objet de la base après ordonnancement.

Personne	Age (A)	Salaire (S)	Crédit (C)	$\mathcal{C}_i$
p_1	22	1200	4	$\emptyset$
p_3	24	1200	3	$\emptyset$
p_2	28	1850	2	$\emptyset$
p_4	35	2200	2	$\{p_5\}$
p_5	38	2000	0	$\{p_4\}$
p_8	41	5000	5	$\{p_6, p_7\}$
p_6	44	3400	1	$\{p_8\}$
p_7	52	3400	5	$\{p_8\}$

TABLE 2.4 – Ordonnancement suivant $A^{\geq}$ et $S^{\geq}$.

Le choix de conserver p_8 entraîne la suppression de p_6 et p_7. Symétriquement, conserver p_6 et p_7 va entraîner la suppression de p_8 comme illustré dans le tableau 2.5.

Personne	Age (A)	Salaire (S)	Crédit (C)	$\mathcal{C}_i$
p_1	22	1200	4	$\emptyset$
p_3	24	1200	3	$\emptyset$
p_2	28	1850	2	$\emptyset$
p_4	35	2200	2	$\{p_5\}$
p_5	38	2000	0	$\{p_4\}$
~~p_8~~	~~41~~	~~5000~~	~~5~~	~~$\{p_6, p_7\}$~~
p_6	44	3400	1	~~$\{p_8\}$~~ $= \emptyset$
p_7	52	3400	5	~~$\{p_8\}$~~ $= \emptyset$

TABLE 2.5 – Conservation de p_6 et p_7.

2.3.5 Approche basée sur les graphes de précédence :

Dans [Di Jorio *et al.*, 2009a], les auteurs considèrent la même définition du support d'un itemset graduel proposée dans l'approche des ensembles de conflits et proposent une nouvelle méthode basée sur les graphes de précédence, appelée GRITE pour *GRadual ITemset Extraction*. Dans cette méthode, les données sont représentées par un graphe dont les noeuds sont définis comme des objets de la base, et les liens expriment les relations de précédence dérivés à partir des attributs pris en compte. Les auteurs adoptent une représentation binaire du graphe par une matrice. En effet, pour un itemset graduel $i_1^{*_1}, \ldots, i_p^{*_p}$, le bit correspondant à l'index du couple d'objets (o, o') est mis à 1 si $\forall j \in [1, p]$ $i_j(o) *_j i_j(o')$, zéro autrement. Le support de l'itemset considéré est défini comme la longueur du chemin le plus long dans le graphe.

Cette approche permet de générer d'une manière efficace des itemsets graduels de taille $p+1$ à partir des itemsets graduels de taille p. En effet, si s est itemset généré à partir de s' et s'', alors sa matrice correspondante M_s=$M_{s'}$ & $M_{s''}$ où le symbole & dénote l'opérateur bit à bit *ET*.

Exemple 6 *La figure 2.3 montre le graphe associé à l'itemset graduel* $S = A^{\geq}S^{\geq}C^{\leq}$ *noté* $\mathcal{L}_{A^{\geq}S^{\geq}C^{\leq}}$. *Chaque noeud du graphe représente un objet de la base de données et chaque flèche entre deux noeuds symbolise la validité de la corrélation des variations décrites par l'itemset* S. *Dans ce graphe, nous avons deux chemins représentant l'ensemble des solutions possibles* $\{(p_1p_3p_2p_4p_6), (p_1p_3p_2p_5)\}$.

Le tableau 2.6 montre la matrice binaire correspondante à l'itemset graduel S. Ce tableau est la projection en mémoire du graphe $\mathcal{L}_{A^{\geq}S^{\geq}C^{\leq}}$. En effet, il s'agit d'une matrice $n \times n$ avec n le nombre de sommets dans le graphe. S'il existe un chemin entre un sommet v et un autre v' alors le correspondant est mis à 1, 0 sinon. Le tableau 2.5 montre la fermeture transitive du graphe $\mathcal{L}_{A^{\geq}S^{\geq}C^{\leq}}$.

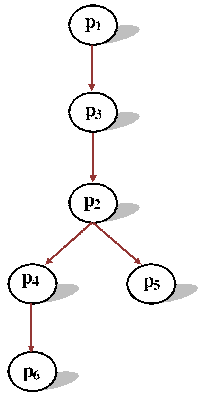

FIGURE 2.3 – $\mathcal{L}_{A^{\geq}S^{\geq}C^{\leq}}$

	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8
p_1	0	1	1	1	1	1	0	0
p_2	0	0	0	1	1	1	0	0
p_3	0	1	0	1	1	1	0	0
p_4	0	0	0	0	0	1	0	0
p_5	0	0	0	0	0	0	0	0
p_6	0	0	0	0	0	0	0	0
p_7	0	0	0	0	0	0	0	0
p_8	0	0	0	0	0	0	0	

TABLE 2.6 – Matrice binaire associée au graphe $\mathcal{L}_{A^{\geq}S^{\geq}C^{\leq}}$.

2.3.6 Approche basée sur le tau de Kendall :

Les auteurs [Laurent *et al.*, 2009], proposent de combiner l'approche de [Berzal *et al.*, 2007] et la définition du support proposée par [Di Jorio *et al.*, 2009a]. Les auteurs proposent d'utiliser les règles graduelles dans le contexte de découverte de corrélations de rangs multiples.

En effet, de par la définition du support, les règles graduelles obligent à conserver l'ordre des objets de la base d'un itemset à l'autre, ce qui les rend particulièrement adaptées à la recherche de rangs. Cependant, les fréquences telles que définies précédemment ne permettent pas de relier sémantiquement les règles au rang. Ainsi, les auteurs considèrent le Kendall *tau ranking correlation coefficient*, qui calcule non pas la longueur du plus long chemin, mais le nombre de paires de n-uplets ordonnables dans la base de données pour être en accord avec le motif graduel considéré. A partir des matrices binaires proposées dans [Di Jorio *et al.*, 2009a], il suffit de sommer le nombre de bits à 1. La méthode proposée se dégage alors de la recherche de la plus grande liste ordonnée d'objets.

Exemple 7 *Pour illustrer l'approche de [Laurent* et al.*, 2009], nous allons prendre l'exemple des quatres premières personnes du tableau 2.3. Le tableau 2.8 décrit les supports et les couples d'objets concordants pour les itemsets graduels* $A^{\geq}S^{\geq}$, $A^{\geq}C^{\leq}$, $S^{\geq}C^{\leq}$ *et* $A^{\geq}S^{\geq}C^{\leq}$.

	p_1	p_2	p_3	p_4	p_5	p_6
p_1	0	1	1	1	1	1
p_2	0	0	0	1	1	1
p_3	0	1	0	1	1	1
p_4	0	0	0	0	0	1
p_5	0	0	0	0	0	0
p_6	0	0	0	0	0	0

TABLE 2.7 – Matrice réduite du graphe de $\mathcal{L}_{A^{\geq}S^{\geq}C^{\leq}}$.

Itemset	Liste des paires concordants	Support
$A^{\geq}S^{\geq}$	$\mathcal{C}_{A^{\geq}S^{\geq}}\{(1,2),(1,3),(1,4),(2,4)\}$	$\frac{4}{6}$
$A^{\geq}C^{\leq}$	$\mathcal{C}_{A^{\geq}C^{\leq}}\{(1,2),(1,4),(2,4)\}$	$\frac{3}{6}$
$S^{\geq}C^{\leq}$	$\mathcal{C}_{S^{\geq}C^{\leq}}\{(1,2),(1,4),(2,4)\}$	$\frac{3}{6}$
$A^{\geq}S^{\geq}C^{\leq}$	$\mathcal{C}_{A^{\geq}S^{\geq}C^{\leq}}\{(1,2),(1,4),(2,4)\}$	$\frac{3}{6}$

TABLE 2.8 – Support et listes des couples d'objets concordants pour quelques motifs graduels.

2.4 Discussion

Dans ce chapitre, nous avons illustré différentes approches pour traiter et extraire les motifs et règles graduels. Dans toutes les approches, que nous venons d'examiner, le problème de la réduction de l'ensemble des motifs extraits n'est pas pris en considération. En effet, le nombre de motifs graduels reste très élevé tout comme le nombre de motifs classiques fréquents. Nous proposons donc de définir une approche permettant de travailler sur une représentation condensée des motifs graduels en se basant sur l'analyse formelle de concepts dont nous rappelons les principales caractéristiques dans le chapitre qui suit. Le tableau 2.9 présente un récapitulatif des différentes approches et méthodes traitant des règles graduelles.

Approche	**Principe d'extraction**	**Automatique/ Manuelle**	**Mesure d'évaluation**	**Domaine d'application**
Dieng et *al.* [Dieng *et al.*, 1993]	À partir d'un discours sémantique avec l'expert	Manuelle	–	Raisonnement automatique
	À partir d'arbres de décisions	Automatique		
Bouchon et *al.* [Bouchon-Meunier, 1990]	À partir d'un discours Règles d'inférence graduelles basées sur les implications floues	Semi-automatique (Système d'aide à la formulation de règles graduelles)	Evaluation des valeurs de vérité des implications floues	Système de raisonnement
Dubois et *al.* [Dubois et Prade, 1992]	Définition d'un contexte formel de règles d'inférence graduelles	Automatique	–	Système de raisonnement
Hüllermeier [Hüllermeier, 2002]	Diagramme de contingence	Automatique	Qualité et coefficients de la régression	Fouille de données
Berzal [Berzal *et al.*, 2007]	Dépendance graduelle avec considération des variations des degrés entre deux objets	Automatique	Pourcentage de couples d'objets vérifiant une variation (Support)	
Fiot et *al.* [Fiot *et al.*, 2008a]	Extraction de motifs séquentiels graduels flous en construisant une base de variation et un graphe de séquences	Automatique	–	
Di Jorio et *al.* [Di Jorio *et al.*, 2008]	Heuristique basée sur les ensembles de conflit	Automatique	Support : Cardinalité de la liste maximale d'objets respectant la gradualité	
[Di Jorio *et al.*, 2009a]	Recherche exhaustive basée sur les graphes de précédence			
Laurent et *al.* [Laurent *et al.*, 2009]	Extraction de couples concordants d'objets respectant la gradualité	Automatique	tau ranking correlation coefficient	

TABLE 2.9 – Calssification des méthodes traitant de la gradualité.

Chapitre 3

Représentations condensées

Sommaire

3.1 Introduction

Dans le domaine de la fouille de données, le problème d'extraction de motifs fréquents a suscité l'intérêt des chercheurs depuis son introduction par [Agrawal *et al.*, 1993]. Plusieurs algorithmes ont été introduits afin d'extraire l'ensemble des motifs fréquents. Néanmoins, le temps de leur extraction ainsi que leur nombre ingérable dans certains cas restent des inconvénients majeurs. Dans l'objectif de remédier à ces inconvénients, les chercheurs ont beaucoup travaillé sur l'extraction d'ensembles de motifs condensés (i.e. ; dont la cardinalité est plus réduite mais avec le même niveau de pertinence que l'ensemble de tous les motifs fréquents). Ces ensembles sont généralement appelés *représentations condensées* de motifs fréquents [Mannila et Toivonen, 1996]. Dans le cadre de l'extraction des itemsets, il existe de nombreuses représentations condensées telles que les représentations closes [Stumme *et al.*, 2000, Pasquier *et al.*, 1999, Zaki et Hsiao, 2002, Pei *et al.*, 2000], les représentations par itemsets maximaux [Bayardo, 1998, Burdick *et al.*, 2005, Lin et Kedem, 1998, Zaki *et al.*, 1997], les représentations par item-

sets non-dérivables [Calders et Goethals, 2002, Calders et Goethals, 2007], les représentations par ensembles libres [Boulicaut *et al.*, 2003], *etc.*

Dans le cadre de cette thèse, nous nous intéressons aux représentations condensées closes basées sur les motifs clos. Toutefois, nous nous proposons de décrire brièvement les autres représentations (*i.e.* ; celles basées sur les motifs maximaux, les motifs non-dérivables et les ensembles libres). Nous commençons par introduire quelques notions de bases utilisées dans ces différentes représentations.

Définition 18 (Contexte d'extraction) *Un contexte d'extraction (ou contexte formel) est un triplet $\mathcal{K} = (\mathcal{O}, \mathcal{I}, \mathcal{R})$, décrivant deux ensembles finis $\mathcal{O}$ (ensemble d'objets ou de transactions) et $\mathcal{I}$ (ensemble d'attributs ou d'items) et une relation $\mathcal{R}$ entre $\mathcal{O}$ et $\mathcal{I}$ telle que $R \subseteq O \times I$. Un couple $(o, i) \in \mathcal{R}$ désigne que l'objet $o \in \mathcal{O}$ possède l'attribut $i \in \mathcal{I}$ (noté $o\mathcal{R}i$).*

Définition 19 (Support d'un motif) *Soit $\mathcal{K} = (\mathcal{O}, \mathcal{I}, \mathcal{R})$ un contexte d'extraction, un sous ensemble X de $\mathcal{I}$ avec $k = |X|$ est un motif ou itemset de taille k. Une transaction (ou un objet) supporte le motif X si elle contient tous les items le composant. Le support d'un motif X représente le pourcentage de transactions de la base qui le supportent. Le support de X est donné par :*

$$Support(X) = \frac{|\{o \in \mathcal{O} | \forall i \in X, (o,i) \in \mathcal{R}\}|}{|\mathcal{O}|}$$

Un motif est dit *fréquent* s'il a un support supérieur à une valeur minimale *minSup* fixée par l'utilisateur.

Définition 20 (Règle d'association) *Une règle d'association est de la forme $R : X \rightarrow Y$ avec $X \cap Y = \emptyset$. $X, Y \in \mathcal{I}$ sont appelés respectivement la prémisse et la conclusion de la règle.*

Une règle d'association est principalement évaluée par deux mesures :

- Le **support** du motif XY permettant de connaître le pourcentage d'objets de la base contenant les motifs X et Y.
- La **confiance** permettant de connaître la probabilité qu'un objet contenant X contienne aussi Y. Cette mesure est donnée par $\text{Conf}(R : X \rightarrow Y) = \frac{Support(XY)}{Support(X)}$

La problématique d'extraction de règles d'associations consiste à trouver à partir d'une base de données l'ensemble de toutes les règles d'association dont le support et la confiance respectent des seuils de support minimal (*minSupp*) et de confiance minimale (*minConf*) fixés par l'utilisateur. Cette problématique est composée de deux étapes : extraction des motifs fréquents puis génération des règles d'association. L'étape la plus coûteuse en terme de temps et de mémoire est celle de l'extraction de l'ensemble de motifs fréquents.

Afin d'optimiser cette étape, des recherches ont été menées pour l'extraction des représentations condensées de l'ensemble de tous les motifs fréquents. La figure 3.1 illustre les étapes de génération des représentations condensées à partir d'une base de données.

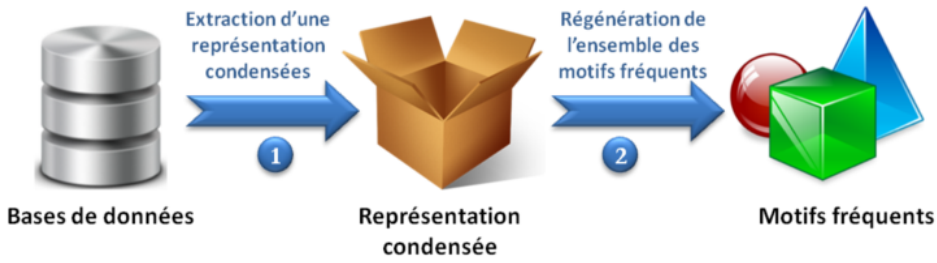

FIGURE 3.1 – Processus d'extraction de représentations condensées.

Dans un cadre plus formel une représentation condensée est définie comme suit :

Définition 21 (Représentation condensée) *Soit $\mathcal{MF}(\mathcal{D},\gamma)$ l'ensemble de motifs fréquents pouvant être extrait à partir de la base de données $\mathcal{D}$ correspondant à un contexte d'extraction $\mathcal{K} = (\mathcal{O},\mathcal{I},\mathcal{R})$ et γ la contrainte du support minimal. Une représentation condensée $\mathcal{RC}(\mathcal{D},\gamma)$ est un ensemble de motifs fréquents tel que :*

$$\mathcal{RC}(\mathcal{D},\gamma) \subset \mathcal{MF}(\mathcal{D},\gamma)$$

L'ensemble des motifs fréquents $\mathcal{MF}(\mathcal{D},\gamma)\backslash\mathcal{RC}(\mathcal{D},\gamma)$ est généré d'une manière efficace à partir de $\mathcal{RC}(\mathcal{D},\gamma)$ sans avoir accès à la base de données $\mathcal{D}$.

Une représentation condensée permettant de déduire tous les motifs fréquents, mais pas leurs valeurs de support, est dite *non-informative*. Dans le cas contraire, elle est dite *informative* et nous distinguons deux cas de figure : si cette valeur du support est déduite avec exactitude alors nous parlons de représentation *exacte*. Si nous ne pouvons qu'approximer la valeur du support alors cette représentation est dite *approximative*.

Ainsi, les représentations condensées permettent d'éviter l'explosion combinatoire du nombre de motifs fréquents, notamment dans certains contextes particuliers (*e.g.*, les bases denses). L'idée est de calculer un ensemble, $\mathcal{RC} \subseteq \mathcal{P}(\mathcal{I})$, le plus concis possible et à partir duquel nous pouvons générer l'ensemble de tous les motifs fréquents sans accéder au contexte d'extraction. Une solution consiste à calculer les bordures positive $\mathcal{BD}^{+}(\mathcal{MF}(\mathcal{D},\gamma))$ et négative

$\mathcal{BD}^-(\mathcal{MF}(\mathcal{D},\gamma))$. La bordure négative contient l'ensemble des plus petits motifs non-fréquents (par rapport à la relation $\subseteq$ entre les motifs) alors que la bordure positive contient l'ensemble des plus grands motifs fréquents.

Définition 22 *Les bordures positive et négative sont respectivement définies par :*

$$\mathcal{BD}^+(\mathcal{MF}(\mathcal{D},\gamma)) = \{I \in \mathcal{MF}(\mathcal{D},\gamma) | \forall J \in \mathcal{P}(\mathcal{I}) : I \subset J \Rightarrow J \notin \mathcal{MF}(\mathcal{D},\gamma)\}$$
$$\mathcal{BD}^-(\mathcal{MF}(\mathcal{D},\gamma)) = \{I \in \mathcal{P}(\mathcal{I}) \backslash \mathcal{MF}(\mathcal{D},\gamma) | \forall J \in \mathcal{P}(\mathcal{I}) : J \subset I \Rightarrow J \in \mathcal{MF}(\mathcal{D},\gamma)\}$$

Outre l'informativité, les représentations condensées se caractérisent aussi par :

- La taille : plus elle est petite plus elle est pertinente (*i.e.* ; la plus petite étant la meilleure) ;
- L'efficacité des algorithmes permettant de la générer : un algorithme d'extraction d'une représentation condensée doit être plus efficace en termes de temps et d'espace mémoire qu'un algorithme d'extraction de l'ensemble de tous les motifs fréquents ;
- La complétude et l'efficacité de la régénération : la régénération est l'étape permettant de déduire tous les motifs fréquents à partir de la représentation condensée. Cette étape doit être efficace en terme de temps de réponse.

3.2 Quelques représentations condensées

Dans cette partie, nous présentons quelques représentations condensées proposées dans le cadre d'extraction des itemsets fréquents.

3.2.1 Représentation par itemsets maximaux

L'extraction d'une représentation condensée basée sur les itemsets maximaux a été proposée par [Bayardo, 1998]. La définition de cette représentation est basée sur la bordure positive $\mathcal{BD}^+(\mathcal{MF}(\mathcal{D},\gamma))$. En effet, un itemset I est dit *maximal* s'il n'existe pas un itemset J tel que $I \subset J$. En effet, $\mathcal{BD}^+(\mathcal{MF}(\mathcal{D},\gamma)) \subset \mathcal{MF}(\mathcal{D},\gamma)$ et tous les sous-ensembles des itemsets fréquents maximaux sont fréquents et peuvent être déduits sans retour aux données. Cependant, cette représentation est approximative, puisqu'elle ne permet pas de déduire les valeurs exactes des supports des itemsets régénérés.

Pour illustrer cette représentation, nous considérons la base donnée par la Table 3.1 comme exemple :

Exemple 3 *À partir de la base de données $\mathcal{D}$ de la Table 3.1 et avec $\gamma = 3$ comme support minimal, l'ensemble des itemsets fréquents est :*

ID Transaction	Item
1	$\{a, b, c\}$
2	$\{b, c\}$
3	$\{b, c, e\}$
4	$\{b, c, d, e\}$
5	$\{a, b, c, d, e\}$

TABLE 3.1 – Base de données $\mathcal{D}$ avec 5 transactions.

$$\mathcal{MF}(\mathcal{D}, \gamma) = \{(b, 5), (c, 5), (e, 3), (bc, 5), (be, 3), (ce, 3), (bce, 3)\}$$

L'ensemble de motifs maximaux fréquents (i.e., la représentation condensée) est donné par :

$$\mathcal{RC}(\mathcal{D}, \gamma) = \{(bce, 3)\}$$

À partir de $\mathcal{RC}(\mathcal{D}, \gamma)$, nous ne sommes pas en mesure de déduire avec exactitude les valeurs du support des itemsets fréquents contenus dans (bce). Toutefois nous pouvons approximer ces valeurs en s'appuyant sur la propriété d'anti-monotonie du support.
En effet, nous pouvons déduire que le support de (ce) est plus grand que celui de (bce) puisque $(ce) \subset (bce)$. Pour déterminer la valeur exacte du support des itemsets fréquents dans $\mathcal{MF}(\mathcal{D}, \gamma)$, il faut faire un balayage du contexte. Plusieurs algorithmes basés sur cette approche ont été proposés, nous pouvons citer les algorithmes PINCER-SEARCH [Lin et Kedem, 1998], MAXCLIQUE et MAXECLAT [Zaki *et al.*, 1997], et MAX-MINER [Bayardo, 1998].

3.2.2 Représentation par itemsets non-dérivables

La notion d'itemset non-dérivable a été introduite par Calders et al. [Calders et Goethals, 2002]. Les itemsets non-dérivables s'appuient sur un ensemble de règles de déduction afin de déduire des bornes pour la fréquence des itemsets. Une règle de déduction du support est définie comme suit :

Définition 23 *Soient le contexte d'extraction $\mathcal{K} = (\mathcal{O}, \mathcal{I}, \mathcal{R})$ et un itemset $I \subseteq \mathcal{I}$. Les inéquations suivantes présentent la règle de déduction $\mathcal{R}_X(I)$ permettent de borner le support de I en fonction de ses sous-ensembles $X \subseteq I$:*

$$Support(I) \leq \sum_{X \subseteq J \subset I} (-1)^{|I \backslash J|+1} Support(J), si |I \backslash X| impair$$

$$Support(I) \geq \sum_{X \subseteq J \subset I} (-1)^{|I \backslash J|+1} Support(J), si |I \backslash X| pair$$

Selon la parité de $|I \backslash X|$, la borne sera supérieure ou inférieure. En effet, si $|I \backslash X|$ est impaire, alors la borne est supérieure, et elle serait inférieure autrement. Ceci dit que si nous disposons du

support de tous les sous-ensembles de I, alors nous pouvons déduire plusieurs bornes supérieures et inférieure du support de I en utilisant $\mathcal{R}_X(I)$.

Nous parlons d'itemset dérivable I lorsque sa plus petite borne supérieure ($LB(I)$) est égale à sa plus grande borne inférieure ($UB(I)$). Dans ce cas, le support de I= $LB(I) = UB(I)$. Un itemset I' est donc non-dérivables lorsque $LB(I') \neq UB(I')$.
Les auteurs [Calders et Goethals, 2002] montrent que la non-dérivabilité est anti-monotone par rapport à la relation d'inclusion des attributs. Ainsi, si I est dérivable, alors ces sur-ensembles le sont aussi.

Exemple 4 *Considérons la base de données $\mathcal{D}$ représentée dans la table 3.1 avec un support minimal $\gamma = 3$. L'ensemble d'itemsets non-dérivables fréquents que nous notons $\mathcal{NF}(\mathcal{D}, \gamma)$ est :*

$$\mathcal{NF}(\mathcal{D}, \gamma) = \{(b, 5), (c, 5), (e, 3)\}$$

L'itemset (bc) ne figure pas dans $\mathcal{NF}(\mathcal{D}, \gamma)$ car il est dérivable. Ainsi, en sélectionnant le sous-itemset vide $\emptyset$ et le sous-itemset (b), nous avons les règles suivantes :

1. $-Support(\emptyset) + Support(b) + Support(c) \leq Support(bc) \Rightarrow 5 \leq Support(bc)$.
2. $Support(b) \geq Support(bc)$.

D'après les règles (1) et (2), nous pouvons conclure que $5 \leq Support(bc) \leq 5$. Ainsi, nous avons pu déduire le support de l'itemset (bce) avec exactitude (*i.e.*, (bce) dérivable) de même pour les itemsets (be), (ce) et (bce).

La représentation condensée par itemsets non-dérivables fréquents constitue une représentation informative exacte de l'ensemble des itemsets fréquents.

3.2.3 Représentation par itemsets δ-libres

Les notions d'itemset δ-libre et de règles d'association δ-forte ont été introduites par [Boulicaut *et al.*, 2000, Boulicaut *et al.*, 2003]. Une règle d'association δ-forte est une règle, dont le nombre d'erreurs effectuées dans la base est borné par un entier $\delta > 0$. Formellement, une règle δ-forte et un itemset δ-libre sont définis par :

Définition 24 *Une règle d'association δ-forte est une règle d'association de la forme $R : I \rightarrow^{\delta} a$, avec $I \subseteq \mathcal{I}$, $a \in \mathcal{I} \setminus I$ et δ un entier naturel. La règle R est valide si $Support(I) - Support(I \cup \{a\}) < \delta$.*
Un itemset $J \subseteq \mathcal{I}$ est un itemset δ-libre si et seulement s'il n'existe pas une règle δ-forte valide $R : I \rightarrow a$ telle que $I \subset J$, $a \in J$ et $a \notin I$. En d'autres termes, un itemset J est δ-libre, avec $\delta \in IN$, si pour chacun de ses sous-itemsets $I \subset J$, $Support(J) + \delta < Support(I)$.

L'ensemble des itemsets δ-libres fréquents, noté par $F_\delta(\mathcal{D})$, permet d'approximer le support des itemsets non δ-libres. En effet, si un itemset I n'est pas δ-libre, son support peut être approximé par le sous-itemset ayant le plut petit support dans l'ensemble des sous-itemsets δ-libres de I. En effet, étant donné l'itemset I fréquent non δ-libre, par définition il existe une règle δ-forte $R : J \rightarrow^\delta a$ telle que $J \subseteq I$ et $a \in I$. De plus, si R est δ-forte valide, alors $I \setminus \{a\}$ est aussi valide. Ce qui nous permet d'approximer le support de I par le support de l'itemset fréquent $I \setminus \{a\}$. En effet, nous avons $Support(I\setminus\{a\}) - \delta \leq Support(I)$, ce qui implique que $Support(I\setminus\{a\})$ est une borne supérieure de $Support(J)$. Dans le cas où il existe plusieurs bornes supérieures, la la plus petite valeur du support des itemsets δ-libres fréquents inclus dans I sera choisie. Ainsi, l'ensemble $F_\delta(\mathcal{D})$ permet d'approximer la fréquence des itemsets fréquents non δ-libres mais il ne permet pas de confirmer le statut de fréquence d'un itemset. Pour y remédier les auteurs proposent de compléter la représentation en rajoutant une bordure qui est l'ensemble des itemsets non-fréquents δ-libres minimaux.

Exemple 5 *Soient la base de données $\mathcal{D}$ représentée dans la table 3.1, un support minimal γ = 3 et δ= 1. L'ensemble des itemsets 1-libres fréquents est :*

$$F_\delta(\mathcal{D}) = \{(b,5),(c,5),(e,3)\}$$

Les itemsets (bc), (be) et (ce) ne sont pas pas 1-libres puisque : $Support(bc) + 1 > Support(c)$, $Support(be) + 1 > Support(e)$ et $Support(ce) + 1 > Support(e)$. De même, (bce) n'est pas 1-libre car, grace à la propriété d'anti-monotonie des itemsets δ-libres, tous ses sous-itemsets ne le sont pas aussi.

3.2.4 Représentation condensée par itemsets clos

Pasquier *et al.* [Pasquier *et al.*, 1998] ont proposé une nouvelle représentation condensée basée sur les itemsets *clos*. Cette approche est basée sur le fait que l'ensemble de motifs clos fréquents est un ensemble générateur de l'ensemble de motifs fréquents [Pasquier, 2000]. L'approche d'extraction de motifs clos fréquents repose sur les fondements mathématiques de l'*analyse formelle de concepts* [Wille, 1982b]. La section suivante est consacrée à la présentation de ces fondements.

Analyse formelle de concepts

- **Notion d'ordre partiel** :

  Soit E un ensemble. *Un ordre partiel* sur l'ensemble E est une relation binaire $\leq$ sur les éléments de E, tel que pour $x, y, z \in E$ nous avons les propriétés suivantes [Davey et Priestley, 2002] :

1. *Réflexivité* : $x \leq x$
2. *Anti-symétrie* : $x \leq y$ et $y \leq x \Rightarrow x = y$
3. *Transitivité* : $x \leq y$ et $y \leq z \Rightarrow x \leq z$

Un ensemble E doté d'une relation d'ordre $\leq$, noté $(E,\leq)$, est appelé *ensemble partiellement ordonné* [Davey et Priestley, 2002].

- **Relation de couverture** :

  Soient E un ensemble ordonné $(E,\leq)$ et x, y deux éléments de E. La relation de couverture entre les éléments de E, notée $\prec$, est définie par $x \prec y$ si et seulement si $x \leq y$ et tel qu'il n'existe pas d'élément $z \in E$ tel que $x \leq z \leq y$ pour $z \neq x$ et $z \neq y$.

  Si $x \prec y$, nous disons que y couvre x ou bien que y est un successeur immédiat de x (et donc x est couvert par y ou x est un prédécesseur immédiat de y) [Davey et Priestley, 2002].

- **Join et Meet** :

  Soit un sous-ensemble $S \subseteq E$ de l'ensemble partiellement ordonné (E,$\leq$). Un élément $u \in E$ est un *majorant*, ou *borne-sup*, de S si pour tout élément $s \in S$, nous avons $s \leq u$. L'ensemble des majorants de S est noté *UB(S)*. D'une manière duale, un élément $v \in E$ est un *minorant*, ou *borne-inf*, de S si pour tout élément $s \in S$, nous avons $v \leq s$. L'ensemble des minorants de S est noté *LB(S)* [Ganter et Wille, 1999].

  $$UB(S) = \{u \in E \mid \forall s \in S,\, s \leq u\}$$
  $$LB(S) = \{v \in E \mid \forall s \in S,\, v \leq s\}$$

  Le *plus petit* majorant d'un ensemble S, s'il existe, est le plus petit élément de l'ensemble $UB(S)$ des majorants de S. Cet élément est noté $Join(S)$ ($\vee S$). D'une manière duale, le *plus grand* minorant d'un ensemble S, s'il existe, est le plus grand élément de l'ensemble $LB(S)$ des minorants de S. Cet élément est noté $Meet(S)$ ($\wedge S$) [Ganter et Wille, 1999].

  Exemple 8 *Soit le treillis de concepts illustré par la figure 3.2. Nous avons :*
  $UB(\{a,b\}) = \{\top, h, i, f\}$
  $LB(\{e,f\}) = \{a, \bot\}$

- **Treillis complet** :

  Un ensemble partiellement ordonné $(E,\leq)$ non vide est un *treillis* si pour tout couple d'éléments $(x,y) \in E$, l'ensemble $\{x,y\}$ possède un plus petit majorant, noté $x \vee y$, et un plus grand minorant, noté $x \wedge y$.

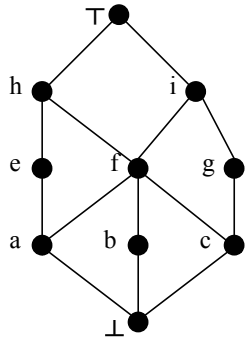

FIGURE 3.2 – Join - Meet

L'ensemble partiellement ordonné $(E,\leq)$ est un *treillis complet* si pour tout sous-ensemble $S \subseteq E$, les éléments Join(S) et Meet(S) existent [Davey et Priestley, 2002].

Théorème 3 ***Théorème fondamental de l'analyse formelle de concepts*** *[Ganter et Wille, 1999] : L'ensemble des concepts formels, extrait à partir d'un contexte formel, constitue un treillis complet quand les concepts formels sont ordonnés par inclusion des extensions (ou par inclusion des intentions).*

- **Join-irreductible et Meet-irreductible** :

  Pour un élément l d'un treillis complet L, nous définissons [Davey et Priestley, 2002] :

$$l_* = \vee\{x \in L | x < l\}$$
$$l^* = \wedge\{x \in L | l < x\}$$

  Un élément l est dit *Join-irréductible* s'il couvre un élément unique. D'une manière duale, l est dit *Meet-irréductible*, s'il est couvert par un seul élément.

  L'ensemble des éléments Join-irréductibles de L est noté par $\mathcal{J}(L)$ et l'ensemble des éléments Meet-irréductible de L est noté par $\mathcal{M}(L)$. Chacun de ces ensembles hérite de la relation d'ordre de L [Davey et Priestley, 2002].

  Exemple 9 *Dans la figure 3.2, b est un élément Meet-irréductible, alors que e est un élément Join-irréductible. Dans la figure 3.3, nous avons* $\mathcal{J}(L) = \{b, c\}$ *et* $\mathcal{M}(L) = \{e, f\}$.

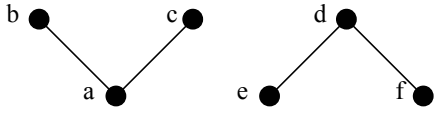

FIGURE 3.3 – Join-irréductible - Meet-irréductible

Correspondance de Galois

- **Opérateurs de cloture** :

  Soit un ensemble partiellement ordonné $(E,\leq)$. Une application ψ de $(E,\leq)$ dans $(E,\leq)$ est appelée un *opérateur de cloture*, si et seulement si elle possède les propriétés suivantes. Pour tout sous-ensemble $S, S' \subseteq E$ [Davey et Priestley, 2002] :

  1. *Isotonie* : $S \leq S' \Rightarrow \psi(S) \leq \psi(S')$
  2. *Extensivité* : $S \leq \psi(S)$
  3. *Idempotence* : $\psi(\psi(S)) = \psi(S)$

  Étant donné un opérateur de cloture ψ sur un ensemble partiellement ordonné $(E,\leq)$, un élément $x \in E$ est un élément *clos* si l'image de x par l'opérateur de cloture ψ est égale à lui-même, *i.e.*, $\psi(x) = x$ [Davey et Priestley, 2002].

- **Opérateurs de la correspondance de Galois** :

  Soit un contexte d'extraction $\mathcal{K} = (\mathcal{O}, \mathcal{I}, \mathcal{R})$. Soit l'application ϕ, de l'ensemble des parties de $\mathcal{O}$[6] dans l'ensemble des parties de $\mathcal{I}$, qui associe à un ensemble d'objets $O \subseteq \mathcal{O}$ l'ensemble des items $i \in \mathcal{I}$ communs à tous les objets $o \in O$ [Ganter et Wille, 1999] :

  $$f : 2^{\mathcal{O}} \rightarrow 2^{\mathcal{I}}$$
  $$f(O) = \{i \in \mathcal{I} | \forall o \in O \wedge o\mathcal{R}i \ \}$$

  Soit l'application g, de l'ensemble des parties de $\mathcal{I}$ dans l'ensemble des parties de $\mathcal{O}$, qui associe à tout ensemble d'items (communément appelé itemset) $I \subseteq \mathcal{I}$ l'ensemble des objets $o \subseteq \mathcal{O}$ contenant tous les items $i \in I$ [Ganter et Wille, 1999] :

  $$g : 2^{\mathcal{I}} \rightarrow 2^{\mathcal{O}}$$
  $$g(I) = \{o \in \mathcal{O} | \forall i \in I \wedge o\mathcal{R}i \ \}$$

  Le couple d'applications $(f,\ g)$ est une *correspondance de Galois* entre l'ensemble des parties de $\mathcal{O}$ et l'ensemble des parties de $\mathcal{I}$ [Barbut et Monjardet, 1970, Ganter et Wille, 1999]. Etant donné une correspondance de Galois, les propriétés suivantes sont vérifiées quelques soient $I,\ I_1,\ I_2 \subseteq \mathcal{I}$ et $O,\ O_1,\ O_2 \subseteq \mathcal{O}$ [Ganter et Wille, 1999] :

  1. $I_1 \subseteq I_2 \Rightarrow g(I_2) \subseteq g(I_1)$;
  2. $O_1 \subseteq O_2 \Rightarrow f(O_2) \subseteq f(O_1)$;

6. L'ensemble des parties d'un ensemble d'éléments $\mathcal{O}$, constitué de tous les sous-ensembles de $\mathcal{O}$, est noté $2^{\mathcal{O}}$.

3. $O \subseteq g(I) \Leftrightarrow I \subseteq f(O) \Leftrightarrow (I,O) \in \mathcal{R}$.

- **Cloture de la correspondance de Galois** :

  Nous considérons les ensembles des parties $2^{\mathcal{I}}$ et $2^{\mathcal{O}}$ dotés de la relation d'inclusion $\subseteq$, *i.e.*, les ensembles partiellement ordonnés $(2^{\mathcal{I}}, \subseteq)$ et $(2^{\mathcal{O}}, \subseteq)$. Les opérateurs $\psi = f \circ g$ de $(2^{\mathcal{I}}, \subseteq)$ dans $(2^{\mathcal{I}}, \subseteq)$ et $\psi' = g \circ f$ de $(2^{\mathcal{O}}, \subseteq)$ dans $(2^{\mathcal{O}}, \subseteq)$ sont des *opérateurs de cloture de la correspondance de Galois* [Barbut et Monjardet, 1970, Ganter et Wille, 1999]. Étant donné une correspondance de Galois, les propriétés suivantes sont vérifiées quelques soient I, I_1, $I_2 \subseteq \mathcal{I}$ et O, O_1, $O_2 \subseteq \mathcal{O}$ [Ganter et Wille, 1999] :

1. $I \subseteq \psi(I)$	1'. $O \subseteq \psi'(O)$
2. $\psi(\psi(I))=\psi(I)$	2'. $\psi'(\psi'(O))=\psi'(O)$
3. $I_1 \subseteq I_2 \Rightarrow \psi(I_1) \subseteq \psi(I_2)$	3'. $O_1 \subseteq O_2 \Rightarrow \psi'(O_1) \subseteq \psi'(O_2)$
4. $\psi(g(I)) = g(I)$	4'. $\psi(f(O)) = f(O)$

- **Treillis de concepts formels (de Galois)** :

  Étant donné un contexte d'extraction $\mathcal{K}=(\mathcal{O},\mathcal{I},\mathcal{R})$, l'ensemble de concepts formels $\mathcal{C}_{\mathcal{K}}$ est un treillis complet $\mathcal{L}_{\mathcal{C}_{\mathcal{K}}} = (\mathcal{C}_{\mathcal{K}}, \leq)$, appelé *treillis de concepts formels (de Galois)*, quand l'ensemble $\mathcal{C}_{\mathcal{K}}$ est considéré avec la relation d'inclusion entre les itemsets [Barbut et Monjardet, 1970, Ganter et Wille, 1999]. La relation d'ordre partiel entre des concepts formels est définie comme suit [Ganter et Wille, 1999] : $\forall\ c_1 = (O_1,I_1)$, $c_2 = (O_2,I_2) \in C_{\mathcal{K}}$, $c_1 \leq c_2 \Longleftrightarrow I_2 \subseteq I_1$ ($\Leftrightarrow O_1 \subseteq O_2$) avec I_1, $I_2 \subseteq \mathcal{I}$ et O_1, $O_2 \subseteq \mathcal{O}$.

Motifs Clos et leurs générateurs minimaux

Dans cette section, nous définissons :
- les motifs clos qui peuvent être ordonnés sous la forme d'un treillis des motifs clos.
- les motifs clos *fréquents* qui peuvent être ordonné sous la forme d'un treillis d'Iceberg de Galois.
- les générateurs minimaux.

- **Motif clos** :

  Étant donné l'opérateur de cloture de la correspondance de Galois ψ, un itemset $l \subseteq \mathcal{I}$ tel que $\psi(l) = l$ est appelé *motif clos*[7]. Un motif clos est donc un ensemble maximal d'items communs à un ensemble d'objets [Pasquier, 2000].

7. Dans la littérature le terme itemset clos est fréquemment utilisé.

Exemple 10 *Soit le contexte $\mathcal{K}$ illustré par la figure 3.4, le motif {BCE} est un motif clos puisqu'il est l'ensemble maximal d'items communs aux objets {2,3,5}. L'itemset {BC} n'est pas un clos car il n'est pas un ensemble maximal d'items communs à certains objets : tous les objets contenant les items B et C, i.e., les objets 2,3 et 5 contiennent également l'item E.*

	A	B	C	D	E
1	×		×	×	
2		×	×		×
3	×	×	×		×
4		×			×
5	×	×	×		×

FIGURE 3.4 – Exemple de contexte formel

- **Ensemble de motifs clos** :

  Soit un contexte d'extraction $\mathcal{K} = (\mathcal{O},\mathcal{I},\mathcal{R})$ et l'opérateur de cloture de la correspondance de Galois ψ. L'ensemble $\mathcal{IF}$ des motifs clos dans le contexte $\mathcal{K}$ est défini comme suit [Pasquier, 2000] :

  $$\mathcal{IF} = \{l \subseteq \mathcal{I} \mid l \neq \emptyset \wedge \psi(l)=l\}.$$

  Le plus petit (minimal au sens de l'inclusion) motif clos contenant un itemset l est obtenu par l'application de l'opérateur ψ à l [Pasquier, 2000].

- **Treillis des motifs clos** :

  L'opérateur de cloture ψ induit une relation d'équivalence sur l'ensemble de parties de $\mathcal{I}$, *i.e.*, l'ensemble de parties est partionné en des sous-ensembles disjoints, appelés aussi *classes d'équivalence*. Dans chaque classe, tous les éléments possèdent la même valeur de support. Les générateurs minimaux d'une classe sont les éléments incomparables (selon la relation d'inclusion) les plus petits, tandis que l'itemset fermé est l'élément le plus large de cette classe. Ces classes d'équivalence sont ordonnées sous forme d'un treillis de concepts formels (de Galois) où chaque concept formel est dans ce cas un motif clos. Ainsi, le couple $\mathcal{L}_{\mathcal{IF}} = (\mathcal{IF},\subseteq)$ est un treillis complet appelé *treillis des motifs clos* [Pasquier, 2000].

- **Motifs clos fréquents** :

  Un motif clos l est dit fréquent si son *support relatif*, $\text{Supp}(l) = \frac{|g(l)|}{|\mathcal{O}|}$, excède un seuil minimum fixé par l'utilisateur noté *minSup* [Pasquier, 2000]. Notons que $|g(l)|$ est appelé *support absolu* de l.

- **Treillis d'Iceberg de Galois** :

  Quand nous considèrons seulement l'ensemble des motifs clos *fréquents* ordonnés par la relation d'inclusion ensembliste, la structure obtenue $(\hat{\mathcal{L}}, \subseteq)$ préserve seulement l'opérateur *Join*. Cette structure forme un semi-treillis supérieur et elle est désignée par *treillis d'Iceberg de Galois* [Bastide *et al.*, 2000b, Stumme *et al.*, 2002, Valtchev *et al.*, 2002].

- **Générateur minimal** :

  Un itemset $g \subseteq \mathcal{I}$ est dit *générateur minimal* d'un motif clos f, si et seulement si $\psi(g) = f$ et il n'existe pas $g_1 \subseteq \mathcal{I}$ tel que $g_1 \subset g$ [Bastide *et al.*, 2000a]. L'ensemble GM_f des générateurs minimaux d'un motif clos f est défini comme suit :

$$GM_f = \{\ g \subseteq \mathcal{I} \mid \psi(g)=f \wedge \nexists\ g_1 \subset g \text{ tel que } \psi(g_1) = f\ \}.$$

3.3 Conclusion

Dans ce chapitre nous avons passé en revue les différentes approches et propositions qui semblent être les plus marquantes dans la littérature dans le cadre d'extraction des représentations condensées.

En raison de la définition même des représentations condensées, toute application ayant besoin des itemsets fréquents trouve un intérêt dans les représentations condensées. En effet, les itemsets fréquents sont générés plus rapidement à partir des représentations condensées notamment à partir de contextes difficiles (*e.g.* ; données denses) ou avec un seuil de support minimal très bas. Les représentations condensées qui ont attiré le plus d'applications diverse sont celles basées sur les propriétés de cloture. C'est pour cette raison que nous focalisons sur l'extraction de représentation condensée basée sur la cloture en association avec l'aspect graduel et flou des données.

Chapitre 4

Nouvelles définitions de la correspondance de Galois pour les motifs flous et graduels

Sommaire

Dans les chapitres précédents nous avons présenté le cadre bibliographique de cette thèse. Dans ce chapitre, nous allons présenter nos différentes contributions pour l'extraction de motifs graduels et flous. Ainsi, dans la première section, nous introduisons une nouvelle représentation condensée des motifs flous basée sur la cloture de la correspondance de Galois. Dans la deuxième section, nous présentons une nouvelle représentation condensée pour les motifs graduels clos. Les différentes contributions ont fait l'objet d'expérimentations afin de pouvoir les valider. Ce chapitre se termine sur une discussion.

4.1 Motifs flous clos

La notion du treillis de Galois [Barbut et Monjardet, 1970], ou treillis de concepts formels [Ganter et Wille, 1999], est à la base d'une famille de méthodes de classification conceptuelle [Wille, 1989]. Cette notion, introduite par Barbut et Monjardet [Barbut et Monjardet, 1970], a été utilisée comme une base pour l'analyse formelle de concepts par Wille [Wille, 1982a]. Les treillis de Galois et l'analyse formelle de concepts (AFC) de relations binaires ont démontré leur utilité dans la résolution de nombreux problèmes. Ainsi, en fouille de données, l'AFC a été employée pour extraire les règles d'association d'une manière efficace [Pasquier, 2000]. En effet, les algorithmes fondés sur l'extraction des itemsets fréquents génèrent un nombre exorbitant de règles rendant leur exploitation une tâche quasi impossible par les experts [BenYahia et Nguifo, 2004]. Cependant, les algorithmes fondés sur l'extraction des itemsets ou motifs clos fréquents ont l'avantage de réduire, de manière substantielle, le nombre de règles découvertes en générant un sous-ensemble réduit de règles (*i.e.*, *représentation condensée* de règles d'association).

Ce constat est valable pour les contextes d'extraction binaires [Gasmi *et al.*, 2005], dans ce chapitre nous allons étudier sa validité pour les contextes d'extraction flous et graduels.

4.1.1 Nouvelle définition de la correspondance de Galois pour les motifs flous

Différentes extensions floues de la correspondance de Galois et du concept formel ont été proposées dans le contexte des sous-ensembles flous [Belohlavèk, 1998, BenYahia et Jaoua, 2001, Jaoua *et al.*, 2000, Pollandt, 1996, Wolff, 1998]. Dans cette section, nous introduisons une nouvelle définition des opérateurs de la correspondance de Galois pour les motifs flous. Cette nouvelle définition va tenir compte des préférences des utilisateurs afin de les impliquer dans le processus d'extraction des règles d'association floues. Ainsi, nous introduisons de nouvelles définitions pour les opérateurs de la correspondance de Galois floue (*i.e.*, $\tilde{f}$ et $\tilde{g}$) et une nouvelle forme du concept formel flou, en considérant que seule la partie *intension* du concept formel flou sera représentée par un sous-ensemble flou. Ainsi, une définition de la contrainte utilisateur sera introduite.

Définition 25 ***Contrainte utilisateur***
Une contrainte utilisateur est un sous-ensemble flou $\tilde{C}$ défini sur un ensemble de référence $\tilde{\mathcal{I}}$ (i.e., ensemble fini d'attributs). L'ensemble flou $\tilde{C}$ est défini par la fonction d'appartenance $\mu_{\tilde{C}}$: $\tilde{\mathcal{I}} \rightarrow$ [0,1] et il est dénoté par :

$$\tilde{C} = \{ i_1^{\mu_{\tilde{C}}(i_1)}, i_2^{\mu_{\tilde{C}}(i_2)}, i_3^{\mu_{\tilde{C}}(i_3)}, \ldots, i_n^{\mu_{\tilde{C}}(i_n)} \}$$

Définition 26 ***Contexte d'extraction formel flou avec contrainte***
Le quadruplet $\mathcal{K}_{\tilde{C}} = (\mathcal{O}, \tilde{\mathcal{I}}, \tilde{\mathcal{R}}, \tilde{\mathcal{C}})$ définit un contexte d'extraction formel flou ***avec contrainte****, tels que $\mathcal{O}$ décrit un ensemble fini d'objets (ou de transactions), $\tilde{\mathcal{I}}$ un ensemble fini flou d'attributs (ou d'items), $\tilde{R}$ une relation binaire floue (i.e., $\tilde{\mathcal{R}} \subseteq \mathcal{O} \times \tilde{\mathcal{I}}$) et $\tilde{\mathcal{C}}$ un ensemble fini flou d'attributs (i.e., la contrainte utilisateur).*
Le couple (o, i^{α}) appartenant à $\tilde{\mathcal{R}}$, signifie que l'attribut (item) i appartenant à $\tilde{\mathcal{I}}$ est vérifié par l'objet (transaction) o appartenant à $\mathcal{O}$ avec un degré α. ***Ce degré α est supérieur ou égal à un seuil minimal donné par $\mu_{\tilde{\mathcal{C}}}(i)$.***

Remarque 1 *Les règles d'association floues extraites doivent respecter la contrainte utilisateur, i.e., que tout item présent dans une règle doit avoir un degré d'appartenance au moins égal au degré indiqué par la contrainte utilisateur.*

Exemple 11 *Un exemple de contexte d'extraction flou avec contrainte $\mathcal{K}_{\tilde{C}} = (\mathcal{O}, \tilde{\mathcal{I}}, \tilde{\mathcal{R}}, \tilde{\mathcal{C}})$ est illustré par la figure 4.1 :*

$\tilde{\mathcal{R}}$	B	C	E	M
t_1	0,5	1,0	0,7	0,5
t_2	0,6	0,7	1,0	0,5
t_3	1,0	0,9	1,0	0,1
t_4	1,0	0,9	0,9	0,1
$\tilde{C}$	0,0	0,2	0,7	0,1

FIGURE 4.1 – Contexte d'extraction flou avec contrainte.

Dans cet exemple et selon la contrainte $\tilde{\mathcal{C}}$, les items "B", "C", "E" et "M" doivent apparaître avec des degrés, respectivement, supérieurs ou égaux à 0,0 ; 0,2 ; 0,7 et 0,1 dans toutes les règles d'association floues extraites. Pour y parvenir, nous devons prendre en compte cette contrainte dès la phase de détermination des générateurs minimaux flous.

Définition 27 ***Opérateurs de la correspondance de Galois floue***
Soit un contexte d'extraction flou avec contrainte $\mathcal{K}_{\tilde{C}} = (\mathcal{O}, \tilde{\mathcal{I}}, \tilde{\mathcal{R}}, \tilde{\mathcal{C}})$. L'application $\tilde{f}_{\tilde{C}}$ définie sur l'ensemble de parties de $\mathcal{O}$ (i.e., P($\mathcal{O}$)) dans l'ensemble de parties de $\tilde{\mathcal{I}}$ (i.e., P($\tilde{\mathcal{I}}$)), associe à un ensemble classique d'objets O, un ensemble flou d'items $\tilde{I}$ pondérés par leurs degrés d'appartenance respectifs, tout en respectant la contrainte $\tilde{C}$.

$$\tilde{f}_{\tilde{C}} : P(\mathcal{O}) \rightarrow P(\tilde{\mathcal{I}})$$
$$\tilde{f}_{\tilde{C}}(O) = \{d^{\alpha} \mid \forall o \in O,\ \alpha = \min\{\mu_{\tilde{\mathcal{R}}}(o, d)\} \wedge \alpha \geq \mu_{\tilde{C}}(d), d \in \tilde{I}\ \}$$

L'application $\tilde{g}_{\tilde{C}}$ définie sur l'ensemble de parties de $\tilde{\mathcal{I}}$ (i.e., P($\tilde{\mathcal{I}}$)) dans l'ensemble de parties de $\mathcal{O}$ (i.e., P($\mathcal{O}$)), associe à tout ensemble flou d'items $\tilde{I}$, un ensemble classique (i.e., non flou) d'objets O en respectant les seuils spécifiés dans l'ensemble flou $\tilde{\mathcal{C}}$ (i.e., la contrainte utilisateur).

$$\tilde{g}_{\tilde{C}} : P(\tilde{\mathcal{I}}) \rightarrow P(\mathcal{O})$$
$$\tilde{g}_{\tilde{C}}(\tilde{I})=\{o \in O | \forall\ d \in \tilde{I},\ \mu_{\tilde{I}}(d) \overset{I_{RG}}{\rightarrow} \{\mu_{\tilde{\mathcal{R}}}(o,d)\} = 1 \wedge \mu_{\tilde{C}}(d) \overset{I_{RG}}{\rightarrow} \{\mu_{\tilde{\mathcal{R}}}(o,d)\} = 1\}$$

Dans la définition ci-dessus, I_{RG} désigne l'implication floue de *Rescher-Gaïnes*, qui est une *R-implication* [Dubois et Prade, 1991]. Ainsi :

$$\tilde{g}_{\tilde{C}}(\tilde{I}) = \{o \in O | \forall d \in \tilde{I},\ \mu_{\tilde{I}}(d) \leq \{\mu_{\tilde{\mathcal{R}}}(o,d)\} \wedge \mu_{\tilde{C}}(d) \leq \{\mu_{\tilde{\mathcal{R}}}(o,d)\}\ \}$$

Remarque 2 *Dans le cas des contextes d'extraction classiques binaires, l'opérateur f de la correspondance de Galois, s'applique sur un ensemble d'objets afin de trouver toutes leurs propriétés (attributs) communes. De même, l'opérateur g s'applique sur un ensemble de propriétés afin de trouver l'ensemble maximal d'objets qui possèdent ces propriétés.*
Dans le cas des données floues, l'opérateur flou $\tilde{f}_{\tilde{C}}$ tel qu'il a été défini ci-dessus, s'applique sur un ensemble classique d'objets alors que l'opérateur flou $\tilde{g}_{\tilde{C}}$, s'applique sur un ensemble flou de propriétés (ou attributs). Ainsi, la fonction $\tilde{f}_{\tilde{C}}$, appliquée à un ensemble classique d'objets O ($\tilde{f}_{\tilde{C}}(O)$), calcule le degré minimal avec lequel une propriété donnée est vérifiée par tous les objets de O, tout en respectant la contrainte $\tilde{C}$. La fonction $\tilde{g}_{\tilde{C}}(\tilde{I})$ retourne tous les objets qui vérifient à la fois les propriétés de $\tilde{I}$ (au moins selon les degrés requis) et la contrainte utilisateur $\tilde{C}$.

Proposition 1 *Les deux fonctions $\tilde{f}_{\tilde{C}}$ et $\tilde{g}_{\tilde{C}}$ telles que définies par la définition 27, constituent des opérateurs de la correspondance de Galois floue. Les opérateurs composites $\tilde{f}_{\tilde{C}} \circ \tilde{g}_{\tilde{C}}$ et $\tilde{g}_{\tilde{C}} \circ \tilde{f}_{\tilde{C}}$ sont des opérateurs de cloture flous définis respectivement sur $P(\mathcal{O})$ et $P(\tilde{\mathcal{I}})$. En effet, étant donné les deux opérateurs de la correspondance de Galois floue ($\tilde{f}_{\tilde{C}}$ et $\tilde{g}_{\tilde{C}}$), tels que définis ci-dessus, les propriétés suivantes sont vérifiées pour tout $\tilde{I}$, $\tilde{I}_i$, $\tilde{I}_j \in \tilde{\mathcal{I}}$:*

($\boldsymbol{A}1$)	$O_i \subseteq O_j \Rightarrow \tilde{f}_{\tilde{C}}(O_i) \supseteq \tilde{f}_{\tilde{C}}\ (O_j)$	($\boldsymbol{B}1$)	$\tilde{I}_i \subseteq \tilde{I}_j \Rightarrow \tilde{g}_{\tilde{C}}(\tilde{I}_i) \supseteq \tilde{g}_{\tilde{C}}\ (\tilde{I}_j)$
($\boldsymbol{A}2$)	$O \subseteq \tilde{g}_{\tilde{C}} \circ \tilde{f}_{\tilde{C}}(O)$	($\boldsymbol{B}2$)	$\tilde{I} \subseteq \tilde{f}_{\tilde{C}} \circ \tilde{g}_{\tilde{C}}(\tilde{I})$
($\boldsymbol{A}3$)	$\tilde{f}_{\tilde{C}}(O) = \tilde{f}_{\tilde{C}} \circ \tilde{g}_{\tilde{C}} \circ \tilde{f}_{\tilde{C}}(O)$	($\boldsymbol{B}3$)	$\tilde{g}_{\tilde{C}}\ (\tilde{I}) = \tilde{g}_{\tilde{C}} \circ \tilde{f}_{\tilde{C}} \circ \tilde{g}_{\tilde{C}}(\tilde{I})$
($\boldsymbol{A}4$)	$O_i \subseteq O_j \Rightarrow \tilde{g}_{\tilde{C}} \circ \tilde{f}_{\tilde{C}}(O_i) \subseteq \tilde{g}_{\tilde{C}} \circ \tilde{f}_{\tilde{C}}(O_j)$	($\boldsymbol{B}4$)	$\tilde{I}_i \subseteq \tilde{I}_j \Rightarrow \tilde{f}_{\tilde{C}} \circ \tilde{g}_{\tilde{C}}(\tilde{I}_i) \subseteq \tilde{f}_{\tilde{C}} \circ \tilde{g}_{\tilde{C}}(\tilde{I}_j)$
($\boldsymbol{A}5$)	$\tilde{g}_{\tilde{C}} \circ \tilde{f}_{\tilde{C}}(\tilde{g}_{\tilde{C}} \circ \tilde{f}_{\tilde{C}}(O)) = \tilde{g}_{\tilde{C}} \circ \tilde{f}_{\tilde{C}}(O)$	($\boldsymbol{B}5$)	$\tilde{f}_{\tilde{C}} \circ \tilde{g}_{\tilde{C}}(\tilde{g}_{\tilde{C}} \circ\ \tilde{f}_{\tilde{C}}(\tilde{I})) = \tilde{f}_{\tilde{C}} \circ \tilde{g}_{\tilde{C}}(\tilde{I})$

Preuve 1 *D'une manière duale, ces propriétés sont valides pour les opérateurs de cloture* $\tilde{f}_{\tilde{C}} \circ \tilde{g}_{\tilde{C}}$ *et* $\tilde{g}_{\tilde{C}} \circ \tilde{f}_{\tilde{C}}$*. Dans ce qui suit, nous démontrons la validité de ces propriétés :*

(*A*1) *Nous avons :*

$\tilde{f}_{\tilde{C}}(O_i) = \{d^{\alpha_i} \mid \forall o, o \in O, \alpha_i = \min\{\mu_{\tilde{\mathcal{R}}}(o,d)\} \wedge \alpha_i \geq \mu_{\tilde{C}}(d), d \in \tilde{I}\}$ *et*

$\tilde{f}_{\tilde{C}}(O_j) = \{d^{\alpha_j} \mid \forall o, o \in O, \alpha_j = \min\{\mu_{\tilde{\mathcal{R}}}(o,d)\} \wedge \alpha_j \geq \mu_{\tilde{C}}(d), d \in \tilde{I}\}$

Si $O_i \subseteq O_j$ *alors* $\alpha_i \geq \alpha_j$*. D'où,* $\tilde{f}_{\tilde{C}}(O_i) \supseteq \tilde{f}_{\tilde{C}}(O_j)$*.*

(*B*1) *Nous avons :*

$\tilde{g}_{\tilde{C}}(\tilde{I}_i) = \{o_i \in O \mid \forall d \in \tilde{I}_i,\ \mu_{\tilde{I}_i}(d) \leq \mu_{\tilde{\mathcal{R}}}(o_i,d) \wedge \mu_{\tilde{C}}(d) \leq \mu_{\tilde{\mathcal{R}}}(o_i,d)\}$ *et*

$\tilde{g}_{\tilde{C}}(\tilde{I}_j) = \{o_j \in O \mid \forall d \in \tilde{I}_j,\ \mu_{\tilde{I}_j}(d) \leq \mu_{\tilde{\mathcal{R}}}(o_j,d) \wedge \mu_{\tilde{C}}(d) \leq \mu_{\tilde{\mathcal{R}}}(o_j,d)\}$

Si $\tilde{I}_i \subseteq \tilde{I}_j \Rightarrow \mu_{\tilde{I}_i}(d) \leq \mu_{\tilde{I}_j}(d)$ *[1]*

Si $o_j \in \tilde{g}_{\tilde{C}}(\tilde{I}_j) \Rightarrow \begin{cases} \mu_{\tilde{I}_j}(d) \leq \mu_{\tilde{\mathcal{R}}}(o_j,d) \\ \\ \mu_{\tilde{C}}(d) \leq \mu_{\tilde{\mathcal{R}}}(o_j,d) \end{cases}$ *[2]*

D'après [1] et [2], nous avons $\mu_{\tilde{\mathcal{R}}}(o_j,\ d) \geq \mu_{\tilde{I}_j}(d) \geq \mu_{\tilde{I}_i}(d)$ *et l'objet* o_j *satisfait la contrainte posée par* $\tilde{C}$*, d'où* $o_j \in \tilde{g}_{\tilde{C}}(\tilde{I}_i)$*. Par conséquent,* $\tilde{g}_{\tilde{C}}(\tilde{I}_i) \supseteq \tilde{g}_{\tilde{C}}(\tilde{I}_j)$*.*

(*A*2) *Nous avons :*

$\tilde{f}_{\tilde{C}}(O) = \{d^{\alpha} \mid \forall o, o \in O, \alpha = \min\{\mu_{\tilde{\mathcal{R}}}(o,d)\} \wedge \alpha \geq \mu_{\tilde{C}}(d), d \in \tilde{I}\}$,

$\tilde{g}_{\tilde{C}} \circ \tilde{f}_{\tilde{C}}\ (O) = \{o \in O \mid \forall d \in \tilde{f}_{\tilde{C}}(O),\ \{\mu_{\tilde{\mathcal{R}}}(o,d)\} \geq \min\{\mu_{\tilde{\mathcal{R}}}(o,d)\} \wedge \mu_{\tilde{C}}(d) \leq \{\mu_{\tilde{\mathcal{R}}}(o,d)\}\}$.

Il est évident que, si $o \in O \Rightarrow o \in \tilde{g}_{\tilde{C}} \circ \tilde{f}_{\tilde{C}}(O) \Rightarrow O \subseteq \tilde{g}_{\tilde{C}} \circ \tilde{f}_{\tilde{C}}(O)$*.*

(*B*2) *Soit* $d \in \tilde{I}$ *avec* $\mu_{\tilde{I}}(d)$*, nous avons :*

$\tilde{g}_{\tilde{C}}\ (\tilde{I}) = \{o \in O \mid \forall d \in \tilde{I},\ \mu_{\tilde{I}}(d) \leq \{\mu_{\tilde{\mathcal{R}}}(o,d)\} \wedge \mu_{\tilde{C}}(d) \leq \{\mu_{\tilde{\mathcal{R}}}(o,d)\}\}$ *(3)*

$\tilde{f}_{\tilde{C}} \circ \tilde{g}_{\tilde{C}}\ (\tilde{I}) = \{d^{\alpha} \mid \forall o, o \in \tilde{g}_{\tilde{C}}(\tilde{I}), \alpha = \min\{\mu_{\tilde{\mathcal{R}}}(o,d)\} \wedge \alpha \geq \mu_{\tilde{C}}(d), d \in \tilde{I}\}$ *(4)*

D'après [3] et [4], $\alpha \geq \mu_{\tilde{I}}(d) \Rightarrow \tilde{I} \subseteq \tilde{f}_{\tilde{C}} \circ \tilde{g}_{\tilde{C}}(\tilde{I})$*.*

(*A*3) *Soit* $\tilde{I} = \tilde{f}_{\tilde{C}}(O)$*, d'après (**B**2), nous avons :*

$\tilde{f}_{\tilde{C}}(O) \subseteq \tilde{f}_{\tilde{C}} \circ \tilde{g}_{\tilde{C}} \circ \tilde{f}_{\tilde{C}}(O)$ ***[1]***

et d'après $O \subseteq \tilde{g}_{\tilde{C}} \circ \tilde{f}_{\tilde{C}}(O)$ *(**A**1), nous avons :*

$\tilde{f}_{\tilde{C}}(O) \supseteq \tilde{f}_{\tilde{C}} \circ \tilde{g}_{\tilde{C}} \circ \tilde{f}_{\tilde{C}}(O)$ ***[2]***

*D'après **[1]** et **[2]*** $\Rightarrow \tilde{f}_{\tilde{C}}(O) = \tilde{f}_{\tilde{C}} \circ \tilde{g}_{\tilde{C}} \circ \tilde{f}_{\tilde{C}}(O)$*.*

(*B*3) *Soit* $O = \tilde{g}_{\tilde{C}}(\tilde{I})$*, d'après (**A**2), nous avons :*

$\tilde{g}_{\tilde{C}}(\tilde{I}) \subseteq \tilde{g}_{\tilde{C}} \circ \tilde{f}_{\tilde{C}} \circ \tilde{g}_{\tilde{C}}(\tilde{I})$ ***[1]***

Nous avons : $\tilde{I} \subseteq \tilde{f}_{\tilde{C}} \circ \tilde{g}_{\tilde{C}}(\tilde{I})$*, d'après (**B**1), nous avons :*

$\tilde{g}_{\tilde{C}}(\tilde{I}) \supseteq \tilde{g}_{\tilde{C}} \circ \tilde{f} g_{\tilde{C}} \circ \tilde{g}_{\tilde{C}}(\tilde{I})$ ***[2]***

*D'après **[1]** et **[2]*** $\Rightarrow \tilde{g}_{\tilde{C}}(\tilde{I}) = \tilde{g}_{\tilde{C}} \circ \tilde{f}_{\tilde{C}} \circ \tilde{g}_{\tilde{C}}(\tilde{I})$*.*

(A4) *Nous avons d'après (**A**1) :*
Si $O_i \subseteq O_j$ *alors* $\tilde{f}_{\tilde{C}}(O_i) \supseteq \tilde{f}_{\tilde{C}}(O_j)$. *Par conséquent,* $\tilde{g}_{\tilde{C}} \circ \tilde{f}_{\tilde{C}}(O_i) \subseteq \tilde{g}_{\tilde{C}} \circ \tilde{f}_{\tilde{C}}\ (O_j)$.

(B4) *Nous avons d'après (**B**1) :*
Si $\tilde{I}_i \subseteq \tilde{I}_j$ *alors* $\tilde{g}_{\tilde{C}}(\tilde{I}_i) \supseteq \tilde{g}_{\tilde{C}}(\tilde{I}_j)$. *Par conséquent,* $\tilde{f}_{\tilde{C}} \circ \tilde{g}_{\tilde{C}}(\tilde{I}_i) \subseteq \tilde{f}_{\tilde{C}} \circ \tilde{g}_{\tilde{C}}\ (\tilde{I}_j)$.

(A5) *Nous avons d'après (**A**3) :*
$\tilde{f}_{\tilde{C}} \circ \tilde{g}_{\tilde{C}}(\tilde{f}_{\tilde{C}}(O)) = \tilde{f}_{\tilde{C}}(O) \Rightarrow \tilde{g}_{\tilde{C}} \circ \tilde{f}_{\tilde{C}}(\tilde{g}_{\tilde{C}} \circ \tilde{f}_{\tilde{C}}(O)) = \tilde{g}_{\tilde{C}} \circ \tilde{f}_{\tilde{C}}(O)$.

(B5) *Nous avons d'après (**B**3) :*
$\tilde{g}_{\tilde{C}} \circ \tilde{f}_{\tilde{C}}(\tilde{g}_{\tilde{C}}(\tilde{I})) = \tilde{g}_{\tilde{C}}(\tilde{I})$
$\Rightarrow \tilde{g}_{\tilde{C}}(\tilde{f}_{\tilde{C}} \circ \tilde{g}_{\tilde{C}}(\tilde{I})) = \tilde{g}_{\tilde{C}}(\tilde{I})$
$\Rightarrow \tilde{f}_{\tilde{C}} \circ \tilde{g}_{\tilde{C}}(\tilde{f}_{\tilde{C}} \circ \tilde{g}_{\tilde{C}}(\tilde{I})) = \tilde{f}_{\tilde{C}} \circ \tilde{g}_{\tilde{C}}(\tilde{I})$.

Avant d'entamer la présentation de notre algorithme (FuzzyClos) permettant d'extraire l'ensemble des motifs flous clos et leurs générateurs minimaux flous associés, nous allons définir quelques notions de base qui vont être utilisées par la suite :

Définition 28 ***Concept formel flou***
Un concept formel flou est le couple $(O, \tilde{I})$, *où* O *et* $\tilde{I}$ *sont reliés avec l'opérateur de la correspondance de Galois floue* $\tilde{f}_{\tilde{C}} \circ \tilde{g}_{\tilde{C}}$ *(i.e.,* $\tilde{f}_{\tilde{C}}(O) = \tilde{I}$ *et* $\tilde{g}_{\tilde{C}}(\tilde{I}) = O$*). O et $\tilde{I}$ sont appelés respectivement* ***extension*** *(ou domaine) et* ***intention*** *(ou co-domaine) du concept formel flou.*

Exemple 12 *Considérons le contexte d'extraction flou avec contrainte illustré par la figure 4.1. Le sous-ensemble O défini sur l'ensemble $\mathcal{O}$ et le sous-ensemble flou $\tilde{I}$ défini sur l'ensemble flou $\tilde{\mathcal{I}}$, tels que* $O = \{t_1, t_2\}$ *et* $\tilde{I} = \{B^{0,5}, C^{0,7}, E^{1}, M^{0,5}\}$ *constituent deux ensembles clos. Par conséquent, le couple $(O, \tilde{I})$ est un* ***concept formel flou****.*

Définition 29 ***Motif flou clos***
Soit un contexte d'extraction flou avec contrainte $\mathcal{K}_{\tilde{C}} = (\mathcal{O}, \tilde{\mathcal{I}}, \tilde{\mathcal{R}}, \tilde{\mathcal{C}})$. *Un sous-ensemble flou $\tilde{I}$ de $\tilde{\mathcal{I}}$ est un motif flou clos si et seulement s'il est égal à sa cloture, i.e.,*
$\tilde{f}_{\tilde{C}} \circ \tilde{g}_{\tilde{C}}(\tilde{I}) = \tilde{I}$.

Proposition 2 *L'ensemble de concepts formels flous, dérivés à partir d'un contexte d'extraction flou avec contrainte* $\mathcal{K}_{\tilde{C}} = (\mathcal{O}, \tilde{\mathcal{I}}, \tilde{\mathcal{R}}, \tilde{\mathcal{C}})$, *formant ainsi un treillis de concepts formels flous, est muni d'une relation d'ordre ($\preceq$), tels que* $\forall$ $(O_1, \tilde{I}_1)$ *et* $(O_2, \tilde{I}_2)$ *deux concepts formels flous nous avons :*

$$(O_1, \tilde{I}_1) \preceq (O_2, \tilde{I}_2) \Leftrightarrow O_1 \subseteq O_2 \text{ et } \tilde{I}_2 \subseteq \tilde{I}_1$$

Preuve 2 *Toute relation d'ordre doit être reflexive, anti-symétrique et transitive.*
L'opérateur de cloture $\tilde{f}_{\tilde{C}} \circ \tilde{g}_{\tilde{C}}$ *défini sur l'ensemble de parties de* $\tilde{\mathcal{I}}$ *satisfait ces trois propriétés :*

- ***Réflexivité*** *: soit $\tilde{I}_1$ un ensemble flou d'attributs défini sur $\tilde{\mathcal{I}}$. Par définition, nous avons $\tilde{I}_1 \subseteq \tilde{I}_1$. D'après la propriété (**B4**) de la Proposition 1, nous avons :*
  $\tilde{f}_{\tilde{C}} \circ \tilde{g}_{\tilde{C}}(\tilde{I}_1) \subseteq \tilde{f}_{\tilde{C}} \circ \tilde{g}_{\tilde{C}}(\tilde{I}_1)$. Ceci prouve la propriété de la reflexivité.
- ***Anti-symétrie*** *: soient $\tilde{I}_1$, $\tilde{I}_2 \in \tilde{\mathcal{I}}$ tels que $\tilde{I}_1 \subseteq \tilde{I}_2$ et $\tilde{I}_2 \subseteq \tilde{I}_1$. D'après la propriété (**B4**) de la Proposition 1 , nous avons :*
  *$\tilde{I}_1 \subseteq \tilde{I}_2 \Rightarrow \tilde{f}_{\tilde{C}} \circ \tilde{g}_{\tilde{C}}(\tilde{I}_1) \subseteq \tilde{f}_{\tilde{C}} \circ \tilde{g}_{\tilde{C}}(\tilde{I}_2)$ **(1)***
  *$\tilde{I}_2 \subseteq \tilde{I}_1 \Rightarrow \tilde{f}_{\tilde{C}} \circ \tilde{g}_{\tilde{C}}(\tilde{I}_2) \subseteq \tilde{f}_{\tilde{C}} \circ \tilde{g}_{\tilde{C}}(\tilde{I}_1)$ **(2)***
  $\tilde{I}_1 \subseteq \tilde{I}_2$ et $\tilde{I}_2 \subseteq \tilde{I}_1 \Rightarrow \tilde{I}_1 = \tilde{I}_2$
  *D'après (**1**) et (**2**) : $\tilde{f}_{\tilde{C}} \circ \tilde{g}_{\tilde{C}}(\tilde{I}_1) = \tilde{f}_{\tilde{C}} \circ \tilde{g}_{\tilde{C}}(\tilde{I}_2)$.*
- ***Transitivité*** *: soient $\tilde{I}_1$, $\tilde{I}_2$, $\tilde{I}_3 \in \tilde{\mathcal{I}}$ tels que $\tilde{I}_1 \subseteq \tilde{I}_2$ et $\tilde{I}_2 \subseteq \tilde{I}_3$.*
  $\tilde{I}_1 \subseteq \tilde{I}_2 \Rightarrow \tilde{f}_{\tilde{C}} \circ \tilde{g}_{\tilde{C}}(\tilde{I}_1) \subseteq \tilde{f}_{\tilde{C}} \circ \tilde{g}_{\tilde{C}}(\tilde{I}_2)$
  $\tilde{I}_2 \subseteq \tilde{I}_3 \Rightarrow \tilde{f}_{\tilde{C}} \circ \tilde{g}_{\tilde{C}}(\tilde{I}_2) \subseteq \tilde{f}_{\tilde{C}} \circ \tilde{g}_{\tilde{C}}(\tilde{I}_3)$
  $\tilde{I}_1 \subseteq \tilde{I}_2$ et $\tilde{I}_2 \subseteq \tilde{I}_3$ donc $\tilde{I}_1 \subseteq \tilde{I}_3$ et par conséquent $\tilde{f}_{\tilde{C}} \circ \tilde{g}_{\tilde{C}}(\tilde{I}_1) \subseteq \tilde{f}_{\tilde{C}} \circ \tilde{g}_{\tilde{C}}(\tilde{I}_3)$.

Remarque 3 *Soit $\mathcal{FC}_{\mathcal{K}}$ l'ensemble de concepts formels flous, dérivés à partir d'un contexte d'extraction formel flou avec contrainte, muni de la relation d'ordre $\preceq$. Le couple $\mathcal{L}_{\mathcal{FC}_{\mathcal{K}}} = (\mathcal{FC}_{\mathcal{K}}, \preceq)$ est un treillis complet, appelé treillis de concepts formels flous.*

Définition 30 ***Générateur minimal flou***
Un motif $\tilde{c} \in \tilde{\mathcal{I}}$ est dit générateur minimal flou d'un motif flou clos $\tilde{I}$, si et seulement si $\tilde{f}_{\tilde{C}} \circ \tilde{g}_{\tilde{C}}(\tilde{c}) = \tilde{I}$ et $\nexists\ \tilde{c}_1 \subset \tilde{c}$ tel que $\tilde{f}_{\tilde{C}} \circ \tilde{g}_{\tilde{C}}(\tilde{c}_1) = \tilde{I}$, sachant que si $\tilde{c}_1 \subseteq \tilde{c} \Rightarrow \forall\ x \in U,\ \mu_{\tilde{c}_1}(x) \leq \mu_{\tilde{c}}(x)$. L'ensemble $\mathcal{GMF}_{\tilde{I}}$ des générateurs minimaux flous d'un motif flou clos $\tilde{I}$ est défini comme suit :

$$\mathcal{GMF}_{\tilde{I}} = \{\tilde{c} \subseteq \tilde{\mathcal{I}} |\ \tilde{f}_{\tilde{C}} \circ \tilde{g}_{\tilde{C}}(\tilde{c}) = \tilde{I} \wedge \nexists\ \tilde{c}_1 \subset \tilde{c} \text{ tel que } \tilde{f}_{\tilde{C}} \circ \tilde{g}_{\tilde{C}}(\tilde{c}_1) = \tilde{I}\ \}.$$

Exemple 13 *Soit le contexte d'extraction flou avec contrainte $\mathcal{K}_{\tilde{C}}$ illustré par la figure 4.1 (page 63). L'motif flou $B^{0,5}\ M^{0,5}$ n'est pas un générateur minimal du motif flou clos $"B^{0,5}\ C^{0,7}\ E^{1}\ M^{0,5}"$, puisqu'il existe un autre motif flou $M^{0,5} \subset B^{0,5}\ M^{0,5}$ tel que $\tilde{f}_{\tilde{C}} \circ \tilde{g}_{\tilde{C}}(B^{0,5}) = "B^{0,5}\ C^{0,7}\ E^{1}\ M^{0,5}"$. Ainsi, $M^{0,5}$ est un générateur minimal flou du motif flou clos $"B^{0,5}\ C^{0,7}\ E^{1}\ M^{0,5}"$.*

Définition 31 ***Iceberg du Treillis de Galois flou***
Soit $\mathcal{MFC}$ l'ensemble des motifs flous clos fréquents extraits d'un contexte d'extraction formel flou avec contrainte $\mathcal{K}_{\tilde{C}}$ muni de la relation d'ordre $\preceq$. Le couple $\mathcal{L}_{\mathcal{MFC}} = (\mathcal{MFC}, \preceq)$, est un sup-demi-treillis appelé Iceberg du treillis de Galois flou.

Exemple 14 *Soit le contexte d'extraction flou $\mathcal{K}_{\tilde{C}}$ donné par la figure 4.1, pour un minSup = 0.25, l'ensemble des générateurs minimaux flous fréquents extraits à partir de ce contexte ainsi*

que leurs domaines et fermetures respectifs sont donnés dans le tableau 4.2. La figure 4.3 illustre l'Iceberg du treillis de Galois flou correspondant.

$\overset{\mathcal{GMFF}}{\Longrightarrow}$

Gén.min (Intension)	Domaine (Extension)	Fermeture	Support
B^1	{t$_3$ t$_4$}	B ^{1} C ^{0,9} E ^{0,9} M ^{0,1}	0.50
$B^{0,6}$	{t$_2$ t$_3$ t$_4$}	B ^{0,6} C ^{0,7} E ^{0,9} M ^{0,1}	0.75
$B^{0,5}$	{t$_1$ t$_2$ t$_3$ t$_4$}	B ^{0,5} C ^{0,7} E ^{0,7} M ^{0,1}	1.00
C^1	{t$_1$}	B ^{0,5} C ^{1} E ^{0,7} M ^{0,5}	0.25
$C^{0,9}$	{t$_1$ t$_3$ t$_4$}	B ^{0,5} C ^{0,9} E ^{0,7} M ^{0,1}	0.75
$C^{0,7}$	{t$_1$ t$_2$ t$_3$ t$_4$}	B ^{0,5} C ^{0,7} E ^{0,7} M ^{0,1}	1.00
E^1	{t$_2$ t$_3$}	B ^{0,6} C ^{0,7} E ^{1,0} M ^{0,1}	0.75
$E^{0,9}$	{t$_2$ t$_3$ t$_4$}	B ^{0,6} C ^{0,7} E ^{0,9} M ^{0,1}	0.75
$E^{0,7}$	{t$_1$ t$_2$ t$_3$ t$_4$}	B ^{0,5} C ^{0,7} E ^{0,7} M ^{0,1}	1.00
$M^{0,5}$	{t$_1$ t$_2$}	B ^{0,5} C ^{0,7} E ^{0,7} M ^{0,5}	0.50
$M^{0,1}$	{t$_1$ t$_2$ t$_3$ t$_4$}	B ^{0,5} C ^{0,7} E ^{0,7} M ^{0,1}	1.00
$B^{0,6}$ $C^{0,9}$	{t$_3$ t$_4$}	B ^{1} C ^{0,9} E ^{0,9} M ^{0,1}	0.50
B^1 E^1	{t$_3$}	B ^{1} C ^{0,9} E ^{1} M ^{0,1}	0.25
$B^{0,6}$ $M^{0,5}$	{t$_2$}	B ^{0,6} C ^{0,7} E ^{1} M ^{0,5}	0.25
$C^{0,9}$ E^1	{t$_3$}	B ^{1} C ^{0,9} E ^{1} M ^{0,1}	0.25
$C^{0,9}$ $E^{0,9}$	{t$_3$ t$_4$}	B ^{1} C ^{0,9} E ^{0,9} M ^{0,1}	0.50
$C^{0,9}$ $M^{0,5}$	{t$_1$}	B ^{0,5} C ^{1} E ^{0,7} M ^{0,5}	0.25
$E^{0,9}$ $M^{0,5}$	{t$_2$}	B ^{0,6} C ^{0,7} E ^{1} M ^{0,5}	0.25

FIGURE 4.2 – Liste des générateurs minimaux flous ainsi que les motifs flous clos associés extraits à partir du contexte flou $\mathcal{K}_{\tilde{C}}$ illustré par la figure 4.1 pour *minSup*= 0.25.

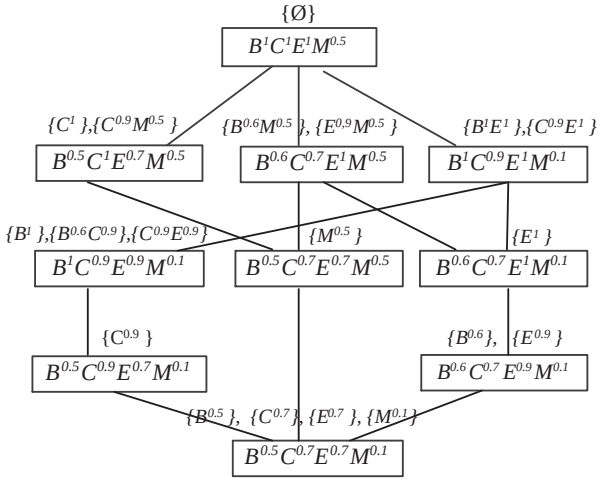

FIGURE 4.3 – Iceberg du treillis de Galois associé au contexte d'extraction flou avec contrainte $\mathcal{K}_{\tilde{C}}$ pour un *minSup* = 0.25.

4.1.2 Algorithme d'extraction des motifs flous clos

Dans cette partie, nous allons présenter un algorithme d'extraction des motifs flous clos, nommé FuzzyClos (Génération des motifs flous clos), à partir d'un contexte d'extraction flou avec contrainte. L'algorithme FuzzyClos est fondé sur la découverte incrémentale des motifs flous clos.
Cet algorithme prend en entrée un contexte d'extraction flou avec contrainte $\mathcal{K}_{\tilde{C}}$ (représenté en mémoire centrale par une structure de données compacte) et un seuil minimal de support *minSup*. Il donne en sortie les motifs flous clos ainsi que leurs générateurs minimaux flous associés.

L'algorithme FuzzyClos, que nous proposons, est un algorithme itératif adoptant la stratégie "*Tester-et-générer*" pour l'exploration de l'espace de recherche (*i.e.*, parcours par niveau). À chaque niveau k, un ensemble de générateurs minimaux flous candidats de taille k est généré à partir de ceux retenus lors de l'itération (k-1) (*i.e.*, générateurs minimaux flous de taille (k-1)) ainsi que leurs domaines respectifs (*i.e.*, application de l'opérateur $\tilde{g}_{\tilde{C}}$). L'ensemble de ces générateurs minimaux flous sera élagué par des heuristiques basées sur des propriétés structurelles des générateurs minimaux flous. Les domaines de ces générateurs seront utilisés afin de construire l'ensemble des motifs flous clos.

L'algorithme FuzzyClos opère en deux étapes :

1. Détermination des générateurs minimaux flous fréquents,
2. Génération des motifs flous clos.

Le pseudo-code et les notations utilisées par l'algorithme FuzzyClos sont, respectivement, présentés dans l'algorithme 1 et le tableau 4.1.

Détermination des générateurs minimaux flous

L'algorithme FuzzyClos commence par déterminer l'ensemble des générateurs minimaux flous candidats, noté par $\mathcal{GMFC}_k$, ainsi que leurs domaines respectifs en adoptant la stratégie *"Tester-et-générer"*. Il effectue un parcours nivelé de l'espace de recherche commençant par déterminer l'ensemble des générateurs minimaux flous candidats de taille 1, *i.e.*, $\mathcal{GMFC}_1$ (*ligne 2*). Cet ensemble sera élagué par rapport à la valeur du support minimale *minSup* et par rapport à la contrainte utilisateur $\tilde{\mathcal{C}}$.

$\mathcal{K}_{\tilde{C}}$	: Contexte d'extraction formel flou avec contrainte $\tilde{\mathcal{C}}$.
$\mathcal{GMFC}_k$	: Ensemble des k-générateurs minimmaux flous candidats.
$\mathcal{GMFF}_k$	: Ensemble des k-générateurs minimaux flous fréquents (*i.e.*, dont le support est supérieur ou égal à *minSup* et satisfaisant la contrainte $\tilde{\mathcal{C}}$).
$\mathcal{GMFF}$	: Ensemble des générateurs minimaux flous fréquents regroupés selon leurs domaines.
$\mathcal{MFC}$	: Ensemble de tous les motifs flous clos.
minSup	: Le support minimal.

TABLE 4.1 – Notations utilisées par l'algorithme FUZZYCLOS

Algorithme : FUZZYCLOS

Données :

– Le contexte d'extraction flou avec contrainte $\mathcal{K}_{\tilde{C}} = (\mathcal{O}, \tilde{\mathcal{I}}, \tilde{\mathcal{R}}, \tilde{\mathcal{C}})$

– *minSup*

Résultat : $\mathcal{MFC}$: L'ensemble des motifs flous clos.

début

- $\mathcal{GMFC}_1 = \{\text{1-motifs flous}\}$;
- CALCUL-DOMAINE($\mathcal{GMFC}_1$) ;
- $\mathcal{GMFF}_0 = \{\emptyset\}$
- $\mathcal{GMFF} = \{\emptyset\}$
- **pour chaque** *($\tilde{g} \in \mathcal{GMFC}_1$)* **faire**
  - **si** *($|\ \tilde{g}.dom| \geq minSup$)* ***et*** *($\tilde{g}.dom \subseteq \tilde{\mathcal{C}}.dom$)* **alors**
    - $\mathcal{GMFF}_1 = \mathcal{GMFF}_1 \cup \tilde{g}$;
- **pour** *(k =1 ; $\mathcal{GMFF}_k \neq \emptyset$; k++)* **faire**
  - $\mathcal{GMFF} = \mathcal{GMFF} \cup \mathcal{GMFF}_k$;
  - $\mathcal{GMFF}_{(k+1)}$ = GEN-NEXT($\mathcal{GMFF}_k$) ;
- $\mathcal{MFC}$=GEN-CLOSURE($\mathcal{GMFF}$) ;
- retourner $\mathcal{MFC}$;

fin

Algorithme 1 : Algorithme d'extraction des motifs flous clos FUZZYCLOS

En effet, l'ensemble $\mathcal{GMFC}_1$ des 1-générateurs minimaux flous candidats est initialisé par l'ensemble des 1-motifs flous du contexte d'extraction flou avec contrainte. Ensuite, la fonction CALCUL-DOMAINE (*ligne 3*) va déterminer les domaines respectifs des éléments de cet ensemble. Cette fonction consiste à appliquer l'opérateur $\tilde{g}_{\tilde{C}}$ pour chaque 1-générateur minimal flou candidat. La cardinalité du domaine de chaque 1-générateur sera comparée à *minSup*. Si elle est supérieure ou égale à *minSup*, alors le générateur est potentiellement fréquent. Un autre test consiste à vérifier si le générateur flou satisfait la contrainte utilisateur $\tilde{\mathcal{C}}$ (*ligne 8*) est effectué. Une fois l'ensemble des 1-générateurs flous fréquents ($\mathcal{GMFF}_1$) est déterminé, il est inséré dans l'ensemble complet des générateurs minimaux flous $\mathcal{GMFF}$ (*ligne 11*) selon leurs domaines, *i.e.*, les générateurs ayant le même domaine, seront regroupés dans un même sous-ensemble.

Ensuite, la procédure GEN-NEXT est appelée pour déterminer l'ensemble des k-générateurs minimaux flous fréquents. Cette procédure prend en entrée l'ensemble des générateurs minimaux flous de taille k et retourne l'ensemble des générateurs minimaux flous fréquents de taille $(k+1)$. La première phase de l'algorithme FUZZYCLOS s'achève lorsqu'il n'y a plus de générateurs minimaux flous candidats à générer. Le pseudo-code de la procédure GEN-NEXT est donné par l'algorithme 2.
Cette procédure comporte trois phases :

1. La première, consiste à appliquer une phase combinatoire qui fait la jointure de deux générateurs minimaux flous fréquents de même taille k, ayant les mêmes premiers $(k-1)$-items flous, produisant ainsi, un nouveau générateur minimal flou potentiellement fréquent de taille $(k+1)$.

2. Lors de la deuxième étape, la fonction CALCUL-DOMAINE est invoquée pour calculer le domaine de chaque générateur minimal flou candidat appartenant à $\mathcal{GMFC}_{(k+1)}$. Cette fonction calcule le domaine du générateur minimal flou résultant de la jointure de deux générateurs minimaux flous fréquents sans avoir à accéder au contexte d'extraction. Ceci est réalisé en se basant sur la proposition suivante :

Proposition 3 $\tilde{g}_{\tilde{C}}(\tilde{I}_1 \cup \tilde{I}_2) = \tilde{g}_{\tilde{C}}(\tilde{I}_1) \cap \tilde{g}_{\tilde{C}}(\tilde{I}_2)$, *pour tout* $\tilde{I}_1$, $\tilde{I}_2 \in \tilde{I}$.

Preuve 3 *Soit* $\tilde{I}_3 = \tilde{I}_1 \cup \tilde{I}_2$, *nous avons donc :*

$$\begin{aligned}
\tilde{g}_{\tilde{C}}(\tilde{I}_3) &= \{g | \forall d \in \tilde{I}_3, \Longrightarrow \mu_{\tilde{\mathcal{R}}}(g,d) \geq \mu_{\tilde{I}_3}(d) \wedge \mu_{\tilde{C}}(d) \leq \mu_{\tilde{\mathcal{R}}}(g,d)\} \\
&= \{g | \forall d \in \tilde{I}_3, \Longrightarrow \mu_{\tilde{\mathcal{R}}}(g,d) \geq \mu_{\tilde{I}_1 \cup \tilde{I}_2}(d) \wedge \mu_{\tilde{C}}(d) \leq \mu_{\tilde{\mathcal{R}}}(g,d)\} \\
&= \{g | \forall d \in \tilde{I}_3, \Longrightarrow \mu_{\tilde{\mathcal{R}}}(g,d) \geq max(\mu_{\tilde{I}_1}(d), \mu_{\tilde{I}_2}(d)) \wedge \mu_{\tilde{C}}(d) \leq \mu_{\tilde{\mathcal{R}}}(g,d)\} \\
&= \{g | \forall d \in \tilde{I}_3, \Longrightarrow \mu_{\tilde{\mathcal{R}}}(g,d) \geq \mu_{\tilde{I}_1}(d) \text{ et } \mu_{\tilde{\mathcal{R}}}(g,d) \geq \mu_{\tilde{I}_2}(d)) \wedge \mu_{\tilde{C}}(d) \leq \mu_{\tilde{\mathcal{R}}}(g,d)\}
\end{aligned}$$

Algorithme : Gen-Next
Données : $\mathcal{GMFF}_k$
Résultat : $\mathcal{GMFF}_{(k+1)}$
début
  /* Phase de jointure*/
  Insert into $\mathcal{GMFC}_{(k+1)}$
  Select $\tilde{g_1}.\text{item}_1^{\alpha_1}$,$\tilde{g_1}.\text{item}_2^{\alpha_2}$,..., $\tilde{g_1}.\text{item}_i^{\alpha_i}$,$\tilde{g_2}.\text{item}_1^{\alpha_1}$,$\tilde{g_2}.\text{item}_2^{\alpha_2}$,..., $\tilde{g_2}.\text{item}_i^{\alpha_i}$
  From $\mathcal{GMFF}_k$ $\tilde{g_1}$, $\mathcal{GMFF}_k$ $\tilde{g_2}$
  Where $\tilde{g_1} \neq \tilde{g_2}$,
    $\tilde{g_1}.\text{item}_1^{\alpha_1} = \tilde{g_2}.\text{item}_1^{\alpha_1}$,
    $\tilde{g_1}.\text{item}_2^{\alpha_2} = \tilde{g_2}.\text{item}_2^{\alpha_2}$,
      ...
    $\tilde{g_1}.\text{item}_{i-1}^{\alpha_{i-1}} = \tilde{g_2}.\text{item}_{i-1}^{\alpha_{i-1}}$,
    $\tilde{g_1}.\text{item}_i^{\alpha_i} \neq \tilde{g_2}.\text{item}_i^{\alpha_i}$

  /* Phase de vérification de l'idéal d'ordre*/
  pour chaque $(\tilde{g} \in \mathcal{GMFC}_{(k+1)})$ **faire**
    pour chaque $(\tilde{g_1}$ ***tel que*** $|\tilde{g_1}| = k$ ***et*** $\tilde{g_1} \subset \tilde{g})$ **faire**
      si $\tilde{g_1} \notin \mathcal{GMFF}_k$ **alors**
        $\mathcal{GMFC}_{(k+1)} = \mathcal{GMFC}_{(k+1)}$ - $\tilde{g_1}$
        arrêt ;

  /* Phase de calcul du domaine et élagage*/
  pour chaque $\tilde{g} \in \mathcal{GMFC}_{(k+1)}$ **faire**
    Calcul-Domaine($\tilde{g}$) ;
    si $(|\tilde{g}.dom| \geq minSup)$ ***et*** $(\nexists\ \tilde{g'} \in \mathcal{GMFF}_k$ *tel que* $(\tilde{g'} \subset \tilde{g})$ ***et*** $(\tilde{g}.dom = \tilde{g'}.dom))$ **alors**
      $\mathcal{GMFF}_{(k+1)} = \mathcal{GMFF}_{(k+1)} \cup \tilde{g}$;
  retourner $\mathcal{GMFF}_{(k+1)}$
fin

Algorithme 2 : Pseudo-code de la procédure Gen-Next

$= \{g|\forall d \in \tilde{I}_3, \Longrightarrow \mu_{\tilde{\mathcal{R}}}(g,d) \geq \mu_{\tilde{I}_1}(d) \wedge \mu_{\tilde{C}}(d) \leq \mu_{\tilde{\mathcal{R}}}(g,d)\}$ *et* $\{g|\forall d \in \tilde{I}_3, \Longrightarrow \mu_{\tilde{\mathcal{R}}}(g,d) \geq \mu_{\tilde{I}_2}(d) \wedge \mu_{\tilde{C}}(d) \leq \mu_{\tilde{\mathcal{R}}}(g,d)\}$

$= \tilde{g}_{\tilde{C}}(\tilde{I}_1) \cap \tilde{g}_{\tilde{C}}(\tilde{I}_2)$

Remarque 4 *Nous remarquons que dans la procédure* Gen-Next*, nous n'avons pas vérifié si les générateurs minimaux flous extraits vérifient la contrainte $\tilde{\mathcal{C}}$. Ceci est dû au fait que ce test a été préalablement effectué lors de la génération des 1-générateurs minimaux flous. Ainsi, tous les générateurs déterminés par* Gen-Next *vérifient certainement la contrainte, puisque leurs domaines ne sont que le résultat d'une intersection de domaines de deux motifs flous vérifiant la contrainte.*

3. Après la phase de jointure, nous passons à la phase de la vérification de l'idéal d'ordre (i.e., tous les sous-ensembles de taille k d'un $(k+1)$-générateur minimal flou fréquent doivent être fréquents) de chaque $\tilde{g} \in \mathcal{GMFC}_{(k+1)}$. Ensuite, l'ensemble des générateurs minimaux flous candidats par rapport à la valeur de *minSup* et par rapport à la définition d'un générateur minimal flou, *i.e.*, pour tout k-générateur minimal candidat flou g_k, nous devons vérifier s'il existe un *(k-1)*-générateur minimal flou fréquent g_{k-1}, tel que $g_{k-1} \subset g_k$, ayant le même domaine. Si un tel cas existe, ce k-générateur flou n'est pas un générateur minimal flou et par conséquent il ne sera pas inséré dans l'ensemble $\mathcal{GMFF}_k$.

Détermination des motifs flous clos

La procédure Gen-Closure prend en entrée l'ensemble $\mathcal{GMFF}$ des générateurs minimaux flous fréquents regroupés selon leurs domaines, et retourne l'ensemble de fermetures relatives à chaque sous-ensemble de ces générateurs minimaux flous *i.e.*, il s'agit d'appliquer l'opérateur $\tilde{f}_{\tilde{C}}$ aux domaines des générateurs minimaux flous fréquents.
Il est à noter que le regroupement des générateurs minimaux flous fréquents dans des sous-ensembles selon leurs domaines, évite le calcul redondant de la cloture.

Exemple 15 *Pour illustrer le déroulement de l'algorithme* FuzzyClos*, nous considérons l'exemple du contexte d'extraction flou avec contrainte $\mathcal{K}_{\tilde{C}} = (\mathcal{O}, \tilde{\mathcal{I}}, \tilde{\mathcal{R}}, \tilde{\mathcal{C}})$ illustré dans la figure 4.1 pour un minSup*=0.25*.*

4.1.3 Mise en oeuvre

Dans cette sous-section, nous décrivons les expérimentations menées concernant l'algorithme FuzzyClos. L'algorithme FuzzyClos a été implémenté en C^{++} et a été testé en terme de temps, de mémoire et de nombres de motifs flous clos extraits. Les expérimentations ont été menées sur un Centrino Dual Core CPU, avec une fréquence d'horloge de $1,66GHz$ et

$\overset{\mathcal{GMFF}_1}{\Longrightarrow}$

Gén.min	Domaine	Support
B^1	$\{t_3\ t_4\}$	0.50
$B^{0,6}$	$\{t_2\ t_3\ t_4\}$	0.75
$B^{0,5}$	$\{t_1\ t_2\ t_3\ t_4\}$	1.00
C^1	$\{t_1\}$	0.25
$C^{0,9}$	$\{t_1\ t_3\ t_4\}$	1.00
$C^{0,7}$	$\{t_1\ t_2\ t_3\ t_4\}$	1.00
E^1	$\{t_2\ t_3\}$	0.75
$E^{0,9}$	$\{t_2\ t_3\ t_4\}$	0.75
$E^{0,7}$	$\{t_1\ t_2\ t_3\ t_4\}$	1.00
$M^{0,5}$	$\{t_1\ t_2\}$	0.50
$M^{0,1}$	$\{t_1\ t_2\ t_3\ t_4\}$	1.00

(a)

$\overset{\mathcal{GMFC}_2}{\Longrightarrow}$

Gén.min	Domaine	Support	Gén.min	Domaine	Support
$B^1\ C^{0,9}$	$\{\ t_3\ t_4\}$	0.25	$B^{0,5}\ M^{0,1}$	$\{t_1\ t_2\ t_3\ t_4\}$	1.00
$B^1\ C^{0,7}$	$\{t_3\ t_4\}$	0.25	$C^1\ E^{0,7}$	$\{t_1\}$	0.25
$B^1\ E^1$	$\{t_3\}$	0.25	$C^1\ M^{0,5}$	$\{t_1\}$	0.25
$B^1\ E^{0,9}$	$\{t_3\ t_4\}$	0.50	$C^1\ M^{0,1}$	$\{t_1\}$	0.25
$B^1\ E^{0,7}$	$\{t_3\ t_4\}$	0.50	$C^{0,9}\ E^1$	$\{t_3\}$	0.25
$B^1\ M^{0,1}$	$\{t_3\ t_4\}$	0.50	$C^{0,9}\ E^{0,9}$	$\{\ t_3\ t_4\}$	0.50
$B^{0,6}\ C^{0,9}$	$\{\ t_3\ t_4\}$	0.50	$C^{0,9}\ E^{0,7}$	$\{t_1\ t_3\ t_4\}$	0.75
$B^{0,6}\ C^{0,7}$	$\{t_2\ t_3\ t_4\}$	0.50	$C^{0,9}\ M^{0,5}$	$\{t_1\}$	0.25
$B^{0,6}\ E^1$	$\{t_2\ t_3\}$	0.50	$C^{0,9}\ M^{0,1}$	$\{t_1\ t_3\ t_4\}$	0.75
$B^{0,6}\ E^{0,9}$	$\{t_2\ t_3\ t_4\}$	0.75	$C^{0,7}\ E^1$	$\{t_2\ t_3\}$	0.50
$B^{0,6}\ E^{0,7}$	$\{t_2\ t_3\ t_4\}$	0.75	$C^{0,7}\ E^{0,9}$	$\{t_2\ t_3\ t_4\}$	0.75
$B^{0,6}\ M^{0,5}$	$\{t_2\}$	0.25	$C^{0,7}\ E^{0,7}$	$\{t_1\ t_2\ t_3\ t_4\}$	1.00
$B^{0,6}\ M^{0,1}$	$\{t_2\ t_3\ t_4\}$	0.75	$C^{0,7}\ M^{0,5}$	$\{t_1\ t_2\}$	0.50
$B^{0,5}\ C^1$	$\{t_1\}$	0.25	$C^{0,7}\ M^{0,1}$	$\{t_1\ t_2\ t_3\ t_4\}$	1.00
$B^{0,5}\ C^{0,9}$	$\{t_1\ t_3\ t_4\}$	0.75	$E^1\ M^{0,5}$	$\{t_2\}$	0.25
$B^{0,5}\ C^{0,7}$	$\{t_1\ t_2\ t_3\ t_4\}$	1.00	$E^1\ M^{0,1}$	$\{t_2\ t_3\}$	0.50
$B^{0,5}\ E^1$	$\{t_2\ t_3\}$	0.50	$E^{0,9}\ M^{0,5}$	$\{t_2\}$	0.25
$B^{0,5}\ E^{0,9}$	$\{t_2\ t_3\ t_4\}$	0.75	$E^{0,9}\ M^{0,1}$	$\{t_2\ t_3\ t_4\}$	0.50
$B^{0,5}\ E^{0,7}$	$\{t_1\ t_2\ t_3\ t_4\}$	1.00	$E^{0,7}\ M^{0,5}$	$\{t_1\ t_2\}$	0.50
$B^{0,5}\ M^{0,5}$	$\{t_1\ t_2\}$	0.50	$E^{0,7}\ M^{0,1}$	$\{t_1\ t_2\ t_3\ t_4\}$	1.00

(b)

$\overset{\mathcal{GMFF}_2}{\Longrightarrow}$

Gén.min	Domaine	Support
$B^1\ E^1$	$\{t_3\}$	0.25
$B^{0,6}\ C^{0,9}$	$\{t_3\ t_4\}$	0.50
$B^{0,6}\ M^{0,5}$	$\{t_2\}$	0.25
$C^{0,9}\ E^1$	$\{t_3\}$	0.25
$C^{0,9}\ E^{0,9}$	$\{t_3\ t_4\}$	0.50
$C^{0,9}\ M^{0,5}$	$\{t_1\}$	0.25
$E^{0,9}\ M^{0,5}$	$\{t_2\}$	0.25

(c)

$\overset{\mathcal{GMFC}_3}{\Longrightarrow}$

Gén.min	Domaine	Support
$B^{0,6}\ C^{0,9}\ M^{0,5}$	$\emptyset$	–
$C^{0,9}\ E^{0,9}\ M^{0,5}$	$\emptyset$	–

(d)

FIGURE 4.4 – **(a)** : liste des 1-motifs flous clos fréquents, **(b)** liste des 2-générateurs minimaux flous candidats, **(c)** liste des 2-motifs flous clos fréquents, **(d)** liste des 3-générateurs minimaux flous candidats.

$2Go$ de mémoire vive. Nous avons utilisé deux différentes catégories de jeux de données : une base provenant du domaine de la biologie et des bases benchmark utilisées en fouille de données. Dans ce qui suit, nous donnons une brève description sur nos jeux de données.

Jeux de données

Le premier jeu de données est le SAGE (Serial Analysis of Gene Expression)[8]. Ce sont des données produites à partir des cellules humaines. Ces données ont été peu exploitées, essen-

8. http ://www.ncbi.nlm.nih.gov/SAGE/index.cgi.

TABLE 4.2 – Description des bases de test.

Base	Type	Nbre d'attributs	Nbre de lignes
SAGE	dense	12 000	56
MUSHROOM	dense	119	8 124
CHESS	dense	75	3 196

tiellement à cause des leurs dimensions [Rioult *et al.*, 2003]. La matrice que nous avons utilisée décrit 56 situations (en ligne) et 12000 gènes (en colonne). Toutefois, travailler avec la totalité des gènes semble être une tâche impossible. C'est la raison pour laquelle nous avons choisi de travailler avec quelques extraits de la base choisis aléatoirement.
Notre deuxième jeux de données est un ensemble de bases benchmark. Nous avons choisi de travailler sur les deux bases "Mushroom" et "Chess". Il est notoire que ces deux bases (*i.e.* ,*"Mushroom"* et *"Chess"*) sont des bases denses (*i.e.*, produisant beaucoup de motifs fréquents même pour des valeurs de *minSup* très élevées). Une brève description de ces différentes bases de test est donnée dans la Table 4.2 .

Expérimentations

Nous avons testé notre algorithme en omettant la contrainte utilisateur (*i.e.*, le pire des cas). Dans la Table 4.3, nous reportons les résultats de nos expérimentations en termes de nombre de motifs flous clos et leurs générateurs minimaux ainsi que le temps nécessaire pour leur extraction.

À partir des résultats présentés dans la Table 4.3 et les courbes des Figures 4.4 et 4.5, nous pouvons constater que :

- Même à partir d'un petit extrait de la base SAGE, le nombre de motifs flous clos extrait est grand pouvant atteindre des milliers même pour des valeurs de *minSup* assez élevées.
- Ce résultat confirme que la base SAGE est très dense. Le nombre de générateurs minimaux flous extraits de la base SAGE a une influence directe sur les performances de notre algorithme FUZZYCLOS. En effet, le temps d'extraction sur la base SAGE est beaucoup plus supérieur à celui alloué aux bases CHESS et MUSHROOM.
- Le nombre des motifs flous clos et leurs générateurs minimaux flous augmente avec la diminution de la valeur du *minSup*.
- Le temps d'extraction des générateurs minimaux flous augmente d'une façon exponentielle avec la diminution de *minSup*. Cependant, le temps de génération des motifs flous clos

Base	*minSup*	#Gén. minimaux flous (Étape 1)	#Motifs flous clos (Étape 2)	Temps (Sec)	
				Étape 1	*Étape 2*
SAGE(1500 gènes)	0.80	496 509	247 521	31 474	291
	0.85	56 502	30 539	577	37
	0.90	10 135	5 824	20	1
	0.95	484	231	1	0
CHESS	0.50	21 142	19 764	2 527	126
	0.60	2 630	1 876	321	17
	0.70	406	365	56	3
	0.80	121	48	8	1
MUSHROOM	0.20	28 934	26 815	4681	228
	0.30	3 344	3 098	1085	61
	0.40	836	695	332	11
	0.50	320	253	109	6
	0.60	187	64	36	2

TABLE 4.3 – Variation du nombre de motifs flous clos et leurs générateurs minimaux flous en fonction de la valeur du *minSup*.

reste raisonnable même pour des seuils de *minSup* assez bas, notamment pour les bases CHESS et MUSHROOM.

- Pour les différentes bases de test, le temps d'extraction des générateurs minimaux est beaucoup plus supérieur à celui d'extraction des motifs flous clos.

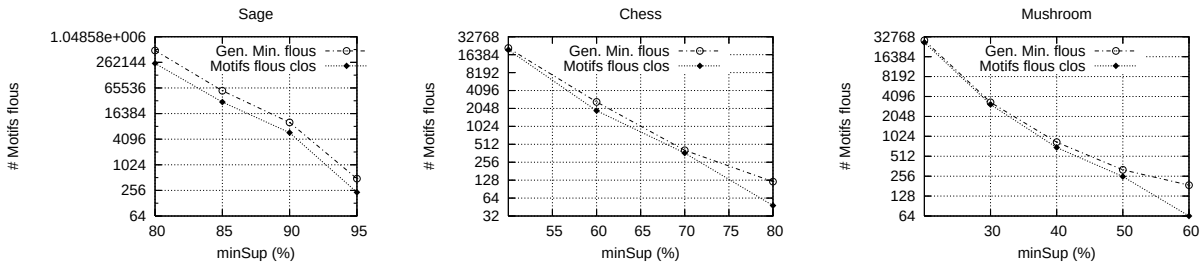

TABLE 4.4 – Variation du nombre de motifs flous clos et leurs générateurs minimaux flous en fonction du *minSup*.

Dans cette section, nous avons introduit la notion de *motif flou clos* et nous avons montré la validité des opérateurs de la correspondance de Galois permettant de l'extraire. Dans la section qui suit, nous introduisons la notion de motif graduel clos. En effet, nous proposons une nouvelle formalisation de la correspondance de Galois pour traiter l'aspect graduel des données.

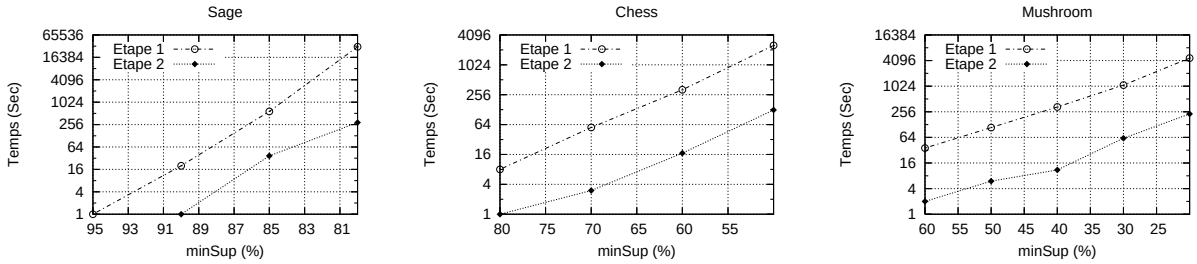

TABLE 4.5 – Évolution du temps d'extraction des motifs flous clos et leurs générateurs minimaux flous en fonction du *minSup*.

4.2 Motifs graduels clos

Dans ce qui suit, nous proposons une représentation condensée des motifs graduels basée sur la notion de la cloture de la correspondance de Galois. Nous introduisons les concepts théoriques associés aux opérateurs de cloture pour les motifs graduels et nous introduisons ainsi la notion de motif graduel clos. Toutefois, nous allons commencer tout d'abord par passer en revue les différentes propositions dédiées à la formalisation et extraction de motifs graduels. Ensuite, nous présentons le cadre théorique pour les motifs graduels clos.

4.2.1 Nouvelles définitions des opérateurs de cloture pour les motifs graduels

Nous formalisons un nouveau système de cloture, à partir des définitions d'un motif graduel de la forme *"Plus/moins* X_1*, ..., plus/moins* X_n*"*, où les X_i sont des items graduels.

Le contexte graduel, contrairement au contexte classique, requiert de considérer tous les objets de la base de données pour traiter un motif. En effet, contrairement aux motifs classiques (e.g. "*beurre,sucre*"), dont on peut dire pour chaque ligne de la base de données s'ils sont présents ou non, un motif graduel (e.g. "*plus il y a de beurre, plus il y a de sucre*") n'est présent dans une base de données qu'au sens de l'ensemble des objets : peut-on ordonner les objets de la base pour qu'ils exhibent une augmentation simultanée sur les valeurs des attributs beurre et sucre ? Cet ordonnancement est défini à partir d'une relation $\preceq$, et nous parlerons alors de séquences d'objets, telles que définie ci-dessous.

Liste d'objets

La recherche des motifs graduels est effectuée à partir d'une base de données avec un ensemble d'objets. Nous définissons alors un ordre sur ces objets en fonction des motifs graduels considérés. Ainsi, nous considèrons un ensemble d'objets (n-uplets) $\mathcal{O}$=$o_1, \ldots, o_n$ d'une base de données où chaque valeur o_i est définie sur un attribut, dont le domaine est muni d'un ordre.

Une séquence d'objets est une liste ordonnée de ces objets notée $\langle o_1, \ldots, o_m \rangle$. Dans ce qui suit, nous proposons de définir un ensemble d'opérations sur ces séquences d'objets.

Définition 32 (Inclusion de séquences) *Soient* $S = \langle o_1, \ldots, o_p \rangle$ *et* $S' = \langle o'_1, \ldots, o'_m \rangle$ *deux séquences d'objets,* S *est* ***incluse*** *dans* S' *(*$S \subseteq S'$*), s'il existe des entiers* $1<i_1<i_2, \ldots, <i_p<m$ *tels que* $o_1 = o'_{i_1}, \ldots, o_p = o'_{i_p}$.

Autrement dit, une séquence d'objets S est incluse dans une autre S' si tous les objets de S apparaissent, *dans l'ordre*, dans S'.

Exemple 6 *Soient* $S_1 = \langle o_1, o_4, o_6 \rangle$, $S_2 = \langle o_2, o_1, o_3, o_4, o_5, o_6 \rangle$, *et* $S_3 = \langle o_2, o_1, o_6, o_3, o_4 \rangle$ *trois séquences d'objets, nous avons* $S_1 \subseteq S_2$ *mais* $S_1 \nsubseteq S_3$.

Définition 33 (Séquence maximale) *Soit* $\mathcal{S}$ *un ensemble de séquences, une séquence d'objets* $S \in \mathcal{S}$ *est dite* ***maximale*** *si* $\nexists\ S' \in \mathcal{S}$, $S' \neq S$ *telle que* $S \subset S'$.

Une séquence est maximale dans un ensemble de séquences si elle est la plus longue (*i.e.*, contient le plus d'objets) dans cet ensemble.

Définition 34 (Intersection de séquences) *L'intersection de deux séquences* S_1 *et* S_2 *est l'ensemble* $\mathcal{S}$ *de sous-séquences maximales telle que chaque séquence de* $\mathcal{S}$ *est une sous-séquence contenue à la fois dans* S_1 *et* S_2, *i.e.,* $S_1 \cap S_2 = \mathcal{S}$, *t.q.,* $\forall\ s_i$ *de* $\mathcal{S}$, $s_i \subseteq S_1$ *et* $s_i \subseteq S_2$ *et* $\nexists\ s'_i \supset s_i$ *tel que* $s'_i \subseteq S_1$ *et* $s'_i \subseteq S_2$.

Exemple 7 *Soient* $S_1 = \{\langle o_1, o_2, o_4, o_7 \rangle\}$ *et* $S_2 = \{\langle o_2, o_5, o_1, o_4, o_6, o_8, o_7 \rangle\}$ *deux séquences. Ainsi, nous avons* $S_1 \cap S_2 = \{\langle o_2, o_4, o_7 \rangle, \langle o_1, o_4, o_7 \rangle\}$.

Définition 35 (Inclusion d'ensembles de séquences) *Soient* $\mathcal{S}$ *et* $\mathcal{S}'$ *deux ensembles de séquences d'objets.* $\mathcal{S}$ *est inclus dans* $\mathcal{S}'$ *(*$\mathcal{S} \preceq \mathcal{S}'$*) si* $\forall\ S \in \mathcal{S}$, $\exists\ S' \in \mathcal{S}'$ *tel que* $S \subseteq S'$.

Exemple 8 *L'ensemble de séquences* $\mathcal{S}_1 = \{\langle o_5, o_6, o_7 \rangle, \langle o_2, o_4, o_7 \rangle\}$ *est inclus dans l'ensemble de séquences* $\mathcal{S}_2 = \{\langle o_5, o_6, o_8, o_7 \rangle, \langle o_1, o_2, 0_4, o_7 \rangle\}$.

À partir de la relation d'inclusion d'ensembles de séquences définie ci-dessus (Définition 35), nous avons la proposition suivante :

Proposition 4 *Soit $\mathcal{S}$ un ensemble de séquences maximales. Ainsi, l'opérateur $\preceq$ définit un ordre partiel sur $\mathcal{P}(\mathcal{S})$.*

Preuve. La relation $\preceq$ définie sur l'ensemble $\mathcal{P}(\mathcal{S})$ est ***réflexive***, ***antisymétrique***, et ***transitive***, *i.e.*, pour chaque S_1, S_2, et S_3 de $\mathcal{P}(\mathcal{S})$, on a :

- $S_1 \preceq S_1$. (**Réflexivité**)
- Soient S_1 et S_2 tels que $\begin{cases} S_1 \preceq S_2 \quad (1) \\ S_2 \preceq S_1 \quad (2) \end{cases}$ Nous avons : $\begin{cases} \forall s_1 \in S_1, \exists s_2 \in S_2 \ tel\ que\ s_1 \subseteq s_2 \quad (1) \\ \forall s_2 \in S_2, \exists s_1 \in S_1 \ tel\ que\ s_2 \subseteq s_1 \quad (2) \end{cases}$

  La clause de maximalité empêchant d'avoir dans le même ensemble deux séquences S_1 et S_2 telles que $S_1 \subset S_2$ ou $S_2 \subset S_1$, la propriété d'**antisymétrie** est donc vérifiée.
- considérons S_1 et S_2 telles que $\begin{cases} S_1 \preceq S_2 \quad (1) \\ S_2 \preceq S_3 \quad (2) \end{cases}$

  Selon *(1)* et *(2)* nous avons $\begin{cases} \forall s_1 \in S_1, \exists s_2 \in S_2 \ tel\ que\ s_1 \subseteq s_2 \quad (3) \\ \forall s_2 \in S_2, \exists s_3 \in S_3 \ tel\ que\ s_2 \subseteq s_3 \quad (4) \end{cases}$

  Grâce à *(3)* et *(4)* nous avons $\forall\ s_1 \in S_1, \exists\ s_3 \in S_3 \ \ tel\ que \ \ s_1 \subseteq s_3$. Nous avons donc $S_1 \preceq S_3$ et la propriété de **transitivité** est donc vérifiée.

■

Opérateurs de cloture graduelle

Dans cette partie, nous introduisons une nouvelle définition de la correspondance de Galois pour les motifs graduels. Ainsi, nous définissons tout d'abord la notion de contexte formel graduel.

Définition 36 (Contexte graduel) *Un contexte formel graduel est défini comme un quadruplet $\mathcal{K}$= ($\mathcal{O}$, $\mathcal{I}$, $\mathcal{Q}$, $\mathcal{R}$) décrivant un ensemble d'objets $\mathcal{O}$, un ensemble fini $\mathcal{I}$ d'attributs (ou*

items), un ensemble fini de valeurs quantitatives $\mathcal{Q}$ et une relation binaire $\mathcal{R}$ (i.e., $\mathcal{R} \subseteq \mathcal{O} \times \mathcal{I}$). Chaque couple $(o, i^q) \in \mathcal{R}$ correspond au fait que la valeur de l'attribut (item) i appartenant à $\mathcal{I}$ pour l'objet O appartenant à $\mathcal{O}$ est q.

Exemple 9 *Par exemple, $\mathcal{K}= (\mathcal{O}, \mathcal{I}, \mathcal{R})$ est un contexte formel graduel associé à la Table 4.6. Nous avons $(o_1, Age^{22}, Salaire^{1200}) \in \mathcal{R}$.*

	Âge	Salaire	Crédit
o_1	22	1200	4
o_2	24	1850	2
o_3	30	2200	3
o_4	28	3400	1

TABLE 4.6 – Contexte formel graduel.

Soit $\mathcal{K}= (\mathcal{O}, \mathcal{I}, \mathcal{Q}, \mathcal{R})$ un contexte formel graduel, nous définissons ci-dessous les deux opérateurs f et g :

$$f : \mathcal{P}(\mathcal{S}) \rightarrow \mathcal{P}(\mathcal{I})$$
$$f(S) = \{i^* | \ \forall\ s \in S, \forall o_l, o_k \in s \text{ t.q. } (o_l, i^{q_1}), (o_k, i^{q_2}) \in \mathcal{R} \text{ et } k < l \text{ nous avons } q_1 * q_2\}$$

La fonction f retourne tous les items graduels, qui respectent toutes les séquences de S.

$$g : \mathcal{P}(\mathcal{I}) \rightarrow \mathcal{P}(\mathcal{S})$$
$$g(I) = \{s \in S | \ s \text{ est maximale dans } S \text{ et } \forall o_l, o_k \in s \text{ t.q. } k < l \text{ et } (o_l, i^{q_1}), (o_k, i^{q_2}) \in \mathcal{R}, \forall i^* \in I \text{ nous avons } q_1 * q_2 \ \}$$

La fonction g retourne l'ensemble des séquences maximales respectant les variations de tous les items de I.

Les deux fonctions g et f sont définies respectivement sur l'ensemble des parties de $\mathcal{I}$ et sur l'ensemble des parties des séquences de $\mathcal{S}$. En effet, étant donné le fait que l'intersection d'un ensemble de séquences d'objets peut résulter en plus d'une séquence, nous considérons l'ensemble des parties de séquences. La fonction f est appliquée sur un ensemble de séquences tandis que g s'applique sur un ensemble d'attributs graduels.

L'ensemble des motifs graduels peut être ordonné par la relation d'inclusion ensembliste classique $\subseteq$ tandis que l'ensemble des séquences est ordonné par la relation $\preceq$.

Exemple 10 *Considérons le contexte illustré par la Table 4.6. Nous avons par exemple* $f(\langle o_1, o_2, o_4\rangle, \langle o_1, o_2, o_3\rangle)$= $\{Age^{\geq}\ Salaire^{\geq}\}$ *et* $g(\{Age^{\geq}\ Credit^{\leq}\})$=$\{\langle o_1, o_2, o_4\rangle, \langle o_1, o_3\rangle\}$.

À partir des définitions et propositions introduites ci-dessus, nous pouvons démontrer que nous construisons un contexte permettant l'utilisation de la correspondance de Galois pour le cas graduel.

Proposition 5 *Pour les ensembles de séquences* S *et* $S' \in \mathcal{S}$*, et les motifs graduels* I *et* I' *les propriétés suivantes sont vérifiées :*

1) $S \preceq S' \Rightarrow f(S') \subseteq f(S)$ **1')** $I \subseteq I' \Rightarrow g(I') \subseteq g(I)$

2) $S \preceq g(f(S))$ **2')** $I \subseteq f(g(I))$

Preuve. Chaque propriété est démontrée ci-dessous.

1) $S \preceq S'$ signifie que $S' = S \cup \{s'_1, \ldots, s'_p\}$. Ainsi chaque item graduel i^* appartenant à $f(S')$ est valide sur $S \cup \{s'_1, \ldots, s'_p\}$ et est donc vérifiée sur S. Ainsi, $f(S)$ inclut tous les items graduels i^* inclus dans $f(S')$, et peut être inclus dans d'autres. Nous avons donc $f(S) \subseteq f(S')$.

2) Nous avons :
- $f(S) = \{i^* | \ \forall\ s \in S, \forall o_l, o_k \in s$ t.q. $(o_l, i^{q_1}), (o_k, i^{q_2}) \in \mathcal{R}$ et $k < l$ on a $q_1 * q_2\}$
- $g(f(S)) = \{s \in S |\ s$ est maximal dans S et $\forall o_l, o_k \in s$ t.q. $k < l$ *and* $(o_l, i^{q_1}), (o_k, i^{q_2}) \in \mathcal{R}, \forall i^* \in f(S)$ nous avons $q_1 * q_2\ \}$

Ainsi, si $s \in S$ alors $s \in g(f(S))$. Nous avons donc : $S \preceq g(f(S))$.

1') Pour chaque $s' \in g(I')$ nous avons $\forall\ i'^* \in I'$, i'^* présente une variation monotone $*$ dans s'. En particulier, pour chaque $i^* \in I'$ puisque $I \subseteq I'$. Ainsi, i^* présente également une variation monotone dans S'. Par conséquent, $s' \in g(I)$ ce qui signifie que $g(I') \preceq g(I)$.

2') Nous avons :
- $g(I) = \{s \in S |\ s$ est maximale dans S et $\forall o_l, o_k \in s$ t.q. $k < l$ *and* $(o_l, i^{q_1}), (o_k, i^{q_2}) \in \mathcal{R}, \forall i^* \in I$ nous avons $q_1 * q_2\ \}$
- $f(g(I)) = \{i^* |\ \forall\ s \in g(I), \forall o_l, o_k \in s$ t.q. $(o_l, i^{q_1}), (o_k, i^{q_2}) \in \mathcal{R}$ et $k < l$ nous avons $q_1 * q_2\}$.

Ainsi, si $i^* \in I$ alors $i^* \in f(g(I))$. Par conséquent, $I \subseteq f(g(I))$.

■

Proposition 6 *Étant donné deux motifs graduels I_1 et $I_2 \in \mathcal{I}$, nous avons $g(I_1 \cup I_2) = g(I_1) \cap g(I_2)$.*

Définition 37 ***Concept graduel formel***
Le couple (S, I), tel que $S \in \mathcal{S}$ et $I \in \mathcal{I}$, est un concept graduel si $f(S) = I$ et $g(I) = S$. O est l'extension et I l'intension du concept graduel.

Définition 38 ***Motif graduel clos***
Considérons le contexte formel $\mathcal{K} = (\mathcal{O}, \mathcal{I}, \mathcal{Q}, \mathcal{R})$, un sous-ensemble graduel $I \subseteq \mathcal{I}$ est un motif graduel clos s'il est égal à sa cloture, i.e., $f \circ g(I) = I$.

Proposition 7 *Les fonctions composées $f \circ g$ et $g \circ f$ forment deux opérateurs de cloture, respectivement définis sur les ensembles de séquences et l'ensemble des motifs graduels.*

Preuve. Considérons les fonctions f et g comme définies précédemment, les propriétés suivantes sont valides :

- **Monotonie** : $I \subseteq I' \Rightarrow f \circ g(I) \subseteq f \circ g(I')$
  Nous avons $I \subseteq I'$, la propriété $1'$) implique que $g(I) \preceq g(I')$, et la propriété 1) permet de déduire que $f \circ g(I) \subseteq f \circ g(I')$.

- **Extensivité** : $I \subseteq f \circ g(I)$
  démontrée par la propriété $2'$).

- **Idempotence** : $f \circ g(I) = f \circ g(f \circ g(I))$
  Grâce à la propriété $2'$) nous avons $f \circ g(I) \subseteq f \circ g(f \circ g(I))$ considérant $S = g(I)$ et grâce à la propriété 2) nous avons $g(I) \preceq g \circ f \circ g(I)$, et la propriété 1) implique que $f \circ g(I) \supseteq f \circ g \circ f \circ g(I)$. Ainsi, nous pouvons conclure que $f \circ g(I) = f \circ g(f \circ g(I))$.

Ces propriétés sont valides de manière duale pour les opérateurs de cloture $f \circ g$ et $g \circ f$. En effet :

– **Monotonie** : $S \preceq S' \Rightarrow g \circ f(S) \preceq g \circ f(S')$
Nous avons $S \preceq S'$, la propriété 1) implique que $f(S') \subseteq f(S)$, et la propriété $1'$) permet de déduire que $g \circ f(S) \preceq g \circ f(S')$.

– **Extensivité** : $S \preceq g \circ f(S)$
Ceci est démontré dans la propriété 2).

– **Idempotence** : $g \circ f(S) = g \circ f(g \circ f(S))$
Selon la propriété 2), nous avons $S \preceq g \circ f(S)$. Grâce à 1), nous avons $f(S) \supseteq f \circ g \circ f(S)$ et grâce à la propriété $1'$), $g \circ f(S) \supseteq g \circ f \circ g \circ f(S)$. En considérant, $I = f(S)$ et grâce à la propriété $2'$) nous avons $f(S) \subseteq f \circ g \circ f(S)$, et la propriété $1'$) permet de déduire que $g \circ f(S) \succeq g \circ f \circ g \circ f(S)$. L'égalité vient du fait que l'opérateur de cloture g, par définition, retourne un ensemble dans lequel aucune séquence n'est une sous-séquence d'une autre.

■

Ces propositions permettent de considérer ce contexte pour la définition et l'extraction de représentations condensées de motifs graduels, comme défini ci-dessous.

Proposition 8 *L'ensemble de concepts graduels formels $\mathcal{GC}_{\mathcal{K}}$ extraits du contexte $\mathcal{K}$ forme un treillis complet $\mathcal{L}_{\mathcal{K}} = (\mathcal{GC}_{\mathcal{K}}, \subseteq)$, que nous nommons treillis de Galois graduel muni d'une relation d'ordre .*

Exemple 11 *Considérons le contexte formel graduel du tableau 4.6. Le treillis des concepts graduels associés est donné par la Figure 4.5.*

Définition 39 ***Générateur graduel minimal*** *Un motif graduel $h \subseteq \mathcal{I}$ est dit générateur graduel minimal d'un motif graduel clos I, si et seulement si $f \circ g(h) = I$ et il n'existe pas $h' \subseteq \mathcal{I}$ tel que $h' \subset h$. L'ensemble $\mathcal{GGM}$ des générateurs graduels minimaux d'un motif graduel clos I est défini comme suit :*

$$\mathcal{GGM} = \{\ h \subseteq \mathcal{I} \mid f \circ g(h) = I \wedge \nexists\ h' \subset h \text{ tel que } f \circ g(h') = I\ \}$$

4.2.2 Algorithme d'extraction de motifs graduels clos

Dans ce chapitre, nous visons à valider l'importance de la réduction du nombre de motifs extraits. En conséquence, nous avons adpoté une approche de type *post-traitement* extrayant les

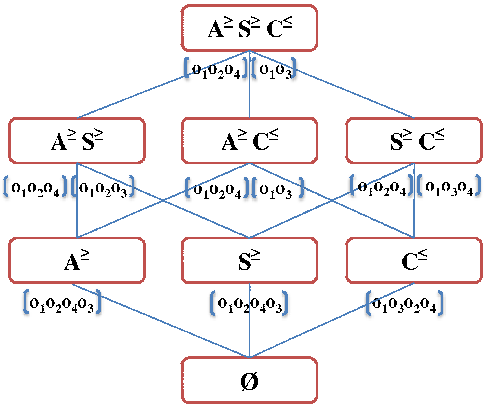

FIGURE 4.5 – Treillis de concepts graduels associés au contexte du tableau 4.6.

$\mathcal{IGF}$	: Ensemble des motifs graduels fréquents .
$\mathcal{GCE}$	: Ensemble de tous les groupes de classes d'equivalence regroupées selon le support.
$\mathcal{IGC}$	: Ensemble des motifs graduels clos.
$\mathcal{GGM}$	: Ensemble des générateurs graduels minimaux d'un motif graduel clos.

TABLE 4.7 – Notations utilisées dans l'algorithme

Données : Ensemble des motifs graduels fréquents ($\mathcal{IGF}$) et leurs supports.
Résultat : Ensembles des motifs graduels clos $\mathcal{IGC}$ et leurs générateurs graduels minimaux respectifs.

```
1  début
2      INSERT INTO GCE SELECT * FROM IGF GROUP BY support ;
3      pour chaque (E ∈ GCE) faire
4          C = MAX(E)
5          tant que (C) ≠ {∅} faire
6              E = E - {C}
7              C.GGM= RECHGEN(C, E )
8              C = MAX(E)
9          IGC = ⋃ {C}
10 fin
```

Algorithme 3 : Algorithme d'extraction des classes d'équivalence.

classes d'équivalence (*i.e.*, l'ensemble des motifs graduels clos et leurs générateurs minimaux) à partir des fréquents trouvés par l'approche de [Di Jorio *et al.*, 2009a].

Pour extraire les différentes classes d'équivalence, nous regroupons les motifs graduels fréquents selon leurs supports. Ainsi, nous construisons des groupes de motifs graduels fréquents selon le support. Chacun de ces groupes peut contenir une ou plusieurs classes d'équivalence. Ceci est expliqué par la double variation $\{\leq, \geq\}$ prise en compte dans notre extraction. En effet, nous pouvons trouver dans un même groupe deux (ou plusieurs) motifs graduels clos *i.e.*, ayant les mêmes items mais de variations $*$ différentes. Le processus d'extraction des classes d'équivalence appartenant à un groupe $\mathcal{E}$ est donné par l'algorithme 3, dans lequel nous procédons comme suit :

1. Déterminer un premier motif graduel maximal $\mathcal{C}$ dans $\mathcal{E}$,
2. Supprimer $\mathcal{C}$ de $\mathcal{E}$,
3. Chercher ses générateurs graduels minimaux en appelant la procédure RechGen. La procédure RechGen retourne tous les générateurs minimaux d'un motif graduel clos. Cette fonction prend en entrée le groupe dans lequel elle va fouiller et le motif graduel clos comme décrit par l'algorithme 4.
4. Itérer jusqu'à extraction de tous les motifs graduels clos et leurs générateurs dans le groupe E.

Ceci est répéter pour chaque groupe E de $\mathcal{GCE}$.

Données : Un motif graduel clos ($\mathcal{C}$) et un groupe de motifs graduels fréquents ($\mathcal{E}$).
Résultat : Ensembles des générateurs graduels minimaux $\mathcal{GGM}$ de $\mathcal{C}$.
1 **début**
2 **pour chaque** *($e \in \mathcal{E}$)* **faire**
3 **si** $e \subset \mathcal{C}$ *et* $\nexists\ e' \subset e$ *tel que* *($e' \in \mathcal{E}$) et ($e' \subset \mathcal{C}$)* **alors**
4 $\mathcal{GGM}$=$\mathcal{GGM} \cup \{e\}$
5 Retourner $\mathcal{GGM}$
6 **fin**

Algorithme 4 : Le pseudo-code de la procédure RechGen.

4.2.3 Mise en oeuvre

Dans ce qui suit, nous rapportons les expérimentations menées pour montrer l'intérêt de notre approche d'extraction des motifs graduels clos. Afin de montrer la concision apportée

par notre méthode, nous comparons le nombre de motifs graduels clos extraits par rapport au nombre de motifs graduels fréquents.

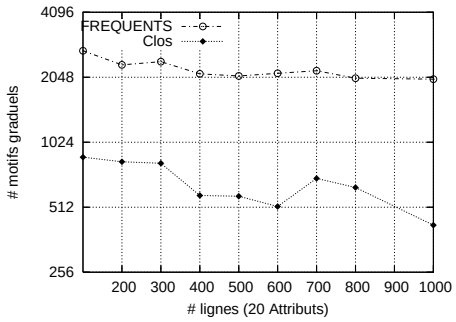

FIGURE 4.6 – Variation du nombre de motifs graduels clos et fréquents en fonction du nombre de lignes.

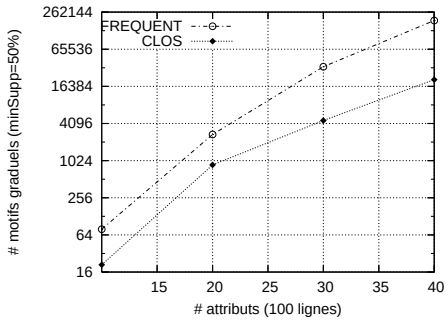

FIGURE 4.7 – Variation du nombre de motifs graduels clos et fréquents en fonction du nombre d'attributs.

Les expérimentations ont été menées sur des jeux de données synthétiques. Les jeux de données synthétiques ont été générés à l'aide d'une version modifiée de IBM Synthetic Data Generation Code for Associations and Sequential Patterns [9]. Notons que ces bases sont très denses et produisent, même pour des valeurs de support minimum élevées, un très grand nombre de motifs fréquents. Dans nos expérimentations, nous nous sommes intéressés à la variation du nombre de motifs graduels clos par rapport au nombre de motifs fréquents en fonction : *(i)* de la valeur de support minimum (*minSup*) ; *(ii)* du nombre de lignes de la base de données et du nombre d'attributs de la base de données, ainsi qu'au temps de calcul.

La Figure 4.6 montre, pour un nombre d'attributs et un *minSup* fixés respectivement à 20 et $0,5$, le nombre de motifs graduels clos et fréquents par rapport au nombre de lignes. Cette courbe montre que le nombre de motifs graduels fréquents et clos varie de manière linéaire avec le nombre de lignes. Notons que l'échelle est logarithmique et que la différence est donc très importante.

Toutefois, ce nombre varie d'une manière exponentielle avec le nombre d'attributs comme le montre la Figure 4.7.

Ces résultats montrent donc l'intérêt de notre approche, avec des différences de nombre de motifs à manipuler et aux experts qui sont considérablement réduits par les opérateurs et algorithmes décrits dans ce chapitre.

Dans cette section, nous visons en effet à valider l'importance de la réduction du nombre de motifs extraits. Comme décrit ci-dessus, l'approche est un *post-traitement* de [Di Jorio *et al.*,

9. www.almaden.ibm.com/software/projects/hdb/resources.shtml

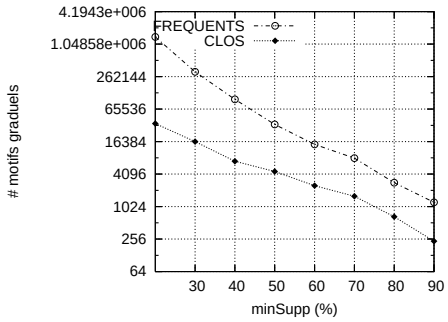

FIGURE 4.8 – Variation du nombre de motifs graduels clos et fréquents en fonction du *minSup*.

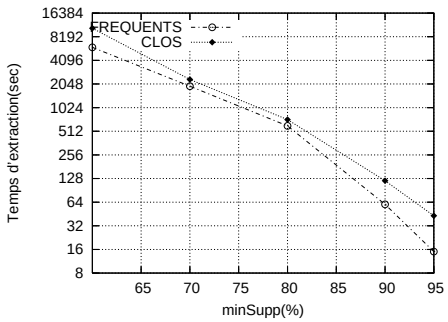

FIGURE 4.9 – Temps de calcul pour les motifs clos et fréquents graduels.

2009a]. Les temps de calcul sont donc un peu plus longs, comme le montre la Figure 4.9. Cependant, une version intégrée de recherche des clos est en cours d'implémentation et permettra d'extraire les clos en des temps comparables aux fréquents.

4.3 Conclusion

Dans ce chapitre nous avons introduit deux nouvelles représentations condensées basées sur la cloture de la correspondance de Galois, une pour les motifs flous clos et une autre pour les motifs graduels clos. Dans le chapitre suivant, nous proposons des approches d'extraction de motifs graduels flous à partir de contextes quantitatifs.

Chapitre 5

Extraction de motifs graduels flous

Sommaire

Dans le chapitre précédent, nous avons introduit deux nouvelles définitions de la correspondance de Galois pour les motifs flous et les motifs graduels. La notion de gradualité a été majoritairement utilisée dans les systèmes de recommandation et de contrôle afin de représenter les corrélations de variations entre le éléments numériques et d'associer des décisions à des situations [Dubois et Prade, 1992, Galichet *et al.*, 2004]. D'une manière générale, un motif graduel flou est de la forme "plus/moins A_1 est F_1, ..., plus/moins A_n est F_n", décrivant la variation d'un attribut A_i associé à une modalité floue F_i elle même décrite par une fonction d'appartenance. Or, il n'existe pas de méthodologie standard pour construire les fonctions d'appartenance, notion de base de la théorie des ensembles flous. Toute la difficulté réside dans le représentation des termes linguistiques par un modèle numériques [Aladenise et Bouchon-Meunier, 1997]. En effet, s'il est facile de comprendre *"très jeune"*, il est difficile de numériser *"très"* ou *"jeune"*.

D'un autre coté, nous nous retrouvons souvent avec des bases de données où la gradualité ne peut plus être extraite à partir de données entre les bornes supérieures et inférieures (Min/Max)

des valeurs numériques de l'attribut, mais plutôt disséminée entre les deux bornes. Ainsi, la règle graduelle "*Plus l'âge d'un patient est proche de 85 ans alors moins son score au test MMS est élevé*" serait incontestablement intéressante pour les chercheurs en psychologie étudiant la maladie d'Alzheimer.

Dans ce qui suit, nous allons tout d'abord introduire la notion de motifs graduels flous et présenter par la suite des méthodes permettant de déterminer automatiquement les modalités flous.

Nous rappelons que notre objectif est d'extraire des motifs graduels à partir de données numériques. Dans certains cas, cette extraction est impossible. En effet, dans certaines bases de données nous ne pouvons pas retrouver de gradualités entre les valeurs des attributs de la base. Par exemple, dans la base numérique illustrée dans le tableau 5 aucun motif graduel pourrait être extrait.

ID	**Age**	**Salaire**	**Nbre de voitures**
1	32	3000	1
2	43	4000	2
3	48	3900	1
4	46	3700	3
5	57	3400	2

TABLE 5.1 – Exemple de base avec attributs numériques.

De même, à partir de données floues, nous ne pouvons pas extraire de gradualités telle que le cas de la base du tableau 5. Cependant, rien n'exclut le fait que d'autres motifs graduels flous pourraient exister dans d'autres modalités floues autres que *"Age jeune" et/ou "Salaire élevé"*. Notre objectif est de localiser les modalités floues susceptibles de contenir certaines gradualités. Pour ce faire, nous avons proposé deux approches différentes. En effet, dans notre première approche, nous proposons de "*fuzzifier*" (*i.e.*, chercher les modalités floues) le contexte d'extraction en se basant sur la médiane des valeurs de chaque attribut de la base. La deuxième approche consiste à chercher des modalités floues suivant la gradualité des attributs en se basant sur les algorithmes génétiques. Ces deux approches sont détaillées dans ce qui suit :

5.1 Proposition d'une approche basée sur la médiane

Dans cette première approche, nous nous intéressons à chercher les modalités floues qui semblent être les plus pertinentes. En effet, au lieu de construire toute la partition d'un attribut numérique, nous nous proposons plutôt de se concentrer sur une modalité floue qui est censée

ID	Age Jeune	Salaire élevé	Nbre de voitures élevé
1	0.8	0.3	0.3
2	0.7	0.8	0.5
3	0.5	0.6	0.8
4	0.6	0.5	0.3
5	0.4	0.4	0.5

TABLE 5.2 – Exemple de base avec attributs flous

recueillir l'identité de l'attribut numérique étant considéré. Pour ce faire, notre idée est de définir les modalités floues, qui décrivent dans quelle mesure la valeur d'un attribut est proche de la médiane de cet attribut.

Par exemple, la base de données illustrée dans le tableau 5, décrit un ensemble de personnes par leurs identificateurs, leurs âges, leurs revenus annuels et le nombre de voitures qu'elles possèdent. À partir de cette base il n'est pas possible de récupérer des motifs graduels. Cependant, si nous considérons les modalités floues "*presque 46*" ans et "*presque* 37,000 euros" comme illustrés par la Figure 5.1, cette base peut être transformée par celle de la Table 5.1. A partir de cette base transformée, le motif graduel flou "*Plus un salarié a presque 46 ans, plus son revenu annuel est presque 37,000 euros*" est vérifié : les lignes de la base peuvent être ordonnées en considérant la liste d'identificateurs $< 1\,;\, 5\,;\, 2\,;\, 3\,;\, 4 >$ et comme il est possible de constater les couples de valeurs des degrés d'appartenance des attributs "*Age presque 46*" et "*Salaire presque 37,000 euros*" peuvent être ordonnées par un ordre de précédence. En effet, si nous considérons qu'un tuple t précède un autre tuple t_0 (dénoté par $t \lhd t_0$), nous pouvons donc ordonner tous les tuples de la base en projetant la base suivant les degrés de "*Age presque 46*" et "*Salaire presque 37,00 euros*" comme suit : $(0,0) \lhd (0,.25) \lhd (.25,.25) \lhd (.5,.5) \lhd (1,1)$.

ID	Age presque 46	Salaire Presque 3700	Nbre de voitures presque 2
1	0.00	0.00	0.50
2	0.25	0.25	1.00
3	0.50	0.50	0.50
4	1.00	1.00	0.50
5	0.00	0.25	1.00

TABLE 5.3 – Base avec attributs flous obtenue à partir de la base de la Table 5

Notre objectif est alors de découvrir automatiquement les modalités floues permettant de découvrir des motifs graduels flous pertinents.

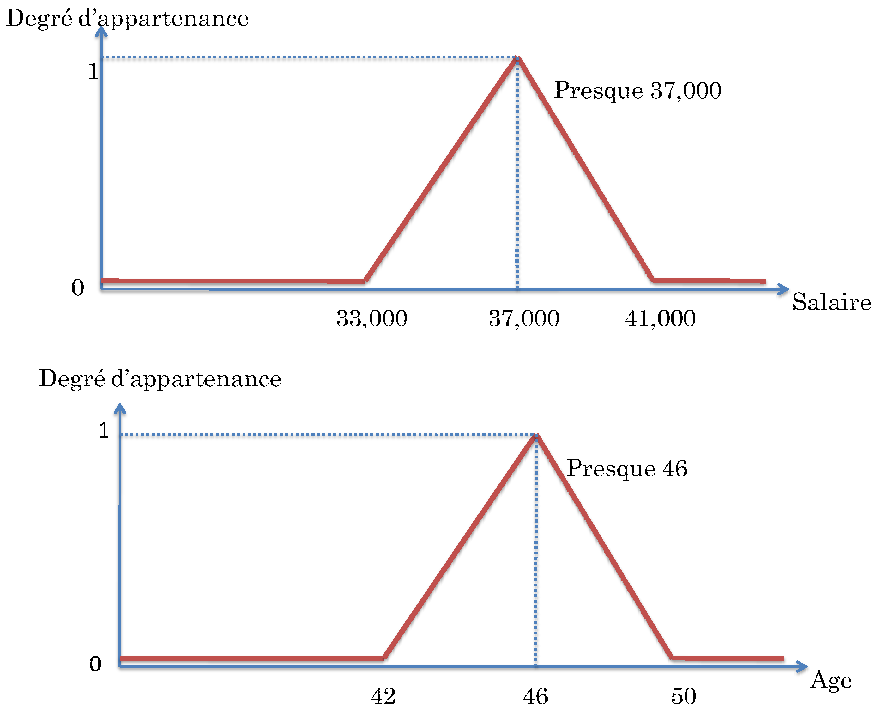

FIGURE 5.1 – Modalités floues autour de la médiane des attributs Age et Salaire

Il est important de noter que ce dans certains domaines, il est difficile de demander à un expert de concevoir une partition floue ou une modalité floue pour chaque attribut numérique de la base (*e.g.*, les bases génomiques contenant les expressions de milliers voir des millions de gènes). Le but est d'obtenir automatiquement la meilleure modalité floue pour chaque attribut à partir de la quelle nous pouvons découvrir des tendances intéressantes qui n'auraient pas été découvertes par les méthodes existante d'extraction de motifs graduels. Ceci peut être justifier soit parce que aucune partition n'a été définie, comme dans les approches classiques [Di Jorio *et al.*, 2008], ou parce que la partition floue étant considérée a été pré-définie et non pas automatiquement obtenue [Berzal *et al.*, 2007].

Comme première approche, nous nous proposons de chercher les tendances graduelles autour de la médiane des valeurs des attributs numériques de la base. Le choix de la médiane est justifié par le fait que cette mesure de position permet de donner la répartition de l'attribut étudié et qui n'est pas affecté par aucune valeur extrême de l'attribut en question. Toutefois, notre approche doit lever les verrous du passage à l'échelle des méthodes traitant des données floues.

Comme les valeurs des attributs de la base sont réparties autour de la médiane, nous considérons la médiane comme le point central de chaque modalité floue pour un attribut donné. Ensuite, les valeurs numériques des attributs sont converties en degrés d'appartenance décrivant la mesure avec laquelle elle sont sont similaires à la valeur de la médiane.

Nous pouvons explorer plusieurs formes des fonctions d'appartenance : fonction triangulaire, trapézoïdale ou gaussienne où le centre est la médiane et le support est tout l'univers qui couvre les valeurs de l'attribut. Nous rappelons que notre objectif est de récupérer les modalités floues les plus intéressantes pour chaque attribut. En effet, il pourrait être le cas que deux attributs décrivent le même type de données (*e.g.*, expression de gène), mais la modalité floue n'est pas la même pour chacun d'eux. Ceci permet de distinguer notre approche de celles classiques et d'être aussi proche que possible du jeu de données.

5.1.1 Algorithme d'extraction de motifs graduels flous basé sur la médiane

Notre approche est appliquée en pré-traitement avant l'application de tout autre algorithme classique de découverte de motifs graduels. Ce pré-traitement a été mis en oeuvre avec le logiciel R [10], qui permet de gérer efficacement de grandes bases de données dans des temps d'exécution raisonnables. Pour extraire les motifs graduels flous pertinents, nous proposons l'algorithme 5 composé de deux sous-algorithmes (*i.e.*, l'algorithme 6 permettant d'obtenir les modalité floues de chaque attribut de la base et l'algorithme 7 se référant à un algorithme classique d'extraction de motifs graduels comme décrit dans [Di Jorio *et al.*, 2009b]).

La fonction $BuildFuzzMod$, telle que décrite dans l'algorithme 6, calcule les paramètres clés de la modalité floue de chaque attribut de la base de données d'origine. Après avoir calculé les paramètres de la modalité, cette fonction retourne le degré d'appartenance de chaque valeur de l'attribut à la modalité correspondante floue qui a été trouvée.

Données : $\mathcal{D}$ contexte d'extraction original, ς minSup.
Résultat : $\mathcal{MGF}$ Ensemble de motifs graduels fréquents, .
début
  $\mathcal{D}' \leftarrow BuildFuzzMod(\mathcal{D})$
  $\mathcal{MGF} \leftarrow$ MineGradualPatterns($\mathcal{D}'$,ς)
  Retourner ($\mathcal{MGF}$)
fin

Algorithme 5 : Extraction de motifs graduels flous.

5.1.2 Mise en oeuvre

Afin d'évaluer notre première approche nous l'avons testée sur plusieurs bases de données. La première base contient des données sur les joueurs de NBA avec 17 atttributs. Une deuxième

10. R est un logiciel libre pour le calcul statistique et graphique et il est disponible à l'adresse http : // www.r-project.org http ://www.r-project.org.

Données : $\mathcal{D}$ contexte d'extraction original.
Résultat : $\mathcal{D}'$ contexte d'extraction transformé.

début
  pour chaque *Attribut* $A \in \mathcal{D}$ **faire**
    TmpMin $\leftarrow$ Min($\mathcal{D}.A$)
    TmpMax $\leftarrow$ Max($\mathcal{D}.A$)
    TmpCenter $\leftarrow$ Médiane($\mathcal{D}.A$)
    /* Construction de la fonction d'appartenance triangulaire $MFunc$ */
    $MFunc \leftarrow$ Triang(TmpMin,TmpCenter,TmpMax)
  pour chaque *Ligne* $l \in \mathcal{D}$ **faire**
    pour chaque *Attribut* $A \in \mathcal{D}$ **faire**
      $\mathcal{D}'[A, l] \leftarrow$ MFunc($\mathcal{D}[A, l]$)
  Retourner ($\mathcal{D}'$)
fin

Algorithme 6 : Pseudo-code de la procedure BuildFuzzMod - Construction des modalités floues.

Données : $\mathcal{D}$ contexte d'extraction original, ς minSup.
Résultat : $\mathcal{MGF}$ ensemble de motifs graduels flous.

début
  $\mathcal{MGF} \leftarrow$ Couple d'attributs avec Support $\geq \varsigma$
  pour chaque *Taille* $k \in [3, ...]$ **faire**
    Construire les motifs candidats de taille k à partir des motifs graduels fréquents de taille $k-1$
    $\mathcal{MGF} \leftarrow \mathcal{MGF} \bigcup$ {Motifs graduels contenant k attributs avec Support $\geq \varsigma$}
  Retourner ($\mathcal{MGF}$)
fin

Algorithme 7 : Pseudo-code de la procédure MineGradualPatterns - Extraction de motifs Graduels.

TABLE 5.4 – Items graduels avant et après le pré-traitement de la base $WINE$

Attribut	Description Classique	Description Floue
Malic Acid	Plus/Moins le Malic Acid	Plus/Moins le Malic Acid est *presque* de 13.05
Ash	Plus/Moins le Ash	Plus/Moins le Ash est *presque* 1.865
Alcalinity of ash	Plus/Moins l'Alcalinity de ash	Plus/Moins l'Alcalinity de ash est *presque* 2.36
Magnesium	Plus/Moins le Magnesium	Plus/Moins le Magnesium est *presque* 19.5
Total phenols	Plus/Moins le Total phenols	Plus/Moins le Total phenols est *presque* 98
Flavanoids	Plus/Moins le Flavanoids	Plus/Moins le Flavanoids est *presque* 2.355
Nonflavanoid phenols	Plus/Moins le Nonflavanoid phenols	Plus/Moins le Nonflavanoid phenols est *presque* 0.34
Proanthocyanins	Plus/Moins the Proanthocyanins	Plus/Moins the Proanthocyanins est *presque* 1.555
Color intensity	Plus/Moins le Color intensity	Plus/Moins le Color intensity est *presque* 4.69
Hue	Plus/Moins le Hue	Plus/Moins le Hue est *presque* 0.965
OD280/OD315 de diluted wines	Plus/Moins le OD280/OD315 of diluted wines	Plus/Moins the OD280/OD315 of diluted wines est *presque* 2.78
Proline	Plus/Moins the Proline	Plus/Moins the Proline est *presque* 673.5

TABLE 5.5 – Nombre de motifs extraits par rapport à la valeur de $minSup$

	wine_crisp	wine_fuzzy	nba_crisp	nba_fuzzy
1	23	38.259	0	275
0.8	149	797.148	0	19.040

décrit le comportement de 500 gènes pour des études biologiques sur le cancer des seins et une troisième, dont nous avons supprimer l'attribut classe, contenant 13 attributs concernant le vin.

Toutes ces trois bases de données ont été prétraitées pour remplacer chaque attribut par une modalité floue décrivant à quelle mesure la valeur initiale est proche de la médiane. Par exemple, la Table 5.4 montre les nouveaux attributs pour la base de données $WINE$.

Dans le tableau 5.5, nous reportons le nombre de motifs graduels pouvant être extraits à partir des trois bases de données que nous avons examinées en considérant l'approche classique [Di Jorio *et al.*, 2009b] et notre approche de fuzzification des données pour deux différentes valeurs de $minSup$. En effet, les motifs graduels classiques sont de la forme "$Plus/moins\ A_1$, ..., $Plus/moins\ A_k$", tandis que les motifs graduels flous sont de la forme "$Plus/moins\ A_1$ *est presque* X, ..., $Plus/moins\ A_k$ *est presque* Y". Afin d'extraire les motifs graduels classiques, nous avons appliqué l'algorithme GRITE proposé par [Di Jorio *et al.*, 2009b]. Nous pouvons constater comme le montre le tableau 5.5, que notre approche permet d'extraire de nouvelles tendances que les méthodes classiques n'ont pas réussi à les extraire.

La table 5.4 présente quelques exemples de motifs extraits par notre méthode. Pour la base de données $WINE$, il n'existe pas une corrélation de variation entre les valeurs des attributs *Total phenols* et *Flavanoids* selon l'approche classique [Di Jorio *et al.*, 2009b]. Cependant, en construisant les modalités floues, nous avons pu découvrir qu'ils sont corrélés lorsque l'on considère *"Plus/moins les Phénols est presque 98 et plus/moins les Flavonoïdes sont presque 2,355"*. En se basant sur cette corrélation, des motifs contenant 9 attributs ont pu être découverts. Pour la base NBA, aucun motif graduel n'est extrait avec l'approche classique pour un $minsup$ égal à 100% alors que 275 motifs graduels flous ont été extraits en considérant les modalités floues.

En effet, les attributs 1 et 12 de la base NBA ont été découverts comme corrélés : *"Plus le Nombre de Parties jouées est presque* 122*, moins le nombre de Buts est presque* 626*"*.
Nous pouvons conclure que notre approche permet de découvrir des connaissances graduelles potentiellement utiles cachées dans de grandes bases de données que l'approche classiques n'a pas réussi à les découvrir.

5.2 Proposition d'une approche basée sur les algorithmes génétiques

Dans la section précédente, nous avons proposé d'extraire des motifs graduels flous à partir de données numériques en construisant les modalités floues autour de la médiane des valeurs des attributs. Ces modalités floues décrivent à quelle mesure la valeur d'un attribut est proche de la médiane de cet attribut.

Les expérimentations ont montré que cette approche a permis d'extraire des connaissances que l'approche classique a échoué de le faire. Cependant, la gradualité pourrait bien être cachée dans une petite partie du domaine des valeurs de l'attribut autre que la partie autour de la médiane. Ainsi, notre deuxième approche consiste à identifier les *meilleures* modalités floues permettant de décrire des motifs graduels flous pertinents. Nous proposons d'utiliser les algorithmes génétiques afin de localiser ces meilleures modalités floues sur tout l'ensemble de données.
Avant d'illustrer notre approche, nous nous proposons de commencer par un petit aperçu sur les algorithmes génétiques.

5.2.1 Généralités sur les algorithmes génétiques

Les algorithmes génétique (AG), développés par John Holland [Holland, 1992], sont des algorithmes d'optimisation stochastique fondés sur les mécanismes de la sélection naturelle et de la génétique. Ils tentent de simuler le processus d'évolution des espèces dans leur milieu naturel tel que énoncé par Darwin. En génétique, un individu est représenté par un code (*i.e.*, le code d'ADN), c'est-à-dire un ensemble de données (appelées *chromosomes*), identifiant complètement l'individu. La reproduction est un mixage aléatoire de chromosomes de deux individus, donnant naissance à des individus enfants ayant une empreinte génétique nouvelle, héritée des parents. La mutation génétique est caractérisée dans le code génétique de l'enfant par l'apparition d'un nouveau chromosome qui n'existait pas chez les parents. Ce phénomène génétique d'apparition de "mutants" est rare mais permet d'expliquer les changements dans la morphologie des espèces.

La disparition de certaines espèces est expliquée par "les lois de survie" selon lesquelles seuls les individus les mieux adaptés survivront et pourront générer une descendance. Les individus peu adaptés auront une tendance à disparaître. C'est une sélection naturelle, qui conduit de génération en génération à une population composée d'individus de plus en plus adaptés à leur environnement. Un AG est une transposition artificielle de ce processus génétique naturel.

En fait, dans un problème d'optimisation, nous disposons d'une population de taille N dans laquelle chaque individu représente une solution potentielle du problème. Ces individus sont convenablement codés et la population va évoluer artificiellement de génération en génération et va être soumise à différents types de mutations et de croisements. Une fonction d'évaluation est nécessaire pour décider de la force de chaque individu de la population à survivre ou non. Un AG est composé des étapes suivantes comme illustré par la Figure 5.2 :

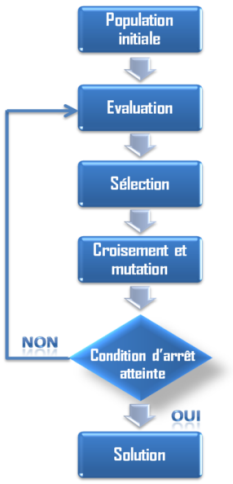

FIGURE 5.2 – Organigramme d'un algorithme génétique.

– **Initialisation** : l'AG démarre avec une population initiale de taille N. Les N individus de la population sont codés selon un principe de codage. Le choix du codage des individus est primordial et conditionne le succès de l'AG. Nous pouvons distinguer deux types de codage : le codage binaire largement utilisé à l'origine et le codage réel désormais employé notamment pour les problèmes à variables continues. La population initiale doit être générée d'une manière non homogène qui servira de base pour les générations futures. Le choix du mécanisme de génération de la population initiale est important pour une convergence plus ou moins rapide vers un optimum global.

– **Evaluation** : chaque individu de la population est évalué selon une fonction d'évaluation

appelée *fitness*. Cette fonction mesure le degré d'adaptation d'un individu à son environnement.

- **Sélection** : il s'agit de choisir dans une population les individus qui survivront à la génération suivante, en fonction d'une valeur d'adaptation (*i.e.*, la valeur de fitness). Il existe plusieurs heuristiques de selection, la méthode la plus connue est la sélection par *roulette biaisée* (roulette wheel selection). Néanmoins, nous pouvons citer d'autres méthodes, telles que la *sélection par tournoi* (tournement selection) ou d'autres méthodes faisant intervenir un *changement d'échelle* (Scalling) et/ou des notions de *voisinage entre chromosomes* (Sharing).

- **Croisement et Mutation** : ces deux opérations sont appliquées afin de garantir la diversification de la population au cours des générations. D'une façon analogue aux chromosomes lors de la reproduction naturelle, l'opérateur de croisement permet de créer de nouveaux individus à partir de deux individus parents en échangeant entre eux des parties de leurs gènes. Ce phénomène nommé *Cross-Over* permet d'explorer l'ensemble des solutions possibles. D'une manière générale, l'opération de croisement est effectuée comme suit :
  La population courante est divisée en deux, un individu est choisi de chaque sous-population. Ces deux individus P_1 et P_2 appelés *parents* participent au croisement avec une probabilité p_c souvent supérieure à $0,5$. Ceci est effectué en choisissant une position de chacun des parents et en échangeant ensuite les deux sous-parties de chacun des deux parents, ce qui génère deux enfants C_1 et C_2. Cette opération est illustrée par la figure 5.3.
  Dans l'exemple de la figure 5.3, le croisement est effectué à une seule position entre P_1 et P_2, il s'agit d'un croisement en un seul point. Il existe bel et bien d'autres types de croisements portant sur plusieurs points (2, 3, etc), il s'agit d'un croisement en multi-points [Man *et al.*, 1996]. La figure 5.4 illustre un exemple de croisement en deux points.
  L'opération de mutation est une modification, qui intervient d'une manière aléatoire sur le génome d'un individu. Elle sert à maintenir une certaine diversité de la population et par conséquent éviter une convergence prématurée de l'algorithme. Elle doit intervenir d'une part sur une partie suffisamment petite de l'individu pour ne pas détruire ses caractéristiques et d'une autre part suffisamment grande pour lui apporter des éléments nouveaux. La figure 5.5 illustre un exemple de mutation.

Un *AG* est itératif, il s'arrête si un nombre d'itérations fixé à l'avance est atteint ou lorsque la population d'individus cesse d'évoluer ou elle n'évolue plus assez rapidement.

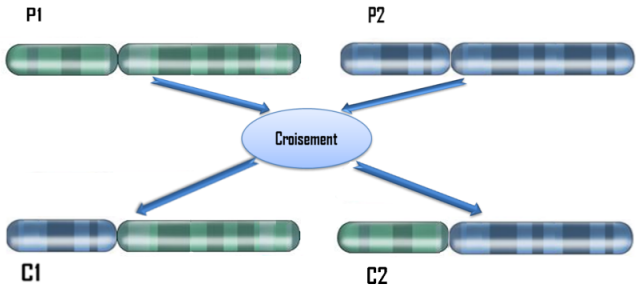

FIGURE 5.3 – Croisement en un point.

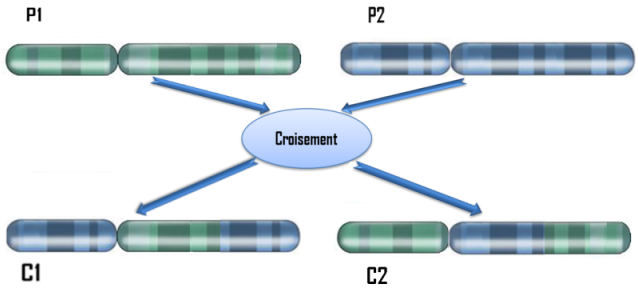

FIGURE 5.4 – Croisement en deux points.

5.2.2 Un algorithme génétique pour les motifs graduels flous

Tout algorithme génétique doit démarrer avec une population initiale contenant des individus décrits par leurs chromosomes. A chaque étape de l'algorithme, des opérations de génétiques (*i.e.*, évaluation, sélection, croisement, mutation) sont appliquées sur les individus et une nouvelle population est alors créée. Dans ce qui suit nous expliquons comment les algorithmes génétiques peuvent aider à extraire des motifs graduels flous pertinents. Nous commençons par introduire ce que pourrait être les individus de la population, et comment ils pourraient être mélangés afin de créer une nouvelle population.

– **Codage des solutions** : notre objectif est d'extraire des motifs graduels flous pertinents à partir des données de la base. Chaque motif graduel flou est définie sur l'ensemble des

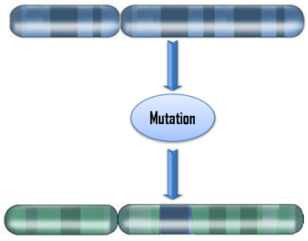

FIGURE 5.5 – Un exemple d'une opération de mutation.

$\emptyset$ ou $(X_1, *_1)$	...	$\emptyset$ ou $(X_i, *_i)$	...	$\emptyset$ ou $(X_m, *_m)$
(a_1, b_1, c_1, d_1)	...	(a_i, b_i, c_i, d_i)	...	(a_m, b_m, c_m, d_m)

TABLE 5.6 – Individu avec m chromosomes - Cas général

attributs X_1, ..., X_i, ..., X_m. Par souci de simplicité, nous nous concentrons sur les fonctions de forme triangulaire, mais notre approche sera la même pour des fonctions d'appartenance triangulaires, gaussiennes ou autres. Chaque fonction d'appartenance est donc décrite par un 3-uplets correspondant aux trois valeurs a_i, b_i, c_i (*i.e.*, a_i et c_i les limites de l'intervalle du support et b_i le support du sous-ensemble flou). La fonction d'appartenance floue n'est pas nécessairement symétrique.

Une solution du problème présente un individu de la population. Un individu est un ensemble de chromosomes représentant m motifs graduels flous, où les ensembles flous correspondants sont définis par les 3 variables des fonctions triangulaires définis sur les m attributs associés à deux variations possibles $*$ correspondant à une augmentation ou une diminution ($* \in \leq, \geq$). Si un attribut ne respecte pas la gradualité du motif selon la fonction d'appartenance qui lui correspond, alors il est remplacé par l'ensemble vide, comme illustré sur les figures 5.6 et 5.7.

– **Initialisation de la population** :
Afin de garantir la diversité de la population initiale, nous allons la générer de sorte à

$\emptyset$	...	$(X_i, \leq)$	...	$(X_m, \geq)$
$(13, 17, 24, 39)$	...	$(25, 26, 43, 58)$	...	$(5, 12, 24, 43)$

TABLE 5.7 – Exemple d'individu

ce qu'elle soit la plus hétérogène possible. Nous proposons de créer le premier individu de la population comme décrit dans [S. Ayouni et Poncelet, 2010]. En effet, chaque chromosome de l'individu représente tout le domaine de l'attribut correspondant. Les autres individus de la population représentent des fonctions d'appartenance floues obtenues par des variations aléatoires latérales des fonctions construites précédemment. Ceci assure assez de diversité dans la population initiale. La taille de la population est l'un des paramètres du système que nous notons σ.

– **Opérations génétiques** :

- **Croisement** :

  A chaque étape de l'algorithme, les individus de la population subissent des opérations génétiques afin de créer de nouveaux (meilleurs) individus. Les chromosomes des individus de la population sont croisés dans le but d'avoir différentes fonctions d'appartenance (*i.e.*, modalités floues) sur tout le domaine de l'attribut. L'opérateur de croisement consiste à prendre deux individus aléatoirement, appelés *parents*, et générer des nouveaux individus appelés *enfants*, dont les paramètres de la fonction de la modalité floue correspond à ceux de l'un des deux parents. Un exemple d'une opération de croisement est illustrée par la figure 5.8.

- **Mutation** :

  L'opérateur de mutation est un mécanisme qui garantit la diversité de la population. La mutation peut agir pour modifier aléatoirement certaines caractéristiques génétiques d'un individus avec un taux de mutation noté MR. Cette modification peut porter soit sur la définition de la fonction d'appartenance, en considérant un changement de sa forme (*i.e.*, par étirement), soit sur le sens de variation du motif graduel, passant d'une augmentation à une diminution ou inversement. Cependant, cette transformation doit être réalisée en considérant certaines contraintes. Par exemple, lorsque nous considérons des fonctions triangulaires, nous devons tenir en compte de la propriété suivante pour chaque attribut $i : A_i \leq B_i \leq c_i$. Comme nous l'avons précédemment indiqué, chaque individu de la population correspond à un motif gra-

Croisement de :

$X_1, \leq$	...	$X_5, \leq$	...	$X_m, \geq$
$(25, 26, 43, 58)$	...	$\emptyset$	...	$(5, 12, 24, 43)$

Avec

$X_1, \geq$	...	$X_5, \geq$	...	$X_m, \geq$
$\emptyset$	...	$(14, 16, 23, 38)$	...	$(38, 43, 72, 87)$

donne

$X_1, \leq$	...	$X_5, \leq$	...	$X_m, \geq$
$(25, 26, 43, 58)$	...	$\emptyset$	...	$(38, 43, 72, 87)$

et

$X_1, \geq$	...	$X_5, \geq$	...	$X_m, \geq$
$\emptyset$	...	$(14, 16, 23, 38)$	...	$(5, 12, 24, 43)$

TABLE 5.8 – L'opération de croisement.

duel flou. A partir d'un individu de la population, il est ainsi possible de récupérer toutes les données qui lui correspond en revenant à la base de données (*i.e.*, le support).

- **Fonction d'évaluation** :

  L'évaluation des individus de la population se base sur la définition d'une fonction (*i.e.*, fitness). Cette fonction d'évaluation est nécessaire à la détermination de la pertinence des solutions potentielles à partir des grandeurs à optimiser. Dans notre algorithme, nous cherchons à extraire des motifs graduels flous pertinents. La pertinence d'un tel motif dépend de son support et de sa longueur (*i.e.*, le nombre d'items graduels flous non vides le formant). Un motif graduel flou est pertinent lorsqu'il a un support et une longueur assez élevé. Ainsi, nous définissons une fonction d'évaluation locale permettant de prendre en compte ces deux paramètres tout en favorisant le paramètre support. Étant donné une population $\mathcal{I} = \{I_1, \ldots, I_\sigma\}$ où chaque individu I_i correspond à un motif graduel flou dont le support est noté par $support(I_i)$ et $Length(I_i)$ sa longueur. Cette fonction est définie comme suit :

$$L\_Fitness(I_i) = -support(I_i) \times log\left(\frac{1}{Length(I_i)}\right)$$

  Cette fonction d'évaluation est utilisée pour sélectionner les meilleurs individus de la population. Ainsi, les individus ayant une valeur de fitness moins élevée que les autres individus ou plus petite qu'une valeur cible seront éliminés.

Nous définissons également une fonction d'évaluation globale, $G\_Fitness(\mathcal{I}$ de toute une population comme étant la moyenne des valeurs des fonctions d'évaluation locales.

$$G\_Fitness(\mathcal{I}) = \left(\frac{1}{\sigma}\right) * \sum_{i=1}^{\sigma} L\_Fitness(I_i)$$

La fonction d'évaluation globale est utilisée comme une condition d'arrêt de l'algorithme. En effet, si la valeur de $G\_Fitness(\mathcal{I})$ stagne après un certain nombre de générations, l'algorithme s'arrête en concluant sur l'ensemble des k-meilleurs motifs .

- **Sélection** : comme nous l'avons déjà mentionné, il existe plusieurs heuristiques de sélection des individus. Cette sélection est généralement relative à la fonction d'évaluation de l'individu. Nous proposons dans notre approche d'utiliser la stratégie élitiste, qui consiste à conserver les meilleurs individus à chaque génération. Cette stratégie de sélection empêche l'individu le plus pertinent de disparaître ou que ses bonnes combinaisons soient affectées par les opérateurs de croisement et de mutation.

Le pseudo-code et les notations utilisés dans notre approche sont respectivement présentés dans le tableau 5.9 et l'algorithme 8.

$\mathcal{D}$	: Base de données initiales.
$\mathcal{MGF}$	: Ensembles de motifs graduels flous.
$\mathcal{MF}$	: Ensemble des fonctions d'appartenance.
$NbrGen$	: Nombre de générations.
CF	: Taux de croisement.
MR	: Taux de mutation.
σ	: Taille de la population.
k	: Nombre des meilleurs motifs graduels flous à extraire :

TABLE 5.9 – Notations utilisées dans l'algorithme génétique

5.2.3 Mise en oeuvre

Afin de valider notre algorithme génétique d'extraction des motifs graduels flous et leurs modalités floues correspondantes nous l'avons testé sur un ensemble de données réelles décrivant des données sur le vin (*i.e.*, la base $Wine$), où l'attribut *Classe* a été supprimé.

Input : $\mathcal{D}$, $minSup$, σ, $NbrGen$, CF, MR, k.

Output : $\mathcal{MGF}$, $\mathcal{MF}$.

begin

Étape1 : Génération aléatoire de la population initiale de taille σ

Étape2 : Evaluation de chaque individu I_i dans la population Pop comme suit :

foreach *($I_i \in Pop$)* **do**

- Transformation de la valeur quantitative de chaque attribut de la base $\mathcal{D}$ en un degré d'appartenance suivant la modalité floue représentée par l'individu
- Calcul du $support(I_i)$ en comptant le nombre de tuples dans $\mathcal{D}$ correspondant à MGF_i représenté par I_i.
- Calcul de la fonction d'évaluation locale de I_i comme suit : $L\_Fitness(I_i) = (-support(I_i) * log\left(\frac{1}{Length(I_i)}\right))$.
- Incrémenter la fonction d'évaluation globale $G\_Fitness$ de la population Pop.

Étape 3 : Création d'une nouvelle population comme suit :

- Application de l'opération de croisement sur les $CF\%$ des individus de la population précédente.
- Application de l'opération de mutation sur les MR individus de la population.
- Insertion des nouveaux individus dans la population.
- Sélection des meilleurs individus de la population selon le mécanisme d'élitisme.

Étape4 : Revenir à **Étape2** pour évaluer la population.

Étape5 : Répéter **Étape3** et **Étape4** jusqu'à satisfaire l'une des conditions suivantes :

- Atteindre $NbrGen$
- Après un certain nombre de stagnations de la valeur de $G\_Fitness$

Étape6 : Insérer l'ensemble des k-meilleurs individus dans $\mathcal{MGF}$ et les modalités floues correspondantes dans $\mathcal{MF}$.

end

Algorithme 8 : Algorithme génétique pour l'extraction de motifs graduels flous.

Suite à une étude empirique, nous avons réussi à fixer les paramètres de notre algorithme génétique à savoir le taux de mutation, le taux de croisement et le taux d'élitisme qui sont respectivement égaux à 0.25, 0.01 et 0.1. La Table 5.10, montre les meilleurs motifs pouvant être extraits à partir de la base $Wine$ pour une population de 100 individus et avec 1000 essais.

Nous avons étudié l'impact de la variation de la taille de la population sur la valeur du

	MGF_1	MGF_2	MGF_3	MGF_4	MGF_5	MGF_6	MGF_7	MGF_8	MGF_9	MGF_{10}
Support	172	171	158	156	146	127	125	122	122	115
Longueur	13	8	13	13	12	13	13	12	12	12

TABLE 5.10 – Les 10-meilleurs motifs graduels flous (MGF) extraits de la base $Wine$.

support du meilleur individus de la population. La Figure 5.6 illustre la courbe de la variation de la taille de la population par rapport au support du meilleur individu de la population.

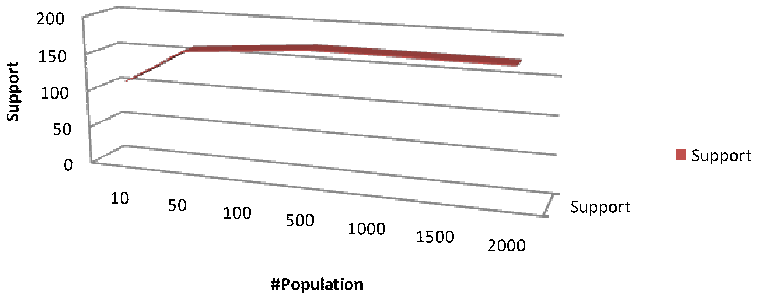

FIGURE 5.6 – Variation du support du meilleur individu de la population par rapport à la variation de la taille de la population.

5.2.4 Discussion

Afin d'extraire les motifs graduels flous à partir d'un contexte quantitatif, il va falloir, tout d'abord, "fuzzifier" ce contexte. Dans ce chapitre, nous avons proposé deux différentes approches d'acquisition des modalités floues des attributs (*i.e.*, fuzzification du contexte). En effet, la première approche est basée sur la médiane des valeurs des attributs alors que la deuxième est basée sur les algorithmes génétiques. Avec l'algorithme génétique, que nous avons proposé, nous avons réussi à extraire à la fois des motifs graduels flous intéressants (*i.e.*, avec des valeurs élevées de support) et des modalités floues pertinentes permettant d'identifier des

co-variations entre les attributs de la base (*i.e.*, gradualité). Dans le chapitre suivant, nous définissons des bases de couverture informatives de toutes les règles graduelles/floues à partir des motifs clos (flous et graduels).

Chapitre 6

Extraction de règles graduelles/floues

Sommaire

6.1 Introduction

Le problème d'extraction de règles d'association consiste à présenter à l'utilisateur un ensemble de règles pouvant l'aider à la prise de décision. Ce problème peut être décomposé en deux sous-problèmes :

- **Extraction des itemsets ou motifs fréquents** : ce sont tous les motifs ayant un support au moins égal à $minSup$.

- Génération de règles d'association valides basées sur l'ensemble de motifs fréquents préalablement extraits : ces règles sont de la forme R : X $\Rightarrow$ Y tel que X $\subset$ Y et confiance(R)$\geq$ *minConf*.

Dans la plupart des contextes réels, le nombre et la taille moyenne des motifs fréquents extraits sont très importants. Par conséquent, le nombre de règles générées est très élevé et peut varier de plusieurs dizaines de milliers à plusieurs millions [Pasquier, 2000]. De plus, il existe souvent des règles de même support et de même confiance, qui s'avèrent redondantes. Ceci remet en cause la pertinence et l'utilité du résultat présenté à l'utilisateur. Ce constat s'aggrave davantage pour les contextes denses ou pour des valeurs de *minSup* basses.

Pour pallier ce problème, une solution consiste à présenter à l'utilisateur un sous-ensemble réduit de règles couvrant toutes les règles valides et véhiculant le maximum de connaissances pertinentes et utiles. Plusieurs approches ont été proposées afin de sélectionner sans perte d'information ce type de règles d'association.
Ces approches reposent sur les travaux issus de la théorie de l'analyse formelle de concepts (*AFC*). Elles proposent d'extraire un sous-ensemble générique, de toutes les règles d'association, appelé *représentation condensée*. Une représentation condensée doit satisfaire certains critères, à savoir [Kryszkiewicz, 2001] :

- Elle doit être accompagnée d'un mécanisme d'inférence (*e.g.*, un système axiomatique), permettant la dérivation de toutes les règles redondantes. Ce mécanisme doit être correct (*i.e.*, le système ne permet de dériver que les règles d'association valides) et complet (*i.e.*, le système permet de retrouver l'ensemble de toutes les règles valides).
- Elle doit assurer la détermination du support et de la confiance de chaque règle dérivée (ou redondante).

Dans ce chapitre, nous proposons de définir une représentation condensée de règles d'association graduelles et floues tout en profitant de ce qui a été réalisé dans le cas des contextes classiques (*i.e*, contextes binaires).
Ainsi, nous proposons, d'abord, de rappeler le formalisme de base des règles d'association graduelles et floues.

6.2 Formalisme de base des règles d'association graduelles/floues

Étant donné un contexte d'extraction graduel ($\mathcal{K} = (\mathcal{O}, \mathcal{I}, \mathcal{Q}, \mathcal{R})$) ou flou ($\mathcal{K}_{\tilde{C}} = (\mathcal{O}, \tilde{\mathcal{I}}, \tilde{\mathcal{R}}, \tilde{\mathcal{C}})$), nous définissons une règle d'association graduelle ou floue $\tilde{R}$ comme étant une relation entre motifs graduels ou flous de la forme :

$$\tilde{R} : \tilde{I}_1 \Rightarrow \tilde{I}_2$$

avec $\tilde{I}_1$, $\tilde{I}_2 \subseteq \tilde{I}$ $(\tilde{I} \subseteq \tilde{\mathcal{I}})$, $\tilde{I}_1 = \{i_1^{\alpha_1}, i_2^{\alpha_2}, ..., i_p^{\alpha_p}\}$ et $\tilde{I}_2 = \{i_q^{\alpha_q}, ..., i_n^{\alpha_n}\}$.

avec $\tilde{I}_1, \tilde{I}_2 \subseteq \tilde{I}(\tilde{I} \subseteq \tilde{\mathcal{I}})$ et

$$\begin{cases} \tilde{I}_1 = \{i_1^{\alpha_1}, i_2^{\alpha_2}, \ldots, i_p^{\alpha_p}\} \text{ et } \tilde{I}_2 = \{i_q^{\alpha_q}, \ldots, i_n^{\alpha_n}\} \text{ si } \tilde{R} \text{ est une règle floue} \\ \tilde{I}_1 = \{i_1^{*_1}, i_2^{*_2}, \ldots, i_p^{*_p}\} \text{ et } \tilde{I}_2 = \{i_q^{*_q}, \ldots, i_n^{*_n}\} \text{ si } \tilde{R} \text{ est une règle graduelle} \end{cases}$$

$\tilde{I}_1$ est la partie *prémisse* de la règle alors que $\tilde{I}_2$ est sa partie *conclusion*. Une règle d'association graduelle ou floue *valide* est celle dont la confiance, résultat du rapport $\frac{support(\tilde{I}_1 \cup \tilde{I}_2)}{support(\tilde{I}_1)}$, est supérieure ou égale à un seuil minimal *minConf* fixé par l'utilisateur.
Nous définissons le *support* et la *confiance* d'une règle d'association graduelle comme suit :

$$Supp(\tilde{R}) = \frac{max(|g(\tilde{I}_1 \cup \tilde{I}_2)|)}{|O|}$$
$$Conf(\tilde{R}) = \frac{max(|g(\tilde{I}_1 \cup \tilde{I}_2)|)}{max(|g(\tilde{I}_1)|)}$$

Nous définissons le *support* et la *confiance* d'une règle d'association floue comme suit :

$$Supp(\tilde{R}) = \frac{|\tilde{g}_{\tilde{C}}(\tilde{I}_1 \cup \tilde{I}_2)|}{|O|}$$
$$Conf(\tilde{R}) = \frac{|\tilde{g}_{\tilde{C}}(\tilde{I}_1 \cup \tilde{I}_2)|}{|\tilde{g}_{\tilde{C}}(\tilde{I}_1)|}$$

Remarque 5 *Dans la règle graduelle ou floue $\tilde{R}$, définie ci-dessus, l'union de la partie prémisse et la partie conclusion forme respectivement un motif graduel clos ou un motif flou clos.*

6.3 Représentations condensées de règles d'association graduelles et floues

L'extraction de règles d'association classiques souffre du problème de génération d'un grand nombre de règles générées. Pour pallier ce problème, plusieurs travaux ont été élaborés afin d'extraire un sous-ensemble générique de toutes les règles (*i.e., représentation condensée*) [Bastide *et al.*, 2000a, Kryszkiewicz, 1998, Kryszkiewicz, 2001, Luong, 2001, Zaki, 2000, Zaki et Hsiao, 2002]. Ces représentations condensées sont particulièrement appropriées pour les contextes denses. Les contextes flous sont des contextes fortement denses. Ainsi, il est incontestablement nécessaire de définir une représentation condensée de toutes les règles d'association floues.

Dans ce qui suit, nous allons adapter les représentations condensées de règles exactes ($\mathcal{GBE}$) et approximatives ($\mathcal{GBA}$) définies par Bastide et *al.* [Bastide *et al.*, 2000a], aux contextes d'extraction graduels/flous. Nous définissons ainsi, une représentation condensée de règles exactes

graduelles /floues ($\mathcal{RCE}$) et une représentation condensée de règles approximatives graduelles/floues ($\mathcal{RCA}$) ainsi que sa couverture transitive ($\mathcal{CTGF}$).

6.3.1 Représentation condensée de règles graduelles/floues exactes ($\mathcal{RCE}$)

Nous introduisons une représentation condensée pour les règles d'association graduelles/floues exactes comme suit :

Définition 40 *Soit $\mathcal{MC}$ l'ensemble des motifs garduels/flous clos fréquents extraits d'un contexte d'extraction graduel $\mathcal{K}$ ou flou $\mathcal{K}_{\tilde{C}}$. Pour chaque motif graduel/flou clos $\tilde{I} \in \mathcal{MC}$, nous désignons par $\mathcal{MG}_{\tilde{I}}$ l'ensemble de ses générateurs minimaux flous. La représentation condensée des règles d'association graduelles/floues exactes $\mathcal{RCE}$ est définie comme suit :*

$$\mathcal{RCE} = \{\ \tilde{R} : \tilde{g} \Rightarrow (\tilde{I} \text{ - } \tilde{g}) \mid \tilde{I} \in \mathcal{MC} \wedge \tilde{g} \in \mathcal{MG}_{\tilde{I}} \wedge \tilde{g} \neq \tilde{I}\ \}$$

Pour toute règle $\tilde{R} : \tilde{g} \Rightarrow (\tilde{I} \text{ - } \tilde{g})$ appartenant à $\mathcal{RCE}$, le motif graduel/flou $\tilde{g}$ est un générateur minimal du motif graduel/flou clos $\tilde{I}$. Ainsi, nous avons $\tilde{f}_{\tilde{C}} \circ \tilde{g}_{\tilde{C}}(\tilde{g}) = \tilde{f}_{\tilde{C}} \circ \tilde{g}_{\tilde{C}}(\tilde{I}) = \tilde{I}$ (*i.e.*, $\tilde{g}$ et $\tilde{I}$ appartiennent à la même classe d'équivalence). Par conséquent, $\tilde{g}$ et $\tilde{I}$ ont la même valeur de support et par suite la confiance de $\tilde{R} = \frac{support(\tilde{I})}{support(\tilde{g})}$ est égale à 1.
La condition $\tilde{g} \neq \tilde{I}$ est nécessaire puisque les règles d'association grauelles/floues générées à partir d'un générateur graduel/flou $\tilde{g}$ d'un motif graduel/flou clos $\tilde{I}$ tel que $\tilde{g} = \tilde{I}$ sont de la forme $\tilde{g} \Rightarrow \emptyset$ et n'appartiennent pas à l'ensemble des règles valides (règle non informative).

Dans ce qui suit, nous proposons un algorithme nommé Gen-GBEF, pour la construction de la représentation condensée de règles d'association graduelles/floues exactes $\mathcal{RCE}$ à partir de l'ensemble des motifs graduels/flous clos et leurs générateurs minimaux graduels/flous associés. Le pseudo-code de l'algorithme Gen-GBEF est donné par l'algorithme 9.

Dans l'algorithme 9 et ce qui quit, l'opérateur h désigne la composition des opérateurs de la correspondance de Galois $f \circ g$ pour le cas graduel et $\tilde{f}_{\tilde{C}} \circ \tilde{g}_{\tilde{C}}$ pour le cas flou.

Données :
L'ensemble des motifs graduels/flous clos $\mathcal{MC}$ et leurs générateurs minimaux flous associés.
Résultat : La base $\mathcal{RCE}$.
début
  $\mathcal{RCE} \leftarrow \emptyset$;
  pour chaque *motif graduel/flou clos* $\tilde{I} \in \mathcal{MC}$ **faire**
    pour chaque *générateur* $\tilde{g} \in \mathcal{F}\tilde{\mathcal{G}}_{\mathcal{I}}$ *tel que* $h(\tilde{g})^{(11)} \neq \tilde{g}$ **faire**
      $\mathcal{RCE} \leftarrow \mathcal{RCE} \cup \tilde{R} : \tilde{g} \Rightarrow (\tilde{I}$ - $\tilde{g})$
  retourner $\mathcal{RCE}$;
fin

Algorithme 9 : Algorithme Gen-GBEF

6.3.2 Représentation condensée de règles graduelles floues approximatives ($\mathcal{RCA}$)

Dans ce qui suit, nous définissons une représentation condensée de règles d'association approximatives graduelles/floues comme suit :

Définition 41 *Soit $\mathcal{MC}$ l'ensemble des motifs graduels/flous clos extraits à partir d'un contexte d'extraction graduel $\mathcal{K}$ ou flou $\mathcal{K}_{\tilde{C}}$ et $\mathcal{M}\tilde{\mathcal{G}}_{\tilde{I}}$ l'ensemble des générateurs minimaux graduels/flous d'un motif graduel/flou clos $\tilde{I}$ appartenant à $\mathcal{MC}$. La base informative des règles d'association graduelles/floues approximatives $\mathcal{RCA}$ est définie comme suit :*

$$\mathcal{RCA} = \{\ \tilde{R} : \tilde{g} \Rightarrow (\tilde{I}_1 \text{ - } \tilde{g}) |\ \tilde{I}, \tilde{I}_1 \in \mathcal{MC},\ \tilde{g} \in \mathcal{M}\tilde{\mathcal{G}}_{\tilde{I}} \wedge \tilde{I} < \tilde{I}_1 \wedge \mathit{Confiance}(\tilde{R}) \geq \mathit{minConf}\ \}$$

Soient $\tilde{g}$ un générateur minimal graduel/flou d'un motif graduel/flou clos $\tilde{I}$ et $F_{\tilde{I}}^{\uparrow}$ l'ensemble de tous les motifs graduels/clos flous couvrant $\tilde{I}$. La règle $\tilde{R} : \tilde{g} \Rightarrow (\tilde{I}_s$ - $\tilde{g})$ appartient à $\mathcal{RCA}$ si et seulement si $\tilde{I}_s \in F_{\tilde{I}}^{\uparrow}$. Puisque $\tilde{g} \subseteq \tilde{I}$ et $\tilde{I} \subset \tilde{I}_s$ ($\forall\ \tilde{I}_s \in F_{\tilde{I}}^{\uparrow}$), alors nous avons $\tilde{g} \subset \tilde{I}_s$. Ainsi, nous pouvons conclure que la *confiance* d'une règle graduelle/floue $\tilde{R}$ est égale à $\frac{support(\tilde{I}_s)}{support(\tilde{g})}$ et le *support* de la règle est égal à $support(\tilde{I}_s)$.

Nous proposons un algorithme nommé Gen-GBAF, pour la génération de la représentation condensée des règles d'association approximatives graduelles/floues $\mathcal{RCA}$ à partir de l'ensemble des motifs graduels/flous clos et l'ensemble des générateurs minimaux graduels/flous. Le pseudo-code de l'algorithme Gen-GBAF est donné par l'algorithme 10.

Données :
- L'ensemble des motifs graduels/flous clos $\mathcal{MC}$ et leurs générateurs minimaux graduels/flous associés ;
- *minConf*.

Résultat : La base $\mathcal{RCA}$.

début
  $\mathcal{RCA} \leftarrow \emptyset$;
  pour chaque *motif graduel/flou clos* $\tilde{I} \in \mathcal{MC}$ **faire**
    pour chaque *générateur* $\tilde{g} \in \mathcal{MG}_{\tilde{I}}$ *tel que* $\tilde{I} < \tilde{I}_1 \wedge \frac{support(\tilde{I}_1)}{support(\tilde{I})} \geq minConf$ **faire**
      $\mathcal{RCA} \leftarrow \mathcal{RCA} \cup \tilde{R} : \tilde{g} \Rightarrow (\tilde{I}_1 \text{ - } \tilde{g})$
  retourner $\mathcal{RCA}$;
fin

Algorithme 10 : Algorithme Gen-GBAF

6.3.3 Couverture transitive de la représentation condensée de règles graduelles floues approximatives ($\mathcal{CTGF}$)

Afin de réduire le nombre de règles génériques approximatives graduelles/floues, nous pouvons déduire, à partir de la définition de la représentation condensée $\mathcal{RCA}$, une autre base ($\mathcal{CTGF}$) qui est sa *couverture transitive*.

Définition 42 *La couverture transitive $\mathcal{CTGF}$ de la représentation condensée $\mathcal{RCA}$ est définie par :*

$$\mathcal{CTGF} = \{ \tilde{R} : \tilde{g} \Rightarrow (\tilde{I}_1 \text{ - } \tilde{g}) | \ \tilde{I}, \tilde{I}_1 \in \mathcal{FC}_{\mathcal{K}} \wedge \tilde{g} \in \mathcal{MG}_{\tilde{I}} \wedge \tilde{I} \subset \tilde{I}_1 \wedge \nexists \tilde{I}_2 \text{ tel que } \tilde{I} \subset \tilde{I}_2 \subset \tilde{I}_1 \wedge Conf(\tilde{R}) \geq minConf \}$$

Les règles de la base $\mathcal{RCA}$ sont de la forme $\tilde{R} : \tilde{g} \Rightarrow (\tilde{I}_1 \text{ - } \tilde{g})$ telles que $\tilde{I}_1$ est un motif graduel/flou clos et $\tilde{g}$ un générateur graduel/flou d'un motif graduel/flou clos $\tilde{I}$, avec $\tilde{I} \subset \tilde{I}_1$ et il n'existe pas un motif $\tilde{I}_2$ tel que $\tilde{I} \subset \tilde{I}_2 \subset \tilde{I}_1$, sont des règles *transitives floues*. La couverture transitive $\mathcal{CTGF}$ de la base $\mathcal{RCA}$ est constituée de règles d'association floues de la forme $\tilde{R} : \tilde{g} \Rightarrow (\tilde{I}_1 \text{ - } \tilde{g})$, où $\tilde{g}$ est un générateur minimal graduel/flou d'un motif fraduel/flou clos $\tilde{I}$ ayant pour successeur immédiat, dans l'Iceberg du treillis de Galois, le motif graduel/flou clos $\tilde{I}_1$. Les règles d'association graduelles/floues transitives appartenant à la base $\mathcal{CTGF}$, possèdent des valeurs de confiance supérieures ou égales à celles des règles non transitives appartenant à $\mathcal{RCA}$. Nous pouvons conclure alors, que la couverture transitive permet de réduire le nombre de règles

graduelles/floues extraites en conservant les règles ayant des valeurs de confiance les plus élevées. Nous proposons un algorithme nommé Gen-CTGF, pour générer la couverture transitive $\mathcal{CTGF}$ de la représentation condensée des règles d'association approximatives graduelles/floues $\mathcal{RCA}$, à partir de l'ensemble des motifs graduels/flous clos et l'ensemble des générateurs minimaux graduels/flous. Le pseudo-code de l'algorithme Gen-CTGF est présenté par l'algorithme 11.

Données :
- L'ensemble des motifs graduels/flous clos $\mathcal{MC}$ et leurs générateurs minimaux graduels/flous associés ;
- *minConf*.

Résultat : La base $\mathcal{CTGF}$.

1 **début**
2 $\quad\mathcal{CTGF} \leftarrow \emptyset$;
3 $\quad$**pour chaque** *motif graduel/flou clos* $\tilde{I} \in \mathcal{MC}$ **faire**
4 $\quad\quad$**pour chaque** *générateur* $\tilde{g} \in \mathcal{MG}_{\tilde{I}}$ *tel que* $\tilde{I}_1$ *est successeur immédiat de* $\tilde{I} \wedge \frac{support(\tilde{I}_1)}{support(\tilde{I})} \geq minConf$ **faire**
5 $\quad\quad\quad\mathcal{CTGF} \leftarrow \mathcal{CTGF} \cup \tilde{R} : \tilde{g} \Rightarrow (\tilde{I}_1 \text{ - } \tilde{g})$
6 $\quad$retourner $\mathcal{CTGF}$;
7 **fin**

Algorithme 11 : Algorithme Gen-CTGF

Remarque 6 *Les représentations condensées des règles d'association graduelles/floues ($\mathcal{RCE}$, $\mathcal{RCA}$ et $\mathcal{CTGF}$) peuvent être obtenues directement à partir de l'Iceberg du treillis de Galois graduel/flou.*

En effet, étant donné un Iceberg du treillis de Galois graduel/flou dont chaque élément (i.e., motif graduel/flou clos) est étiqueté par la liste de ses générateurs minimaux graduels/flous, les règles génériques exactes graduelles/floues sont des implications "intra-noeud" avec une confiance égale à 1. En revanche, les règles génériques approximatives graduelles/floues sont des implications "inter-noeuds" assorties d'une mesure de confiance et qui mettent en jeu deux classes d'équivalences comparables (i.e., selon l'opérateur de l'ordre partiel). Les règles d'association transitives (i.e., les règles appartenant à $\mathcal{CTGF}$) sont aussi des implications "intra-noeuds" sauf que la conclusion d'une règle non transitive graduelle/floue provient d'une classe d'équivalence "directement supérieure" à sa prémisse.

6.4 Dérivation des règles d'association floues redondantes

Par définition, une approche d'extraction de représentation condensée est dite *sans perte d'informations* si elle satisfait les deux critères suivants [Gasmi *et al.*, 2005] :

1. ***Dérivabilité*** : le mécanisme d'inférence associé permettant la dérivation de toutes les règles redondantes doit être *correct* (*i.e.*, il ne permet la dérivation que des règles valides) et *complet* (*i.e.*, l'ensemble de toutes les règles peut être dérivé).
2. ***Informativité*** : la représentation condensée de règles d'association floues doit permettre de retrouver avec exactitude le support et la confiance des règles dérivées (*i.e.*, redondantes).

Dans ce qui suit, nous allons définir la notion de *redondance* des règles d'association graduelles/floues. Ensuite, nous allons prouver dans la section 6.4.2, que l'approche d'extraction du couple ($\mathcal{RCE}$, $\mathcal{RCA}$) est une approche sans perte d'information.

6.4.1 Notion de redondance :

La notion de redondance pour les règles d'association graduelles/floues est définie comme suit :

Définition 43 *Soit $\mathcal{GFAR}$ l'ensemble de toutes les règles graduelles/floues extraites d'un contexte d'extraction graduel $\mathcal{K}$ ou flou $\mathcal{K}_{\tilde{C}}$. Une règle graduelle/floue $\tilde{R} : \tilde{I}_1 \overset{s,c}{\Rightarrow} \tilde{I}_2 \in \mathcal{GFAR}$ est dite redondante (ou dérivable) par rapport à (de) $\tilde{R}' : \tilde{I_1}' \overset{s,c}{\Rightarrow} \tilde{I_2}'$, si $\tilde{R}$ satisfait les conditions suivantes :*

1. *$Support(\tilde{R}) = Support(\tilde{R}') = s \wedge Confiance(\tilde{R}) = Confiance(\tilde{R}') = c$.*
2. *$\tilde{I_1}' \subseteq \tilde{I}_1 \wedge \tilde{I}_2 \subset \tilde{I_2}'$.*

D'après la définition 43, une règle d'association graduelle/floue est redondante si elle renvoie la même information ou une information moins générale que l'information renvoyée par une autre règle de même *utilité* et de même *pertinence*. Les règles d'association graduelles/floues non redondantes sont donc définies en tenant compte des deux mesures *support* et *confiance*. Une règle d'association graduelle/floue $\tilde{R}'$ est non redondante, s'il n'existe pas une autre règle d'association graduelle/floue $\tilde{R}$ ayant les mêmes valeurs de support et de confiance, dont la prémisse est un sur-ensemble de la prémisse de $\tilde{R}'$ et la conclusion est un sous-ensemble de la conclusion de $\tilde{R}'$.

6.4.2 Mécanismes d'inférence

Dans ce qui suit, nous proposons un *système axiomatique* permettant de retrouver l'ensemble de toutes les règles d'association graduelles/floues redondantes valides, nous prouvons également la correction et la complétude de ce système.

Proposition 9 *Soient le couple ($\mathcal{RCE}$, $\mathcal{RCA}$) et $\mathcal{GFAR}$ l'ensemble de toutes les règles graduelles/floues valides pouvant être extraites d'un contexte d'extraction graduel $\mathcal{K}$ ou flou $\mathcal{K}_{\tilde{C}}$ par le biais du système axiomatique suivant :*

Augmentation à gauche *:*

- *Si $\tilde{R} : X \overset{s,c}{\Rightarrow} Y \in \mathcal{RCE}$ et $Z \subset Y$, alors $\tilde{R}' : XZ \overset{s,c}{\Rightarrow} (Y - Z) \in \mathcal{GFAR}$, $\forall\ Z \subset Y$ (i.e., $\tilde{R}'$ est valide).*
- *Si $\tilde{R} : X \overset{s,c}{\Rightarrow} Y \in \mathcal{RCA}$ alors $\tilde{R}' : XZ \overset{s,c}{\Rightarrow} (Y - Z) \in \mathcal{GFAR}$, tel que support($XZ$) = support($X$) et $Z \subset Y$.*

Décomposition à droite *:*

- *Si $\tilde{R} : X \overset{s,c}{\Rightarrow} Y \in \mathcal{RCE}$ alors $\tilde{R}' : X \overset{s,c}{\Rightarrow} Z \in \mathcal{GFAR}$, $\forall\ Z \subset Y$.*
- *Si $\tilde{R} : X \overset{s,c}{\Rightarrow} Y \in \mathcal{RCA}$ alors $\tilde{R}' : X \overset{s,c}{\Rightarrow} Z \in \mathcal{GFAR}$, tel que support($XZ$) = support($XY$) et $Z \subset Y$.*

*Ce système axiomatique est **correct** et **valide**.*

Preuve 4 *La preuve de la proposition ci-dessus est divisée en deux parties comme suit :*

a . Correction du système axiomatique

*Pour démontrer la correction du système axiomatique, que nous avons proposé, il suffit de prouver que l'ensemble de toutes les règles d'association graduelles/floues, dérivées à partir du couple ($\mathcal{RCE}$, $\mathcal{RCA}$), sont **valides** (i.e., leurs supports et confiances sont, respectivement, supérieurs ou égaux à minSup et minConf).*

Augmentation à gauche *:*

- *Si $\tilde{R} : X \Rightarrow Y \in \mathcal{RCE}$ alors confiance($\tilde{R}$)= $\frac{support(XY)}{support(X)}$ =1, i.e., support(XY)= support(X). Soit la règle $\tilde{R}' : XZ \Rightarrow (Y - Z)$ tel que $Z \subset Y$, puisque $X \subset XZ$ alors support(X) $\geq$ support(XZ). Nous avons donc confiance($\tilde{R}'$) = $\frac{support(XY)}{support(XZ)} \geq$ confiance($\tilde{R}$) et par conséquent $\tilde{R}'$ est une règle d'association floue valide.*
  La confiance d'une règle est une mesure statistique appartenant à l'intervalle $[0, 1]$. Nous avons confiance($\tilde{R}$)= 1 et confiance($\tilde{R}'$) $\geq$ confiance($\tilde{R}$), nous pouvons alors conclure que confiance($\tilde{R}'$) = 1.

- *Si* $\tilde{R} : X \overset{c}{\Rightarrow} Y \in \mathcal{RCA}$ *alors confiance*($\tilde{R}$)= $\frac{support(XY)}{support(X)}$ =*c. Soit la règle graduelle/floue* $\tilde{R}' : XZ \overset{c'}{\Rightarrow} (Y - Z)$ *tel que* $Z \subset Y$. *Puisque,* $Z \subset Y$ *alors support*($\tilde{R}'$)= $|\tilde{g}_{\tilde{C}}(XZ \cup (Y - Z))|$ = *support*(XY) =*support*($\tilde{R}$). *D'autre part, nous avons support*(XZ) = *support*(X), *donc confiance*($\tilde{R}'$) = $\frac{support(XY)}{support(XZ)}$ = c' = $\frac{support(XY)}{support(X)}$ = *confiance*($\tilde{R}$) = c .

Décomposition à droite *:*

- *Si* $\tilde{R} : X \Rightarrow Y \in \mathcal{RCE}$ *alors confiance*($\tilde{R}$)= $\frac{support(XY)}{support(X)}$ =1, *i.e., support*(XY)= *support*(X). *Soit la règle* $\tilde{R}' : XZ \Rightarrow (Y - Z)$ *tel que* $Z \subset Y$, *puisque* $X \subset XZ$ *alors support*(X) $\geq$ *support*(XZ). *Nous avons donc confiance*($\tilde{R}'$) = $\frac{support(XY)}{support(XZ)}$ $\geq$ *confiance*($\tilde{R}$) *et par conséquent* $\tilde{R}'$ *est une règle d'association graduelle/floue valide.*
  La confiance d'une règle est une mesure statistique appartenant à l'intervalle $[0, 1]$.
  Nous avons confiance($\tilde{R}$)= 1 *et confiance*($\tilde{R}'$) $\geq$ *confiance*($\tilde{R}$), *nous pouvons alors conclure que confiance*($\tilde{R}'$) = 1.
- *Si* $\tilde{R} : X \overset{c}{\Rightarrow} Y \in \mathcal{RCA}$ *alors confiance*($\tilde{R}$)= $\frac{support(XY)}{support(X)}$ =*c. Soit la règle graduelle/floue* $\tilde{R}' : XZ \overset{c'}{\Rightarrow} (Y - Z)$ *tel que* $Z \subset Y$.
  Puisque $Z \subset Y$ *alors support*($\tilde{R}'$)= $|\tilde{g}_{\tilde{C}}(XZ \cup (Y - Z))|$ = *support*(XY)
  =*support*($\tilde{R}$). *D'autre part, nous avons support*(XZ) = *support*(X), *donc confiance*($\tilde{R}'$) = $\frac{support(XY)}{support(XZ)}$ = c' = $\frac{support(XY)}{support(X)}$ = *confiance*($\tilde{R}$) = c.

b . Complétude du système axiomatique

Pour prouver la complétude du système axiomatique proposé il suffit de montrer que s'il est appliqué aux représentations condensées $\mathcal{RCE}$ *et* $\mathcal{RCA}$, *il permet alors la dérivation de* ***toutes*** *les règles d'association graduelles/floues valides pouvant être extraites à partir d'un contexte d'extraction flou.*

Soit $\tilde{R} : X \overset{s,c}{\Rightarrow} Y - X$ *une règle d'association graduelle/floue valide entre deux itemsets flous* X *et* Y *:*

- *Si* $\tilde{R}$ *est une règle d'association graduelle/floue* ***exacte*** *(i.e., confiance*($\tilde{R}$)= c = 1), *nous avons nécessairement* $X \subset Y$ *et puisque confiance*($\tilde{R}$)=1, *nous avons support*(X)= *support*(Y). *Nous pouvons alors conclure que h*(X)= *h*(Y)= $\tilde{I}$. *Le motif graduel/flou* $\tilde{I}$ *est un motif graduel/flou clos* $\in \tilde{\mathcal{FC}}$ *et il existe forcément une règle d'association graduelle/floue* $\tilde{R}'$:$\tilde{g} \Rightarrow (\tilde{I}$ - $\tilde{g}) \in \mathcal{RCE}$ *telle que* $\tilde{g}$ *est un générateur minimal graduel/flou de* $\tilde{I}$ *pour lequel nous avons :*
  – $\tilde{g} \subset X$ *et* $\tilde{g} \subset Y$, *(application de l'axiome d'augmentation) ;*
  – $\tilde{g} = X$ *et* $\tilde{g} \subset Y$, *(application de l'axiome de décomposition).*
  Nous démontrons que la règle $\tilde{R}$ *et son support peuvent être déduits à partir de la règle* $\tilde{R}'$.

Puisque $\tilde{g} \subseteq X$ et $\tilde{g} \subset Y \subseteq \tilde{I}$, la règle d'association floue $\tilde{R}$ peut être reconstruite à partir de $\tilde{R}'$ (en appliquant l'axiome d'augmentation ou de décomposition). De h(X)= h(Y)= $\tilde{I}$, nous pouvons déduire que support($\tilde{R}$)= s = support(Y)= support(h(Y))= support($\tilde{R}'$).

– *Si $\tilde{R}$ est une règle d'association graduelle/floue* ***approximative*** *(i.e., confiance($\tilde{R}$)= s $\leq$ 1), nous avons nécessairement $X \subset Y$ et puisque confiance($\tilde{R}$) $\leq$ 1, nous avons h(X) $\subset$ h(Y). Quelques soient les motifs graduels/flous X et Y, il existe un générateur minimal graduel/flou $\tilde{g_1}$ d'un motif graduel/flou clos $\tilde{I_1}$ tels que $\tilde{g_1} \subset X \subset$ h(X)= $\tilde{I_1}$ et un générateur d'un motif graduel/flou clos $\tilde{I_2}$ tels que $\tilde{g_2} \subset Y \subseteq$ h(Y)= $\tilde{I_2}$. Puisque $X \subset Y$, nous avons $X \subseteq \tilde{I_1} \subset Y \subseteq \tilde{I_2}$ et la règle $\tilde{R}'$: $\tilde{g_1} \overset{s',c'}{\Rightarrow} (\tilde{I_2}$ - $\tilde{g_1}) \in \mathcal{RCA}$. Nous démontrons que la règle $\tilde{R}$ ainsi que son support et sa confiance peuvent être déduits de la règle $\tilde{R}'$, de son support et de sa confiance. Puisque, $\tilde{g_1} \subset X \subseteq \tilde{I_1} \subset Y \subseteq \tilde{I_2}$ alors la prémisse et la conclusion de la règle d'association graduelle/floue $\tilde{R}$ peuvent être reconstruites à partir de $\tilde{R}'$. De plus, nous avons h(Y)= $\tilde{I_2}$ et par suite support($\tilde{R}$)= s = support(Y)= support($\tilde{I_2}$)= support($\tilde{R}'$)= s'. Puisque, $X \subseteq \tilde{I_1}$ alors nous avons support(X)= support($\tilde{I_1}$) et nous pouvons déduire que :*
confiance($\tilde{R}$) = c= $\frac{support(Y)}{support(X)} = \frac{support(\tilde{I_2})}{support(\tilde{I_1})}$ = confiance($\tilde{R}'$) = c'.

6.5 Mise en oeuvre

Dans cette section, nous montrons la validité et la compacité de nos représentations condensées de règles graduelles/floues. En effet, d'après la table 6.1, nous pouvons constater que :

– Même pour une valeur élevée de *minSup* (*i.e.*, 0,95), le nombre règles d'associations floues (RAFs) exactes et transitives reste tout de même important allant de 688 règles (pour un extrait de 100 gènes) à 1.957.257 règles (pour un extrait de 900 gènes).
– Le nombre de règles floues condensées est beaucoup plus inférieur à celui de toutes les règles d'association floues. En effet, le nombre total de règles d'association floues pouvant être extraites à partir d'un contexte a été évalué à $2^{2\times l}$, où l est la longueur du plus long itemset fréquent [Zaki, 2004]. Ainsi, si nous considérons un extrait de 200 gènes (les items), le nombre de RAFs pouvant être extraites, sera égal à 2^{400} (un chiffre que nous ne pouvons même pas lire).

La table 6.2, illustre la compacité du couple ($\mathcal{CTGF}$, $\mathcal{RCE}$) par rapport au nombre total de toutes les RAFs redondantes à partir des deux bases Chess et Mushroom.

TABLE 6.1 – Compacité des règles floues exactes *vs* le nombre de gènes de la base SAGE (*minSup*=95%).

#Gènes	Motifs flous clos	RAFs Trans. ($\mathcal{CTGF}$)	RAFs Exact. ($\mathcal{RCE}$)	RAFs redondantes exact.	Taux de compacité
400	2 649	105 169	10 572	8 379 892	0.0012
500	3 049	189 534	16 990	16 863 554	0.0010
600	4 084	448 869	33 562	40 019 440	0.0008
700	5 391	1 018 551	64 282	89 500 100	0.0007
800	6 373	1 512 339	87 563	139 423 724	0.0006
900	7 029	1 854 399	102 858	184 348 722	0.0005

TABLE 6.2 – Taux de compacité du couple ($\mathcal{CTGF}$, $\mathcal{RCE}$) pour les bases CHESS et MUSHROOM.

Base	*minSup* %	*minConf* %	$\lvert\mathcal{CTGF}\rvert + \lvert\mathcal{RCE}\rvert$	RAFs redondantes	Taux de compacité
CHESS	70	100	490	58 484	0.0083
MUSHROOM	50	100	1 692	189 564	0.0089

6.6 Conclusion

Dans ce chapitre, nous avons proposé des algorithmes d'extraction des représentations condensées informatives de règles d'association graduelles et floues. Nous avons également introduits les mécanismes d'inférence permettant la dérivation de toutes les règles graduelles/floues redondantes à partir des représentations condensées sans perte d'information.

Conclusion et perspectives

Depuis son introduction dans les années 90 [Agrawal *et al.*, 1993], le problème d'extraction de règles d'association et de motifs ou itemsets fréquents, a suscité beaucoup d'intérêt dans la communauté de la fouille de données. En effet, plusieurs problèmes liés à l'extraction des motifs fréquents (*i.e.*, le nombre des motifs extraits et le temps de leurs extraction) ont été étudiés et plusieurs approches ont été proposées dans le cadre de ce sujet. D'un autre côté, plusieurs sémantiques ont été attribuées aux motifs fréquents, ce qui a donné lieu à des nouvelles notions telles que la gradualité, la temporalité, le flou, *etc.*
Dans ce travail de thèse, nous nous sommes intéressés aux motifs portant sur la gradualité et le flou. Ainsi, nous avons proposé des approches d'extraction de ces types de motifs en essayant de réduire leurs nombres d'une manière efficace et sans perdre de l'information.

1 Synthèse des travaux entrepris

Dans le cadre de cette thèse, nous nous sommes intéressés à l'extraction de motifs flous et graduels en se basant sur la notion de la cloture de la correspondance de Galois. Ainsi, nous avons introduit les notions de motifs graduels clos et de motifs flous clos. Nous avons également introduit la notion de motif graduel flou.

Le premier chapitre de ce mémoire de thèse a dressé une introduction au contexte et à la problématique traitée dans cette thèse. Les motivations de ce travail ainsi que les principales contributions y sont également présentées.

Le chapitre 2 de cette thèse, est consacré à présenter les notions de bases et les fondements mathématiques relatifs aux sous-ensembles flous ainsi que les méthodes d'acquisition des modalités floues proposées.

Dans le chapitre 3, nous avons introduit la formalisation des motifs et règles graduels. En effet, nous avons dressé un panel de tous les travaux existants portant sur l'extraction et la

recherche de motifs et règles graduels.

Dans le chapitre 4, nous avons présenté la notion de représentations condensées décrivant un ensemble réduit de tous les motifs pouvant être extraits. Nous avons présenté les représentations les plus marquantes dans la littérature (*i.e.*, celles basées sur les itemsets maximaux, sur les itemsets clos, etc).

Dans le chapitre 5, nous avons introduit nos différentes contributions, que nous pouvons récapituler, comme suit :

- **Extraction de motifs flous clos** : la correspondance de Galois classique ne permet pas d'extraire des motifs flous ayant une certaine particularité par rapport aux motifs classiques. En effet, les motifs flous maintiennent de l'information en outre (*i.e.*, les degrés d'appartenance) par rapport aux motifs classiques décrivant une relation de présence ou d'absence dans une transaction. Ainsi, pour extraire les motifs flous clos, il est évident que de nouveaux opérateurs de la correspondance de Galois doivent être définis. Ainsi, nous avons pu extraire des motifs flous clos et nous avons montré la validité de notre correspondance de Galois floue.
- **Extraction de motifs graduels clos** : nous avons introduit une nouvelle correspondance de Galois pour extraire les motifs graduels clos. Pour y parvenir, nous avons introduit la notion d'ensemble de séquences et nous avons défini les différentes opérations ensemblistes pouvant être effectuées sur ces ensembles.
- **Extraction de motifs graduels flous** : un motif graduel flou de la forme "*plus/moins* A_1 *est* F_1*, ..., plus/moins* A_n *est* F_n", décrit une variation d'un attribut A_i associé à une modalité floue F_i elle même décrite par une fonction d'appartenance. Pour extraire un tel motif, il fallait tout d'abord obtenir la modalité floue F_i. Ainsi, nous avons proposé deux approches d'acquisition de ces modalités. La première est basée sur la médiane des valeurs des attributs. La deuxième consistait à définir un algorithme génétique permettant d'obtenir à la fois les modalités floues et les motifs graduels flous les plus pertinents pouvant être extraits à partir de ces modalités flous. Les différentes expérimentations, que nous avons menées sur des bases synthétiques ou réelles, ont montré que notre méthode a permis d'extraire des motifs graduels flous pertinents (*i.e.*, avec des valeurs de support et des longueur élevées) que les méthodes classiques n'ont pas réussi à le faire.

Dans le chapitre 6, nous nous sommes intéressés à l'extraction de représentation condensées de règles graduelles/floues. En effet, nous avons défini trois bases génériques de toutes les règles graduelles/floues. Nous avons également présenté le mécanisme d'inférence permettant

de déduire toutes les règles redondantes à partir de ces bases génériques.

2 Perspectives

Plusieurs perspectives relatives à nos différentes contributions sont envisageables. Nous discutons des perspectives à court terme qui visent à améliorer l'efficacité d'extraction des motifs flous.

2.1 Extension de l'algorithme génétique

Dans le cinquième chapitre, nous avons proposé un algorithme génétique permettant d'obtenir les modalités floues les plus pertinentes permettant d'extraire des motifs graduels flous avec des valeurs de support élevées. Nous pourrons changer d'objectif et extraire les modalités floues les plus pertinentes permettant d'extraire le plus de règles graduelles floues avec des valeurs de confiance élevées ou encore les règles les plus discriminates permettant de classifier les données de la base. Pour atteindre cet objectif, il va falloir redéfinir la représentation des individus de la population, la fonction d'évaluation (*i.e.*, *fitness*) et les opérations génétiques (*i.e.*, mutation et croisement).

2.2 Acquisition de toute la partition floue d'un attribut

Pour extraire les motifs graduels flous, nous avons proposé de travailler sur une seule modalité floue de chaque attribut. Nous proposons de travailler sur toute une partition floue d'un attribut au lieu de se concentrer sur une seule fonction d'appartenance de l'attribut. Cette exploration nous permettra de définir d'autres sémantiques d'élagage des motifs graduels flous en considérant la relation entre les différentes modalités floues de la partition.

2.3 Restitution des motifs et règles graduelles/floues

La majorité des travaux de la communauté s'intéressent très fortement aux premières étapes du processus de l'ECD (i.e., sélection, transformation et fouille de données). Toutefois, des efforts considérables doivent être accordés à l'étape d'interprétation et de visualisation des résultats. Bien qu'il existait plusieurs plateformes permettant une visualisation graphique des résultats de certains algorithmes tel que WEKA [12], il reste de proposer d'autres outils permettant de faire intervenir l'utilisateur au coeur de la navigation. En effet, ces outils doivent permettre à l'utilisateur de choisir quels motifs il veut visualiser, de quelle manière il veut les dérouler, les

12. http ://www.cs.waikato.ac.nz/ml/weka

compresser ou encore les mettre en relation avec d'autres motifs. Cet objectif, au confluent de la fouille de données et de l'interaction homme-machine, est une perspective à long terme.

2.4 Amélioration de l'extraction des représentations condensées de règles graduelles/floues

Les représentations condensées de règles graduelles/floues, que nous avons définies dans le chapitre 6, sont basées sur les motifs graduels/flous clos et leurs générateurs minimaux. Dans le but de minimiser la taille de ces représentations, nous allons étudier la possibilité de réduire le nombre des générateurs minimaux graduels/flous en introduisant la notion de générateurs minimaux succincts [Dongy *et al.*, 2005]. Ceci nous permettra de parler de générateurs minimaux succincts graduels/flous qui peuvent servir à l'extraction d'une nouvelle représentation condensée de règles graduelles/floues qui soit plus compacte que le couple de bases ($\mathcal{RCE}$, $\mathcal{RCA}$).

Bibliographie

[Agier *et al.*, 2007] Agier, M., Petit, J. et Suzuki, E. (2007). Unifying framework for rule semantics : Application to gene expression data. *Fundam. Inf.*, 78:543–559.

[Agrawal *et al.*, 1993] Agrawal, R., Imielinski, T. et A.Swami (May 1993). Mining associations rules between sets of items in large databases. *In Proceedings of the 1993 ACM SIGMOD Conference*, Washington DC, USA.

[Agrawal et Srikant, 1994] Agrawal, R. et Srikant, R. (1994). Fast algorithms for mining association rules in large databases. *In Proceedings of the 20th Intl. Conference on Very Large Databases*, pages 487–499, Santiago, Chile.

[Aladenise et Bouchon-Meunier, 1997] Aladenise, N. et Bouchon-Meunier, B. (1997). Acquisition de connaissances imparfaites : mise en évidence d'une fonction d'appartenance. *Revue Internationale de Systémique*, 11(1):109–127.

[Alcalá-Fdez *et al.*, 2009] Alcalá-Fdez, J., Alcalá, R., Gacto, M. et Herrera, F. (2009). Learning the membership function contexts for mining fuzzy association rules by using genetic algorithms. *Fuzzy Sets and Systems.*, 160(7):905–921.

[Ayres *et al.*, 2002] Ayres, J., Gehrke, J., Tiu, T. et FlannicK, J. (2002). Sequential pattern mining using a bitmap representation. *In Proceedings of the SIGKDD'02*, Edmonton, Alberta, Canada. ACM.

[Barbut et Monjardet, 1970] Barbut, M. et Monjardet, B. (1970). *Ordre et classification. Algèbre et Combinatoire*. Hachette, Tome II.

[Bastide, 2000] Bastide, Y. (2000). Data mining : algorithmes par niveau, techniques d'implantation et applications. Thèse de doctorat, Ecole Doctorale Sciences pour l'Ingénieur de Clermont-Ferrand, Université Blaise Pascal, France.

[Bastide *et al.*, 2000a] Bastide, Y., Pasquier, N., Taouil, R., Lakhal, L. et Stumme, G. (2000a). Mining minimal non-redundant association rules using frequent closed itemsets. *In Proceedings of the International Conference DOOD'2000, LNAI, volume 1861, Springer-Verlag, London, UK*, pages 972–986.

[Bastide *et al.*, 2000b] Bastide, Y., Taouil, R., Pasquier, N., Stumme, G. et Lakhal, L. (2000b). Mining frequent patterns with counting inference. *Proceedings of The Sixth ACM-SIGKDD International Conference on Knowledge Discovery and Data Mining, Boston, Massachusetts, USA*, 2(2):66–75.

[Bayardo, 1998] Bayardo, R. (1998). Efficiently mining long patterns from databases. *In Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data*, SIGMOD'98, pages 85–93, Seattle.

[Bellman *et al.*, 1966] Bellman, R., Kalaba, L. et Zadeh, L. (1966). Abstraction and pattern classification. *Journal of Mathematical Analysis and Applications*, 13:1–7.

[Belohlavèk, 1998] Belohlavèk, R. (1998). Fuzzy concepts and conceptual structure : induced similarities. *In Proceedings of the Joint Conference on Information Science. USA, North Carolina*, volume 1, pages 179–182.

[BenYahia et Jaoua, 2000] BenYahia, S. et Jaoua, A. (2000). A top-down approach for mining fuzzy association rules. *In Proceedings of the 8th Int. Conf. Information Processing and Management of Uncertainty in Knowledge-Based Systems. (Madrid, Spain)*, pages 952–959.

[BenYahia et Jaoua, 2001] BenYahia, S. et Jaoua, A. (March 2001). Discovering knowledge from fuzzy concept lattice. *In* A. Kandel, M. L. et Bunke, H., éditeurs : *Data mining and Computational Intelligence, Studies in Fuzziness and Soft Computing*, volume 68, pages 167–190. Physica–Verlag.

[BenYahia et Nguifo, 2004] BenYahia, S. et Nguifo, E. M. (2004). Approches d'extraction de règles d'association basées sur la correspondance de galois. *In* Boulicault, J.-F. et Cremilleux, B., éditeurs : *Revue d'Ingénierie des Systèmes d'Information (ISI), Hermes-Lavoisier*, volume 3–4, pages 23–55.

[Berry et Linoff., 2004] Berry, M. J. A. et Linoff., G. S. (2004). *Data Mining Techniques : For Marketing, Sales, and Customer Relationship Management, Second Edition.* Wiley Publishing.

[Berzal *et al.*, 2007] Berzal, F., Cubero, J., Sánchez, D., Vila, M. et Serrano, J. (2007). An alternative approach to discover gradual dependencies. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 15(5):559– 570.

[Bezdek, 1981] Bezdek, J. (1981). *Pattern Recognition with Fuzzy Objective Function Algorithms.* Kluwer Academic Publishers, Norwell, MA, USA.

[Bilgiç et Turksen, 1995] Bilgiç, T. et Turksen, I. (1995). Measurement-theoritic justification of fuzzy set connectives. *Fuzzy Sets and Systems*, 76(3):289–308.

[Black, 1937] Black, M. (1937). Vagueness. *philosophy of Science*, 4:427–455.

[Boose et Otto, 1985] Boose, J. et Otto, K. (1985). A knowledge acquisition program for expert systems based on personal construct psychology. *International journal of Man-Machine Studies*, 23:495–525.

[Bosc *et al.*, 1997] Bosc, P., Lietard, L. et Pivert, O. (1997). Gradualité, imprécision et dépendances fonctionnelles. *In* Ferrié, J., éditeur : *13ème Journées Bases de Données Avancées (BDA'97).*

[Bosc *et al.*, 1999] Bosc, P., Pivert, O. et Ughetto, L. (1999). On data summaries based on gradual rules. *In Proceedings of the 6th International Conference on Computational Intelligence, Theory and Applications : Fuzzy Days*, pages 512–521, London, UK. Springer-Verlag.

[Botzheim *et al.*, 2002] Botzheim, J., Mori, B., Koczy, L. et Ruano, A. (2002). bacterial algorithm applied for fuzzy rule extraction. *In Proc. of IPMU'02*, pages 1021–1026.

[Bouchon-Meunier, 1990] Bouchon-Meunier, B. (1990). Acquisition numérique/symbolique de connaissances graduelles. *In Actes des 3èmes journées nationales PRC-GDR Intelligence Artificielle, Paris*, pages 127–138.

[Bouchon-Meunier, 1995] Bouchon-Meunier, B. (1995). *La logique floue et ses applications.* Edition Addison-Wesley France, SA.

[Boulicaut *et al.*, 2000] Boulicaut, J., Bykowski, A. et Rigotti, C. (2000). Approximation of frequency queries by means of free-sets. *In Proceedings of the Int. Conf. on Principles of Data Mining and Knowledge Discovery (PKDD'01)*, pages 75–85.

[Boulicaut *et al.*, 2003] Boulicaut, J., Bykowski, A. et Rigotti, C. (2003). Free-sets : A condensed representation of boolean data for the approximation of frequency queries. *Data Mining and Knowledge Discovery*, 7(1):5–22.

[Breuker et de Greef, 1993] Breuker, J. et de Greef, P. (1993). Modelling system-user cooperation. *In* Schreiber, A., Wielinga, B. et Breuker, J., éditeurs : *KADS : a Principled Approach to Knowledge Engineering*, pages 47 – 70. Academic Press, London.

[Burdick *et al.*, 2005] Burdick, D., Calimlim, M., Flannick, J., Gehrke, J. et Yiu, T. (2005). Mafia : A maximal frequent itemset algorithm. *IEEE Trans. Knowl. Data Eng.*, 17(11):1490–1504.

[Calders et Goethals, 2002] Calders, T. et Goethals, B. (2002). Mining all non-derivable frequent itemsets. *In* Elomaa, T., Mannila, H. et Toivonen, H., éditeurs : *Proceedings of the 6th European Conference on Principles of Data Mining and Knowledge Discovery, PKDD 2002, LNCS, volume 2431, Springer-Verlag, Helsinki, Finland*, pages 74–85.

[Calders et Goethals, 2007] Calders, T. et Goethals, B. (2007). Non-derivable itemset mining. *Data Mining and Knowledge Discovery*, 14(1):171–206.

[Chan et Au, 1997a] Chan, K. et Au, W. (1997a). An effective algorithm for mining interesting quantitative association rules. *In Proceedings of the 12th ACM Symp. on Applied Computing (SAC'97), San Jose, CA*.

[Chan et Au, 1997b] Chan, K. et Au, W. (1997b). Mining fuzzy association rules. *In Proceedings of the Sixth Int'l Conf. on Information and Knowledge Management (CIKM'97), Las Vegas, Nevada, USA*, pages 209–215.

[Chen *et al.*, 2000] Chen, G., Wei, Q. et Kerre, E. E. (2000). Fuzzy data mining : Discovery of fuzzy generalized association rules. *In G. Bordogna and G. Pasi, editors, Recent Issues on Fuzzy Databases. Springer-Verlag.*

[Chen et Otto, 1995] Chen, J. et Otto, K. (1995). Constructing membership functions using interpolation and measurement theory. *Fuzzy Sets and Systems*, 73:313–327.

[Choong *et al.*, 2009] Choong, Y. W., Jorio, L. D., Laurent, A., Laurent, D. et Teisseire, M. (2009). CBGP : Classification based on gradual patterns. *In Proceedings of the 2009 International Conference of Soft Computing and Pattern Recognition*, SOCPAR '09, pages 7–12, Washington, DC, USA. IEEE Computer Society.

[Davey et Priestley, 2002] Davey, B. et Priestley, H. (2002). *Introduction to Lattices and Order*. Cambridge University Press.

[Delgado *et al.*, 2003] Delgado, M., Marin, N., Martin-Bautista, M., Sanchez, D. et Vila, M. (2003). Mining fuzzy association rules : An overview. *BISC International Workshop on Soft Computing for Internet and Bioinformatics.*

[Derbel *et al.*, 2008] Derbel, I., Hachani, N. et Ounelli, H. (2008). Membership functions generation based on density function. *In Proceedings of the International Conference on Computational Intelligence and Security (CIS'08)*, pages 96–101.

[Di Jorio *et al.*, 2008] Di Jorio, L., Laurent, A. et Teisseire, M. (2008). Fast extraction of gradual association rules : A heuristic based method. *In Proceedings of the IEEE/ACM Int. Conf. on Soft computing as Transdisciplinary Science and Technology, CSTST'08*, Cergy, France.

[Di Jorio *et al.*, 2009a] Di Jorio, L., Laurent, A. et Teisseire, M. (2009a). Extraction efficace de règles graduelles. *In Actes de EGC'09.*

[Di Jorio *et al.*, 2009b] Di Jorio, L., Laurent, A. et Teisseire, M. (2009b). Mining frequent gradual itemsets from large databases. *In Proceedings of the Int. Conf. on Intelligent Data Analysis, IDA'09*, Lyon France.

[Dieng *et al.*, 1993] Dieng, R., Corby, O. et Lapalut, S. (1993). Acquisition of gradual knowledge. *In Proceedings of EKAW*, pages 407–426.

[Dimitrov et Rykov, 2004] Dimitrov, B. N. et Rykov, V. (2004). On reliability of hierarchical systems with gradual failures. *Journal of Mathematical Sciences*, 123(1):3802–3815.

[Dongy *et al.*, 2005] Dongy, G., Jiangy, C., Peiz, J., Li, J. et Wong, L. (2005). Mining succinct systems of minimal generators of formal concepts. *In Proceedings of Database Systems for Advanced Applications (DASFAA'2005), Beijing, China*, pages 175–187.

[Dârlea, 2010] Dârlea, G.-L. (2010). Un système de classification supervisée à base de règles implicatives. Thèse de doctorat, Université de Savoie, France.

[Dubois, 1991] Dubois, D. (1991). Fuzzy sets and their applications : Vilem novak, translated from czechoslovakian. bristol and philadelphia : Adam hilger, 1989, 248 pages. *Mathematical Social Sciences*, 21(2):193–197.

[Dubois et Prade,] Dubois, D. et Prade, H. Fuzzy sets and systems, theory and applications. *Academic Press, New York*.

[Dubois et Prade, 1986] Dubois, D. et Prade, H. (1986). Weighted minimum and maximum operations in fuzzy set theory. *Information Sciences*, (39):205–210.

[Dubois et Prade, 1991] Dubois, D. et Prade, H. (1991). Fuzzy sets in approximate reasoning, part II : Logical approaches. *Fuzzy Sets and Systems Journal*, 40:203–244.

[Dubois et Prade, 1992] Dubois, D. et Prade, H. (1992). Gradual inference rules in approximate reasoning. *Information Sciences*, 61(1-2):103–122.

[Dubois et Prade, 1995] Dubois, D. et Prade, H. (1995). Bases de règles floues en commande : Une discussion critique. Technical report irit/59-48-r, Institut de Recherche en Informatique de Toulouse (IRIT), France.

[Dubois et Prade, 1996a] Dubois, D. et Prade, H. (1996a). Logique floue, interpolation et commande. *Journal Européen des systèmes Automatisés*, 30:607–644.

[Dubois et Prade, 1996b] Dubois, D. et Prade, H. (1996b). What are fuzzy rules and how to use them ? *Fuzzy Sets and Systems*, (84):169–185.

[Dubois *et al.*, 1995] Dubois, D., Prade, H. et Grabisch, M. (1995). *Gradual rules and the approximation of control laws*, pages 147–181. John Wiley & Sons, Inc., New York, NY, USA.

[Elloumi, 2002] Elloumi, S. (2002). Apprentissage Supervisé par Localisation de Concepts dans les Contextes Flous ou Réels. Thèse de doctorat en informatique, Faculté des Sciences de Tunis, Département des Sciences Informatiques, Université d'El Manar.

[Fayyad *et al.*, 1996] Fayyad, U., Piatetsky-Shapiro, G. et dhraic Smyth, P. (1996). From data mining to knowledge discovery : An overview. *In* Fayyad, U., G.Piatetsky-Shapiro, Smyth, P. et Uthurusamy, R., éditeurs : *Advances in Knowledge Discovery and Data Mining*, pages 1–30. AAAI Press.

[Fiot *et al.*, 2007] Fiot, C., Laurent, A. et Teisseire, M. (2007). From crispness to fuzziness : Three algorithms for soft sequential pattern mining. *IEEE T. Fuzzy Systems*, 15(6):1263–1277.

[Fiot *et al.*, 2008a] Fiot, C., Masseglia, F., Laurent, A. et Teisseire, M. (2008a). Gradual trends in fuzzy sequential patterns. *In Proceedings of the Int. Conf. on Information Processing and Management of Uncertainty in Knowledge-based Systems (IPMU)*, Malaga, Spain.

[Fiot *et al.*, 2008b] Fiot, C., Masseglia, F., Laurent, A. et Teisseire, M. (2008b). Ted and eva : expressing temporal tendencies among quantitative variables using fuzzy sequential patterns. *In Proc. 17th IEEE Internat. Conf. on Fuzzy Systems (Fuzz IEEE'08)*.

[Frawley *et al.*, 1992] Frawley, W. J., Piatetsky-Shapiro, G. et Matheus, C. J. (1992). Knowledge discovery in databases - an overview. *Artificial Intelligence Magazine*, 13:57–70.

[Fu *et al.*, 1998] Fu, A., Wong, M., Sze, S. et Wong, W. (1998). Finding fuzzy sets for the mining of fuzzy association rules for numerical attributes. *In Proceedings of the first International Symposium of Intelligent Data Engeneering and Learning (IDEAL'98)*, pages 263–268.

[Galichet *et al.*, 2003] Galichet, S., Dubois, D. et Prade, H. (2003). Fuzzy interpolation and level 2 gradual rules. *In Proceedings of the EUSFLAT Conf.*, pages 506–511.

[Galichet *et al.*, 2004] Galichet, S., Dubois, D. et Prade, H. (2004). Imprecise specification of ill-known functions using gradual rules. *International Journal of Approximate Reasoning*, 35:205–222.

[Ganter et Wille, 1999] Ganter, B. et Wille, R. (1999). *Formal Concept Analysis*. Springer-Verlag.

[Gasmi *et al.*, 2005] Gasmi, G., BenYahia, S., Nguifo, E. M. et Slimani, Y. (2005). Discovering factual and implicative generic association rules. *In Actes de la Conférence Francophone sur l'Apprentissage Automatique (CAp'05), Nice, France.*

[Gyenesei, 2000] Gyenesei, A. (2000). A fuzzy approach for mining quantitative association rules. Rapport technique 336, Turku Center of Computer Science, University of Turku Departement of Computer Science.

[Hachani, 2010] Hachani, N. (2010). Détermination automatique des fonctions d'appartenance et interrogation flexible et coopérative des bases de données. Thèse de doctorat, Faculté des Sciences de Tunis, Tunisie.

[Han et Kamber, 2000] Han, J. et Kamber, M. (2000). *Data Mining : Concepts and Techniques*. Morgan Kaufmann.

[Hart, 1992] Hart, A. (1992). *A knowledge acquisition for expert systems.* 2nd edition, McGraw-Hill.

[Holland, 1992] Holland, J. (1992). *Adaptation in natural and artificial systems : an introductory analysis with applications to biology, control, and artificial intelligence.* MIT Press, Cambridge Mass., 1st MIT press ed. édition.

[Hüllermeier, 2002] Hüllermeier, E. (2002). Association rules for expressing gradual dependencies. *In Proceedings of the 6th European Conf. on Principles of Data Mining and Knowledge Discovery, PKDD'02*, pages 200–211. Springer-Verlag.

[Isabela *et al.*, 2002] Isabela, D., Lluis, G. et Sandra, S. (2002). Restoring consistency in systems of fuzzy gradual rules using similarity relations. *In Proceedings of the 16th Brazilian Symposium on Artificial Intelligence : Advances in Artificial Intelligence*, SBIA '02, pages 386–396, London, UK, UK. Springer-Verlag.

[Jaoua *et al.*, 2000] Jaoua, A., Alvi, F., Elloumi, S. et BenYahia, S. (2000). Galois connection in fuzzy binary relations : applications for discovering association rules and decision making. *In Proceedings of the Intl. Conference RELMICS'2000, Canada*, numéro 5, pages 141–149.

[Kaya et Alhajj, 2006] Kaya, M. et Alhajj, R. (2006). Utilizing genetic algorithms to optimize membership functions for fuzzy weighted association rules mining. *Applied Intelligence*, 24(1): 7–15.

[Klement *et al.*, 2002] Klement, E., Mesiar, R. et Pap, E. (2002). Triangular norms. *Kluwer Academic Publishers.*

[Kryszkiewicz, 1998] Kryszkiewicz, M. (1998). Representative association rules and minimum condition maximum consequence association rules. *In Proceedings of Second European Symposium on Principles of Data Mining and Knowledge Discovery (PKDD), 1998, LNCS, volume 1510, Springer-Verlag, Nantes, France*, pages 361–369.

[Kryszkiewicz, 2001] Kryszkiewicz, M. (2001). Concise representation of frequent patterns based on disjunction-free generators. *In Proceedings of the IEEE International Conference on Data Mining (ICDM) 2001, San Jose, California, USA*, pages 305–312.

[Kuok *et al.*, 1998] Kuok, C., Fu, A. et Wong, M. (1998). Mining fuzzy association rules in databases. *SIGMOD Record*, 27(1):41–46.

[Laurent *et al.*, 2009] Laurent, A., Lesot, M.-J. et Rifqi, M. (2009). Graank : Exploiting rank correlations for extracting gradual dependencies. *In Proceedings of the Eighth International Conference on Flexible Query Answering Systems (FQAS'09)*, Roskilde, Denmark.

[Lin et Kedem, 1998] Lin, D. et Kedem, Z. M. (1998). Pincer-search : A new algorithm for discovering the maximum frequent sets. *In In 6th Intl. Conf. Extending Database Technology*, pages 105–119.

[Luong, 2001] Luong, V. P. (2001). Raisonnement sur les règles d'association. *In Actes des 17ème Journées Bases de Données Avancées BDA'2001, Agadir (Maroc), Cépaduès Edition*, pages 299–310.

[Man *et al.*, 1996] Man, K., Tang, K. et Kwong, S. (1996). Genetic algorithms : concepts and applications. *IEEE Transactions on Industrial Electronics*, 43(5):519–534.

[Mannila et Toivonen, 1996] Mannila, H. et Toivonen, H. (1996). Multiple uses of frequent sets and condensed representations (extended abstract). *In Proceedings of KDD*, pages 189–194.

[Mannila et Toivonen, 1997] Mannila, H. et Toivonen, H. (1997). Levelwise search and borders of theories in knowledgediscovery. *Data Min. Knowl. Discov.*, 1:241–258.

[Masseglia, 2002] Masseglia, F. (2002). *Algorithmes et applications pour l'extraction de motifs séquentiels dans le domaine de la fouille de données : de l'incrémental au temps réel.* Thèse de doctorat, Université Versailles-UVSQ.

[Masseglia *et al.*, 2004] Masseglia, F., Poncelet, P. et Teisseire, M. (2004). Pre-processing time constraints for efficiently mining generalized sequential patterns. volume 0, pages 87–95, Los Alamitos, CA, USA. IEEE Computer Society.

[Miyazaki *et al.*, 2001] Miyazaki, J., Akutsu, S., Satow, N., Hirao, C. et Yao, Y. (2001). The gradual expression of troponin t isoforms in chicken wing muscles. *J Muscle Res Cell Motil*, 22(8):693–701.

[Pasquier, 2000] Pasquier, N. (2000). Datamining : Algorithmes d'extraction et de réduction des règles d'association dans les bases de données. Thèse de doctorat, Ecole Doctorale Sciences pour l'Ingénieur de Clermont Ferrand, Université Clermont Ferrand II, France.

[Pasquier *et al.*, 1998] Pasquier, N., Bastide, Y., Taouil, R. et Lakhal, L. (1998). Pruning closed itemset lattices for association rules. *In* Bouzeghoub, M., éditeur : *Proceedings of 14th Intl. Conference Bases de Données Avancées, Hammamet, Tunisia*, pages 177–196.

[Pasquier *et al.*, 1999] Pasquier, N., Bastide, Y., Touil, R. et Lakhal, L. (1999). Discovering frequent closed itemsets. *In* Beeri, C. et Buneman, P., éditeurs : *Proceedings of 7th International Conference on Database Theory (ICDT'99), LNCS, volume 1540, Springer-Verlag, Jerusalem, Israel*, pages 398–416.

[Pei *et al.*, 2000] Pei, J., Han, J. et Mao, R. (2000). Closet : An efficient algorithm for mining frequent closed itemsets. *In Proceedings of the 2000 ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery.*

[Plaza *et al.*, 1986] Plaza, E., Alsina, C., de Mantaras, R. L., Aguilar, J. et Agusti, J. (1986). Consensus and knowledge acquisition. *In Proceedings of the fifth International Conference on Information Processing and management of Uncertainty in Knowledge-Based Systems IPMU, Paris, France*, pages 294–306.

[Pollandt, 1996] Pollandt, S. (1996). *Fuzzy-Concepts. Formal Concept Analysis for imprecise data.* Edition Springer Verlag, Berlin.

[R. Srikant ant R. Agrawal,] R. Srikant ant R. Agrawal, title = Mining Sequential Patterns : Generalizations and Performance Improvements, b. . P. a. . A. y. . S.

[Raymond et Jiawei, 1994] Raymond, T. N. et Jiawei, H. (1994). Efficient and effective clustering methods for spatial data mining. *In Proceedings of the 20th International Conference on Very Large Data Bases*, VLDB '94, pages 144–155, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[Rioult *et al.*, 2003] Rioult, F., Robardet, C., Blachon, S., Crémilleux, B., Gandrillon, O. et Boulicaut, J. (2003). Mining concepts from large sage gene expression matrices. *In Proceedings of the Int. Conf. on Knowledge Discovery In Databases, Cavtat-Dubrovnik (Croatia).*, pages 107–118.

[S. Ayouni et Poncelet, 2010] S. Ayouni, S. Ben Yahia, A. L. et Poncelet, P. (2010). Fuzzy gradual patterns : What fuzzy modality for what result ? *In Proceedings of the International Conference on Soft Computing and Pattern Recognition (SoCPaR'10)*, Cergy, France.

[Salleb, 2003] Salleb, A. (2003). Recherche de motifs fréquents pour l'extraction de règles d'association et de caractérisation. Thèse de doctorat, Laboratoire d'Informatique Fondamentale d'Orléans LIFO, Université d'Orléans, France.

[Srikant et Agrawal, 1996] Srikant, R. et Agrawal, R. (1996). Mining quantitative association rules in large relational tables. *In Proceedings of the ACM-SIGMOD Int. Conf., Montreal, Canada*, pages 1–12.

[Stumme *et al.*, 2000] Stumme, G., Taouil, R., Bastide, Y., Pasquier, N. et Lakhal, L. (2000). Fast computation of concept lattices using data mining techniques. *In* Bouzeghoub, M., Klusch, M., Nutt, W. et Sattler, U., éditeurs : *Proceedings of 7th Intl. Workshop on Knowledge Representation Meets Databases (KRDB'00), Berlin, Germany*, pages 129–139.

[Stumme *et al.*, 2002] Stumme, G., Taouil, R., Bastide, Y., Pasquier, N. et Lakhal, L. (2002). Computing iceberg concept lattices with Titanic. *Journal on Knowledge and Data Engineering (KDE)*, 2(42):189–222.

[Takagi et Hayashi, 1991] Takagi, H. et Hayashi, I. (1991). Non-driven fuzzy reasoning. *International Journal of Approximate Reasoning.*

[Valtchev *et al.*, 2002] Valtchev, P., Missaoui, R., Godin, R. et Meridji, M. (2002). Generating frequent itemsets incrementally : two novel approaches based on Galois lattice theory. *Journal Expt. Theoretical Artificial Intelligence*, 14(1):115–142.

[Wang, 1988] Wang, E. (1988). Are grades of memebership probabilities ? *Fuzzy Sets and Systems*, 25:325–348.

[Wang, 1983] Wang, P. (1983). From the fuzzy statistics to the falling random subsets. *Wang P.P., Advances in Fuzzy Sets, Possibility Theory and Applications*, 9:81–96.

[Wille, 1982a] Wille, R. (1982a). Restructuring lattice theory : An approach based on hierarchies of concepts. pages 445–470. Reidel Edition.

[Wille, 1982b] Wille, R. (1982b). Restructuring lattices theory : An approach based on hierarchies of concepts. *I. Rival, editor, Ordered Sets, Dordrecht-Boston*, pages 445–470.

[Wille, 1989] Wille, R. (1989). Knowledge acquisition by methods of formal concept analysis. *In* Diday, E., éditeur : *Data analysis, learning symbolic and numeric knowledge*. Nova Science, New York.

[Wilson et Sutcliffe, 2007] Wilson, D. et Sutcliffe, G., éditeurs (2007). *Proceedings of the Twentieth International Florida Artificial Intelligence Research Society Conference, May 7-9, 2007, Key West, Florida, USA*. AAAI Press.

[Wolff, 1998] Wolff, K. E. (1998). Conceptual interpretation of fuzzy theory. *In Proceedings of the 6th European Congress on Intelligent techniques and soft computing, Aachen*, volume I, pages 555–562.

[Yamakawa et Furukawa, 1992] Yamakawa, T. et Furukawa, M. (1992). A design algorithm of membership functions for a fuzzy neuron using example-based learning. *In Proc. of the First IEEE Conference on Fuzzy Systems*, San Diego.

[Zadeh, 1965] Zadeh, L. (1965). Fuzzy sets. *Information and Control Journal*, 8:338–353.

[Zadeh, 1977] Zadeh, L. (1977). Fuzzy sets and their application to pattern classification and clustering analysis. *Academic Press, New York.*

[Zadeh, 1996] Zadeh, L. (1996). Fuzzy logic = computing with words. *IEEE Transaction on Fuzzy Systems*, 4.

[Zaki, 2000] Zaki, M. J. (2000). Generating non-redundant association rules. *In Proceedings of the 6th ACM-SIGKDD International Conference on Knowledge Discovery and Data Mining, Boston, Massachusetts, USA*, pages 34–43.

[Zaki, 2004] Zaki, M. J. (2004). Mining non-redundant association rules. *In Data Mining and Knowledge Discovery*, volume 9, pages 223–248.

[Zaki et Hsiao, 2002] Zaki, M. J. et Hsiao, C. J. (2002). Charm : An efficient algorithm for closed itemset mining. *In Proceedings of the 2nd SIAM International Conference on Data Mining, Arlington, Virginia, USA*, pages 34–43.

[Zaki *et al.*, 1997] Zaki, M. J., Pathasarathy, S., Ogihara, M. et Li, W. (1997). New algorithms for fast discovering association rules. *In Proceedings of the third international conference on Knowledge Discovery and Datamining*, pages 283–286.

[Zhang, 1993] Zhang, L. (1993). Strucrural and functional quantization of vagness. *Fuzzy Sets and Systems*, 55:51–60.

[Zhao et Bhowmick, 2003] Zhao, Q. et Bhowmick, S. S. (2003). Sequential pattern mining : A survey. *Rapport technique, Centre For Andvenced Information systems, Nanyang Technological University, Singapore.*

Résumé : L'Extraction de Connaissances dans les bases de Données est un processus qui vise à extraire un ensemble réduit de connaissances à fortes valeurs ajoutées à partir d'un grand volume de données. La fouille de données, l'une des étapes de ce processus, regroupe un certain nombre de taches, telles que : le clustering, la classification, l'extraction de règles d'associations, etc. La problématique d'extraction de règles d'association nécessite l'étape d'extraction de motifs fréquents. Nous distinguons plusieurs catégories de motifs : les motifs classiques, les motifs flous, les motifs graduels, les motifs séquentiels. Ces motifs diffèrent selon le type de données à partir desquelles l'extraction est faite et selon le type de corrélation qu'ils présentent. Les travaux de cette thèse s'inscrivent dans le contexte d'extraction de motifs graduels, flous et clos. En effet, nous définissons de nouveaux systèmes de clôture de la connexion de Galois relatifs, respectivement, aux motifs flous et graduels. Ainsi, nous proposons des algorithmes d'extraction d'un ensemble réduit pour les motifs graduels et les motifs flous. Nous proposons également deux approches d'extraction de motifs graduels flous, ceci en passant par la génération automatique des fonctions d'appartenance des attributs. En se basant sur les motifs flous clos et graduels clos, nous définissons des bases génériques de toutes les règles d'association graduelles et floues. Nous proposons également un système d'inférence complet et valide de toutes les règles à partir de ces bases.
Mots clés : Fouille de données, Ensembles flous, Règles graduelles, Analyse Formelle de Concepts.

Title : Survey and Extraction of fuzzy gradual rules : Definition of efficient algorithms.
Abstract : Knowledge Discovery in Databases is a process aiming at extracting a reduced set of valuable knowledge from a huge amount of data. Data mining, one step of this process, includes a number of tasks, such as clustering, classification, of association rules mining, etc. The problem of mining association rules requires the step of frequent patterns extraction. We distinguish several categories of frequent patterns : classical patterns, fuzzy patterns, gradual patterns, sequential patterns, etc. All these patterns differ on the type of the data from which the extraction is done and the type of the relationship that represent. In this thesis, we particularly contribute with the proposal of fuzzy and gradual patterns extraction method. Indeed, we define new systems of closure of the Galois connection for, respectively, fuzzy and gradual patterns. Thus, we propose algorithms for extracting a reduced set of fuzzy and gradual patterns. We also propose two approaches for automatically defining fuzzy modalities that allow obtaining relevant fuzzy gradual patterns. Based on fuzzy closed and gradual closed patterns, we define generic bases of fuzzy and gradual association rules. We thus propose a complet and valid inference system to derive all redundant fuzzy and gradual association rules.
Keywords : Datamining, Fuzzy sets, Gradual rules, Formal Concept Analysis.

Discipline : Informatique.

Laboratoires :

– Unité de Recherche en Programmation, Algorithmique et Heuristique, Faculté des Sciences de Tunis, Tunisie.
– Laboratoire d'Informatique de Robotique et de Micro-électronique de Montpellier
Université Montpellier II - CNRS (UMR 5506)
161 rue Ada - 34392 Montpellier cedex 5 - France