REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

Université MOULOUD MAMMERI de Tizi Ouzou

Faculté de génie électrique et d'Informatique

Département d'Informatique

Spécialité : Système Informatique

Pour l'obtention d'un diplôme de master en informatique

Thème intitulé :

Ordonnancement et classement (seuillage) d'un voisinage d'un treillis en analyse formelle de concepts flous

Encadré par : M^{r} Y. Djouadi

Présenté par : M^{elle} Abdous Nadia ep Nouraoui

02/10/2011

Remerciements

Mes premiers remerciements et ma grande gratitude s'expriment envers mon encadreur *Mr Yassine Djouadi*.

Ses conseils et ses encouragements ont permis à ce travail d'aboutir à la réussite. Ses capacités scientifiques et ses compétences étaient mon grand support.

La liberté qu'il m'a accordée et les responsabilités qu'il m'a confiées ont beaucoup contribué à ma formation et à mon autonomie de travail.

Ses critiques et suggestions ont été d'une précieuse aide pour la réalisation de ce projet.
Je le remercie chaleureusement pour sa pédagogie, sa patience, sa disponibilité et son dévouement.

Faire ce projet sous sa direction était pour moi un grand honneur et un immense bonheur.

J'aimerais adresser un remerciement particulier à Mademoiselle *Berkane Tessadite*, enseignante à l'université de Tizi Ouzou, pour son aide, sa gentillesse et son soutien tout au long de ces années.

J'exprime ma gratitude et ma reconnaissance à messieurs les membres de jury , ainsi je les remercie chaleureusement d'avoir pris le soin et le temps de bien juger ce travail.

J'adresse également mes sincères remerciements à ma famille, ma belle famille ainsi qu'à mes amis de m'avoir aidé à surmonter tous les obstacles et à me forger à travers les difficultés vécues durant toute cette période de travail.

A ceux qui ont été continuellement présents, qui m'ont épaulée par leur aide, soutien et encouragement.

Soyez assurés de mon profond respect.

Mlle Abdous Nadia

Dédicaces

Je dédicace ce modeste travail à :

- *Mes parents, ce travail vous doit beaucoup, qu'il soit pour vous le témoignage de mon infinie reconnaissance pour ces années de compréhension et d'effort commun.*
- *Mon mari Aghiles qui m'a soutenu et motivé dans mes études.*
- *Mon frère Hocine et ma sœur Fatiha qui m'ont toujours encouragé.*
- *Mon beau père, ma belle mère, mes belles sœur plus particulièrement à Dihia.*
- *Mes grands parents, ma tante Saliha, mes oncles, mes tantes ainsi que tous mes cousins et cousines.*
- *Ma très chère cousine et amie Lamia qui m'a beaucoup aidé*
- *Mes amis.*

Ce mémoire présente l'aboutissement du soutien et des encouragements qu'ils m'ont prodigués tout au long de ma scolarité, qu'ils en soient remercier par cette modeste dédicace.

Nadia

Table des matières

Chapitre I: Introduction générale

Introduction……1

Chapitre II: Analyse formelle de concepts

Introduction……4
II.1 Présentation intuitive de l`AFC……4
II.1.1 Concepts et contexte formel……4
II.1.2 Treillis de concepts formels……6
II.2 Fondements théoriques……9
II.2.1 Rappels Mathématiques……9
II.2.2 Théorie des treillis……10
II.2.3 Analyse de concepts formels……12
II.3 Domaines d'application de L'AFC……14
II.3.1 AFC et recherche d'information……15
II.3.2 L'extraction de règles d'associations……20
Conclusion ……26

Chapitre III : Théorie des ensembles flous

Introduction……27
III.1 Notions de base……27
II.1.1 Définitions usuelles……28
II.1.2 La cardinalité d'un ensemble flou……31
III.1.3 Ensembles Flous Discrets……31
III.2 Les opérateurs algébriques flous……32
III.2.1 Intersection et Union d'ensembles flous……33
III.2.2 Le complément……34
III.2. 3 Le produit cartésien……35
III.2.4 L'égalité……35
III.2.5 L'inclusion……35
III.2. 6 Relations floues et composition de relations floues……36

III.3 Les implications floues……37
II.3.1 S-implications……38
II.3.2 R-implication……39
III.4 Distances entre ensembles flous……41
III.4.1 Distance de Hamming……42
III.4.2 La distance Euclidienne……43
III.4.3 La distance Euclidienne pondérée……43
III.4.3 Distance de Hausdorff et variantes……43
Conclusion……45

Chapitre IV : Analyse formelle de concepts flous

Introduction……46
IV.1. contexte formel flou et concepts formels flous……47
IV.2 Rappels mathématiques……48
IV.3 Approches existantes……49
IV.3.1 Méthodes à seuil(s) ……49
IV.3.2 Approches “mono-face”(one Sided)……50
IV.3.3 Approches generals……52
IV.3.3.1 Approche de Burusco……52
IV.3.3.2 Approche Belohlavek……54
IV.3.3.3 Approche de Georgesco et Popescu……56
Conclusion……56

Chapitre V : Contribution

Introduction à la contribution……57
V.1 fondements théoriques……57
V.2 présentation de la contribution……59
V.2.1 Construction de l’ensemble des concepts formels……59
V.2.2 Ordonnancement linéaire du treillis des CFs……64
V.4 Présentation de l’application……66
V.4.1 Environnement technique de développement……66

V.4.2 Technique d'implémentation……..69
Conclusion……..71

Chapitre VI : Conclusion et Perspective

Conclusion et perspective……70

Introduction

L'analyse formelle de concepts est un domaine à la frontière entre la représentation et la découverte de connaissances, l'intelligence artificielle et enfin la théorie mathématique des treillis [Birkhoff, 1949]. La théorie de l'analyse formelle de concepts (abréviation AFC) a été introduite par Wille en 1982 [Wille, 1982]. La consolidation mathématique de cette théorie a ensuite été proposée par Ganter en 1999 [Ganter, 1999]. A partir de cette consolidation, cette théorie a été largement utilisée dans divers domaines : psychologie, sociologie, médecine, biologie, linguistique, informatique, etc…. [Wolff, 1993]

L'analyse formelle de concepts permet de découvrir des structure hiérarchisées (treillis) de concepts formels à partir de relations binaires appelées contextes formels. De manière plus précise, un contexte formel consiste en une relation *R* entre un ensemble d'objets et un ensemble de propriétés. Cette relation est souvent représentée sous la forme d'une table, en lignes les objets et en colonnes les propriétés. Une cellule (intersection d'une ligne et d'une colonne) de cette table indique si l'objet correspondant a la lige vérifie ou ne vérifie pas la propriété correspondant a la colonne. Un concept formel consiste en une paire $\langle Objets, Proprietés \rangle$où les ensembles objets et propriétés se décrivent mutuellement. L'ensemble de tous les concepts formels est naturellement équipé d'une relation d'ordre. Ganter a prouvé que l'ensemble de tous les concepts formels formait un treillis complet [Ganter, 1999].

Dans la proposition originale [Ganter, 1999], [Wille, 1982] les auteurs considèrent des relations binaires et Booléennes sur la base des suppositions suivantes : i) soit l'objet vérifie la propriété soit il ne la vérifie pas, ii) il est toujours connu si l'objet vérifie ou ne vérifie pas la propriété correspondante. De ce fait, l'AFC initialement proposée considère uniquement des relations booléennes mais aussi complètement renseignées. Il s'avère que dans la réalité certaines relations entre objets et propriétés peuvent être partiellement connues, incertaine, imprécises, vagues, floues ou carrément inconnues. Par exemple en recherche d'information une relation d'incidence *Terme* × *Document* peut contenir des valeurs floues.

Pour prendre en considération des contextes formels contenant des valeurs floues, l'AFC a été généralisée au delà du cadre booléen et a donné naissance à l'analyse formelle de

concepts floue. De nombreux travaux de recherche ont proposé des méthodes de découverte des concepts formels flous. De la même manière que dans le cas classique (Booléen), l'ensemble des concepts formels flous forme un treillis complet. Il est à noté toutefois que certaines conditions algébriques doivent être remplies pour que l'on puisse obtenir un treillis complet.

Une structure de treillis est quelque peu délicate à manipuler et à interpréter du fait qu'une telle structure n'offre qu'un ordre partiel. Aussi, il n'est pas possible d'ordonnancer les éléments du treillis selon un ordre total. En recherche d'information par exemple où les treillis de concepts formels sont souvent utilisés il n'est pas possible d'ordonnancer (ranking) les réponses d'une requête.

Ce travail apporte une contribution théorique originale pour obtenir une structure partielle du treillis de concepts formels sous forme linéaire c.-à-d. équipé d'un ordre total. Nous proposons cela en considérant la notion de concept formel cible à partir duquel un ordre linéaire sera établie sur tout le voisinage du concept formel cible. Cet ordre linéaire sera obtenu par la projection de la distance de Hausdorff dans un espace métrique. Les applications de notre proposition théorique peuvent être multiples. En recherche d'information basée sur l'AFC, notre méthode permet l'ordonnancement des résultats d'une requête. Un autre type d'application consisterai en la réduction de la complexité de navigation en effet, il est beaucoup plus simple de parcourir une structure linéaire qu'une structure sous forme de treillis.

Pour cela notre mémoire est organisé comme suit :
Le premier chapitre consiste en une introduction générale.
Le deuxième chapitre porte sur l'analyse de concepts formels. Dans ce chapitre nous commençons par une représentation intuitive de l'AFC, Nous donnons par la suit une représentation basée sur des fondements mathématiques ainsi que des notions de fermeture de connexion de Galois. Et en fin nous donnons quelques domaines d'application de l'AFC.
Dans le troisième chapitre, nous présentons la théorie des ensembles flous introduite par Lotfi Zadeh en 1965. On y détaille plus particulièrement les notions de base nécessaire à la compréhension de cette théorie. Nous commençons par donner une définition détaillée des ensembles flous par la suite nous donnons les opérateurs ensembliste flous lesquels dépendent de la définition sous-jacente d'une norme et d'une conorme triangulaire.

Dans ce chapitre nous avons particulièrement mis l'accent sur les implications flous notamment les R-implication et enfin nous donnons quelques distances appliquées aux ensembles flous.

Le quatrième chapitre porte sur l'analyse formelle de concepts flous. Dans cette partie nous donnons les notions de base d'une AFC floue qui repose sur une connexion de Galois flou en présentant les différentes approches existantes.

En fin, le dernier chapitre qui est l'occasion de présenter notre contribution.

Chapitre II: Analyse formelle de concepts

Sommaire

Introduction 4

II.1 Présentation intuitive de l`AFC 4

II.1.1 Concepts et contexte formel 4

II.1.2 Treillis de concepts formels 6

II.2 Fondements théoriques 9

II.2.1 Rappels Mathématiques 9

II.2.2 Théorie des treillis 10

II.2.3 Analyse de concepts formels 12

II.3 Domaines d'application de L'AFC 14

II.3.1 AFC et recherche d'information 15

II.3.2 L'extraction de règles d'associations 20

Conclusion 26

Introduction

Analyse Formelle de Concepts appelé aussi L`Analyse de Concepts Formels (AFC) est un domaine de recherche en pleine expansion qui offre un cadre formel pour développer des modèles d`analyse de données et de représentation de connaissances. Les structures conceptuelles de l`AFC, tel que les treillis de concepts, constituent de bons outils de classification et de support de découverte de ressources, de fouille de données, de recherche d'information etc.…

Au début du présent chapitre nous donnons un aperçu de l'AFC, suivront après des notions mathématiques sous-jacentes à la notion de l'AFC ainsi que ses composantes, et enfin nous présenterons quelques domaines d'application de cette théorie.

II.1 Présentation intuitive de l`AFC

La théorie de l'AFC est un domaine des mathématiques qui consiste à restructurer la théorie des treillis [Birkhoff, 1967] afin de faciliter son utilisation dans des applications du monde réel. L'objectif à travers la mise en place de l'AFC est « d'*atteindre une théorie structurée qui expose les pensées formelles selon des interprétations significatives et permettre ainsi des communications et des discussions critiques de leurs contenus* » [WILLE, 1996]. Pour cela L`AFC a été centrée autour de la notion de ***concepts formels***, lesquels sont extraits a partir de contextes restreints [DAU et KLINGER, 2005]. Ces contextes appelés «***contextes formels***» constituent le point de départ de l`AFC.

II.1.1 Concepts et contexte formel

D'un point de vue philosophique, un concept est considéré comme l'unité de base de la pensées humaine. Dans la théorie de l'AFC, un concept est une représentation générale et abstraite d`une certaine réalité [WILLE, 1982], [GANTER et WILLE, 1999]. Il est alors appelé concept formel et il peut être défini comme un groupement d`individus (objets) et de leurs propriétés (attributs) communes. Il est formé d'un ensemble maximal d'objets appelés ***extension*** et d'un ensemble maximal d'attributs appelés ***intension,*** de telle sorte que tous les objets de l'extension partagent tous les attributs de l'intension et vice-versa.

Ces concepts sont extraits d'une table de relation binaire entres les objets et les attributs (propriétés). Cette table est appelée un ***contexte formel***.

Un contexte formel peut être représenté sous la forme d'un tableau où les lignes correspondent aux objets et les colonnes correspondent aux attributs. Les cases (cellules) du tableau sont remplies comme suit : si le ième objet est en relation avec le jème attribut alors la case intersection de la ligne i et de la colonne j contient le symbole " ×" sinon la case est vide. Le tableau 1 [Wolff, 1993] ci-dessous donne un bon exemple de contexte formel représentant certains animaux et certaines de leurs propriétés. Les animaux retenus comme objets dans cet exemple sont : ***lion, moineau, aigle, lapin, autruche***. *Les attributs* retenus sont : prédateur, vole, oiseau, mammifère.

ANIMAUX	prédateur	Vole	Oiseau	mammifère
Lion	×			×
Moineau		×	×	
Aigle	×	×	×	
Lapin				×
Autruche			×	

Tableau1 : contexte formel ANIMAUX

Pour mieux comprendre la notion de ***concept formel,*** considérons chaque objet O_i parmi les objets retenus, et regardons dans la table tous les objets autre que O_i qui partagent les mêmes attributs que ceux de O_i. Appelons A l'ensemble des objets résultant y compris O_i et, B l'ensemble des attributs partagés. Le couple $\langle A, B \rangle$ est appelé ***concept formel*** où A est appelé l'***extension*** et B est appelé l'***intension*** du concept formel $\langle A, B \rangle$.

Par exemple si on considère O_i = ***moineau***, seul l'objet **aigle** partage les mêmes attributs que O_i . Ainsi A et B correspondent respectivement a l'ensemble {**moineau, aigle**} et l'ensemble

{vole, oiseau}. La paire ({**moineau, aigle**}, {vole, oiseau}) forme ***un concept formel.*** L'ensemble {moineau, aigle} (resp. {Vole, oiseau}) est appelé extension (resp. intension). Selon ce principe, nous dégageons huit concepts formels à partir du contexte formel présenté dans le tableau 1. L'ensemble de ces concepts formels peuvent être organisés hiérarchiquement en une structure appelée ***treillis de concepts formel*** (cf. fig.1).

II.1.2 Treillis de concepts formels

Un treillis de concepts formels correspond à un diagramme dans lequel les nœuds correspondent aux concepts formels et les liens à des relations d'ordre entre ces concepts formels. Cet ordre se fait selon deux critères duaux liés aux extensions et aux intensions des concepts, à savoir les concepts les plus généraux sont situés en haut du treillis alors que les concepts les plus spécifiques sont situés en bas du treillis.

Le diagramme ci-dessous correspond au ***contexte formel*** donné dans la section précédente.

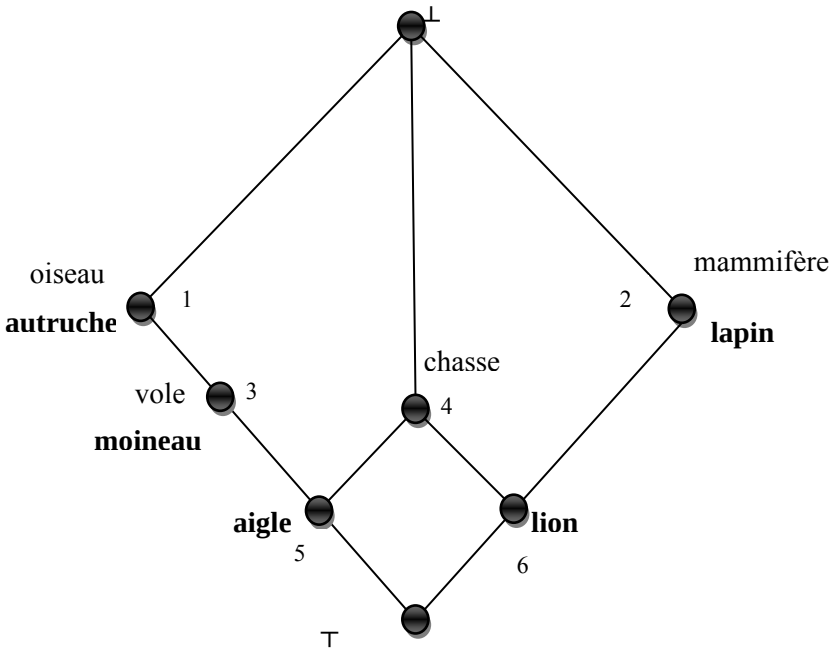


Fig 1: treillis de concepts formels

Sur ce diagramme figure les huit concepts identifiés précédemment numérotés et deux autres concepts particuliers notés $\top et \perp$.

Ces concepts formels sont organisés de manière à ce qu'un parcours ascendant des concepts formels du treillis se traduit à chaque étape par la diminution progressive du nombre d'attributs dans les intensions des concepts formels et l'augmentation progressive du nombre d'objets dans leurs extensions. De façon duale, un parcours descendant se traduit à chaque étape par l'augmentation progressive du nombre d'attributs dans les intensions des concepts et la diminution progressive du nombre d'objets dans leurs extensions.

On dira alors, qu'un concept $\langle A_1, B_1 \rangle$ est inférieur au concept $\langle A_2, B_2 \rangle$ si et seulement si A_1 est inclus dans A_2 et B_2 est inclus dans B_1, on dira alors que $\langle A_1, B_1 \rangle$ est un ***sous-concept*** de $\langle A_2, B_2 \rangle$ et que $\langle A_2, B_2 \rangle$ est un ***super-concept*** de $\langle A_1, B_1 \rangle$ [J. VILLERD, 2008].

L'extension (resp. intension) d'un concept formel est composée de tous les objets (resp. attributs) qui étiquettent ce concept formel et ses sous-concepts (resp. super-concepts) [M. KAYTOUE et A. NAPOLI, 2009]. Par exemple en appliquant cette règle au concept 3 (cf. figure 1) on trouvera que son extension est égale à {**moineau**, **aigle**} et son intension est égale à {vole, oiseau}.

Certains concepts formels sont incomparables [J. VILLERD, 2008] c'est-à-dire que la notion de sous-concept/super-concept n'est pas réalisable, ainsi aucune relation d'inclusion n'est identifiable entre les intensions et extensions de ces concepts formels, a l`exemple des concepts 3 et 4. L'ordre partiel ainsi introduit dans ce cas, est complété par un concept maximal $\top$ (top) et un concept minimal $\perp$ (bottom) afin de former un treillis

Le concept formel $\top$ est défini comme contenant tous les objets en extension et le concept formel $\perp$ comme contenant tous les attributs en intension. L'intension de $\top$ contient donc les attributs communs à tous les objets et l'extension de $\perp$ contient les objets possédant tous les attributs [J. VILLERD, 2008]. Dans le diagramme donné, l'intension de $\top$ et l'extension de $\perp$ sont toutes deux des ensembles vides.

Afin d`éviter cette situation nous rajoutons un attribut à notre contexte formel, soit l`attribut animal. Les cases de la colonne résultante seront toutes marquées par «×», par conséquent l'intention du top concept sera égale a {animal}. Le treillis correspondant à cette deuxième réalité est donne dans la figure 2.

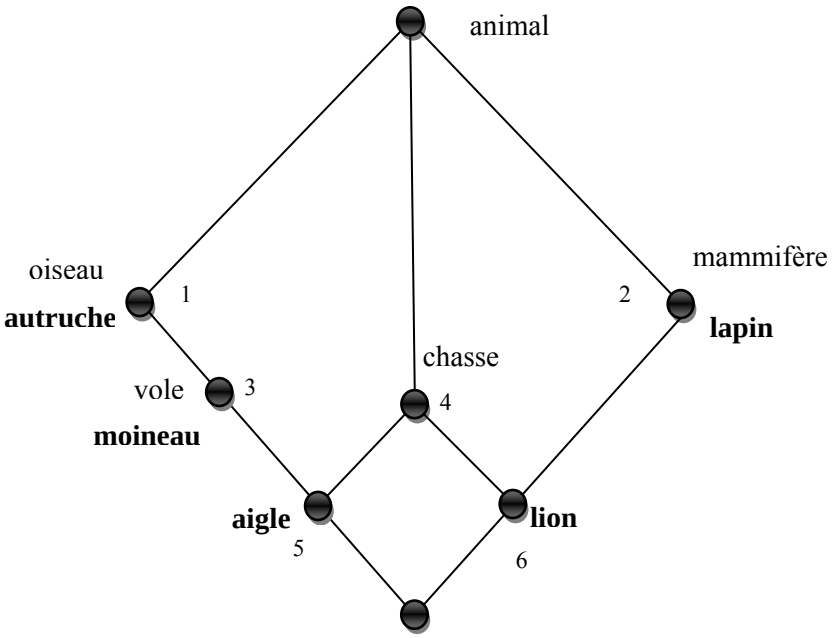


Fig 2: treillis de concepts formels après rajout d'un attribut

A partir de ce treillis on pourrait comprendre que « chaque animal qui vole est un oiseau ». Pour démontrer le contraire, ajoutant à notre contexte l'objet « **abeille »** qui a comme attribut « vole ». Le treillis résultant [Wollf, 1993] est représenté dans la figure 3 :

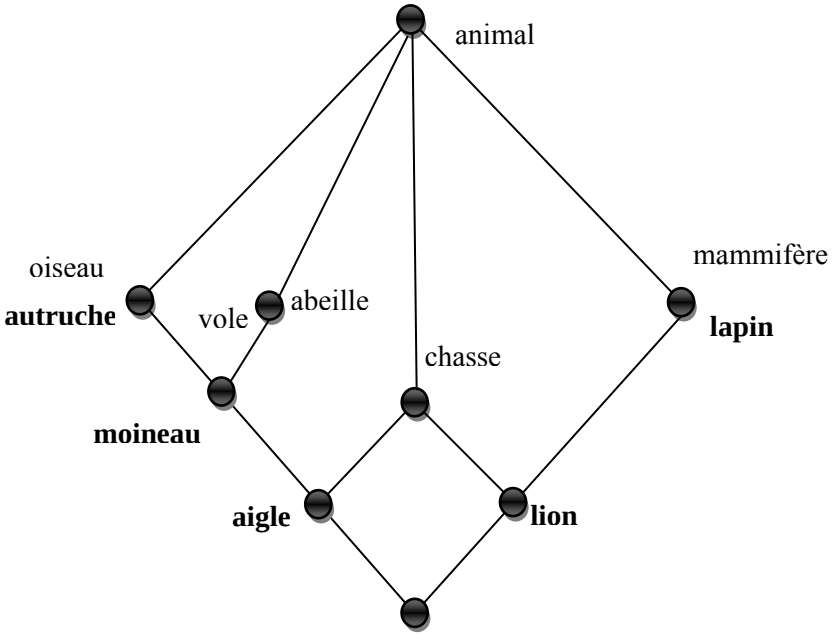


Fig 3: treillis de concepts formels après rajout d'un objet

II.2 Fondements théoriques

La théorie de l'AFC nécessite la connaissance à priori de certaines notions mathématiques sous-jacentes. A savoir, les notions de relations d'ordre, d'ensembles ordonnés, de connexions de Galois et d'operateurs de fermeture.

La section Ci-dessous donne les définitions de ces notions, lesquelles sont extraites des thèses de référence [MESSAI, 2009] et [PASQUIER, 2000].

II.2.1 Rappels Mathématiques

Relation binaire : Une relation binaire $\boldsymbol{R}$ entre deux ensembles $\boldsymbol{M}$ et $\boldsymbol{N}$ est un ensemble de couples d'éléments (m , n) tels que $m \in \boldsymbol{M}$ et $n \in \boldsymbol{N}$, i.e. $\boldsymbol{R}$ est un sous ensemble du produit cartésien $\boldsymbol{M} \times \boldsymbol{N}$.

(m , n) $\in \boldsymbol{R}$ (aussi noté par $m\boldsymbol{R}n$) signifie que l'élément m est en relation $\boldsymbol{R}$ avec l'élément n.

Relation d'ordre partiel : Une relation binaire ***R***, souvent notée ≤, sur un ensemble *E* est dite relation d'ordre partiel (ou simplement relation d'ordre) sur *E* si elle vérifie les conditions suivantes pour tous *x, y, z* ∈ *E*:

1. $x \leq x$ (*Réflexivité*)
2. *si* $x \leq y$ *et* $y \leq x$ *alors* $x = y$ (*Antisymétrie*)
3. *: si* $x \leq y$ *et* $y \leq z \in R$ *alors* $x \leq z$ (*Transitivité*)

Ensemble ordonné *:* Un ensemble partiellement ordonné (ou simplement ensemble ordonné) est un couple (*E* , ≤) où E est un ensemble et " ≤ " est une relation d'ordre sur *E.*

Relation de couverture* (*successeur, prédécesseur*) :** Soit (*E* , ≤) un ensemble ordonné et *x, y* deux éléments de *E*. La relation de couverture entre les éléments de *E*, notée ≺, est définie par : $x \prec y$ si et seulement si $x \leq y$ et 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 dirons que *y couvre x* ou que *y* est un *successeur immédiat de x.* ***La couverture de *x* est formée par tous ses successeurs

Tout ensemble ordonné (*E* , ≤), peut être représenté graphiquement par un diagramme appelé "diagramme de Hasse'' (ou diagramme de couverture), et il est obtenu comme suit :

1. Tout élément de *E* est représenté par un petit cercle dans le plan
2. Si *x, y* ∈ *E* et $x \prec y$ alors le cercle correspondant à *y* doit être au-dessus de celui correspondant à *x* et les deux cercles sont reliés par un segment.

À partir d'un tel diagramme on peut lire la relation d'ordre comme suit : $x \leq y$ si et seulement s'il existe un chemin ascendant qui relie le cercle correspondant à *x* au cercle correspondant à *y*.

II.2.2 Théorie des treillis

Join et meet : Soient (*E* , ≤) un ensemble ordonné et *S* un sous ensemble de *E.* Un élément $u \in E$ est un majorant de *S* si pour tout élément $s \in S$ nous avons $s \leq u$. L'ensemble des majorants de *S* est noté UB(*S*) (*upper-bound*). De façon duale, un élément $v \in E$ est un minorant de *S* si pour tout élément $s \in S$ nous avons $v \leq s$. L'ensemble des minorant de *S* est noté LB(*S*) (*Lowe-bound*).

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

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*)** (ou $\bigvee S$). Dualement, le plus grand des minorants 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*)** (ou $\bigwedge S$).

$$\textbf{\textit{Join}}(S) = \text{Min}\,\{UB(S)\} = \bigwedge UB(S)$$

$$\textbf{\textit{Meet}}(S) = \text{Max}\,\{LB(S)\} = \bigvee LB(S)$$

Dans le cas ou $S = \{x, y\}$, $\bigvee S$ ***et*** $\bigwedge S$ *sont aussi notés par* $x \vee y$ *et* $x \wedge y$ *respectivement.*

Treillis et treillis complet : Un treillis est un ensemble partiellement ordonné $(E, \leq)$ tel que ***Join*** $(\{x, y\})$ et ***Meet*** $(\{x, y\})$ existent pou tout couple d'éléments $x, y \in E$. Un treillis est dit complet si ***Join*(*S*)** et ***Meet*(*S*)** existent pour tout sous ensemble *S* de *E*. En particulier, un treillis complet admet un élément maximal (top) noté par $\top$ et un élément minimal (Bottom) noté par $\bot$.

Fermeture : On appelle opérateur de fermeture sur un ensemble ordonné $(E, \leq)$, toute application $\Phi : E \rightarrow E$ qui vérifie les propriétés suivantes pour tout $x, y \in E$:

1. $x \leq \Phi(x)$ (Φ *est extensive*)
2. *Si* $x \leq y$ *alors* $\Phi(x) \leq \Phi(y)$ (Φ *est monotone croissante*)
3. $\Phi(x) = \Phi(\Phi(x))$ (Φ *est idempotente*)

Un élément $x \in E$ est dit ***fermé*** *pour* Φ *si et seulement si* $x = \Phi(x)$.

Système de fermeture : Un système de fermeture (dit aussi système de fermés) sur un ensemble E est un ensemble de parties de *E* contenant *E* et fermé pour l'intersection.

Formellement, $\mathfrak{C}$ est un système de fermeture si les conditions suivantes sont vérifiées :

- $\emptyset \in E$
- $E \in \mathfrak{C}$
- $\mathfrak{A} \subset \mathfrak{C}$ alors $\bigcap \mathfrak{A} \in E$ (*l'intersection de toute suite de fermés de* $\mathfrak{A}$ *appartient à* $\mathfrak{C}$)

La notion de fermeture est liée à la définition des treillis. En effet, l'ensemble des fermés dans un ensemble E étant donné un opérateur de fermeture sur E forme un treillis complet [Birkhoff]. Inversement, tout treillis complet est isomorphe au treillis des fermés d'un opérateur de fermeture et on a le théorème suivant.

Théorème : *Si* $\mathfrak{C}$ *est un système fermé alors* $(\mathfrak{C},\subseteq)$ est un treillis complet avec $\bigwedge S = \bigcap S$ et $\bigvee S = \Phi_{\mathfrak{C}}(\bigcup S)\ \forall\ S \subset \mathfrak{C}$. Inversement, tout treillis complet est isomorphe au treillis des fermés d'un opérateur de fermeture. [Messai,2009]

Connexion de Galois : Soit $\varphi : X \to Y$ et $\psi : Y \to X$ deux applications entre deux ensembles ordonnés $(X, \leq_X)$ et $(Y, \leq_Y)$. Φ et ψ forment une ***connexion de Galois*** entre $(X, \leq_X)$ et $(Y, \leq_Y)$ si elles vérifient les conditions suivantes pour tous $x, x_1, x_2 \in X$ et $y, y_1, y_2 \in Y$:

- *Si* $x_1 \leq_X x_2$ *alors* $\varphi(x_2) \leq_Y \varphi(x_1)$
- *Si* $y_1 \leq_Y y_2$ *alors* $\psi(y_2) \leq_X \psi(y_1)$
- $x \leq_X \psi(\varphi(x))$ *et* $y \leq_y \varphi(\psi(y))$

II.2.3 Analyse de concepts formels

Un contexte formel est un triplet $K = (O, P, \boldsymbol{R})$ où O est un ensemble d'objets, P est un ensemble d'attributs et $\boldsymbol{R}$ est une relation binaire entre O et P appelée relation d'incidence de K et vérifiant $\boldsymbol{R} \subseteq O \times P$. Un couple $(x, y) \in \boldsymbol{R}$ (noté aussi $x\boldsymbol{R}y$) signifie que l'objet $x \in O$ possède l'attribut $y \in P$.

Dans la proposition initiale de Wille, cette relation est Booléenne (i.e. un objet possède totalement la propriété ou ne la possède pas du tout) et, est complètement renseignée (i.e. il est toujours connu si un objet possède ou non une propriété) [Djouadi et al, 2009].

Opérateur de dérivation de Galois

Soit $K = (O, P, \boldsymbol{R})$ *un contexte formel*, étant donné un objet $x \in O$, et une propriété $y \in P$, on définit l'ensemble des propriétés possédées par l'objet x *par* $\boldsymbol{R}(x) = \{y \in P \mid x\boldsymbol{R}y\}$ ($x\boldsymbol{R}y$ signifie que x possède la propriété y), et l'ensemble des objets possédant la propriété y par $\boldsymbol{R}(y) = \{x \in O \mid x\boldsymbol{R}y\}$.

En AFC, on définit des correspondances entre les ensembles 2^O et 2^P. Ces correspondances sont appelés ***opérateur de dérivation de Galois.*** L'opérateur de Galois le plus couramment utilisé en AFC est quelquefois appelé *opérateur de suffisance* [Djouadi et al, 2009].

Etant donné $X \subseteq O$ et $Y \subseteq P$, l'opérateur de suffisance, noté $(.)^{\Delta}$, permet d'exprimer l'ensemble des propriétés satisfaites par tous les objets de X :

$$X^{\Delta} = \{y \in P \mid \forall x \in O(x \in X \rightarrow x\boldsymbol{R}y)\}$$
$$= \{y \in P \mid X \subseteq \boldsymbol{R}(x)\}$$
$$= \cap_{x \in X} \boldsymbol{R}(x)$$

Et l'ensemble des objets satisfaisant toutes les propriétés de Y :

$$Y^{\Delta} = \{x \in O \mid \forall y \in P(y)\ (y \in Y \rightarrow x\boldsymbol{R}y)\}$$
$$= \{x \in O \mid Y \subseteq \boldsymbol{R}(x)\}$$
$$= \cap_{y \in Y} \boldsymbol{R}(y)$$

Connexion de Galois dans un contexte formel

Soit $K = (O, P, \boldsymbol{R})$ un contexte formel, et soit $(.)^{\Delta}: 2^O \rightarrow 2^P$, $(.)^{\Delta}: 2^P \rightarrow 2^O$ deux *operateurs de suffisance*, la composition asymétrique de ces operateurs produit deux ensembles. L'opérateur $((.)^{\Delta})^{\Delta} : 2^O \rightarrow 2^O$ permet d'associer à un ensemble d'objets X l'ensemble maximal d'objets dans O ayant les attributs communs aux objets de X. Cet ensemble est noté par $((X)^{\Delta})^{\Delta}$. De façon duale, le second opérateur $((.)^{\Delta})^{\Delta} : 2^P \rightarrow 2^P$ permet d'associer à un ensemble d'attributs Y l'ensemble maximal d'attributs dans P communs aux objets ayant les attributs dans Y. Cet ensemble est noté par $((Y)^{\Delta})^{\Delta}$.

Ces deux opérateurs définissent deux fermetures respectivement sur l'ensemble des parties de O, 2^O, et sur l'ensemble des parties de P, 2^P. Les ensembles $((X)^{\Delta})^{\Delta}$ et $((Y)^{\Delta})^{\Delta}$ sont des fermés pour ces deux opérateurs respectifs. L'ensemble des fermés de 2^O (resp. 2^P) muni de l'inclusion ensembliste est un treillis complet.

Les opérateurs de dérivation $(.)^{\Delta}: 2^O \rightarrow 2^P$ et $.^{\Delta}: 2^P \rightarrow 2^O$ forment une bijection[1] entre les ensembles de fermés de 2^O et 2^P et définissent un isomorphisme entre les deux treillis respectifs : à chaque fermé X dans 2^O correspond un unique fermé Y dans 2^P et vice versa. De

[1] une **bijection** est une application *bijective*. Une application est **bijective** si et seulement si tout élément de son *ensemble d'arrivé* a un et un seul *antécédent*, c'est-à-dire est *image* d'exactement un élément de son ensemble de départ, ou encore si elle est *injective* et *surjective*

cette façon, les opérateurs de dérivation $((.)^{\Delta})^{\Delta}\ 2^{O} \rightarrow 2^{O}$ et ${}_{\Delta}{}^{\Delta}: 2^{P} \rightarrow 2^{P}$ forment une ***connexion de Galois*** entre $(2^{O}, \subseteq)$ et $(2^{P}, \subseteq)$.

Concept formel : *Soit* $K = (O, P, \boldsymbol{R})$ un contexte formel. Un concept formel est un couple $\langle A, B \rangle$ tel que $A \in O$, $B \in P$ et $A^{\Delta} = B$ et $B^{\Delta} = A$. A et B sont respectivement appelés extension et intension du concept formel $\langle A, B \rangle$L'ensemble des concepts formels associés au contexte formel $K = (O, P, \boldsymbol{R})$ est noté par $\mathfrak{B}(K)$.

Les concepts de $\mathfrak{B}(K)$ sont ordonnés par une relation de d'ordre hiérarchique entre concepts notée par " $\leq$ " et définie comme suit.

Ordre sur les concepts :

Soient $\langle A_1, B_1 \rangle$ et $\langle A_2, B_2 \rangle$ deux concepts formels de $\mathfrak{B}(K)$. $\langle A_1, B_1 \rangle \leq \langle A_2, B_2 \rangle$ si et seulement si $A_1 \subseteq A_2$ (ou de façon duale $B_2 \subseteq B_1$). $\langle A_2, B_2 \rangle$ est dit ***super-concept*** de $\langle A_1, B_1 \rangle$ et $\langle A_1, B_1 \rangle$ est dit ***sous-concept*** de $\langle A_2, B_2 \rangle$. La relation " $\leq$ " est dite relation de ***subsomption***.

Treillis de concepts : La relation " $\leq$ " permet d'organiser les concepts formels en un treillis complet $(\mathfrak{B}(K), \leq)$ appelé treillis de concepts formels ou encore treillis de Galois et noté par $L(K)$. Les operateurs Meet et Join dans $\mathfrak{B}(K)$ sont décrits par le théorème fondamental de Ganter et Wille [Ganter, 1999] :

$$\bigwedge(X_i, Y_i) = (\cap X_i, ((\cup Y_i)^{\Delta})^{\Delta})$$
$$\bigvee(X_i, Y_i) = (((\cup X_i)^{\Delta})^{\Delta}, (\cap Y_i))$$

II.3 Domaines d'application de L'AFC

L'ACF s'appuie sur un formalisme mathématique rigoureux, la théorie des treillis, et l'adapte afin de permettre d'analyser des données, sous la forme d'un contexte formel, et de les structurer en un treillis de concepts. La structure de treillis facilite la visualisation des relations sous-jacentes entre les données et l'interprétation de ses relations. Le fondement mathématique de l'ACF et les structures conceptuelles qu'elle permet de dériver [Godin et al., 1995b] ont été exploités dans plusieurs domaines d'analyse et d'exploitation de données tels

que la classification, la recherche d'information [Carpineto and Romano, 2005], la sélection de services Web [Azmeh et al., 2008], la construction d'ontologies [Bendaoud et al., 2008], l'extraction de connaissances [Lakhal and Stumme, 2005], l'apprentissage machine [Kuznetsov, 2005], l'ingénierie des logiciels [Tilley et al., 2005, Godin and Valtchev, 2005], la linguistique [Priss, 2005], etc. Les principales approches développées dans le cadre de ces domaines sont décrites en détails dans [Ganter et al., 2005]. Dans la suite de cette section nous nous intéressons plus particulièrement aux applications de l'ACF à la recherche d'information ainsi qu'aux règles d'association.

II.3.1 AFC et recherche d'information

L'utilisation des treillis de concepts formels pour la recherche d`information est motivée par l'étroites analogie entre une relation binaire (i.e. contexte formel) {Objets}×{Proprietes}et une matrice d'incidence Documents×Termes.

La recherche d'information (RI) a été l'une des premières applications phare des treillis de concepts à la découverte de ressources. Les premiers travaux [Carpineto and Romano, 1993], [Godin et al, 1993], [Godin et al, 1995] ont étudié la possibilité d'utiliser les treillis de concepts comme support pour la recherche documentaire. Des collections de documents sont alors représentées sous la forme de contextes formels. Les objets du contexte formel sont des documents et les attributs sont les termes d'indexation de ces documents. Chaque concept formel du treillis correspondant est vu comme un couple formé par une requête, dont les mots clés sont les termes contenus dans l'intension du concept, et l'ensemble de documents pertinents pour cette requête sont les documents contenus dans l'extension du concept. Le critère de pertinence dans ce cas étant celui considéré dans le cas de la recherche booléenne (conjonctive) à savoir la vérification de tous les critères spécifiés dans la requête. Ceci justifie l'interprétation donnée aux concepts formels du treillis puisque les objets dans l'extension d'un concept formel partagent tous les attributs dans son intension.
Le calcul de la réponse à une requête donnée revient à identifier, dans le treillis, le concept formel dont l'intension est identique à la requête. Les liens de spécialisation/généralisation entre les concepts permettent d'effectuer une recherche progressive dans le treillis. Cette façon de procéder suppose que la requête existe déjà dans le treillis. Pour assurer cette condition, des algorithmes de construction incrémentale de treillis de concepts sont utilisés pour l'insertion des requêtes dans un treillis déjà construit. De cette manière, un premier mode

de recherche par treillis a été défini : ***la recherche par interrogation***. La structure hiérarchique des treillis de concepts permet la définition d'un deuxième mode de recherche : ***la recherche par navigation***. Ces deux modes sont détaillés dans les sections suivantes.

Interrogation

Ce mode de recherche est facilité par la mise en place d'algorithmes performants pour la construction incrémentale des treillis de concepts. La définition de requêtes consiste à spécifier directement les termes d'indexation qui décrivent le(s) document(s) à trouver. La requête est ensuite insérée dans le treillis. La recherche des documents pertinents revient à localiser le concept formel le plus général incorporant les termes spécifiés dans la requête.

Navigation

Ce mode de recherche exploite la structure hiérarchique des treillis. Il consiste à explorer librement les concepts en s'appuyant sur la visualisation du treillis par son diagramme de Hasse. Cette forme d'interaction tire profit d'une caractéristique importante de la cognition humaine : "il est plus facile de reconnaître quelque chose d'intéressant que de le décrire". Le diagramme de Hasse est utilisé comme structure de base pour la recherche. Il offre une interface de navigation permettant de suivre les liens de spécialisation/généralisation entre concepts pour spécialiser ou élargir graduellement l'espace de recherche. Dans le cas général, le scenario de recherche peut être présenté comme suit : partant du concept le plus général du treillis qui représente la classe de tous les documents avec un ensemble de termes communs souvent vide, on effectue une spécialisation graduelle en suivant les liens descendants dans le treillis. Chaque pas dans le treillis est équivalent à l'ajout d'un nombre minimal de nouveaux termes qui spécifient la description des documents à trouver. Ceci entraîne la restriction de l'espace de recherche à un sous ensemble de documents. Cette opération est répétée jusqu'à l'identification du sous ensemble minimal de documents recherchés. Dans d'autres cas, partant d'une description très précise de documents, on peut relaxer progressivement cette description en suivant les liens ascendants du treillis. Chaque étape correspond à la suppression d'un ensemble minimal de termes ce qui entraîne l'augmentation du nombre de documents qui satisfont la description allégée.

De manière générale, on peut effectuer une navigation libre dans le treillis en suivant les liens descendants et/ou ascendants en fonction du besoin en précision lors du passage d'un concept à l'autre dans le treillis.

La combinaison des deux modes

L'utilisation des treillis de concepts permet de définir deux modes de recherche, par interrogation et par navigation, qui peuvent être combinés [Cole et al., 2003],[Ducrou, 2007].

Le treillis de concepts sert d'espace de recherche commun et assure la cohérence des résultats de recherche des deux modes. De plus, le passage d'un mode à l'autre peut se faire à tout moment pendant la recherche. La recherche se ramène ainsi à effectuer une combinaison libre de spécification directe de requête, résultant en un saut dans le concept le plus général incorporant les termes de la requête, et la navigation libre en suivant les liens entre les concepts du treillis.

Classement des résultats de la recherche par treillis

Dans les premières approches de recherche d'information par treillis de concepts citées plus hauts, les treillis sont utilisés dans la perspective de la recherche booléenne. Une requête consiste en une conjonction de termes. Les documents pertinents sont ceux indexés par tous les termes de la requête. Plus tard, de nouveaux critères de pertinence ont été défini. En conséquence, des documents n'apparaissant pas dans le même concept que la requête deviennent à leurs tours pertinents. Cependant leur pertinence est inférieure à ceux dans le concept représentant la requête. De ce fait il est essentiel de pouvoir classer l'ensemble de documents en réponses de façon à refléter cette différence de pertinence. La structure hiérarchique des treillis a été exploitée pour déduire automatiquement un tel classement. Dans [Carpineto and Romano, 2000], Carpineto et Romano proposent une approche qui s'appuie sur la similarité entre les documents et la requête. L'idée est la suivante : les documents en extension du concept représentant la requête dans le treillis sont classés en première position. Les documents dans les extensions des concepts joignables en un seul pas à partir de la requête sont classés en deuxième position. Les documents classés dans les positions supérieurs sont obtenus de la même façon en incrémentant le nombre de pas dans le treillis à chaque fois.

Nous estimons que ce type de classement est relativement empirique. A ce titre, notre proposition apporte une contribution théoriquement fondée pour le classement des résultats d'une requête en RI.

Approches récentes de recherche d'information par treillis

Pendant les dernières années, les approches de recherche d'information par treillis ont donné lieux à des méta-moteurs de recherche d'information. Ces méta-moteurs se placent entre les utilisateurs et les moteurs de recherche tels que Google et Yahoo. Leur rôle principal

est d'améliorer la présentation des résultats de recherche aux utilisateurs. Le fonctionnement général est le suivant.

Tout d'abord, une requête utilisateur est passée à un moteur de recherche. Le résultat retourné par le moteur de recherche, sous forme de liens vers des documents et un ensemble de "synsets" (mots décrivant le document : souvent une ou deux lignes du document qui contiennent un ou plusieurs mots clés de la requête), est transformé sous la forme d'un contexte formel. Les objets sont les liens et les attributs sont formés par l'ensemble des éléments des synsets utilisés. Après construction du treillis correspondant à ce contexte, on propose à l'utilisateur le nouveau classement du résultat fourni par le treillis. Ce type d'approche est implémenté dans plusieurs systèmes opérationnels tels que CREDO [Carpineto and Romano, 2004b], CREDINO [Carpineto et al, 2006], FooCA [Koester, 2006], Insighter [Jaoua, 2007], CRECHAINDO [Nauer and Toussaint, 2007] et SearchSleuth [Ducrou and Eklund, 2007].

CREDO est le premier système proposé. Après classification du résultat d'une requête sous la forme d'un treillis de concept, il affiche la partie supérieure du treillis sous la forme d'une hiérarchie de liens similaire à celle des hiérarchies de fichiers dans les systèmes hiérarchiques. Chaque nœud de la hiérarchie est un mot qui indexe un certain nombre de documents dans le résultat. Ce nombre est aussi indiqué avec le mot clé. Le système Credino est une adaptation du système CREDO pour les PDA.

Le Système FooCA est le seul système à visualiser le résultat retourné par le moteur de recherche, Google, sous la forme d'un contexte formel (une matrice) et le diagramme de Hasse du treillis correspondant à ce contexte. A travers cette visualisation, FooCA permet à

l'utilisateur d'apporter ses modifications au contexte en supprimant des mots clés qui lui paraissent inutiles pour sa recherche. Les modifications apportées sont ensuite propagées au treillis pour la visualisation de la nouvelle classification.

Le système CreChainDo reprend le même principe et les même fonctionnalités que CREDO et leur rajouter la possibilité d'interagir et de modifier le contexte en y incluant les retours de l'utilisateur. Il s'agit des fonctionnalités similaires à celle de FooCA mais avec soumission d'une nouvelle requête à Google et reconstruction du treillis après chaque modification apportée au contexte.

Le système Insighter [Jaoua, 2007]diffère des autres par la représentation de chaque concept du treillis par un seul mot, le plus représentatif de l'intension du concept. Le choix du mot représentatif s'appuie sur une heuristique qui consiste à construire un graphe biparti complet (ou biclique) entre les mots utilisés dans les documents et à effectuer la pondération des arcs du graphe selon les co-occurrences des pairs de mots. Le mot le plus représentatif d'un ensemble de mots est celui dont les liens ont les poids maximaux.

Le système SearchSleuth [Ducrou and Eklund, 2007] fournit une catégorisation supplémentaire du résultat en exploitant les relations entre le concept requête et les autres concepts du treillis. Cette catégorisation consiste à répartir les concepts du treillis sur quatre ensembles. Le premier ensemble, affiché comme résultat principal, est formé par les liens aux documents qui forment l'extension du concept requête. Le deuxième et le troisième ensemble sont formés respectivement par les sous concepts et les super-concepts de la requête dans le treillis. Le quatrième ensemble est formé par la fratrie du concept requête dans le treillis, les concepts ayant un parent commun avec la requête. Contrairement au premier ensemble, les autres ensembles ne sont pas affichés mais sont accessibles via des liens de raffinement en fonction du besoin en spécialisation, généralisation ou la combinaison des deux.

Recherche dans des domaines spécifiques

En plus des systèmes de recherche d'information sur le web cités dans la section précédente, d'autres approches de découvertes de ressources par treillis de concepts formels ont aussi été définies pour la découverte de ressources dans des domaines particuliers ou encore pour la découverte de types particuliers de ressources. Parmi ces approches nous pouvons mentionner les systèmes de gestion de messagerie électronique HireMail et

MailSleuth [II et al., 2003], les systèmes de recherche d'images Image-Sleuth [Ducrou and Eklund, 2008], Camelis 24 [Ferré, 2007] et de séquences vidéo [Mimouni and Slimani, 2006] et le système de recherche de bugs dans des fichiers de code source JAVA [Tilley et al., 2005], [Cellier, 2008]. Le principe général de ces systèmes est le même que ceux de la recherche d'information sur le web. Cependant, la spécificité de chaque système provient de sa définition de la pertinence qui peut dépendre de la particularité du domaine étudié.

II.3.2 L'extraction de règles d'associations

Introduite par Agrawal et al. L'extraction de règles d'association est l'un des principaux problèmes du processus d'extraction de connaissance dans les bases de données. Ce problème fut développé pour l'analyse de bases de données de transactions de ventes, chacune constituée d'une liste d'articles achetés, afin d'identifier les groupes d'articles achetés le plus fréquemment ensemble.

Une règle d'association sera par exemple : " Les clients qui achètent du pain ont tendance à acheter du lait" une telle règle d'association permettra de définir une stratégie commerciale dans le but d'augmenter les ventes en plaçant par exemple le pain et le lait dans le même rayonnage du magasin. Afin de ne gérer que les relations significatives entre les ensembles d'articles des mesures d'utilité (le support) et de précision (la confiance), empruntées aux statistiques, sont associées à chaque règle d'association. Les règles générées sont celles dont le support et la confiance sont supérieurs ou égaux à des seuils minimaux définis par l'utilisateur en fonction de ses objectifs et du type de données traitées.

Pasquier [Pasquier, 2002] à défini une nouvelle sémantique basée sur la connexion de Galois pour le problème de l'extraction de règles d'associations. En utilisant les opérateurs de la fermeture de la connexion de Galois, il a défini les itemsets fermés, qui forment le treillis des itemsets fermés et les itemsets fermés fréquents. Il a également démontré que l'ensemble des itemsets fermés fréquents constituent un ensemble générateur pour les itemsets fréquents (i.e. ensembles des attributs regroupés ensemble selon un support) et les règles d'association. Cela signifie que les règles d'association peuvent être générées sans accéder à la base de données

Sémantique des règles d'association

Une règle d'associations est une implication de la forme $A \Longrightarrow B$, qui signifie que "toute transaction contenant A tend à contenir B". Les mesures habituelles pour évaluer les règles d'association sont le support et la confiance.

Définition (support et confiance)

Soit $I = \{i_1, i_2, \ldots, i_m\}$ un ensemble de m littéraux, appelé *items* (*article en français*).

Soit $T = \{t_1, t_2, \ldots, t_n\}$ une base de donnée de n transactions, chaque transaction ti étant constituée d'un sous-ensemble $J \subset I$ d'items et identifiée par un identifiant unique appelé TID. Un sous-ensemble $J \subset I$ de taille k est appelé un *k-itemset.* Une transaction t_i « contient un *itemset* J si et seulement si $J \subset t_i$. Le support d'un *itemset* J est le pourcentage de transactions de T qui contiennent J :

$$Support\ (J) = \frac{|\{t \in T \mid J \subset I\}|}{|T|}$$

Un *itemset* dont le support est supérieur ou égal a un seuil minimal de support défini par l'utilisateur est appelé *itemset fréquent.*

La confiance d'une règle d'association $r : J_1 \Longrightarrow J_2$ est la probabilité conditionnelle qu'une transaction contienne J_2 sachant qu'elle contient J_1 :

$$Confiance\ (r) = \frac{Support\ (J_1 \cup J_2)}{Support\ (J_1)}$$

Etant donné une base de données de transactions B, le problème de l'extraction des règles d'association dans B consiste à déterminer l'ensemble des règles d'association dont le support et la confiance sont au moins égaux à des seuils minimaux de support *minsupport* et de confiance *minconfiance* définis par l'utilisateur. Ce problème peut être décomposé en deux sous-problèmes :

- Déterminer l'ensemble des *itemsets fréquents* dans B, c'est-à-dire les *itemset* dont le support est supérieur ou égal à *minsupport*.
- Pour chaque itemset fréquent J_1, générer toutes des règles d'association de la forme $r : J_2 \rightarrow J_1 - J_2$ telles que $J_2 \subset J_1$ et dont la confiance est supérieure ou égale a minconfiance.

Etapes de l'extraction de règles d'association

L'extraction de règles d'association est un processus itératif et interactif [Fayyad et al., 1996]. Ce processus peut se décomposer en quatre phases qui sont représentées dans la figure 4

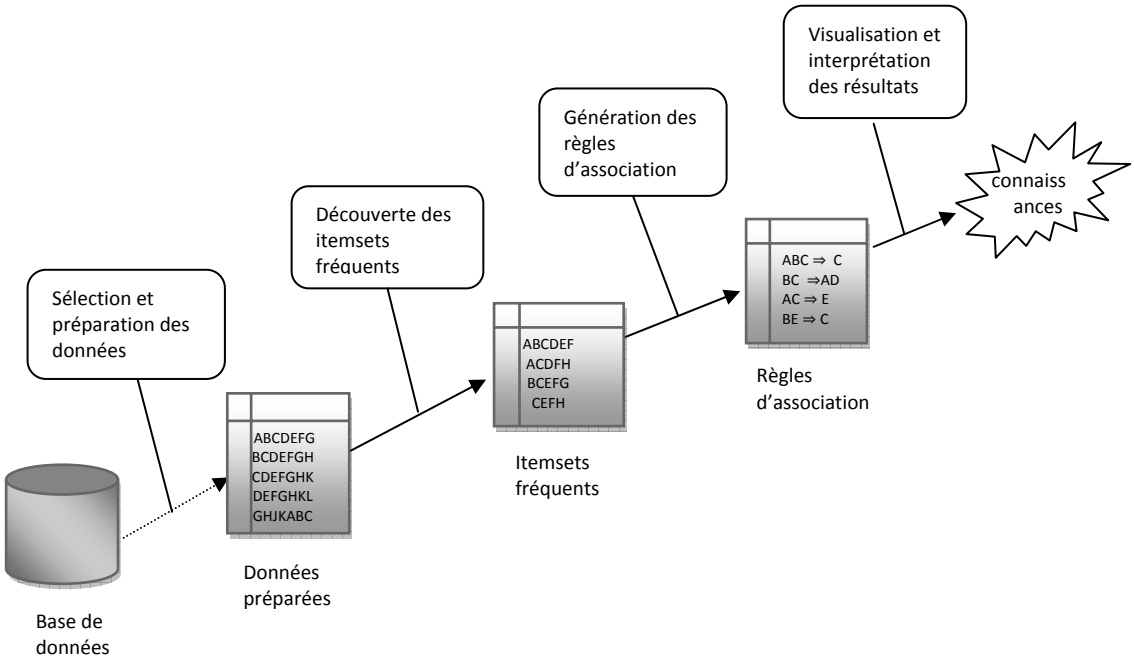


Fig.4 : Etapes du processus d'extraction de règles d'association.

Problème du temps d'extraction des règles d'association

Les temps de réponse de l'extraction des règles d'association dépendent principalement des temps d'extraction des *itemsets fréquents* car plusieurs balayages du contexte doivent être réalisés en comptant pour chaque itemset fréquent potentiel le nombre d'objets du contexte dans lesquels il est contenu. Le nombre d'itemsets à considérer ($2^{|I|}$) et la taille des jeux de données (contexte d'extraction) étant importants des algorithmes permettant de minimiser le nombre d'itemsets candidats (itemsets potentiellement fréquents) considérés et le nombre de balayages du contexte ont été proposés.

Il existe actuellement trois grandes tendances pour minimiser le temps d'extraction de règles d'associations dans une base de données, à savoir : ***l'approche orientée fréquents***, ***l'approche orientée fermés***, ***l'approche orientée maximaux***. Dans le cadre de notre mémoire nous présentons ***l'approche orientée fermés*** car basée sur la connexion de Galois. Cette approche utilise les opérateurs de fermeture de la connexion de Galois, afin de définir les *itemsets*

fermés fréquents qui forme *le treillis des itemsets fermés fréquents* constituant un ensemble générateur pour les itemsets fréquents et les règles d'association. Cela signifie que les règles peuvent être générées sans accéder à la base de données lorsque les itemsets fermés fréquents sont déterminés.

Ainsi une nouvelle décomposition du problème de l'extraction des règles d'association est proposée consistant à extraire les itemsets fermés fréquents au lieu des itemsets fréquents et la génération des règles d'association à partir des itemsets fréquents. Cette décomposition permet d'améliorer les temps de réponse car dans la plupart des cas le nombre d'itemsets fermés fréquents est bien inferieur au nombre d'itemsets fréquents

Algorithme de découverte des ensembles fermés fréquents

Deux algorithme, nommés Close et A-Close, ont été proposé [Pasquier, 2000] basés sur les opérateurs de fermeture de la connexion de Galois permettant la détermination efficace des ensemble fermés, selon cette fermeture, dans les bases de données, de plus ces deux algorithme prennent en compte la notion de support des ensembles fermés, et permettent donc de déterminer les itemsets fermés fréquents, a partir desquels les iemsets fréquents et les règles d'association sont dérivés de manière directe, sans accéder au jeu de données.

Itemsets fermés

Etant donne l'operateur de fermeture δ, un itemset $\iota \subseteq I$ *tel que* $\delta(\iota) = \iota$ est appelé *itemset fermé*. Un itemset ferme est donc un ensemble maximal d'items communs à un ensemble d'objets.

Exemple

Soit le contexte d'extraction de règles d'association ***D*** constitué de six objet, chacun identifier par son OID, et cinq items :

Objet	items			
1	A	C	D	
2	B	C	E	
3	A	B	C	E
4	B	E		

5	A	B	C	E
6	B	C	E	

Dans ce contexte, l'itemset {BCE} est un itemset fermé car il est l'ensemble maximal d'items communs aux objets {2, 3, 5, 8}. L'itemset {BC} n'est pas un itemset fermé car il n'est pas un ensemble maximal d'items communs a certains objets : tous les objets contenant les items B et C (objets 2, 3,5 et 6) contiennent également 1'item E. Dans une base de données de ventes, cela signifie que les clients achètent *au plus* les articles B, C et E et que tous les clients qui achètent les articles B et C achètent également l'article E.

Ensemble des itemsets fermés

Soit un contexte d'extraction $D = (O, I, \boldsymbol{R})$ et l'opérateur de fermeture de la connexion de Galois ψ. L'ensemble Γ des itemsets fermés dans le contexte D est :

$$\Gamma = \{l \subseteq I \mid l \neq \emptyset \wedge \psi(l) = l\}$$

Treillis des itemsets fermés

Soit Γ l'ensemble des itemsets fermés dans un contexte $D= (O, I, \boldsymbol{R})$ et la relation d'ordre partiel ⊆ sur les éléments de Γ. Pour chaque couple d'éléments $l1, l2 \in \Gamma$ nous avons $l1 \leq l_2$ si et seulement si $l1 \subseteq l_2$. Nous appelons alors l_1 *sous-ensemble fermé* de l_2 et l_2 *sur-ensemble fermé* de l_1. L'ensemble partiellement ordonné (Γ,⊆) forme un treillis complet car pour tout sous-ensemble $S \subseteq \Gamma$ il existe un plus petit majorant $Join(S)$ et un plus grand minorant *Meet(*S) définis comme suit :

$$\boldsymbol{Join(S)} = \psi(\bigcup_{l \in S} l)$$

$$\boldsymbol{Meet(S)} = \bigcap_{l \in S} l$$

Exemple

Le treillis [Pasquier, 2000] des itemsets fermés associé au contexte *D* est représenté dans la figure 1.4. Ce treillis contient 8 itemsets et sa hauteur est égale à cinq.

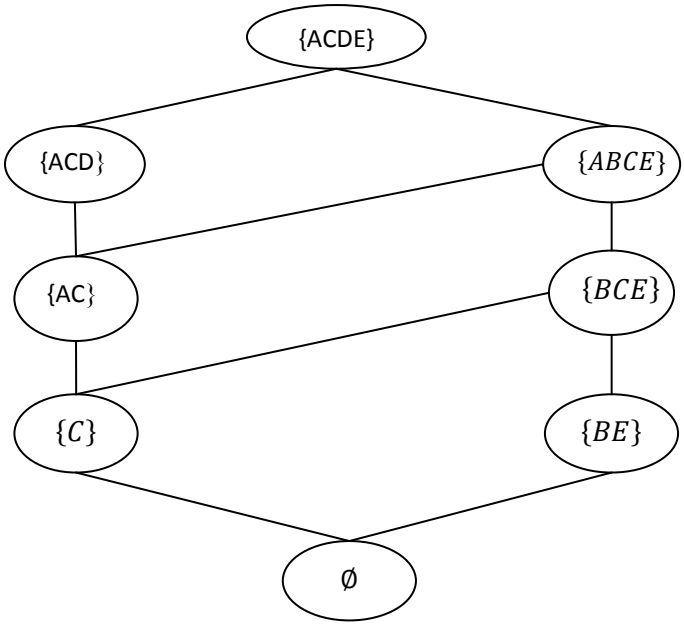


Fig. 4: Treillis des itemsets fermés associé au contexte D

Extraction des itemsets fermés fréquents

La découverte des itemsets fermés fréquents consiste à extraire du treillis des itemsets, selon la fermeture de la connexion de Galois, les itemsets dont le support est supérieur ou égal au seuil minimal de support minsupport. Pasquier dans [Pasquier, 2000] a proposé deux algorithmes d'extraction des itemsets fréquents nommés Close et A- Close. Ces algorithmes réalisent un parcours en largeurs du treillis des itemsets fermés pour extraire les itemsets fermés fréquents.

Extraction de règles d'association

Étant donnée un ensemble *F* d'itemsets fréquents dans un contexte d'extraction *D* pour un seuil minimal de support *minsupport*, la génération des règles d'association pour un seuil minimal de confiance *minconfiance* est un problème exponentiel dans la taille de *F*.

Définition (ensemble de règles d'association)

Soit un ensemble *F* d'itemsets fréquents dans un contexte d'extraction *D* pour un seuil minimal de support *minsupport*. Étant donné un seuil minimal de confiance *minconfiance*, l'ensemble *AR* des règles d'association valides dans B est :

$$AR = \{ r: l_2 \longrightarrow (l_1 - l_2) \mid l_1, l_2 \in F \wedge l_2 \subset l_1 \wedge \frac{support(l_1)}{support(l_2)} \geq minconfiance \}$$

En pratique, la génération des règles d'association est réalisée de manière directe, sans accéder au contexte d'extraction, et le coût de cette phase en temps d'exécution est donc faible comparé au coût de l'extraction des itemsets fréquents. Le principe général de la génération des règles d'association est le suivant :

Pour chaque itemsets fréquent l_1 dans F, tous les sous-ensembles l_2 de l_1 sont déterminés et la valeur du rapport $\frac{support(l_1)}{support(l_2)}$ est calculée. Si cette valeur est supérieure ou égale au seuil de confiance *minconfiance* alors la règle d'association $l_2 \longrightarrow (l_1 - l_2)$ est générée. Un algorithme efficace de génération des règles d'association a été proposé par Agrawal et al dans [Agrawal et Srikant, 1994].

Pasquier a proposé une nouvelle façon pour l'extraction des règles d'association en Utilisant les itemsets fermés fréquents. Il a défini des bases[2] pour les règles d'association. Ces bases sont des sous ensembles de l'ensemble des règles d'association valides, permettent d'amélioration la pertinence et l'utilité de l'ensemble de règles extraites

Conclusion

Dans ce chapitre nous avons d'abord présenté l'analyse de concepts formels de façon intuitive, par la suite nous avons donné des rappels mathématiques nécessaires à la définition et la compréhension des structures conceptuelles de l'AFC. Nous avons essentiellement mis l'accent sur certaine propriétés algébriques telle que les connexions de Galois, les operateurs

[2]Le terme « base » est utilisé au sens d'ensemble générateur non redondant
Une base pour un ensemble de règle E est un sous-ensemble de E ne contenant aucune règle redondante et à partir duquel toutes les règles appartenant à E peuvent être réduite.

de dérivation de Galois ainsi que les propriétés de fermeture. Enfin nous avons donné quelques domaines d'application de l'analyse de concepts formels. Le chapitre suivant présente la théorie des ensembles flous qui permettra d'étendre l'analyse de concepts formels classique.

Chapitre III : Théorie des ensembles flous

Sommaire

Introduction 27

III.1 Notions de base 27

III.1.1 Définitions usuelles 28

III.1.2 La cardinalité d'un ensemble flou 31

III.1.3 Ensembles Flous Discrets 31

III.2 Les opérateurs algébriques flous 32

III.2.1 Intersection et Union d'ensembles flous 33

III.2.2 Le complément 34

III.2. 3 Le produit cartésien 35

III.*2.4* L'égalité 35

III.2.5 L'inclusion 35

III.2. 6 Relations floues et composition de relations floues 36

III.3 Les implications floues 37

III.3.1 S-implications 38

III.3.2 R-implication 39

III.4 Distances entre ensembles flous 41

III.4.1 Distance de Hamming : 42

III. 4.2 La distance Euclidienne : 43

III.4.3 La distance Euclidienne pondérée : 43

III.4.3 Distance de Hausdorff et variantes 43

Conclusion 45

Introduction

La théorie des ensembles flous a été introduite par Lotfi Zadeh en 1965[Zadeh, 1965]. Cette théorie permet de généraliser la notion d'appartenance totale et la non appartenance totale classiquement utilisée. En effet, cette théorie permet d'introduire une transition graduelle et graduée du concept d'appartenance d'un élément à un ensemble. Les ensembles flous ont reçu une attention particulière ces dernières années dans différents domaine. Parmi ces domaine, nous pouvons citer le contrôle flou, logique floue, la représentation de connaissance, le traitement de connaissances incomplètes (c.-à-d. imprécise, incertaine, vague, floue, manquante). Les ensemble flous sont aussi utilisés pour manipuler des informations sous forme linguistique (variable et modificateur linguistique).

Un ensemble flou est généralement défini par une fonction d'appartenance qui généralise la notion usuelle de fonctions indicatrice. Aussi, dans ce chapitre nous commencerons par définir cette fonction d'appartenance puis nous donnerons les bases théoriques sur lesquelles reposent les ensembles flous, notamment certains concepts flous essentiels comme les propriétés des ensembles flous ainsi que les opérations flous sur ces ensembles.

III.1 Notions de base

Sous-ensemble flou

Soit X un univers du discours, un sous-ensemble flou E dans X est défini par une application de X dans [0, 1], cette application est appelée ***fonction d'appartenance***, et est notée $\boldsymbol{\mu_E}$. Pour chaque élément $x \in X$, la valeur de la fonction d'appartenance $\mu_E(x)$ est appelée ***degré d'appartenance*** de l'élément x au sous-ensemble E.

Un ensemble flou discret peut être représenté comme suit :

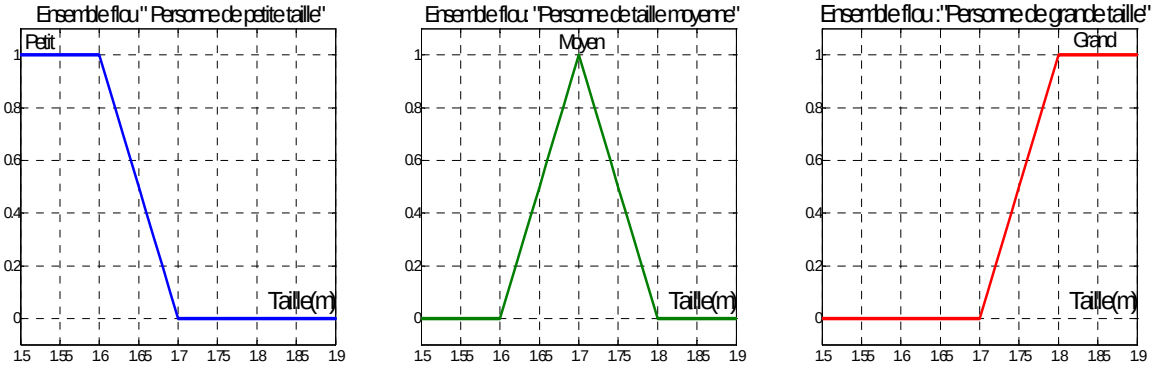
$$\boldsymbol{E = \mu_E(x_1)/x_1 + \mu_E(x_2)/x_2 + \ldots + \mu_E(x_n)/x_n}$$

Ou bien $$\boldsymbol{E = \{\mu_E(x_1)/x_1, \mu_E(x_2)/x_2, \ldots, \mu_E(x_n)/x_n\}}$$

Le degré d'appartenance $\mu_E(x)$ exprime la caractéristique de transition graduelle et non brutale entre l'appartenance complète et la non appartenance totale de l'élément x a l'ensemble E tel que :

$$\mu_E(x) = \begin{cases} 0 & \text{Non appartenance totale} \\ t \in [0,1[& \text{Plus } \mu_E(x) \text{ se rapproche de 1, plus } x \text{ appartient à } E \\ 1 & \text{Appartenance complète} \end{cases}$$

Exemple d'ensemble flou [Russel,…]



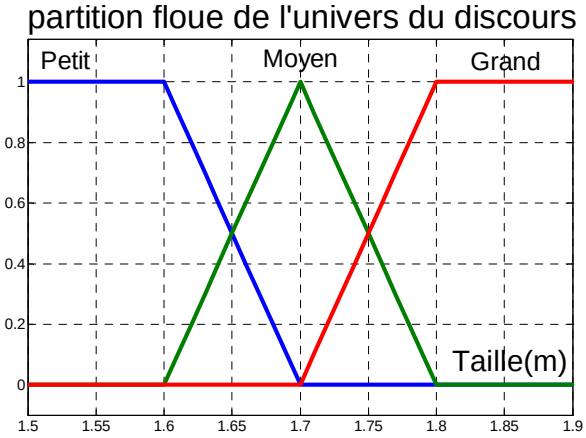


Fig. 1 : Sous-ensembles flous

III.1.1 Définitions usuelles

Nous donnons dans ce qui suit certaine caractéristiques et propriétés inhérentes aux ensembles flous.

a- Le noyau

Le noyau d'un sous-ensemble flou E de X, noté $Noy(E)$, est l'ensemble de tous les éléments lui appartenant totalement c'est-à-dire dont le degré d'appartenance à E vaut 1, il est représente par :

$$Noy(E) = \{x \in X \mid \mu_E(x) = 1\}$$

b- **Le support**

Le support d'un sous-ensemble flou E de X, noté $Supp(E)$, est l'ensemble de tous les éléments qui lui appartiennent au moins un petit peu c'est-à-dire dont le degré d'appartenance à E est différent de 0, il est représenté par :

$$Supp(E) = \{x \in X \mid \mu_E(x) > 0\}$$

c- **La hauteur**

La hauteur d'un sous-ensemble flou E de X, noté $h(E)$, est la valeur maximale atteinte sur tout le support de E c'est donc la borne supérieure de la fonction d'appartenance $\mu_E(x)$, d'ou :

$$h(E) = sup \{ \mu_E(x) \mid x \in X \}$$

Ensemble Flou Normalisé

Un ensemble flou est dit normalisé s'il est de hauteur 1.

d- **Coupe de niveau α (α-coupe)**

La coupe (resp. coupe stricte) de niveau α de l'ensemble flou E noté E_α (resp. $E_{\overline{\alpha}}$) est l'ensemble usuel (classique) composé des éléments dont le degré d'appartenance à E est supérieur ou égal (resp strictement supérieur) à α. Elles sont définies comme suit :

$$E_\alpha = \{x \mid x \in X \text{ et } \mu_E(x) \geq \alpha\}$$
$$E_{\overline{\alpha}} = \{x \mid x \in X \text{ et } \mu_E(x) > \alpha\}$$

D'où les propriétés suivantes sont valide [Dubois, 2000]

- $E_1 = Noy(E)$
- $E_{\overline{0}} = Supp(E)$
- L'ensemble des coupes de niveau d'un ensemble E sont emboités au sens où :

Si $\alpha > \grave{\alpha}$ *alors* $E_\alpha \subseteq E_{\grave{\alpha}}$

Une α-coupe d'un ensemble flou à deux vues : une vue horizontale (*cf. Fig. 2*) et une vue verticale (*cf. Fig. 3*)

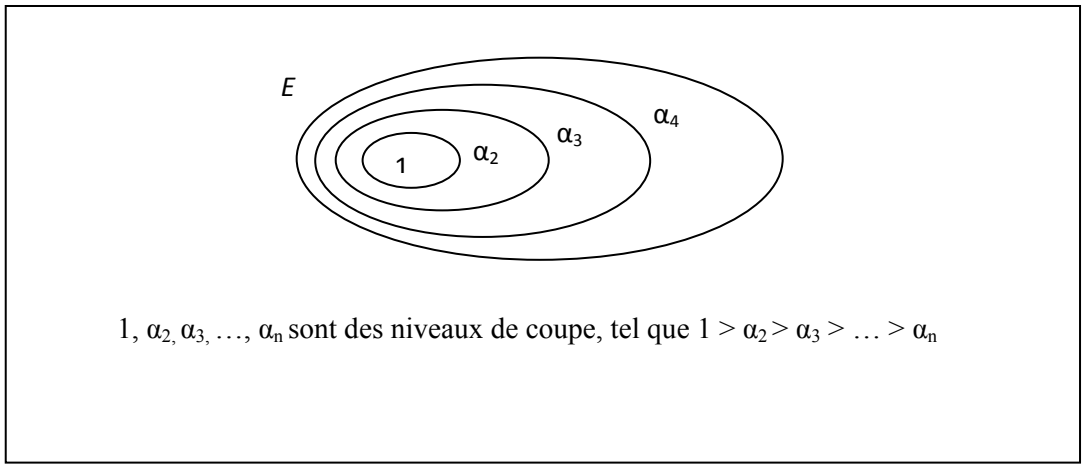


Fig. 1: Vue horizontale d'un ensemble flou: α-coupe

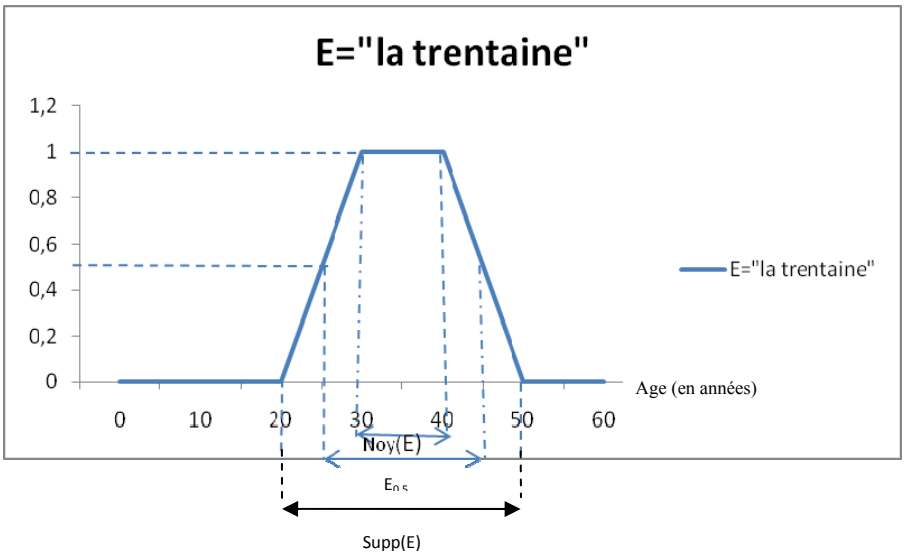


Fig. 2 : Vue verticale d'un ensemble flou

Les coupes de niveau α établissent un lien entre ensemble flou et ensemble classique qui permet d'exprimer un ensemble flou par ses interprétations plus au moins exigeante ou relaxées en terme de α-coupe. Une propriété intéressante des α-coupes est l'ensemble des éléments propres d'une coupe qui définissent une partition de l'ensemble flou

Ensemble des éléments propres d'une coupe

L'ensemble des éléments propres d'une coupe, noté $\boldsymbol{P_E(\alpha)}$, est l'ensemble des éléments de E_α qui n'appartiennent à aucune coupe de niveau $\grave{\alpha}$ strictement supérieure a α, on a alors :

$$\boldsymbol{P_E(\alpha)} = \{x \mid x \in E \; et \; \forall \; \grave{\alpha} > \alpha, \; x \notin \; E_\alpha\}$$

Un ensemble flou peut être reconstitué à partir de ses α-coupes en considérant les éléments propres de chaque coupe. L'ensemble initial est alors constitué des éléments de chaque ensemble de $\boldsymbol{P_E(\alpha)}$ affecté du degré α

III.1.2 La cardinalité d'un ensemble flou

La cardinalité d'un sous-ensemble flou E de X, notée $Card(E)$, est le nombre d'éléments appartenant à E pondéré par leurs degrés d'appartenance.

Pour un ensemble flou fini, on a donc : $Card(E) = \sum_{x \in X} \mu_E(x)$

De Lucas et Temini (1972) ont introduit cette définition et l'on appelée « the power of fuzzy set ». Elle évalue « combien » d'éléments contient l'ensemble E. Cette définition suppose que :

- Les degrés d'appartenance sont numériques
- Le support de E existe

Dans le cas de support infini, on peut utiliser l'intégrale de la fonction d'appartenance d'un ensemble flou sur son support, s'il existe, on aura donc :

$$Card\,(E) = \int \mu_E(x) \; dx$$

III.1.3 Ensembles Flous Discrets

Si le support d'un ensemble flou est fini, il est possible de donner la fonction d'appartenance par l'énumération de ses valeurs, ainsi par exemple dans le référentiel

X = {a, b, c, d}, on note habituellement E = {a / 0.1, b / 0.5, c / 0.3, d / 0} (ou en couples, en indices ou encore avec le «deux points») le sous-ensemble flou formellement défini par les degrés d'appartenance :

μ_E (a) = 0.1, μ_E (b) = 0.5, μ_E (c) = 0.3 et μ_E (d) = 0.

III.2 Les opérateurs algébriques flous

Afin de pouvoir manipuler aisément les ensembles flous, Zadeh a proposé des operateurs ensemblistes flous qui permettent aussi d'être étendus à la logique flous. Nous pouvons constater que la définition des opérateurs ensemblistes proposés par Zadeh n'est pas fixé à priori mais dépend de la définition sous-jacente d'une T-norme et /ou d'une T-conorme. Nous commençons par donner la définition de ces deux derniers operateurs.

Norme triangulaire

Une norme triangulaire (ou **T-norme**) est une fonction $\otimes$: [0,1]× [0,1] ⟶ [0,1] qui vérifie, pour tout *x, y, z, t* dans [0, 1], les propriétés suivantes :

- Commutativité : $x \otimes y = y \otimes x$
- Associativité : $x \otimes (y \otimes z) = (x \otimes y) \otimes z$
- Monotonie : $x \otimes y \leq z \otimes t$ *si* $x \leq z$ *et* $y \leq t$
- 1 comme élément neutre : $x \otimes 1 = x$

Co-norme triangulaire

La co-norme triangulaire (ou **T-conorme**) est une fonction $\oplus$ d :[0,1]× [0,1] ⟼ [0,1] qui vérifie, pour tout *x, y, z, t* dans [0, 1], les propriétés suivantes :

- Commutativité : $x \oplus y = y \oplus x$
- Associativité : $x \oplus (y \oplus z) = (x \oplus y) \oplus z$
- Monotonie : $x \oplus y \leq z \oplus t$ *si* $x \leq z$ *et* $y \leq t$
- 0 comme élément neutre : $x \oplus 0 = x$

Il existe une infinité de norme et de co-norme, les plus courantes sont données dans le tableau ci-dessous :

Norme : T(x,y)	**Co-norme associée**	**Nom**
Min (*x, y*)	Max (*x, y*)	**Zedeh**
xy	*x+y-xy*	**probabiliste**
Max (x+y-1, 0)	Min(x+y, 1)	**Lukasiewicz**
xy / γ+ (1-γ)(x+y-xy)	*x+y-xy-xy (1-γ) / 1-(1-γ)xy*	**Hamacher**
x si y=1 *y si x=1* *0 sinon*	*x si y=0* *y si x=0* *0 sinon*	**Weber**

III.2.1 Intersection et Union d'ensembles flous

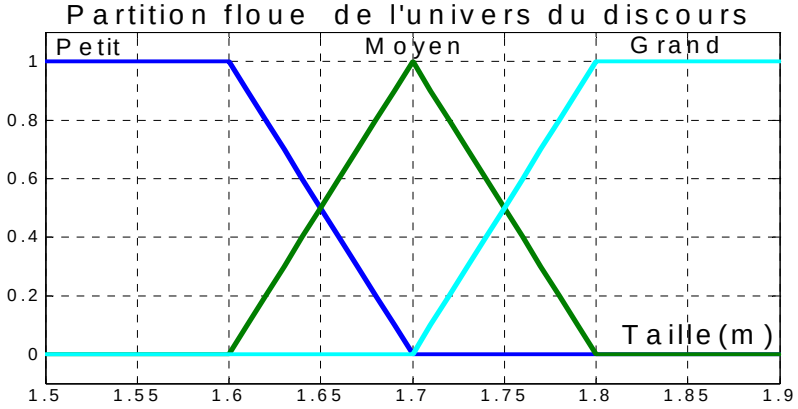
Soit *E* et *F* deux ensembles flous. L'intersection (resp. l'union) des ensembles *E et F* est un ensemble flou *G* dont la fonction d'appartenance est définie à partir des fonctions d'appartenance de *E* et *F* comme suit :

Intersection: $\forall x \in X, \mu_G(x) = \mu_{E\cap F}(x) = \mu_E(x) \otimes \mu_F(x)$

Union: $\forall x \in X, \mu_G(x) = \mu_{E\cup F}(x) = \mu_E(x) \oplus \mu_F(x)$

Exemple

Soit A l'ensemble des personnes petites et B l'ensemble des personnes moyennes de l'univers du discours donné précédemment. Les figures ***fig.4*** et ***fig.5*** donne respectivement l'ensemble des personnes petite **ET** moyenne et l'ensemble des personnes petites **OU** moyennes



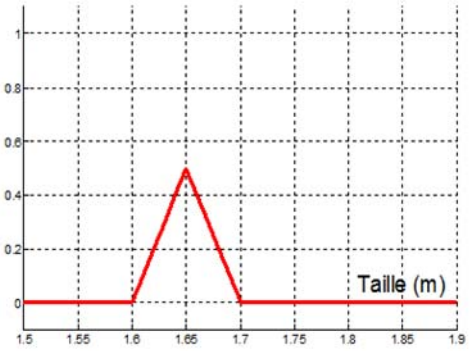


Fig.4 : ensemble flou : personne petite ET moyenne

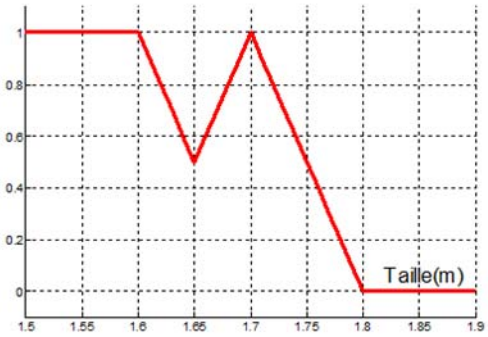


Fig.5 : ensemble flou : personne petite OU moyenne

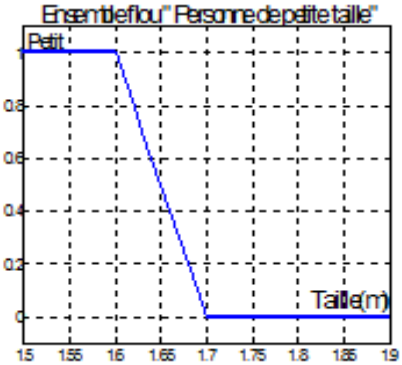
III.2.2 Le complément

Le complémentaire d'un ensemble flou *F,* noté F^c, est défini par sa fonction d'appartenance de la manière suivante :

$$\forall x \in X \quad \mu_{F^c} = 1 - \mu_F(x)$$

Exemple

Soit A l'ensemble flou des personnes petites. La figure 5 donne le complément de A, à savoir l'ensemble flou des personnes **NON** petites



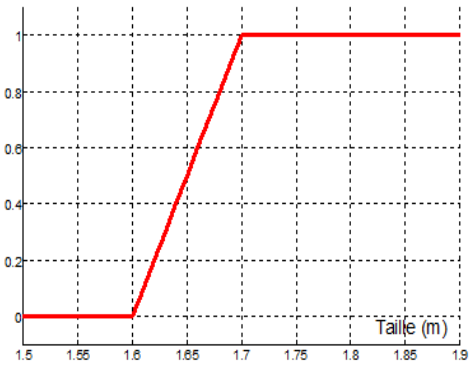


Fig. 6 : Ensemble flou : « personnes NON petites »

III.2. 3 Le produit cartésien

Les problèmes considérés sont souvent décrits dans plusieurs univers de référence $X_1, X_2, \ldots, X_n$. Il peut être intéressant de pouvoir raisonner dans un univers de références X global composé de chacun des univers initiaux. X correspond donc au produit cartésien de $X_1, X_2, \ldots, X_n$:

$X = X_1 \times X_2 \times \ldots \times X_n$ et ses éléments x sont des n-uplet : $x = (x_1, x_2, \ldots, x_n)$.

Le produit cartésien de n sous-ensembles flous $E_1, E_2, \ldots, E_n$ définis respectivement sur les univers $X_1, X_2, \ldots, X_n$ est un sous-ensemble flou E défini sur X par sa fonction d'appartenance :

$$\forall\, x = (x_1, x_2, \ldots, x_n) \in X,\ \mu_E(x) = (\mu_{E1}(x) \otimes \mu_{E2}(x), \ldots, \mu_{En}(x))$$

III.*2.4* L'égalité

Deux sous ensembles flous E et F sont égaux si leurs fonctions d'appartenance sont égales a tout point de X

$$\boldsymbol{E = F \Rightarrow \forall\, x \in X,\ \mu_E(x) = \mu_F(x)}$$

III.2.5 L'inclusion

Une façon habituelle de définir l'inclusion d'un ensemble A dans un ensemble B repose sur l'expression *:* $\boldsymbol{(A \subseteq B) \Leftrightarrow (\forall\, x \in X, (x \in A) \Rightarrow (x \in B))}$

qui peut être écrit également en termes de contraintes entre valeurs des fonctions caractéristiques de A et B sous la forme *:* $\boldsymbol{(A \subseteq B) \Leftrightarrow (\forall\, x \in X, f_A(x) \leq f_B(x)).}$

Cette dernière s'étend canoniquement a deux ensembles flous E et F et conduit a :

$$\boldsymbol{(E \subseteq F) \Leftrightarrow (\forall\, x \in X, \mu_E(x) \leq \mu_F(x))}$$

Une définition plus exigeante de l'inclusion est donnée par :

$$\boldsymbol{(E \subseteq F) \Leftrightarrow (\forall\, x \in X, (x \in Supp(E)) \Rightarrow (x \in Noy(F)))}$$

La figure ci-dessous donne un exemple d'inclusion de deux ensembles flous *A* et *B*

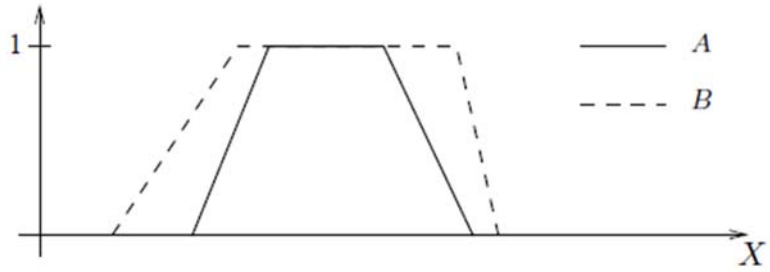


Fig. 7 : Inclusion de deux ensembles flous

III.2. 6 Relations floues et composition de relations floues

Soit deux ensemble de référence *X* et *Y*, on peut alors représenter une relation ***R*** entre *X* et *Y* par un sous-ensemble flou de $X \times Y$, dont la fonction d'appartenance μ_R est définie par :

$\mu_R : X \times Y \rightarrow [0, 1]$. La relation ***R*** est notée ***R***(*X*, *Y*)

La composition de deux relations floues $R_1(X, Y)$ et $R_2(Y, Z)$ définit une relation
R = R1∘ R2 sur $X \times Z$ dont la forme de la fonction d'appartenance μ_R est définie par :

$$\forall (x, z) \in (X \times Z), \mu_R(x, z) = \text{Sup}_{y \in Y} \text{ Min } (\mu_{R1}(x, y), \mu_{R2}(y, z))$$

Composition Sup-Inf

Soit *E* un ensemble flou dans *X*. La composition Sup-Inf des relations floues ***R***(*X*, *Y*) est un ensemble flou *F*, notée par $F = R \circ E$. Elle est définie par la fonction d'appartenance $\mu_{R \circ E}$ comme suit :

$$\mu_{R \circ E}(y) = \vee_x \{\mu_E(x) \wedge \mu_R(x,y)\} \text{ ou } \vee = sup \text{ et } \wedge = inf$$

Propriétés des relations floues : Une relation floue ***R****(X, Y)* est dite :

- **Réflexive** : Ssi $\forall x \in X \, \mu_R(X, X) = 1$,
- **Symétrique** : Ssi $\forall x_1, x_2 \in X \, \mu_R(x_1, x_2) = \mu_R(x_2, x_1)$

- **T-Transitive** : $\forall x_1, x_2, x_3 \in X\ \mu_R(x_1, x_3) \geq \otimes\ (\mu_R\ (x_1, x_2), \mu_R\ (x_2, x_3))$

Relations floues particulières

- **Relation** de **ressemblance** : Une relation floue est dite une relation de *ressemblance* (ou relation de *tolérance)* si elle est réflexive et symétrique.

- **Relation de similarité** : Une relation de similarité est une extension de la relation d'équivalence, enrichie de versions graduelles de la réflexivité, symétrie et transitivité (i,e. une relation de ressemblance qui vérifie la propriété de Min-transitivité [Dubois 2000]. La *Min-Transitivité* est définie par :

$$\forall x_1, x_2, x_3 \in X\ \mu_R(x_1, x_3) \geq Min\ (\mu_R\ (x_1, x_2), \mu_R\ (x_2, x_3))$$

Remarque :

- Une relation de similarité peut être aussi définie avec Max-min-transitivité définie par: $\forall x_1, x_2, x_3 \in X\ \mu_R(x_1, x_3) \geq Max_{y \in X}\ \{Min\ (\mu_R\ (x_1, x_2), \mu_R\ (x_2, x_3))\}$

- La transitivité Max-min est une propriété très restrictive et elle ne s'applique pas sur tous les problèmes.

III.3 Les implications floues

D'ordinaire, l'implication $(P \longrightarrow Q)$ exprime une liaison entre les valeurs p et q. Elle rend *vrai* quand la proposition Q est vraie ou quand la proposition P est fausse, d'ou les valeurs de vérité données dans la table **Tab. 1**

Q / P	**Vrai**	**Faux**
Vrai	Vrai	Faux
Faux	Vrai	Vrai

Tab. 1. Table de vérité de l'implication matérielle $P \Rightarrow Q$

Plusieurs approches d'extension de l'implication peuvent être envisagées lorsque les propositions *p* et q prennent, non plus une valeur booléenne, mais une valeur de l'intervalle unité. Toute implication floue est une fonction (notée $\Rightarrow_f$) définie par :

$$\Rightarrow_f : [0,1] \times [0,1] \rightarrow [0,1]$$
$$(p, q) \quad \longmapsto \quad (p \Rightarrow_f q)$$

Et elle d'autant plus vrai (resp. fausse) que son résultat est proche de 1 (resp. 0). Les implications floues peuvent être représentées selon plusieurs classifications. Dans la suite elles sont reparties en trois familles couvrantes les plus fréquentes (et utiles). Leurs principales caractéristiques sont précisées ainsi que leur sémantique afin d'en permettre un *usage rationnel.*

III.3.1 S-implications

L'appellation S-implication vient de l'expression anglaise Strong **Implication.** On définit la classe des S-implications (notées $\Rightarrow_{S\text{-}I}$) a partir de l'expression : **((non** P) **ou** *Q)* de la manière suivante :

$$P \Rightarrow_{S\text{-}I} Q = \oplus(1 - p, q)$$

Ou la disjonction (ou) est généralisée par une co-norme ($\oplus$).

II existe donc une infinité de S-implications et les trois plus communes sont données dans la table **Tab.2.**

Nom	Symbole	Co-norme sous-jacente	Valeur de verite
Kleene-Dienes	$\Rightarrow_{K\text{-}D}$	$\oplus(p, q) = \text{Max}(p, q)$	$\text{Max}(1 - p, q)$
Reichenbach	$\Rightarrow_{Rb}$	$\oplus(P, q) = p + q - pq$	$1 - p + p * q$
Lukasiewicz	$\Rightarrow_{Lu}$	$\oplus(p,q) = Mim(p + q, 1)$	$\text{Min}(1 - p + q, 1)$

Tab.2. Les trois principales S-implications

De plus, l'ordre suivant sur les S-implications précédentes est valide [Amroune,...]

$$(P \Rightarrow_{\text{K-D}} Q) \leq (P \Rightarrow_{\text{Rb}} Q) \leq (P \Rightarrow_{\text{Lu}} Q)$$

L'implication de Kleene-Dienes est la plus petite des S-implications (puisqu'elle est construite à partir de la plus petite des co-normes). La plus grande (notée $\Rightarrow_{\text{S-M}}$) est trouvée en prenant la co-norme de Weber, soit:

$$P \Rightarrow_{\text{S-M}} Q = \begin{cases} (1-p) & si\ q = 0 \\ q & si\ p = 0 \\ 1 & sinon \end{cases}$$

Remarque :

- Comme pour 1'implication usuelle, on a : $(P \Rightarrow_{\text{S-I}} \textbf{faux}) = \perp(1\text{-}p, 0) = 1\text{-}p$.
- Les implications de cette classe sont leur propre contraposée :
  $\textbf{non } Q \Rightarrow_{\text{S-I}} \textbf{non } P = \perp(1-(1-q), 1-p) = \perp(q, 1-p) = \perp(1-p, q) = P \Rightarrow_{\text{S-I}} Q$

III.3.2 R-implication

La seconde classe d'implications floues regroupe les *R*-implications, ainsi dénommées parce qu'elles utilisent le principe de *résiduation.*

Dans le cas classique on a :

$(P \text{ et } (P \Rightarrow Q)) \Leftrightarrow (P \text{ et } (\text{non } P \text{ ou } Q)) \Leftrightarrow ((P \text{ et non } P) \text{ ou } (P \text{ et } Q)) \Leftrightarrow (P \text{ et } Q)$. Ce qui fait que la valeur de vérité de la proposition (P et ($P \Rightarrow 0$) est inferieure ou égale a celle de la proposition Q, i.e., $((p \text{ et } (p \Rightarrow q)) \leq q)$.

Dans le cas flou cette inégalité sert de point de départ pour chercher l'élément maximal la vérifiant, soit:

$$P \Rightarrow_{\text{R-i}} Q = \text{SUP}_{[0,1]} \{u \mid \otimes(p, u) \leq q\} \ *$$

$\otimes$: étant une norme triangulaire.

II y a donc une infinité de R-implications dont la forme générale est:

$$P \Rightarrow_{\text{R-i}} Q = \begin{cases} 1 & si\ p \leq q \\ \otimes(p,q) & sinon \end{cases}$$

Les plus courantes figurent dans la table Tab.3.avec la norme triangulaire (génératrice) qui leur est associée. L'implication de Rescher-Gaines peut être vue comme une *R*-implication particulière prenant en argument des propositions floues en rendant un résultat booléen, dérivant d'une formule voisine de la formule (*), a savoir :

$$P \Rightarrow_{\text{R-G}} Q = \text{SUP}_{[0,1]} \{u \mid \otimes(p, u) \leq q\}$$

$$= \begin{cases} 1 & si\ p \leq q \\ 0 & sinon \end{cases}$$

Et ce, pour toute norme⊗. De façon analogue aux S-implications, ces implications peuvent être ordonnées : $(p \Rightarrow_{\text{R-G}} q) \leq (p \Rightarrow_{\text{Gö}} q) \leq (p \Rightarrow_{\text{Gg}} q) \leq (p \Rightarrow_{\text{Lu}} q)$.

Nom	Symbole	Valeur de vérité	Norme sous-jacentes
Gödel	$\Rightarrow_{\text{Gö}}$	1 si $p \leq q$ q sinon	$\otimes(p, q) = \min(p, q)$
Goguen	$\Rightarrow_{\text{Gg}}$	1 si $p < q$ q/p sinon	$\otimes(p,q) = pq$
Lukasiewicz	$\Rightarrow_{\text{Lu}}$	1 si $p < q$ $1 - p + q$ sinon	$\otimes(p, q) = \max(p+ q - 1, 0)$

Tab. 3. les trois principales R-implications

L'implication de Gödel est la plus petite ***R***-implication propre (i,e., en excluant celle de Rescher-Gaines), puisque elle est construite a partir de la plus grande norme triangulaire. La plus grande (notee =>R-M) est celle construite à partir de la plus petite norme (celle de Weber), soit:

$$P \Rightarrow_{\text{R-G}} Q = \begin{cases} 1 & si\ p \leq q \\ q & si\ p > q\ et\ p = 1 \\ 1^- & si\ p > q\ et\ p = 1 \end{cases}$$

Ou 1^- désigne une valeur limite strictement inferieure a 1 (le comportement discontinu de cette implication provient de celui de la norme triangulaire sous-jacente).

Sémantique associée aux implications floues

Au-delà de ces aspects, il apparait particulièrement important de s'intéresser a la signification qui peut être attribuée aux diverses implications floues. Ceci permet en particulier de procéder à un choix approprié et d'en expliciter le sens. Les S-implications (Kleene-Dienes, Reichenbach et Lukasiewicz) présentent la caractéristique de garantir un *niveau de satisfaction minimal* de $(1-p)$. De plus, pour I' implication de Kleene-Dienes, p joue le rôle d'un niveau d'importance attribué à la conclusion Q.

En effet, si p est faible, on "voit" seulement les valeurs élevées de Q, contrairement au cas où p est élevé pour lequel Q est "vue" sur une plage de valeurs large ($1-p$ est faible quand p est grand et donc l'intervalle $[1-p, 1]$ sur lequel on "voit" Q est alors grand).

Pour ce qui est des R-implications, la valeur de la prémisse P joue le rôle d'un seuil qui conduit à la totale satisfaction quand il est atteint ou dépassé par la valeur de la conclusion Q et à une satisfaction partielle sinon. Dans ce dernier cas, le degré obtenu *ne dépend que de la conclusion* avec I' implication de Godel, varie avec le *ratio* entre la valeur de la conclusion et celle de la prémisse avec I' implication de Goguen et avec leur *écart* (ou distance) avec I' implication de Lukasiewicz

III.4 Distances entre ensembles flous

Une distance entre deux sous-ensembles flous d'un même ensemble de référence est une évaluation de ce qui sépare ces deux sous-ensembles. Elle permet d'apprécier leur ressemblance ou leur dissemblance. Il existe plusieurs types de distances : la distance de Hamming, la distance Euclidienne, et la distance Hausdorff.etc…

Supposons que A, B, C sont trois sous-ensembles flous dans l'ensemble de référence E Chaque mesure de distance doit satisfaire les trois axiomes suivantes :

- $D(A, B) \geq 0$ (Une distance n'est jamais une quantité négative)
- $D(A, B) = D(B, A)$ (La distance entre A et B est égale à la distance B et A)
- $D(A, A) = 0$ (La distance entre A et A est toujours nulle)

Si l'on associe à une distance un opérateur de composition entre deux distances, on doit avoir la propriété suivante appelée **inégalité triangulaire**

D (A, C)≤ $D(A,B)$* D (B, C)

Où * est l'opérateur considérée, par exemple la somme pour la distance de Hamming.

III.4.1 Distance de Hamming :

La définition générale de la distance de Hamming est la somme des valeurs absolues des différences entre les fonctions d'appartenance des objets correspondants. Si A et B sont deux sous-ensembles flous dans le même ensemble de référence $E=\{x_1, x_2, x_3 \ldots x_n\}$, la distance de Hamming entre eux est :

$$D(A, B) = \sum_{i=1}^{n} |\mu_A(Xi) - \mu_B(Xi)|$$

Ou n est le nombre d'éléments dans l'ensemble de référence E. Cette définition s'étend très bien aux ensembles classiques.

Pour tenir compte du nombre d'élément de l'ensemble de référence, on préfère utiliser la notion de « distance de Hamming relative » en divisant le résultat par le nombre d'éléments de l'ensemble de référence, soit :

$$\delta(A,B) = \frac{1}{n}\sum_{i=1}^{n} |\mu_A(X_i) - \mu_B(X)_i|$$

Exemple :

Soit deux sous-ensembles flous du même ensemble de référence

$E = \{a, b, c, d, e, f, g\}$

$A = \{a/0.5, b/1, c/0.7, d/0, e/0.2, f/0.5, g/0.9\}$

$B = \{a/0.8, b/0.2, c/0.7, d/0.3, e/0.1, f/0.1, g/0.6\}$

La distance de Hamming est : $D(A,B) = 2.3$

La distance relative de Hamming est : $\delta(A, B) = \frac{D(A,B)}{7} = 0.328$

Si A et B sont deux sous-ensembles continus, la distance de Hamming est généralisée en utilisant la surface de Hamming. Elle est définie sur tout le segment tel que la réunion des deux sous-ensembles possède une fonction d'appartenance non nulle.

La formule générale dans ce cas est :

$$D(A, B) = \int_a^b |\mu_A(X) - \mu_B(X)| dx$$

La fonction d'appartenance d'un ensemble rectangulaire ayant la même base et la même surface que la réunion de deux sous-ensembles correspondra à la distance de hamming relative.

III. 4. 2 La distance Euclidienne :

La distance Euclidienne mesure la ressemblance entre deux sous-ensembles flous

$$D_{Eu}\ (A, B) = \sqrt{\sum_{i=1}^{n}(\mu_A(X_i) - \ \mu_B(X_i))^2}$$

III.4.3 La distance Euclidienne pondérée :

$$D_{Ep}(A, B) = \sqrt{\sum_{i=1}^{n}\ W_i{}^2 . (\mu_A(X_i) - \ \mu_B(X_i))^2}$$

Wi étant un réel

La distance Euclidienne pondérée est comparable à la distance Euclidienne mais permet l'introduction de coefficient Wi pour donner plus d'importance à la différence

$\mu_A(X_i) - \ \mu_B(X_i)$ pour un X_i donné. Les Wi permettent aussi de normaliser le résultat de cette distance entre 0 et 1.

le choix des coefficient Wi dépend de l'application et ils sont souvent fixés par plusieurs essais. Cette distance semble donc en général difficile à utiliser

III.4.3 Distance De Hausdorff Et Variantes

En mathématiques, et plus précisément en géométrie, la **distance de Hausdorff** est un outil topologique qui mesure l'éloignement de deux sous-ensembles d'un espace métrique sous-jacent.

Cette distance est utilisée pour le traitement d'images, elle indique si deux formes sont les mêmes ou si elles sont différentes, la distance quantifie ces dissemblances. En dimension 2, la distance de Hausdorff permet de numériser une image ou encore de reconnaître une forme

L'intérêt des distance de Hausdorff en traitement d'image est multiple et couvre plusieurs approches : distance d'un point a un ensemble, distance entre deux objets et distance entre

deux points dans un ensemble. Le deuxième type de distance a fait l'objet de nombreux travaux dans la littérature notamment la distance entre ensembles flous.

Considérant deux ensembles de points *A* et *B*, la distance de Hausdorff directe *h* peut être écrite comme :

$$h(A, B) = max_{a\in A}(min_{b\in B}||a - b||) \ldots\ldots\ldots\ldots\ldots\ldots (1)$$

La distance entre un point a et un ensemble de n points $B = \{b_1, b_2, \ldots, b_n\}$ est noté :

$$D(a, B) = min_{b\in B}||a - b|| \ \ldots\ldots\ldots\ldots\ldots\ldots (2)$$

On remarque que la fonction (1) n'est pas symétrique, elle ne peut donc pas être une distance au sens métrique. La symétrie est habituellement restaurée en prenant le maximum entre $h(A, B)$ *et* $h(B, A)$ la définition de la distance indirecte de Hausdorff :

$$H(A, B) = max\ \{ h(A, B), h(B, A)\}$$

Comme il a été montré dans [jpt-m06], il existe des variantes aux définitions précédentes qui donnent de meilleurs résultats pour la reconnaissance de contours. Nous donnons ci-dessous quelques façons de calculer les distances directes $h(A, B)$ *et* $h(B, A)$:

$h_1(A, B) = \ min_{a\in A} D(a, B)$

$h_2(A, B) = K^{th}_{a\in A} D(a, B)$

$h_3(A, B) = \ max_{a\in A} D(a, B)$

$h_4(A, B) = \frac{1}{N_a}\sum_{a\in A} D(a, B)$

De plus quatre façons de construire la distance indirect $H(A, B)$ sont proposées dans [Simon et Tarel, 2006] :

$H_1(A, B) = \min (h(A, B), h(B, A))$

$H_2(A, B) = \max (h(A, B), h(B, A))$

$H_3(A, B) = \frac{h(A,B) + h(B,A)}{2}$

$H_4(A, B) = \frac{N_a h(A,B) + N_b h(B,A)}{N_a + N_b}$

On combinant les distances *h* et les fonctions *H* on obtient différentes variantes de la distance de Hausdorff. La combinaison de H_2 et h_3 défini la distance de Hausdorff classique (*HD*), définie comme suit :

Soit deux figure définies grâce à un ensemble de points *A* et *B*. La distance de Hausdorff classique [Truck, 2002] entre *A* et *B* s'écrit :

$$H(A, B) = max \{ h(A, B), h(B, A)\}$$

Avec $h(A, B) = max_{a\in A}\{min_{b\in B}\ d(a, b)\}$ *et* $h(B, A) = max_{b\in B}\{min_{a\in A}\ d(b, a)\}$

Ou d(a,b) est une distance métrique ou distance euclidienne tel que :

$d(a, b)=\sqrt[2]{|a-b|}$

Autrement dit, on cherche, pour chaque point a de *A*, la plus petite distance entre chaque point *b* de *B* et a. On garde la plus grande distance trouvée parmi tous les *a*. Puis pour chaque point *b* de *B*, on cherche la plus petite distance entre chaque point *a* de *A*. On garde la plus grande distance trouvée parmi tous les *b* et on prend la plus grande des deux distances obtenues. C'est cette distance qui est appelée *distance de Hausdorff*.

Conclusion

Dans ce chapitre nous avons fait une présentation détaillée de la théorie des ensembles flous, nous avons présenté les notions de bases nécessaires à la compréhension de cette dernière. Nous avons particulièrement mis l'accent sur les implications floues notamment les R-implications qui sont à la base de la spécification d'operateurs de dérivation de Galois flous tel que nous allons le voir dans chapitre qui suit.

Chapitre IV : Analyse formelle de concepts flous

Sommaire

Introduction 46

IV.1. contexte formel flou et concepts formels flous 47

IV.2 Rappels mathématiques 48

IV.3 Approches existantes 49

IV.3.1 Méthodes à seuil(s) [MESSAI, 2009] 49

IV.3.2 Approches “mono-face” (one Sided) 50

IV.3.3 Approches générales 52

IV.3.3.1 Approche de Burusco…….52

IV.3.3.2 Approche Belohlavek…….54

IV.3.3.3 Approche de Georgesco et Popescu……..……..56

Conclusion 56

Introduction

L'ACF offre un cadre formel pour le développement d'applications efficaces d'analyse de données. De plus les structures conceptuelles de l'ACF, telles que les treillis de concepts, peuvent être exploitées à la fois comme outils de classification, support de découverte de ressources, de fouille de données, etc. Ces caractéristiques sont à l'origine du succès des approches d'analyse, de représentation et d'exploitation de données qui s'appuient sur l'ACF. Cependant, avant de profiter de la classification en treillis de concepts, il est indispensable de ramener les données manipulées, souvent complexes et hétérogènes, à un contexte formel. Dans les applications réelles les attributs décrivant les objets prennent différentes valeurs pour les différents objets et parfois même plusieurs valeurs pour un même objet. Dans ce cas le modèle de base proposé par Wille [Wille, 1982] s'avère inapproprié.

L'exemple de la table 1[Djouadi, 2009] illustre différents cas de données incomplètes :

	Jeune	*Anglais*	*Marié*
Pierre	1	o.9	×
Sophie	0.7	]0,1]	?
Mike	o.6	0	
Nahla	1	[0.2, 0.4]	(0.7; 0.0)

Table 1 Contexte formel avec données incompletes

La première colonne de la Table 1 indique le degré de satisfaction de la propriété graduelle *jeune*, la seconde, le niveau de maîtrise de l'*Anglais*. Ce niveau peut être connu de manière précise ou seulement apprécié sous forme d'intervalles. Par exemple, on sait seulement qu'il est faux que *Sophie* ne parle pas du tout *Anglais*.

La troisième colonne illustre la présence d'incomplétude et d'incertitude : alors que Pierre est marié et Mike ne l'est pas, rien ne peut être affirmé pour Sophie. La paire (0.7;0.0) exprime que la certitude que Nahla ne soit pas mariée est de 0.0, et de 0.7 qu'elle le soit. Remarquons que l'entrée " ?" corresponds à (0 ; 0).

A ce titre, plusieurs approches ont proposé d'étendre l'analyse de concepts formels classiques à des relations non booléennes. Certaine de ces approches sont basées sur la théorie des ensembles flous [Zadeh, 1965] et permettant de modéliser des relations floues ou la notion

de satisfaction de la relation est maintenant représentée par un degré d ($d \in [0,1]$). Notre mémoire s'inscrit dans ce type d'approche et nous nous limitons aux valeurs floues (telle que les valeurs de la première colonne de la table 1)

Dans le présent chapitre nous allons présenter les notions essentielles de l'analyse de concepts formels concernant les contextes formels flous que nous appellerons ***analyse de concepts formels flous.***

IV.1. contexte formel flou et concepts formels flous

Contexte formel flou

L'analyse de concepts formels flous repose sur la notion de contexte formel flou. Un contexte formel flou consiste en une relation binaire floue entre un ensemble d'objets et un ensemble d'attributs. Formellement, un contexte formel flou ou L-contexte est un quadruplet $K = (L, O, P, R)$ où la relation floue $R \in L^{O \times P}$ est une fonction définie : $O \times P \rightarrow L$.

Une relation floue R est représentée sous forme d'une table, généralement les lignes représentent les objets et les colonnes représentent les propriétés. Chaque cellule de la table exprime une valeur appartenant à L.

Concept formel flou

Un concept formel flou correspond à la paire $\langle \tilde{A}, \tilde{B} \rangle$ où $\tilde{A}$ est un ensemble flou ($\tilde{A} \in L^O$), $\tilde{B}$ est un ensemble flou ($\tilde{B} \in L^P$) telle que $\tilde{A}^{\Delta} = \tilde{B}$ et $\tilde{B}^{\Delta} = \tilde{A}$. On peut remarquer que $\tilde{A}^{\Delta\Delta} = \tilde{A}$ et $\tilde{B}^{\Delta\Delta} = \tilde{B}$.

L'ACF floue consiste à induire tous les concepts formels flous $\langle \tilde{A}, \tilde{B} \rangle$, L'ensemble de tous les concepts formels flous est aussi équipé d'une relation d'ordre définie comme suit : $\langle \tilde{A}_1, \tilde{B}_1 \rangle \leq \langle \tilde{A}_2, \tilde{B}_2 \rangle$ ssi $\tilde{A}_1 \subseteq \tilde{A}_2$ ou $\tilde{B}_2 \subseteq \tilde{B}_1$ où l'inclusion floue est ponctuellement définie par l'inégalité des fonctions d'appartenance respective autrement dit :

$\tilde{A}_1 \subseteq \tilde{A}_2 \Leftrightarrow \forall x \in O : (\tilde{A}_1(x) \leq \tilde{A}_2(x))$.

Il a été aussi montré que, sous certaines conditions topologiques et algébriques, l'ensemble de tous les concepts formels flous forme un treillis complet[Belohlavek 2005]. Parmi ces conditions, nous pouvons citer l'ouverture et la fermeture topologique que nous définissons ci-après.

L'AFC floue repose sur la notion de connexion de Galois floue tel qu'il existe plusieurs approches dans la littérature définissant cette connexion Ces approche utilisant généralement toutes un treillis residué $\mathbb{L}$ = (L, ∧, ∨, ∗, ⟶) comme structure algébrique pour le calcul des degrés [Belohlavek 2005]. Avant de nous étendre sur l'aspect algébrique, nous donnons les définitions suivantes.

IV.2 Rappels mathématiques

Opérateur de fermeture flou

Un opérateur de fermeture flou défini sur un ensemble $\mathcal{U}$ est une fonction $\Phi: L^{\mathcal{U}} \rightarrow L^{\mathcal{U}}$ satisfaisant ∀ U, V ∈ $L^{\mathcal{U}}$:

1. U⊆ $V \Rightarrow \phi(U) \subseteq \phi(v)$
2. U⊆ $\phi(U)$
3. $\phi\big(\phi(U)\big) = \phi(U)$

Opérateur d'ouverture flou

Un opérateur d'ouverture flou défini sur un ensemble $\mathcal{U}$ est une fonction $\Psi : L^{\mathcal{U}} \rightarrow L^{\mathcal{U}}$ satisfaisant ∀ $U, V \in L^{\mathcal{U}}$:

- U⊆ $V \Rightarrow \Psi(U) \subseteq \Psi(\mathrm{V})$
- $\Psi(U) \subseteq U$
- $\Psi\big(\Psi(\mathrm{U})\big) = \Psi(\mathrm{U})$

Monoïde

Etant donné un ensemble E et un opérateur ∗, on dit que (E, ∗, e) est un monoïde si
Les trois propriétés suivantes sont vérifiées :

- La stabilité : ∀ *x, y* ∈ *E alors* $x * y \in E$
- L'associativité : ∀ *x, y, z* ∈ *E alors* $x *(y* z) = (x * y) * z$
- L'élément neutre :∃ e ∈ *E,* ∀ *x* ∈ *E alors* $x* e = e * x = x$

Un monoïde est dit commutatif ssi ∀ *x, y* ∈ *E alors* $x * y= y * x$

Antitonie et isotonie

Soient U et V deux univers, f et g deux opérateurs de fermeture, tels que f :$\mathrm{L}^{\mathcal{U}} \rightarrow \mathrm{L}^{\mathcal{V}}$ et g :$\mathrm{L}^{\mathcal{V}} \rightarrow \mathrm{L}^{\mathcal{U}}$.∀ A, B∈ U :

Isotonie : $A \subseteq B \Rightarrow$ fog(A) ⊆ fog(B)

Antitonie: $A \subseteq B \Rightarrow \text{fog}(B) \subseteq \text{fog}(A)$

Treillis residué

Un treillis résidué décrit par une structure algébrique $\mathbb{L}$= (L, ∧, ∨, ∗, ⟶) est un treillis complet ou (L, ∗) est un monoide commutatif et la paire (∗, ⟶) verifie le principe de résiduation. L'implication ⟶ correspond soit a une S-implication soit a une R-implication.

IV.3 Approches existantes

Il existe dans la littérature plusieurs approches concernant l'AFC floue. Burusco et al [Burusco, 1994] ont été les premiers auteurs à considérer des contextes formels flous. Ils ont proposé l'utilisation des S-implication. Plusieurs approches sont venues par la suite [Burusco 2000] [Belohlavek 2000] [Belohlavek 2005]. Elles utilisent généralement toutes un treillis résidué $\mathbb{L}$ = (L, ∧, ∨, *, ⟶) comme structure algébriques pour le calcul des degrés.

IV.3.1 Méthodes à seuil(s) [MESSAI, 2009]

La méthode dite "*à seuil variable*" consiste à fixer un seuil δ au départ puis transformer le contexte flou en un contexte monovalué en enlevant toutes les valeurs inférieures à δ et en remplaçant par "×" (pour indiquer la relation entre l'objet et l'attribut correspondant de la même manière que dans le cas de l'ACF classique) celles supérieures à δ.

Exemple

	y_1	y_1	y_3
x_1	0.4	0.3	0.7
x_2	1	0.2	0.0
x_3	0.7	0.8	0.0

$\delta = 0.6$

	y_1	y_1	y_3
x_1			×
x_2	×		
x_3	×	×	

Dans la méthode dite "*à fenêtre*" [Zhou et al., 2007], deux bornes sont considérées à la fois et définies pour chaque attribut : une borne supérieure et une borne inférieure. Le contexte multivalué est transformé en contexte monovalué en enlevant les valeurs qui ne sont pas entre les deux bornes et en remplaçant celles qui le sont par "×". Dans [Quan et al., 2004] l'idée est similaire à celle de la méthode "*à seuil variable*" dans la mesure où un seuil est choisi et le contexte multivalué est transformé en contexte monovalué. La différence réside dans le fait qu'une fois le treillis de concepts construit à partir du nouveau contexte monovalué, les valeurs supérieures aux seuils fixés sont considérées à nouveau pour être utilisées comme poids des attributs dans les intensions des concepts du treillis. D'autres méthodes de discrétisation similaires sont détaillées dans [Pensa et al., 2004].

IV.3.2 Approches "mono-face" (one Sided)

Cette approche est basée sur un contexte formel (O, *P, R)* définissant une relation floue *R* entre un ensemble fini d'objets *O* et un ensemble flou d'attributs *P* appartient à l'intervalle$[0, 1]$. Dans [Yahia and Jaoua, 2001], un contexte flou est défini comme suit :

Un contexte flou est un triplet (*O, P, R*) *où X est un ensemble d'objets, Y est un ensemble d'attributs flous et I une relation binaire floue (*$R : O \times R \rightarrow [0, 1]$*). Chaque couple* $(x, y)^{\alpha} \in I$ *signifie que l'objet* $x \in O$ *a l'attribut* $y \in P$ *à un degré supérieur ou égal à* $\alpha \in [0, 1]$.

Dans le cadre de la recherche d'information [$Latiri$] considère une relation entre un ensemble fini de documents (représentent les objets) et un ensemble flou de termes de corpus (représentent les attributs) ces termes sont caractérisés par des poids qui appartiennent généralement à l'intervalle$[0, 1]$. Les concepts formels résultants d'une telle relation peuvent être interprétés comme des paires ({réponse}, {requête}) où la requête correspond à l'intension du concept tandis que les réponses correspond à son extension. [$Messai$ 2005]. Dans[ben Yahia 2007], l'auteur avait introduit dans le contexte formel une ligne supplémentaire qui correspond à une contrainte c prédéfinie par un utilisateur. L'ensemble de ces auteurs sont basés dans leurs études sur une implication de Rescher Gaïnes, cette dernière donne comme résultat 0 ou 1.

Exemple : Soit le contexte formel flou "$terme \times document$"

	T_1	T_2	T_3	T_4
D_1	o.5	1	0.7	0.5
D_2	0.6	0.7	1	0.5
D_3	1	0.9	1	0.1
D_4	1	0.9	0.9	0.1

Table 2 : Contexte formel flou "$terme \times document$"

Soit "2^o" l'ensemble des parties de O et "2^P" l'ensemble des parties de P "2^P". Les opérateurs de dérivation entre "2^o" et "2^P" sont définis comme suit :

$$A\uparrow = \{y^\alpha \in P \mid \forall x \in A,\ \alpha = min\ (I(x, y)\}$$

$$\tilde{B}\downarrow = \{x \in O \mid \forall y^\alpha \in B,\ I(x, y) \geq \alpha\}$$

Où $I(x, y) = x \rightarrow_{RG} y$

$$= \begin{cases} 1 & si\ x \leq y \\ 0 & sinon \end{cases}$$

L'opérateur $\uparrow$ ($\uparrow: 2^o \rightarrow 2^P$) appliqué à l'ensemble non flou de documents $A \in O$ permet d'obtenir un ensemble flou de termes qui pondérés par leurs degrés d'appartenance respectifs. Inversement, l'opérateur $\downarrow$ ($\downarrow: 2^P \rightarrow 2^O$) appliqué à l'ensemble flou de terme $\tilde{B} \in P$ permet d'obtenir un ensemble non flou de documents qui sont annotés par les termes de la requête. Autrement dit l'opérateur $\downarrow$ de la connexion de Galois floue peut être défini comme la fonction de correspondance qui va apparier les documents avec tous les documents pertinents avec tous les termes d'une requête floue donnée, sans pour autant évaluer leurs degrés de pertinences respectifs.[Latiri]

Dans l'exemple de contexte flou donné dans la table 2 on a :

$\{x1, x2\}\uparrow = \{a/0.5, b/0.7, c/0.7, d/0.5\}$ et $\{a/0.5, b/0.7, c/0.7, d/0.5\}\downarrow = \{x1, x2\}$.

Les opérateurs $\uparrow$ et $\downarrow$ forment une connexion de Galois entre 2^o et 2^P [YAHIA and JAOUA, 2001]

Les concepts formels flous issus du contexte formel flou donné sont organisés en ***treillis de concepts flous*** en s'appuyant sur la relation super-concept/sous-concept définie de façon similaire à celle dans le cadre de l'ACF. Le treillis de concepts flous correspondant au contexte table 1est donné dans la figure ci-dessous.

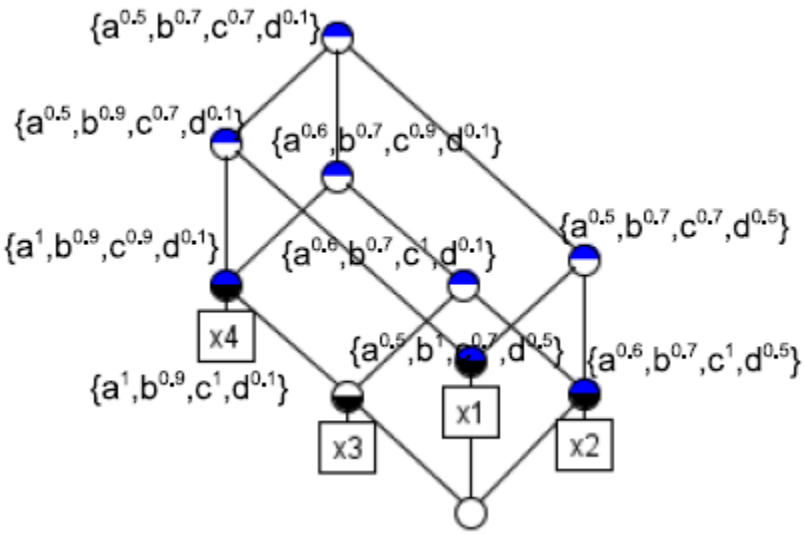


Fig1 : ***Treillis de concepts formels flous***

Dans les approches mono-face one sided, l'extension d'un concept formel flou est un ensemble d'objets alors que son intension est un ensemble flou d'attributs. Ceci exprime que les valeurs dans un contexte formel flou sont utilisées comme degrés de vérités qui s'appliquent uniquement aux attributs. Cette façon de considérer les contextes flous est à l'origine de l'appellation "mono-facette" de ces approches [Belohlavek et Vychodil, 2005].

IV.3.3 Approches générales

Dans [Belohlávek, 1999] et [Belohlavek, 2002], la définition de contexte flou est plus générale que celle dans le cas mono-facette. La généralisation consiste à considérer que l'ensemble des objets est lui aussi un ensemble flou et que l'ensemble des valeurs de vérité est un ensemble L quelconque ($L \in [0, 1]$).

IV.3.3.1 Approche de Burusco

Burusco et Fuentes Gonzales sont les premiers auteurs de l'analyse de concepts formels dans le cadre flou. Leur approche est basée principalement sur une S-implication.

Soit **L** = $\langle L, \leq, ', \oplus, 0, 1\rangle$ une structure algébrique tel que $\langle L, \leq, 0, 1\rangle$ est un treillis complet compris entre deux valeurs 0 et 1, $'$ est l'operateur de négation de Zadeh et $\oplus$ est une t-conorme (rappelons nous que une t-conorme $\oplus$ est un opérateur binaire, associatif, commutatif et il accepte ''0'' comme élément neutre) pour les deux ensemble flous $\widetilde{A} \in O$ (*O* est l'ensemble d'objets) et $\widetilde{B} \in P$ (*P* est l'ensemble d'attributs) l'auteur avait défini alors les deux opérateurs de dérivation $\uparrow : L^O \rightarrow L^P$ et $\downarrow : L^P \rightarrow L^O$ par :

$$\tilde{A}^{\uparrow}(y) = \bigwedge_{x \in X} (\tilde{A}(x)' \oplus R(x,y))$$
$$\tilde{B}^{\downarrow}(x) = \bigwedge_{y \in Y} (\tilde{B}(y)' \oplus R(x,y))$$

Un treillis de concepts formels flous $\mathcal{L}(\mathcal{K})$ est décri par :

$$\mathcal{K} = \{\langle \widetilde{A}, \widetilde{B} \rangle \in L^O \times L^P / \tilde{A}^{\Delta} = \widetilde{B} \ et \tilde{B}^{\Delta} = \widetilde{A}, \widetilde{A} \in O \ et \widetilde{B} \in P \}$$

Et $\preccurlyeq$ definit un ordre partiel dans $\mathcal{K}$ entre ces concepts formels flous tel que :
$\langle \widetilde{A}_1, \widetilde{B}_1 \rangle \preccurlyeq \langle \widetilde{A}_2, \widetilde{B}_2 \rangle$ ssi $\widetilde{A}_1 \subseteq \widetilde{A}_2$ (ou $\widetilde{B}_2 \subseteq \widetilde{B}_1$).

Dans le papier [Burusco 1998] l'auteur a démontré que si l'implication utilisée vérifiée la condition : $a \leq c \Longrightarrow a \rightarrow b \geq c \rightarrow b$ alors les deux propriétés suivante sont vérifiées

- $\widetilde{X} \leq \tilde{Y} \Rightarrow \tilde{X}^{\Delta} \geq \tilde{Y}^{\Delta}, \forall \tilde{X}, \tilde{Y} \in O$
- $\tilde{A} \leq \tilde{B} \Rightarrow \tilde{A}^{\Delta} \geq B^{\Delta}, \forall \tilde{A}, \tilde{B} \in P$

Après l'approche classique de Wille [Wille 1982] sur l'analyse de concepts formels, Burusco s'inscrit parmi les premiers auteurs qui ont ouvert la voie sur une nouvelle théorie de l'analyse de concepts formels floue.
Néanmoins cette approche a été critiquée par Belohlavek [Belohlavek 2005] sous une confirmation sur certaines propriétés utilisées par Burusco et qui ne vérifiées pas la propriété de fermeture. (Autrement dit la propriété d'extensivité « *tout ensemble est inclus ou égal à sa fermeture* » n'est pas vérifiée par certaines S-implication.)

IV.3.3.2 Approche Belohlavek

Pollant [pollant, 1997] et Belohlavek [Belohlavek, 1998] sont parmi les premiers auteurs qui avaient proposé d'utiliser un treillis résidué comme une structure algébrique, cependant leur approche est basée sue une R-implication.

Pour assurer la généricité de l'approche, Belohlavek propose un cadre formel pour l'ensemble L qui soit une généralisation du cas binaire. Et comme la généralisation vise à étendre les résultats de l'ACF aux contextes flous, Belohlavek propose de considérer l'equivalent des opérateurs de la logique classique dans la logique floue.

Le point de départ consiste à considérer que :

- l'ensemble $(L, \leq)$ est partiellement ordonné et a respectivement 0 et 1 comme plus petit et plus grand élément. $(L, \bigwedge, \bigvee, 0, 1)$ est donc un treillis complet ayant 1 comme top et 0 comme bottom.
- L'opérateur de conjonction, entre les éléments de L, qui généralise la conjonction classique est noté par $\otimes$. Cet opérateur vérifie : $1 \otimes 1 = 1$ et $1 \otimes 0 = 0 \otimes 1 = 0 \otimes 0 = 0$.
- Tout comme la conjonction classique, l'opérateur $\otimes$ est commutatif et associatif et a 1 comme élément neutre. Par définition, les propriétés précédentes signifient que $(L, \otimes, 1)$ est un monoïde commutatif.
- La généralisation de la règle d'inférence *Modus ponens* dans le cas de la logique floue aboutit à la définition de l'opérateur d'implication "$\longrightarrow$" qui étend l'implication de la logique classique "$\Rightarrow$".
- Les deux opérateurs $\otimes$ et $\longrightarrow$ sont dits opérateur de conjonction flou et opérateur d'implication flou, respectivement, et vérifient pour tout triplet $a, b, c \in L$,

$$a \otimes b \leq c \text{ si et seulement si } b \leq a \longrightarrow c.$$

Étant donné toutes ces propriétés, la structure algébrique formée par l'ensemble des valeurs de vérité, L, et les opérateurs définis sur L est $\mathbf{L} = (L, \bigwedge, \bigvee, \otimes, \longrightarrow, 0, 1)$. **L** est une algèbre appelée ***treillis résidué*** et l'ensemble L est appelé ***support*** de **L**.

Étant donné une structure **L** de valeurs de vérité, un contexte flou, appelé aussi **L**-contexte par Belohlavek, est défini comme suit :

Contexte flou *:* un ***contexte flou*** ou ***L****-contexte* est un quadruplet (L, O, P, $\mathbf{R}$) où O est un ensemble d'objets flous, P est un ensemble d'attributs flous et R une relation binaire floue ($R : O \times P \rightarrow L$) dite aussi **L**-relation entre O et P. La valeur $R(x, y) \in L$ est interprétée comme le degré de vérité pour lequel l'objet $x \in O$ possède l'attribut $y \in P$.

Les opérateurs de dérivation sont définis entre les ensembles flous d'objets, notés par L^X, et les ensembles flous d'attributs, notés par L^Y, comme suit pour tous sous ensembles flous $\tilde{A} \in L^X$ et $\tilde{B} \in L^Y$:

$$A\uparrow(y) = \bigwedge_{x \in X} (\tilde{A}(x) \rightsquigarrow R(x,y))$$

$$\tilde{B}\downarrow(x) = \bigwedge_{y \in Y} (\tilde{B}(y) \rightsquigarrow R(x,y))$$

Tel que :

$\rightsquigarrow$: une R-implication.$\tilde{A}^{\uparrow}$

$\tilde{A}\uparrow(y)$: le degré de vérité de "l'attribut y est partagé par tous les objets de $\tilde{A}$"

$\tilde{B}\downarrow(x)$: le degré de vérité de "l'objet x possède tous les attributs de $\tilde{B}$".

Concept flou : *u*n concept flou ou **L**-concept est un couple $\langle \tilde{A}, \tilde{B} \rangle$ tel que $\tilde{A} \in L^X$, $\tilde{B} \in L^Y$, $\tilde{A}\uparrow = \tilde{B}$ et $\tilde{B}\downarrow = \tilde{A}$. $\tilde{A}$ est l'extension du **L**-concept et $\tilde{B}$ est son intension.

Les **L**-concepts d'un contexte flou sont organisés par l'intermédiaire d'une relation d'ordre partiel définie de la même manière que dans le cas de l'ACF en remplaçant les inclusions entre ensemble d'éléments discrets par les inclusions entre ensembles flous. Étant donné deux **L**-concepts, $(\tilde{A}_1, \tilde{B}_1)$ et $(\tilde{A}_2, \tilde{B}_2)$, $(\tilde{A}_1, \tilde{B}_1)$ est dit un sous-concept de $(\tilde{A}_2, \tilde{B}_2)$ (auquel cas $(\tilde{A}_2, \tilde{B}_2)$ est dit super-concept de $(\tilde{A}_1, \tilde{B}_1)$) et on note $(\tilde{A}_1, \tilde{B}_1) \leq (\tilde{A}_2, \tilde{B}_2)$ si et seulement si $\tilde{A}_1 \subseteq \tilde{A}_2$ (ou de manière équivalente $\tilde{B}_2 \subseteq \tilde{B}_1$).

L'ensemble des **L**-concepts d'un **L**-contexte organisé par l'intermédiaire de la relation d'ordre partiel "$\leq$" forme un treillis de concepts flous ou **L**-treillis noté par $\mathrm{B}(\mathcal{K})$.

Les **L**-concepts et le **L**-treillis correspondant à un **L**-contexte dépendent du choix de la structure **L**.

IV.3.3.3 Approche de Georgesco et Popescu

Inspiré de l'approche de Belohlavek, Georgesco [Georgesco 2004] avait étendu l'analyse de concepts formels floue pour considérer le cas d'une algèbre non commutative. Dans cette approche, on considère " $*$ " un opérateur de conjonction non commutatif et deux

implications résiduées : " ↬ " une implication résiduée à gauche vérifie la propriété : a≤ $b \leftarrow c$ ssi b∗a≤ c et " ⇝ " une implication résiduée à droite vérifie la propriété : a≤ $b \rightsquigarrow c$ ssi a∗b≤ c. Pour un ensemble d'objets flou $\widetilde{A}$ on définit deux ensembles de propriétés en appliquant deux opérateurs de dérivation $\tilde{X}^{\uparrow}$ et$\tilde{X}^{\Uparrow}$, ces deux opérateurs sont définis comme suit :

$\tilde{A}^{\Uparrow}(x) = \overline{\tilde{A}^{\uparrow}(x)} = \overline{V_{x\in O}\tilde{A}(x) * \mathcal{R}(x,a)} = \wedge_{x\in O}\,\overline{\tilde{A}(x) * \mathcal{R}(x,a)} = \wedge_{x\in O}\,\tilde{A}(x) * \bar{\mathcal{R}}(x,a)$

De même pour un ensemble de propriétés flou $\tilde{B}$ on définit deux ensembles d'objets $\tilde{A}^{\downarrow}$ et $\tilde{A}^{\Downarrow}$. Ces opérateurs sont donnés comme suit :

$\tilde{B}^{\Downarrow}(x) = \tilde{B}^{\downarrow} = \wedge_{a\in P}\,\mathcal{R}(x,a) \rightarrow \overline{\tilde{B}(a)} = \wedge_{a\in P}\,\tilde{B}(a) \rightarrow \bar{\mathcal{R}}\,(x,a)$

Comparé au cas commutatif, on aura $\tilde{A}^{\uparrow\Downarrow\uparrow} = \tilde{A}^{\uparrow}$ et$\tilde{B}^{\Downarrow\uparrow\Downarrow} = \tilde{B}^{\Downarrow}$, résultant ainsi deux opérateurs de fermeture (↑⇓, ⇓↑) et (⇑↓, ↓⇑), alors on dit que l'ensemble d'objets $\tilde{X}$ est fermé par rapport à (↑⇓, ⇓↑) et l'ensemble des propriétés fermé par rapport à (⇑↓, ↓⇑). Selon l'auteur un concept formel dans ce cas étudié, doit avoir une extension et deux intentions décrivant la même extension, alors un concept formel est décrit par un triplet

$(\tilde{A}, \tilde{B}, \tilde{C})$ tel que$\tilde{A}^{\uparrow} = \tilde{A}$, $\tilde{B}^{\Downarrow} = \tilde{A}$, $\tilde{A}^{\Uparrow} = \tilde{C}$ et $\tilde{C}^{\downarrow} = A$.

Un treillis résidué est une structure algébrique tel que :

- $\langle L, \mathrm{V}, \Lambda, 0,1\rangle$ est un treillis limité par les deux valeurs 0 et 1.
- $\langle L, *, 1\rangle$ est un monoïde.
- Deux implications l'une résiduée à gauche (noté par « ↬ ») et l'autre résidué à droite (noté par « ⇝ »).

Dans un treillis résidué complet on site quelques propriétés suivantes [Georgescu 2004] :

- $x\leq a$ ssi $x \rightsquigarrow a = 1$ et ssi $x \leftarrow a = 1$
- $x \rightsquigarrow 1 = x \leftarrow 1 = 1; 1 \rightsquigarrow x = 1 \rightsquigarrow x = x$
- ⇝, ↬ sont des antitone sur le 1^{er} argument et isotone sur le second argument.
- $*$ est isotone.

Conclusion

L'AFC permet de construire une structure mathématique qui sert de support à la découverte de connaissance. L'AFC est un formalisme bien établi pourvu d'outils pour la construction, visualisation et l'interprétation de treillis de concepts.

Dans de nombreux cas les données ne sont pas binaires mais complexes et hétérogènes ainsi dans le chapitre présenté ci-dessus nous avons présente l'extension de l'analyse de concepts formels au cas flou qui repose sur une connexion de Galois floue, pour laquelle nous avons donné les différentes définitions et approches existantes dans la littérature.

Chapitre V : Contribution

Sommaire

Introduction à la contribution 57

V.1 fondements théoriques 57

V.2 présentation de la contribution 59

V.2.1 Construction de l'ensemble des concepts formels 59

V.2.2 Ordonnancement linéaire du treillis des CFs 64

V.4 Présentation de l'application 66

V.4.1 Environnement technique de développement 66

V.4.2 Technique d'implémentation 69

Conclusion 71

Introduction à la contribution

L'analyse de concepts formelle à été centré autour de la notion de concepts formels. Ces concepts formels sont extraient à partir d'une relation binaire (contexte formel) entre un ensemble d'objet et un ensemble d'attributs.

Comme nous l'avons vu précédemment plusieurs approches ont été proposé dans la littérature pour étendre cette relation binaire à une relation flou, c.à.d. que la relation entre un objet et une propriété est exprimé par un degré appartenant à l'intervalle$[0, 1]$.

Actuellement, la tendance générale de l'AFC floue consiste à utiliser une algèbre résiduée pour maintenir les propriétés de fermeture.

Après avoir présenté dans les chapitres précédents les notions nécessaires à l'accomplissement de notre travaille, ce chapitre est maintenant l'occasion de présenter notre contribution à savoir « l'ordonnancement et classement (seuillage) d'un treillis en analyse de concepts formels floue ».

Dans le présent chapitre, nous commencent par donner des fondements théoriques pour spécifier le choix de l'implication utiliser pour l'extraction de l'ensemble des concepts formels flous, ensuite viens la procédure utilisée pour la construction de ces derniers.

V.1 fondements théoriques

Pour la génération de concepts formels flous, on applique l'implication de Gödel au contexte considéré

Afin de justifier le choix de l'implication utilisée, nous donnons ces quelques propriétés.

Etant donné un contexte R défini comme suit :

	y_1	y_2	y_3	y_4
x_1	0.9	0.7	0.6	0.2
x_2	0.7	0.4	0.7	0.8
x_3	0.6	0.8	1.0	0.0

Table V.1 : Contexte formel flou

L'implication de Gödel est une implication résiduée, elle est définie comme suit :

$$x \longrightarrow y = \begin{cases} 1 \; si \; x \leq y \\ y \; sinon \end{cases}$$

Propriété 1 :

Soit K= (L, O, P, R) un L contexte tel que L= $[0, 1]$ pour l'implication de Gödel l'ensemble des concepts formels est fini.

Démonstration :

Soit *CF* l'ensemble des concepts formels, q_{min} est l'élément le plus petit de R soit (A, B) ∈ CF sachant que B= A^{Δ} alors :

$$A^{\Delta} = \begin{cases} \text{soit } 1 \\ \text{soit } q \end{cases}$$

Si $A^{\Delta} = 1$ alors $A^{\Delta\Delta} = 1^{\Delta} = q_{min} \in R$

Si $A^{\Delta} = q$ alors $A^{\Delta} = q^{\Delta} = \begin{cases} 1 \\ q \end{cases}$, donc A∈ R

Alors on peut dire que pour tout (A, B)∈ *CF*, A∈ R or R est un ensemble fini alors CF est fini

Propriété 2

(q_{min}, 1) et (1,q_{min}) sont des concepts formels quel que soit le contexte.

Démonstration

Nous avons : $X^{\Delta} = \{a \in A \setminus \forall x \in X(x \in X \rightarrow xRa)\}$

$1^{\Delta} = \bigwedge(1 \rightarrow 1Ra) = \bigwedge q = q_{\min}$

$1^{\Delta\Delta} = (q_{\min})^{\Delta} = 1$ (car $q_{\min} \leq q$) alors (1, $q_{\min}$) est une fermé

De même : $(q_{\min})^{\Delta} = 1$ et $(q_{\min})^{\Delta\Delta} = 1^{\Delta} = q_{\min}$ alors $(q_{\min}, 1)$ est un fermé.

V.2 présentation de la contribution

V.2.1 Construction de l'ensemble des concepts formels

Soit un *L* contexte formel (*L*, *O*, *P*, ***R***) ou *O* est l'ensemble des objets, *P* est l'ensemble des propriétés et ***R*** une relation floue entre objet\ propriété.

Pour un ensemble d'objets$\tilde{X}$, nous définissons l'ensemble $\tilde{X}^{\Delta}$ d'attributs satisfait par tous les objets de $\tilde{X}$, de meme pour l'ensemble de propriétés $\tilde{Y}$, nous définissons l'ensemble $\tilde{Y}^{\Delta}$d'objet satisfaisant toutes les propriétés de $\tilde{X}$ comme suit :

$$\tilde{X}^{\Delta}(y) = \bigwedge_{x \in O} (\tilde{X}(x) \longrightarrow_{G\ddot{o}} R(x, y))$$

$$\tilde{Y}^{\Delta}(x) = \bigwedge_{y \in P} (\tilde{Y}(y) \longrightarrow_{G\ddot{o}} R(x, y)$$

La paire duale d'opérateur$\langle (.)^{\Delta}, (.)^{\Delta} \rangle$ forme une connexion de Galois qui permet d'induire des concepts formels

V.2.1.1 Formulation mathématique :

Considérant le contexte formel représenté dans la table V.1

Soit la R implication de G*ö*del définie comme suit :

$$x \longrightarrow_{G\ddot{o}} y = \begin{cases} 1 \; si \; x \leq y \\ y \; sinon \end{cases}$$

Soit CF l'ensemble des concepts formel à construire

Tout concept formel $\langle \tilde{X}, \tilde{Y} \rangle$ est tel que :

1) $\tilde{X} \in c_1 \times c_2 \times \ldots \times c_n$

$c_1 = \{0.9, 0.7, 0.6\}$

$c_2 = \{0.7, 0.4, 0.8\}$

$c_3 = \{0.6, 0.7, 1.0\}$

$c_4 = \{0.2, 0.8, 0.0\}$

2) $\tilde{Y} \in L_1 \times L_2 \times \ldots \times L_m$

$L_1 = \{0.9, 0.7, 0.6, 0.2\}$

$L_2 = \{0.7, 0.4, 0.7, 0.8\}$

$L_3 = \{0.6, 0.8, 1.0, 0.0\}$

Pour tout $\tilde{Y} \in L_1 \times L_2 \times \ldots \times L_m$ soit à calculer

$$\tilde{Y}^{\Delta}(x) = \bigwedge_{y \in p} (Y(y) \longrightarrow_{G\ddot{o}} R(x, y))$$

$$= \tilde{X} \in c_1 \times c_2 \times \ldots \times c_n$$

Ensuite pour chaque $\tilde{X}$ obtenu on calcul

$$\tilde{X}^{\Delta}(y) = \bigwedge_{x \in O} (\tilde{X}(x) \longrightarrow_{G\ddot{o}} R(x, y))$$

$$= \tilde{Y'} \in L_1 \times L_2 \times \ldots \times L_m$$

Si $\tilde{Y} = \tilde{Y'} \Longrightarrow CF \ni \langle \tilde{X}, \tilde{Y} \rangle$ sinon CF$\nexists \langle \tilde{X}, \tilde{Y} \rangle$

Exemple :

Soit $\tilde{Y} = \{y_1 \setminus 0.6,\ y_2 \setminus 0.8, y_3 \setminus 0.6,\ y_4 \setminus 0.0\}$

$$\tilde{Y}(x_1) \longrightarrow_{G\ddot{o}} R(x_1, y) = \wedge \begin{cases} 0.6 \longrightarrow 0.9 = 1 \\ 0.8 \longrightarrow 0.7 = 0.7 \\ 0.6 \longrightarrow 0.6 = 1 \\ 0.0 \longrightarrow 0.2 = 1 \end{cases}$$

= 0.7

$$\tilde{Y}(x_2) \longrightarrow_{G\ddot{o}} R(x_2, y) = \bigwedge \begin{cases} 0.6 \longrightarrow 0.7 = 1 \\ 0.8 \longrightarrow 0.4 = 0.4 \\ 0.6 \longrightarrow 0.7 = 1 \\ 0.0 \longrightarrow 0.8 = 1 \end{cases}$$

= 0.4

$$\tilde{Y}(x_3) \longrightarrow_{G\ddot{o}} R(x_3, y) = \bigwedge \begin{cases} 0.6 \longrightarrow 0.6 = 1 \\ 0.8 \longrightarrow 0.8 = 1 \\ 0.6 \longrightarrow 1.0 = 1 \\ 0.0 \longrightarrow 0.0 = 1 \end{cases}$$

= 1.0

$\tilde{X} = \{x_1 \setminus 0.7, x_2 \setminus 0.4, x_3 \setminus 1.0\}$

Soit à calculer $\tilde{X}^{\Delta}$

$$\tilde{X}(y_1) \longrightarrow R(x, y_1) = \bigwedge \begin{cases} 0.7 \longrightarrow 0.9 = 1 \\ 0.4 \longrightarrow 0.7 = 1 \\ 1.0 \longrightarrow 0.6 = 0.6 \end{cases}$$

=0.6

$$\tilde{X}(y_2) \longrightarrow R(x, y_2) = \bigwedge \begin{cases} 0.7 \longrightarrow 0.7 = 1 \\ 0.4 \longrightarrow 0.4 = 1 \\ 1.0 \longrightarrow 0.8 = 0.8 \end{cases}$$

=0.8

$$\tilde{X}(y_3) \longrightarrow R(x, y_3) = \bigwedge \begin{cases} 0.7 \longrightarrow 0.6 = 0.6 \\ 0.4 \longrightarrow 0.7 = 1.0 \\ 1.0 \longrightarrow 1.0 = 1.0 \end{cases}$$

=0.6

$$\tilde{X}(y_1) \longrightarrow R(x, y_1) = \bigwedge \begin{cases} 0.7 \longrightarrow 0.9 = 1 \\ 0.4 \longrightarrow 0.7 = 1 \\ 1.0 \longrightarrow 0.6 = 0.6 \end{cases}$$

=0.6

$$\tilde{X}(y_4) \longrightarrow R(x, y_4) = \bigwedge \begin{cases} 0.7 \longrightarrow 0.2 = 0.2 \\ 0.4 \longrightarrow 0.8 = 1.0 \\ 1.0 \longrightarrow 0.0 = 0.0 \end{cases}$$

=0.0

$\tilde{Y}' = \{y_1 \setminus 0.6, y_2 \setminus 0.8, y_3 \setminus 0.6, y_4 \setminus 0.0\}$

$\tilde{Y} = \tilde{Y}' \Leftrightarrow \tilde{Y} = \tilde{X}^{\Delta} \Leftrightarrow \tilde{Y} = (\tilde{Y}^{\Delta})^{\Delta} \Leftrightarrow CF \ni \langle \{\dots\}, \{\dots\} \rangle$

V.2.1.2 Algorithmique

Algorithme génération concept formel
Entrée : Le contexte flou K= (L, O, P, R) Sortie : CF ensemble de concepts formel flou Début Insert into CP $\boldsymbol{L_1 \times L_2 \dots \times L_m}$ /*calculer $\tilde{\boldsymbol{Y}}^{\Delta}$ et $\tilde{\boldsymbol{Y}}^{\Delta\Delta}$ */ Pour chaque Y∈ $\boldsymbol{CP}$ $\boldsymbol{Y^{\Delta}}$ = **calculer** (**Y**); $\boldsymbol{Y^{\Delta\Delta}}$ = **calculer** ($\mathbf{Y}^{\Delta}$); Fin pour

/ *sélectionner les concepts formels flous*/

Insert into CF

Select Distinct Y, $\boldsymbol{Y}^{\Delta\Delta}$ from CP

where (Y=$\boldsymbol{Y}^{\Delta\Delta}$);

Calculer l'opérateur Δ

```
Procédure calculer (A);
Variable e, min: réelles
Début
    Pour i ← 1 à NC faire
        min ← 1;
        Pour j ← 1 à NC faire
            Si A[i] ≤ K[i, j]
                e ← 1
            Sinon
                e ← K[i, j]
            Fin Si
        Fin Pour
        Si min > e alors  min ← e;
    Fin Si
      A^Δ[i] ← P;
    Fin Pour
    Retourner (A^Δ)
Fin.
```

V.2.2 Ordonnancement linéaire du treillis des CFs

Une fois l'ensemble des concepts formels flou est induis, en sachant que chaque concept est un ensemble flou, une fonction d'approximation est appliquée à chaque concept formel par rapport à un concept formel cible choisi par l'utilisateur. Cette fonction permet d'établir un ordre linéaire du treillis sur tout le voisinage du concept formel cible.

Cet ordre linéaire sera établi par la projection de la distance de Hausdroff qui est défini comme suit :

$$\mathrm{H}(A,B) = \max\{\max_{a\in A}(\min_{b\in B} d(a,b)\quad ,\max_{b\in B}(\min_{a\in A} d(a,b)\quad \}$$

Après le choix de la distance, la fonction d'approximation est la suivante:

- L'utilisateur choisi un concept formel cible à partir du quel il souhaite avoir la structure linéaire du treillis
- L'utilisateur fixe un seuil ε
- Former l'ensemble E contenant tous les concepts formels flous dont la distance soit inférieur ou égale à ε
- E étant le résultat attendu à savoir la pseudo linéarisation du treillis de concepts formels flous.

V.2.2.1 Formulation mathématique

CF : ensemble des concepts formels

$\langle \tilde{A}, \tilde{B} \rangle$: concept cible.

E : ensemble du voisinage de $\langle \tilde{A}, \tilde{B} \rangle$

$\forall \langle \widetilde{\mathrm{X}}_{\mathrm{i}}, \widetilde{\mathrm{Y}}_{\mathrm{i}} \rangle \in CF, H_i = H\langle \tilde{B}, \tilde{Y}_i \rangle$

Soit $\max = \max H_i$

$\forall\ H_i, H'_i = \frac{\mathrm{H_i}}{\max}$

$\exists \ni\ H'_i$ tel que $H'_i \leq \varepsilon$

V.2.2.2 Algorithmique

Algorithme ordonnancement des concepts

Entrée : CF ensemble de concepts formels flous

$\langle \tilde{A}, \tilde{B} \rangle$ Concepts cible

ε *réel*

Sortie : E ensemble du voisinage de $\langle \tilde{A}, \tilde{B} \rangle$

Début

max= 0;

Pour chaque $\langle \tilde{X}_i, \tilde{Y}_i \rangle \in CF\ faire$

H[i] = $calcul\ (\tilde{B}, \tilde{Y}_i)$;

Si H[i] > max alors

max= H[i]

Fin Si

Fin Pour

Pour (i= 0; $i < nbrconcept\ formel; i++$) faire

H[i] = $H[i]$ max ;

Fin Pour

Trier (H) ;

While ($H_i \leq \varepsilon$)

Insert H_i to E

Fin While

Fin.

Calcul de la distance de Hausdorff

```
Procedure calcul(conceptCible, X)
Variable Min, d : réel ;
Debut
    Min = 0 ;
    Pour i← 1 à tailleConcept  faire
        d= calculd(ConceptCible[i], X) ;
        Si (d < min) alors
            Min = d;
        finSi
    finPour
    retourner(Min) ;
Fin.

Procedure calculd(c, X)
Variable Max , b: réel ;
Debut
    Max = 1 ;
    Pour i← 1 à tailleConcept  faire
        b= |c − X[i]| ;
        Si ( b> max)
                max =b;
        finSi
    finPour
    retourner(Min) ;
Fin.
```

V.4 Présentation de l'application

V.4.1 Environnement technique de développement

Notre prototype de système a été réalisé sur un PC ayant les caractéristiques suivantes :

- Un microprocesseur Pentium 4, fréquence de l'horloge 2.00 GHz.
- mémoire RAM de 3 Go.
- disque dur 250 Go.
- Ecran LCD couleur 15 pouces.
- Avec Windows Vista professionnel comme système d'exploitation, eclipse comme environnement de développement, et Java comme langage de programmation.

V.4.1.1 Présentation du Langage de programmation utilisé (java)

Java est un langage de programmation informatique orienté objet crée par James Gosling et Patrick Naughton, employees de Sun Microsystems, avec le soutien de Bill Joy (cofondateur de SunMicrosystems en 1982), présenté officiellement le 23 mai 1995 au SunWorld

Java possède un certain nombre de caractéristiques qui ont largement contribué à son énorme succès :

- **Java est interprété** : Le source est compilé en pseudo code ou byte code puis exécuté par un interpréteur Java « la Java Virtual Machine (**JVM**) ». Ce concept est à la base du slogan de Sun pour Java : **WORA** (Write Once, Run Anywhere : écrire une fois, exécuter partout). En effet, le byte code, s'il ne contient pas de code spécifique à une plate-forme particulière peut être exécuté et obtenir quasiment les même résultats sur toutes les machines disposant d'une JVM.
- **Java est portable (Il est indépendant de toute plate-forme) :** il n'y a pas de compilation spécifique pour chaque plate forme. Le code reste indépendant de la machine sur laquelle il s'exécute. Il est possible d'exécuter des programmes Java sur tous les environnements qui possèdent une Java Virtual Machine. Cette indépendance est assurée au niveau du code source grâce à Unicode et au niveau du byte code.
- **Java est orienté objet** : Comme la plupart des langages récents, Java est orienté objet. Chaque fichier source contient la définition d'une ou plusieurs classes qui sont utilisées les unes avec les autres pour former une application. Java n'est pas complètement objet car il définit des types primitifs (entier, caractère, flottant, booléen,...).
- **Java est simple :** Le choix de ses auteurs a été d'abandonner des éléments mal compris ou mal exploités des autres langages tels que la notion de pointeurs (pour éviter les incidents en manipulant directement la mémoire), l'héritage multiple et la surcharge des opérateurs, ...

- **Java est fortement typé :** Toutes les variables sont typées et il n'existe pas de conversion automatique qui risquerait une perte de données. Si une telle conversion doit être réalisée, le développeur doit obligatoirement utiliser un cast ou une méthode statique fournie en standard pour la réaliser.
- **Java assure la gestion de la mémoire :** L'allocation de la mémoire pour un objet est automatique à sa création et Java récupère automatiquement la mémoire inutilisée grâce au « **garbage collector** » qui restitue les zones de mémoire laissées libres suite à la destruction des objets.

V.4.1.2 Présentation de l'environnement de développement utilisé (eclipse)

Eclipse est un environnement de développement libre. Le projet a été initié par IBM pour remplacer, en utilisant Java, l'IDE *Visual Age*, basé sur Smalltalk. Dès l'origine du projet, IBM a voulu offrir une solution multiplateforme, pouvant être exécutée sur les différents systèmes d'exploitation de ses clients. De même le projet s'est voulu extensible par le biais de plugins.

Le nom serait un jeu de mots : le créateur de Java est Sun (en français : « soleil »), concurrent qu'IBM semble vouloir « éclipser ».

En juin 2007, parait la version 3.3 appelée ***Europa.*** 310 développeurs répartis dans 19 pays ont écrit les 17 millions de lignes de codes qui la composent. Cette version porte officiellement 21 projets.

La version 3.5 appelée ***Galileo***, parue en juin 2009 porte en outre cette fois-ci 33 projets internes allant de modélisation à l'analyse des performances. Plus de 380 membres appartenant à 44 organisations différentes ont contribué à l'élaboration de cette version comportant 24 millions de lignes de code. Six versions de test et d'évaluation, appelées *Milestone* étaient déjà parues, la première en août 2008, la seconde en septembre 2008, puis Novembre 2008 et Décembre 2008 et enfin la dernière en février 2009 puis Mars 2009.

La version 3.6, appelée ***Helios***, sortie en juin 2010, est déclinée en 12 paquetages, selon les usages pour le développement (C/C++, JavaScript, PHP,...). Elle s'appuie sur 77 projets. Les principales évolutions relevées comprennent : un nouveau paquetage adapté au développement en C/C++) dans l'environnement système d'exploitation Linux, une

nouvelle gestion des plugins, le support de Git, le support des dernières évolutions Java (dont Servlet 3.0, JPA 2.0, JSF 2.0,EJB 3.1), une amélioration du support JavaScript (en mettant en place un cadre pour intégrer des débogueurs tels que Rhino ou Firebug), Eclipse Xtext 1.0, (environnement pour créer des langages spécifiques -domain specific languages, DSL-), une nouvelle version de Acceleo 3.0 (OMG Model-to-text -MTL-).

La version actuelle, appelée ***Indigo***, est sortie le 22 juin 2011. Elle s'inscrit dans le plan de développement (Release) suivi depuis juin 2007, qui prévoit une version annuelle en juin et deux versions intermédiaires, positionnées en septembre et février.

V.4.2 Technique d'implémentation

Interface graphique

Pour le développement de notre interface graphique nous avons utilisé l'API SWING.

Swing propose de nombreux composants dont certains possèdent des fonctions étendues, une utilisation des mécanismes de gestion d'événements performants et une apparence modifiable à la volée (une interface graphique qui emploie le style du système d'exploitation Windows ou Motif ou un nouveau style spécifique à Java nommé Metal).
Les composants Swing forment une nouvelle hiérarchie parallèle à celle de l'AWT[1]. L'ancêtre de cette hiérarchie est le composant JComponent. Presque tous ces composants sont écrits en pur Java : ils ne possèdent aucune partie native sauf ceux qui assurent l'interface avec le système d'exploitation : JApplet, JDialog, JFrame, et JWindow. Cela permet aux composants de toujours avoir la même apparence quelque soit le système sur lequel l'application s'exécute.

SWING possède plusieurs package, parmi ces package nous avons utilisé :

- Javax.swing.JFram : Classe pour la création de fenêtre
- javax.swing.event : Classes et interfaces pour les événements spécifiques à Swing.
- javax.swing.table : Classes définissant un composant pour la présentation de données sous forme de tableau
- javax.swing.text Classes et interfaces de bases pour les composants manipulant du texte
- javax.swing.JButton : Classe pour la création de boutons

L'application

Pour le développement de notre application nous avons utilisé ce qu'on appelle les « collections ». Les collections sont des objets qui permettent de gérer des ensembles d'objets. Ces ensembles de données peuvent être définis avec plusieurs caractéristiques : la possibilité de gérer des doublons, de gérer un ordre de tri, etc. ...

Pour combler le manque d'objets adaptés, la version 2 du J.D.K. apporte un framework complet pour gérer les collections. Cette bibliothèque contient un ensemble de classes et interfaces. Elle fourni également un certain nombre de classes abstraites qui implémentent partiellement certaines interfaces.

Les interfaces à utiliser par des objets qui gèrent des collections sont :

- Collection : interface qui est implémentée par la plupart des objets qui gèrent des collections
- Map : interface qui définit des méthodes pour des objets qui gèrent des collections sous la forme clé/valeur
- Set : interface pour des objets qui n'autorisent pas la gestion des doublons dans l'ensemble
- List : interface pour des objets qui autorisent la gestion des doublons et un accès direct à un élément
- SortedSet : interface qui étend l'interface Set et permet d'ordonner l'ensemble
- SortedMap : interface qui étend l'interface Map et permet d'ordonner l'ensemble

Le framework propose plusieurs objets qui implémentent ces interfaces et qui peuvent être directement utilisés :

- TreeSet : arbre qui implémente l'interface SortedSet
- ArrayList : tableau dynamique qui implémente l'interface List
- LinkedList : liste doublement chaînée (parcours de la liste dans les deux sens) qui implémente l'interface List
- HashMap : HashTable qui implémente l'interface Map

- TreeMap : arbre qui implémente l'interface SortedMap

Le framework définit aussi des interfaces pour faciliter le parcours des collections et leur tri :

- Iterator : interface pour le parcours des collections
- ListIterator : interface pour le parcours des listes dans les deux sens et modifier les éléments lors de ce parcours
- Comparable : interface pour définir un ordre de tri naturel pour un objet
- Comparator : interface pour définir un ordre de tri quelconque

Conclusion

Cette partie à été l'occasion de présenter notre contribution ainsi que l'aspect pratique de notre système .Nous avons montré le fonctionnement principale de l'application en donnant l'aspect mathématique et algorithmique suivi pour la construction de l'ensemble des concept formels flous à partir d'un contexte formel flou saisi par l'utilisateur et pour linéarisation du treillis.

Nous estimons avoir réalisé un logiciel qui répond aux objectifs que nous sommes fixés

Conclusion et perspectives

Les travaux effectués dans le cadre de ce mémoire porte sur l'analyse formelle de concepts flous, plus précisément il s'agit de la linéarisation partielle d'un treillis de concepts formels flous qui possède une structure peu délicate à manipuler et à interpréter. La linéarisation est établie sur tout le voisinage d'un concept formel cible choisi par l'utilisateur, cette approche est caractérisée par deux apports par rapport à l'état de l'art.

Le premier apport consiste à étendre l'analyse formelle de concepts pour le traitement de données complexes et hétérogènes. Nous avons mis en évidence l'intérêt de plusieurs extensions de l'AFC classique concernant la prise en compte de la gradualité des propriétés (éventuellement la gradualité de règles liant ces propriétés) et de l'incertitude relative à la possession des propriétés par des objets.

En effet, il existe dans la littérature plusieurs approches définissant la connexion de Galois flous. Certain de ces approches sont basées sur la théorie des ensembles flous [Zadeh, 1965] ou la notion de satisfaction est maintenant représentée par un degré d appartenant à l'intervalle [0 , 1], ces approches utilisent generalement toutes un treillis résidué comme structure algébrique pour le calcul des degrés [Belolavek, 2005]

Le deuxième apport consiste à appliquer la notion de distance mathématique sur les concepts formel afin d'avoir une structure linéaire partielle du treillis de concepts formels flous, Sachant que chaque concepts formel est un ensemble flous. Nous utilisons l'implication de Godel pour définir l'ensemble des concepts formels flous, l'avantage de cette dernière et que son espace de fermeture est un ensemble fini, elle fournit une méthode constructive efficace pour l'ensemble de tous les fermés. Une fois que l'ensemble des fermés est induit, on considère la notion de concept cible à partir duquel un ordre linéaire est défini sur tout le voisinage du concept cible en appliquant la distance de Hausdorff.

La distance de Hausdorff est utilisée généralement dans le traitement d'image, elle indique si deux formes sont les même ou si elles sont l'AFC s'appuie sur formalisme mathématique rigoureux, la théorie des treillis, et l'adapte afin de permettre d'analyser des données, sous la forme d'un contexte formel, est de les structurer en un treillis de concepts, pour cela les perspectives de ce travail porte sur deux points :

- ✓ La proposition d'un algorithme de construction du treillis à partir d'contexte formel flous, et cela afin de facilité la visualisation des relations sous-jacentes entre les données et l'interprétation de ses relation.
- ✓ La proposition d'un algorithme pour le parcours du treillis.

Référence bibliographique

[Agrawal et Srikant, 1994]R. Agrawal and R. Srikant. *Fast algorithms for mining association rules in large databases*. in Proceeding of the 20th international conference on very large data Bases (VLDB'94), pages 478 – 499. Morgan Kaufmann, septembre 1994.

[Azmeh et al., 2008] Azmeh, Z., Huchard, M., Tibermacine, C., Urtado, C., and Vauttier, S. (2008). Wspab : *A tool for automatic classification and selection of web services using formal concept analysis*. ECOWS'08 IEEE Sixth European Conference on Web Services, pages 31–40.

[BELOHLAVEK and VYCHODIL, 2005]. Belohlávek, R. and Vychodil, V. (2005). *What is a fuzzy concept lattice* ? In 3rd Internationnal Conference on Concept Lattices and Their Applications CLA 2005, September 7-9, pages 34 – 45, Olomouc, Czech Republic.

[Bendaoud et al., 2008] Bendaoud, R., Napoli, A., and Toussaint, Y. (2008). *Formal concept analysis : A unified framework for building and refining ontologies*. In [Gangemi and Euzenat, 2008], pages 156–171.

[Carpineto and Romano, 1993] Carpineto, C. and Romano, G. (1993). Galois : *An ordertheoretic approach to conceptual clustering*. Proceedings of 10th International Conference on Machine Learning, Amherst, pages 33–40.

[Carpineto and Romano, 2005] Carpineto, C. and Romano, G. (2005). *Using concept lattices for text retrieval and mining*. In [Ganter et al., 2005a], pages 161–179.

[Cole et al., 2003] Cole R., Eklund P., Stumme G., *Document Retrieval for Email Search and Discovery using Formal Concept Analysis* , Journal of Applied Artificial Intelligence (AAI), vol. 17, n° 3,

[Dau and Klinger, 2005] Dau, F. and Klinger, J. (2005). *From formal concept analysis to contextual logic*. In [Ganter et al., 2005b], pages 81–100.

[Ducrou, 2007] : Ducrou J., DVDSleuth : *A Case Study in Applied Formal Concept Analysis for Navigating Web Catalogs , Conceptual Structures : Knowledge Architectures for Smart Applications*, 15th International Conference on Conceptual Structures (ICCS 2007), vol. 4604 of LNCS, Springer, p. 496-500, 2007. p. 257-280, 2003.

[Dubois, 2000] D. Dubois, H. Prade. *Fundamental of fuzzy sets. In: Kluwer Academic Publisher*, 2000

[Ganter et al., 2005] Ganter, B., Stumme, G., and Wille, R., editors (2005b). *Formal Concept Analysis, Foundations and Applications, volume 3626 of Lecture Notes in Computer Science. Springer*.

[Ganter and Wille, 1999] Ganter, B. and Wille, R. (1999). *Formal Concept Analysis*. Springer, mathematical foundations edition.

[Godin et al, 1993] Godin, R., Missaoui, R., and April, A. (1993). *Experimental comparison of navigation in a galois lattice with conventional information retrieval methods*. International Journal of Man-machine Studies, 38 :747–767.

[Godin et al, 1995] Godin, R., Mineau, G., and Missaoui, R. (1995). *Incremental structuring of knowledge bases*. In Ellis, G., Levinson, R. A., Fall, A., and Dahl, V., editors, Proceedings of the 1st International Symposium on Knowledge Retrieval, Use, and Storage for Efficiency (KRUSE'95), Santa Cruz (CA), USA, pages 179–193. Department of Computer Science, University of California at Santa Cruz.

[Godin and Valtchev, 2005] Godin, R. and Valtchev, P. (2005). *Formal concept analysis-based class hierarchy design in object-oriented software development*. In [Ganter et al., 2005a], pages 304–323.

[Kuznetsov, 2005] Kuznetsov, S. O. (2005). *Galois connections in data analysis : Contributions from the soviet era and modern russian research*. In [Ganter et al., 2005a], pages 196–225.

[M. KAYTOUE et A. NAPOLI, 2009] M. Kaytoue et A. Napoli. *Classification de données numériques par treillis de concepts et structures de patrons* (2009)

[Lakhal and Stumme, 2005] Lakhal, L. and Stumme, G. (2005). *Efficient mining of association rules based on formal concept analysis*. In [Ganter et al., 2005a], pages 180–195.

[Messai, 2009] N. Messai. *Analyse de concepts formels guidée par des connaissances de domaine : Application à la découverte de ressources génomiques sur le Web* (2000)

[Pasquier, 2000] N. Pasquier. Data Minnig : *Algorithme d'extraction et de réduction des règles d'association dans les bases de données* (2000)

[Pensa et al., 2004] Pensa, R. G., Leschi, C., Besson, J., and Boulicaut, J.-F. (2004). Assessment of discretization techniques for relevant pattern discovery from gene expression data. In [Zaki et al., 2004], pages 24–30.

[Priss, 2005] Priss, U. (2005). *Linguistic applications of formal concept analysis*. In [Ganter et al., 2005a], pages 149–160.

[Russel, ….] B. Russel. Cours logique floue (2003)

[Tilley et al., 2005] Tilley, T., Cole, R., 0002, P. B., and Eklund, P. W. (2005). *A survey of formal concept analysis support for software engineering activities*. In [Ganter et al., 2005a], pages 250–271.

[Truck, 2002] Isis TRUCK. *Approches symbolique et floues des modificateurs linguistiques et leur lien avec l'agrégation. Application : le logiciel FLOUS* (2000)

[J. Villerd, 2008] *Représentations visuelles adaptatives de connaissances associant projection multidimensionnelle* (MDS) et analyse de concepts formels (FCA).

[Wille, 1996] Wille, R. (1996). *Restructuring mathematical logic : an approach based on peirce's pragmatism*. In Ursini, A. and Agliano, P., editors, Logic and algebra, pages 267–281, New York. Marcel Dekker.

[wille, 1982] R. Wille. *Restructuring lattice theory : an approach based on hierarchies of oncepts*. Ordered sets, pages 445_470, 1982.

[Wolff, 1993] Karl Erich Wolff. *A First Course In Formal Concept Analysis How To Understand Line Diagrams* (1993)

[YAG 80] R.R.Yager. An *approach to inference in approximation reasoning*. In journal of Man-Machine studies, vol.19, pp. 195-227, 1980

[Yahia and Jaoua, 2001] Yahia, S. B. and Jaoua, A. (2001). *Discovering knowledge from fuzzy concept lattice*. pages 167–190.

[Zhou et al., 2007] Zhou, W., Liu, Z., Zhao, Y., and Xie, Z. (2007). *Clustering-based reduction algorithm on the structure of fuzzy concept lattice*s. In 5th International Conference Formal Concept Analysis, ICFCA07, Supplementary Volume, pages 131–145, Clermont-Ferrand, France.