

NO d'ordre :

UNIVERSITE DE M'SILA

Faculté des Mathématiques et de l'informatique

Département d'informatique

MEMOIRE

Présenté pour l'obtention du diplôme de Magistère

Spécialité : Informatique

Option : Systèmes d'Information et de Connaissance

Par :

Hamani Mohamed Said

SUJET

***Une approche à base d'ontologies
floues pour la fouille de données***

Soutenu publiquement le 10/06/2010 devant le jury composé de :

Président : Pr. Boufaida Mahmoud, Université de Constantine

Rapporteur : Dr. Maamri Ramdane, Université de Constantine

Examineur : Pr. Bouderah Brahim, Université de M'sila

Examineur : Pr. Boufaida Zaizette, Université de Constantine

Examineur : Pr. Benmohammed Mohamed, Université de Constantine

Promotion : 2009/2010

Remerciement

Mes remerciements vont tout d'abord à Mr Maamri Ramdane, Maître de conférence de l'université de Constantine pour avoir accepté de diriger ce travail, pour ses précieux conseils, ses orientations sa compréhension et sa patience.

Je remercie Mr Boufaida Mahmoud, professeur de l'université de Constantine, qui me fait l'honneur de présider ce jury.

Je remercie Mr Bouderah Brahim, professeur de l'université de M'sila, qui me fait l'honneur d'avoir accepté d'examiner ce travail.

Je remercie Mme Boufaida Zizette, professeur de l'université de Constantine, qui me fait l'honneur d'avoir accepté d'examiner ce travail.

Je remercie Mr Benmohammed Mohamed, professeur de l'université de Constantine, qui me fait l'honneur d'avoir accepté d'examiner ce travail.

Je remercie, autre fois, Mr Guemra Mohamed pour avoir relu ce mémoire et avoir enlevé les fautes d'orthographe que j'avais laissées.

Mes remerciements vont également à tous ceux qui ont contribué à la réalisation de ce travail.

TABLE DES MATIERES

Introduction Générale

1. Contexte de l'étude et problématique	1
2. Plan de mémoire	3

Partie 1 : Etat de l'art

Chapitre 1: Fouille de données

1. Introduction	7
2. Extraction de connaissances à partir de données (ECD)	8
2.1. Le Processus de l'ECD	9
2.1.1. Préparation des données	9
2.1.2. Fouille de données	10
2.1.2.1 Tâches de fouille de données	10
2.1.2.2 Classification des méthodes de fouille de données ...	12
2.1.3. Interprétation des résultats	12
2.3. Modèle CRISP-DM	13
3. Règles d'association	15
3.1. L'extraction des motifs fréquents	15
3.2. Génération des règles d'associations	16
3.3 L'algorithme d'extraction des règles 'Apriori'	17
3.4. Mesures d'intérêts des règles d'associations	18
3.4.1. Mesures d'intérêts objectives	19
3.4.2. Mesures d'intérêts subjectives	24
3.5. L'intérêt de classement des règles d'associations	24
4. Conclusion	26

Chapitre 2: Représentation de la connaissance et ontologie

1. Introduction	28
2. Représentation de la connaissance	29
2.1. Langages de représentation des connaissances	29
3. Ontologie	31

3.1. Composantes d'une ontologie	31
3.2. Le cycle de vie des ontologies	33
3.3. Construction d'ontologies	33
3.4. Typologie des ontologies	34
3.5. Langages de représentation des ontologies	35
3.6. Ontologie et représentation des connaissances	37
3.7. Ontologie floue	38
4. Similarités entre concepts	40
4.1. Mesures reposant sur la distance	41
4.2. Mesures reposant sur le contenu en information	43
4.3. Mesures Mixtes	43
5. Logique floue	44
5.1. Variables linguistiques floues	44
5.2. Les ensembles flous et la fonction d'appartenance	44
5.3. Opérations logiques sur les ensembles flous	46
6. Conclusion	48

Chapitre 3: L'utilisation des ontologies dans le processus de fouille de données

1. Introduction	50
2. Ontologies en fouille de données	51
2.1. Les ontologies dans le cycle de vie CRISP-DM	52
2.2. Travaux existants sur l'intégration des ontologies dans l'ECD	53
3. Travaux existants sur les mesures d'intérêt et évaluation	54
4. Conclusion	59

Partie 2 : Contributions

Chapitre 4: Utilisation des ontologies floues dans le classement des règles intéressantes

1. Introduction	62
2. Présentation de l'approche	63
2.1. Principe de l'approche	63
2.2. Description de l'approche	63
2.3. Notre approche dans le modèle CRISP-DM	64

3. Présentation du système	65
3.1. Processus de classement des règles d'association	65
3.2. Représentation des connaissances du système	68
3.2.1. Représentation des règles d'associations	68
3.2.2. Représentation d'une ontologie floue	69
4. Présentation de l'algorithme de classement des règles	71
4.1. Calcul de la distance sémantique entre concepts	71
4.2. L'ontologie floue et le calcul de distance	72
4.3. Pondération des relations taxonomique	72
4.4. Calcul de la distance d'une règle d'association	73
4.5. Présentation des résultats	73
4.6. Algorithme Pseudo code	73
5. Conclusion	75

Chapitre 5: Etude de cas

1. Introduction	77
2. Présentation du cas d'étude théorique	78
2.1. Application de l'algorithme de classement des règles	78
2.1.1. Construction de la matrice des relations pondérées	79
2.1.2. Construction de la matrice du plus court chemin	80
2.1.3. Calcul des distances des règles	81
2.1.4. Présentation des résultats	83
2.2. Interprétation des résultats	83
3. Présentation du cas d'étude réel	84
3.1. Extraction des règles d'associations	88
3.2 Application de l'algorithme de classement des règles	89
3.2.1 Cas d'ontologie ordinaire	89
3.2.2. Cas d'ontologie Floue	92
4. Conclusion	97

Conclusion générale	98
----------------------------------	----

Références	100
-------------------------	-----

Annexe	109
---------------------	-----

LISTE DES FIGURES

Figure 1-1 Présentation classique du processus d'ECD	8
Figure 1-2 Modèle CRISP-DM	13
Figure 1-3 Présentation des sous ensembles de l'ensemble {a, b, c, d}.....	16
Figure 2-1 Le cycle de vie d'une ontologie	33
Figure 2-2 La fonction d'appartenance pour un ensemble ordinaire et floue	45
Figure 2-3 Les fonctions d'appartenance	46
Figure 3-1 Domaine de connaissance et CRISP-DM	51
Figure 4-1 Système de classement des règles d'associations	65
Figure 4-2 Processus de classement des règles d'associations	67
Figure 4-3 Extrait d'un Document PMML	68
Figure 4-4 Exemple de relation avec degré de confiance	69
Figure 4-5 Exemple de nœud blanc	70
Figure 4-6 Exemple de taxonomie	71
Figure 5-1 Ontologie floue de nourriture	78
Figure 5-2 Identification des concepts de l'ontologie	79
Figure 5-3 Relations de l'ontologie	79
Figure 5-4 Graphe pondéré de nourriture	80
Figure 5-5 Matrice des relations pondérées	80
Figure 5-6 Matrice du plus court chemin	81
Figure 5-7 Les règles en entrées	81
Figure 5-8 Classement des règles et leurs distances	83
Figure 5-9 Extrait des résultats du processus d'extraction des règles	88
Figure 5-10 Extrait du résultat du premier test	89
Figure 5-11 Extrait du résultat du second test	90
Figure 5-12 Extrait du résultat de classement 'Ontologie floue'	96

LISTE DES TABLEAUX

Tableau 1-1 Liste des mesures objectives	20
Tableau 5-1 Tableau des distances des règles	83
Tableau 5-2 Les attributs de la base de données recensement du revenu	84
Tableau 5-3 Ontologie de recensement de revenu	87
Tableau 5-4 Concepts flous de l'ontologie de recensement	92
Tableau 5-5 Ontologie floue de recensement de revenu.....	95

Introduction générale

1. Contexte de l'étude et problématique

Il est chaque jour plus facile de collecter des données mais notre capacité à en extraire des informations à forte valeur ajoutée reste limitée. La fouille de données est l'extraction automatique des motifs représentant des connaissances stockées dans de larges bases de données. La découverte de connaissance dans les bases de données (data mining) a été défini dans [FPSS96] comme étant "le processus non trivial d'identification, à partir de données, de motifs valides, nouveaux, utiles et compréhensibles". L'objectif principal de l'exploration de données est de découvrir des motifs intéressants à partir de données. Les processus d'exploration de données sont des processus itératifs et interactifs que l'on peut résumer en 5 macro-étapes : la sélection et consolidation des sources de données brutes, la préparation de ces données pour produire des contextes d'extraction, les extractions proprement dites de motifs ou de modèles, le post-traitement et l'interprétation des motifs ou modèles calculés et enfin l'exploitation des résultats.

Il a été observé que la plupart des algorithmes de data mining génèrent un grand nombre de motifs qui sont valides mais évidents ou peu intéressants. Pour cette raison des méthodes de filtrages et de classement de motifs sont nécessaires pour sélectionner les motifs les plus intéressants.

Des mesures d'intérêt dites objectives, telles que la confiance et le support [AIS93a], ont été introduites et utilisées pour l'évaluation des propriétés statistiques des modèles extraits (la force statistique du pattern). Bien que les mesures objectives soient utiles, elles sont insuffisantes dans la détermination de l'intérêt d'une règle. Il a été discuté dans [MM95, PS91, UFS91, Zim91] qu'en plus des mesures d'intérêt objectives, les mesures subjectives telles que l'inattendu "unexpectedness" (les règles sont plus intéressantes si elles sont inconnues à l'utilisateur ou en contradiction avec ses connaissances ou ses attentes) [MM95, UFS91, Zim91] et "actionability" (Les règles sont intéressantes si l'utilisateur peut faire quelque chose à son avantage) [DT93, PS91, UFS91, Zim91] sont également importantes.

L'intérêt d'une règle est essentiellement subjectif [KMR+94, LH96, ST96]. Des mesures d'intérêts dites subjectives comme l'inattendu d'un motif dépendent du décideur et pas seulement de la force statistique du motif. Les mesures d'intérêt

subjectives confrontent les modèles extraits à des connaissances exprimées sur le domaine.

Les motifs inattendus ont été étudiés dans [ST95, ST96, LH96, LHC97, Suz97, CSD98, Sub98, BT98, PT98, PT99, P99] où l'extraction des règles est limitée aux règles inattendues. L'inattendu est défini par rapport à un système d'attente à priori. Les motifs inattendus ont été étudiés dans [ST95, ST96, LH96, LHC97, PT98, PT99] et définis par rapport à la croyance de l'utilisateur. Une règle est considérée intéressante si elle affecte le niveau de conviction de l'utilisateur. Dans [ST95, ST96] L'inattendu est défini en termes probabilistes alors que dans [LH96], il est défini comme une distance et il est basé sur une comparaison syntaxique entre une règle et une conviction. Une règle et une conviction sont distants si le conséquent de la règle et le conséquent de la conviction sont similaires, mais les antécédents sont éloignés, ou vice versa. Dans [PT98] l'inattendu est défini du point de vue de contradiction logique entre une règle et une conviction où le motif qui contredit une connaissance préalable est inattendu.

La plupart des recherches sur l'inattendu portent sur une comparaison syntaxique ou sémantique entre une règle et une conviction. Dans notre approche, on propose une mesure d'intérêt en fonction de la distance sémantique de la règle, basée sur les connaissances préalables représentées par une ontologie. Le classement des règles sur la base de mesures l'intérêt est une approche intuitive à la résolution des problèmes de qualité et quantité des règles.

2. Notre contribution

On propose une approche pour le classement des règles les plus intéressantes en fonction de leurs distances conceptuelles. On définit la distance conceptuelle d'une règle, comme étant la distance sémantique entre son antécédent et son conséquent basée sur les connaissances à priori du domaine représentées par une ontologie floue. L'intégration des connaissances du domaine dans le processus du data mining peut aider à choisir les informations appropriées (prétraitement), diminuer l'espace d'hypothèses (traitement) et évaluer la pertinence des motifs découverts et leurs expressions dans des termes concis et à différents niveaux d'abstraction (post-traitement).

Une ontologie permet une représentation formelle d'un domaine de connaissance sous la forme d'une terminologie hiérarchisée munie de relations sémantiques [Gru93a]. Elle est organisée comme une hiérarchie DAG (Directed

Acyclic Graph) où les concepts hautement reliés sont regroupés ensemble dans la hiérarchie. Plus les concepts sont éloignés, plus ils sont moins reliés.

Du fait que la connaissance est lexicalement imprécise et non catégorique, elle contient de l'ambiguïté et le flou, l'ontologie floue a été introduite pour s'occuper de la connaissance floue [WY01] où les concepts sont en relation entre eux dans l'ontologie, avec un degré d'appartenance μ ($0 \leq \mu \leq 1$).

L'idée de base derrière notre approche est que plus un pattern est constitué d'éléments conceptuellement distants, plus il est surprenant ; que tout pattern surprenant est inattendu et que tout pattern inattendu est intéressant. Un classement des patterns par ordre décroissant selon leurs degrés d'intérêts en fonction de la distance conceptuelle, facilite l'accès aux patterns les plus intéressants. Les règles d'associations de rang élevé issues de classement à base de mesures subjectives (sur la base d'inattendu) sont indicatives de connaissances nouvelles et inconnues.

Un algorithme de classement des patterns à base de degrés d'intérêts est présenté. On va prendre pour cas d'application les règles d'associations. Les règles que nous considérons sont de la forme « body $\rightarrow$ head » où « body » et « head » sont des conjonctions de concepts en vocabulaire de l'ontologie. Dans un réseau sémantique «IS-A», la forme la plus simple de déterminer la distance entre deux nœuds de concepts élémentaires, A et B, est le chemin le plus court qui relie A et B, c'est-à-dire le nombre minimal d'arcs qui séparent A et B [RMBB89] ou la somme des poids des arcs le long de ce chemin le plus court [RS95]. Pour comparer des groupes de concepts, on va utiliser, la distance de Hausdorff [HAR93].

Notre travail rentre dans la phase post traitement du processus du data mining qui suppose que la tâche de découverte de patterns a été exécutée et a éliminé les patterns qui ne satisfaisaient pas les critères objectifs à savoir le support et la confiance. Une étude de cas est présentée pour validation de notre approche. Nous allons terminer ce travail par une conclusion et des perspectives pour des travaux futurs.

3. Plan de mémoire

Le mémoire est organisé en un plan de lecture divisé en (05) chapitres :

Le **Chapitre 1** introduit le processus de fouille de données, à savoir le prétraitement, le traitement et le post traitement. On expose les différentes tâches des phases de prétraitement, traitement ainsi que les mesures d'intérêts pour l'évaluation des connaissances dans la phase post traitement.

Le **Chapitre 2** décrit la représentation de la connaissance ainsi que la construction et la représentation de l'ontologie.

Le **Chapitre 3** décrit les différents travaux d'intégrations et d'utilisations des ontologies dans le processus de data mining. Un état de l'art des travaux sur le classement et le filtrage des règles selon des mesures subjectives est décrite à la fin de ce chapitre.

Le **chapitre 4** décrit la partie théorique de notre contribution pour le classement des règles d'associations selon leurs degrés d'intérêts. Une présentation générale de l'approche et de son emplacement dans le processus de data mining sera décrite, ainsi qu'une présentation détaillée de l'algorithme de calcul des distances et de classement des règles.

Le **chapitre 5** présente deux études de cas pour appliquer et valider notre approche par le biais de notre implémentation de l'algorithme de classement des règles. Le premier cas d'étude est théorique avec un nombre réduit de règles et de concepts d'une ontologie floue de nourriture, ce qui permet de suivre le déroulement de l'algorithme étape par étape pour valider les résultats de l'approche. Le deuxième cas d'étude porte sur une base de données réelle de recensement des revenus où les règles ont été extraites en utilisant une implémentation de l'algorithme 'Apriori'. Ce deuxième cas va permettre de valider notre approche sur un nombre acceptable de règles et de concepts. À la fin de chaque étude de cas une interprétation des résultats issus de l'application de notre algorithme est donnée.

Enfin, Nous allons terminer ce mémoire par une conclusion générale dans laquelle, nous allons présenter un bilan du travail effectué ainsi que les perspectives et les travaux futurs, méritant d'être effectués pour améliorer et approfondir cette étude.

Partie 1 : Etat de l'art

Chapitre 1

Fouille de données

1. Introduction
2. Extraction de connaissances à partir de données (ECD)
 - 2.1. Le Processus de l'ECD
 - 2.1.1. Préparation des données
 - 2.1.2. Fouille de données
 - 2.1.2.1 Tâches de fouille de données
 - 2.1.2.2 Classification des méthodes de fouille de données
 - 2.1.3. Interprétation des résultats
 - 2.3. Modèle CRISP-DM
3. Règles d'association
 - 3.1. L'extraction des motifs fréquents
 - 3.2. Génération des règles d'associations
 - 3.3. L'algorithme d'extraction des règles 'Apriori'
 - 3.4. Mesures d'intérêts de règles d'associations
 - 3.4.1. Mesures d'intérêts objectives
 - 3.4.2. Mesures d'intérêts subjectives
 - 3.5 L'intérêt de classement des règles d'associations
4. Conclusion

1. Introduction

La croissance rapide dans la production et le stockage des données, ont attiré l'attention à un besoin d'outils automatiques pour assister à la transformation de ces données en informations et connaissances utiles qui peuvent être utilisées pour différentes applications telles que l'administration d'affaires, le contrôle de production ou l'analyse du marché,...etc. Grâce aux techniques actuelles d'extraction de connaissances, les bases de données volumineuses sont devenues des sources riches et fiables pour la génération et la validation de connaissances.

Data mining, «knowledge discovery in databases (KDD) », fouille de données ou extraction de connaissances à partir de données (ECD) sont des termes qui désignent l'extraction automatique des motifs représentant des connaissances stockées dans de large base de données. L'ECD est un champ interdisciplinaire qui a émergé pendant la fin des années 1980, Il se situ à l'intersection de nombreuses disciplines comme l'apprentissage automatique, la reconnaissance des formes, les bases de données, les statistiques, la représentation de connaissances, l'intelligence artificielle, les systèmes experts, etc.

2. Extraction de connaissances à partir de données (ECD)

Le terme ECD est communément confondu avec le data mining ou « Fouille de Données ». Ceci s'explique par le fait que le data mining est l'étape principale du processus de l'ECD. L'ECD a été défini, comme étant «le processus non trivial d'identification, à partir de données, de motifs valides, nouveaux, utiles et compréhensibles» [FPSS96]. Concrètement il s'agit de l'utilisation de méthodes de fouille de données, associées à une préparation des données préalables, et à une interprétation des résultats de fouille, afin d'extraire des connaissances pertinentes au regard des objectifs visés par l'analyste. Le terme processus implique que l'ECD comprend plusieurs étapes. Par non trivial, on veut dire que l'extraction des connaissances n'est pas aussi simple que le calcul d'une moyenne ou de la variance d'un ensemble de nombres. Les motifs sont des descriptions des données, exprimés dans un langage spécifique, ils doivent être valides, ayant un degré d'exactitude acceptable si on les applique sur de nouveaux cas. Ils doivent être nouveaux au système et utiles, donnant ainsi une valeur ajoutée à l'utilisateur. Finalement, ils doivent être compréhensibles, ayant un sens pour le preneur de décision. L'ECD est un processus qui comprend plusieurs étapes, dont certaines impliquent une prise de décision de l'utilisateur, i.e. l'analyste qui conduit le processus.

Les deux grand objectifs du data mining peuvent être définis comme prédiction et description [FPSS96]. La prédiction utilise quelque attributs ou variables dans la base de données pour prévoir des valeurs futures d'autres variables. La description caractérise les propriétés générales des données dans la base de données et met l'accent sur la compréhension et l'interprétation de ces dernières.

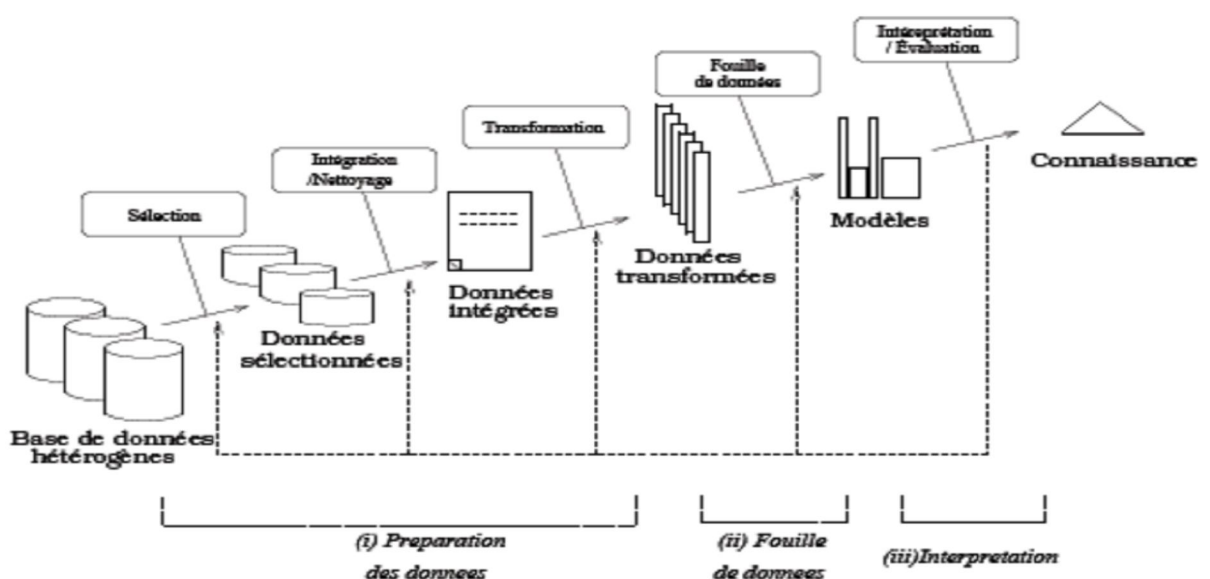


Figure 1-1 Présentation classique du processus de l'ECD [FPSS96]

2.1. Le Processus d'ECD

Le processus de l'ECD est un processus itératif et interactif assisté par l'utilisateur. Itératif car les résultats produits peuvent être réutilisés lors des itérations suivantes du processus. Interactive car le processus est réalisé sous le contrôle d'un expert du domaine étudié : l'analyste. C'est lui qui guide le processus en fonction de ses objectifs, de ses propres connaissances du domaine, et des résultats obtenus lors des précédentes itérations de l'extraction.

Nous distinguons trois étapes principales dans le processus, représentées dans la (Figure 1-1) :

1. La préparation des données incluant leur sélection, leur intégration, et leur nettoyage
2. L'opération de fouille de données proprement dite, conduisant à l'extraction de connaissances.
3. L'interprétation des d'informations extraites en terme de connaissances. Cette étape identifie les modèles intéressants représentant les connaissances, en se basant non seulement sur des mesures d'intérêt mais aussi sur l'avis de l'expert.

Dans ce qui suit on va présenter une description plus détaillée des différentes étapes.

2.1.1. Préparation des données

La préparation des données (prétraitement) est définie par l'ensemble des opérations qui permettent de convertir les données brutes en données préparées et adaptées à la méthode de fouille envisagée. Le prétraitement de données consiste en toute action effectuée sur les données avant l'application d'une technique de data mining (e.g. la classification automatique, le classement et les règles d'association). L'intérêt principal de cette étape est d'améliorer la qualité des données selon la méthode de fouille choisie et ainsi d'améliorer l'efficacité du processus d'ECD. Les opérations de préparation peuvent être de différents types : l'intégration des données, le nettoyage des données, la réduction des données, la transformation des données. L'ordre de ces opérations varie souvent selon la stratégie d'ECD adoptée. Un processus de pré-traitement de données peut être décrit selon les étapes suivantes :

2.1.1.1. La sélection de données

Cette phase concerne donc le filtrage de données. Par filtrage, nous voulons dire la réduction de la dimensionnalité des données (élimination d'attributs sans intérêt, ou ayant beaucoup de valeurs erronées et manquantes), ainsi que la réduction la taille des données (des enregistrements). La deuxième réduction peut être faite par des techniques statistiques d'échantillonnage, s'il s'avère que la masse de données est trop grande, et que l'application du Data Mining serait ainsi très coûteuse en terme de temps CPU et d'espace mémoire.

2.1.1.2. Le nettoyage et intégration et des données

Cette tâche consiste en la détection et la suppression des erreurs, du bruit, de l'incohérence de données pour améliorer leur qualité et l'intégration de données de sources multiples.

2.1.1.3. La transformation de données

Plusieurs algorithmes de data Mining sont contraignants sur la forme des données qu'ils acceptent. Cette étape consiste à préparer les données brutes et à les convertir en données appropriées. La transformation se fait par attribut (toutes les valeurs d'un attribut doivent être transformées).

La discrétisation de variables continues est un exemple de transformation d'attributs. Il s'agit de transformer un attribut continu en divisant son domaine en intervalles finis. Ainsi, le domaine de l'attribut transformé devient un ensemble de valeurs discrètes.

L'agrégation de données est un autre type de transformation. L'agrégat d'un attribut est la transformation de ce dernier par une règle ou équation. Imaginons que l'ont veut analyser les salaires annuels des employés, et que l'on dispose seulement des salaires mensuels. Un nouvel attribut agrégat serait le salaire multiplié par douze.

2.1.2. Fouille de données

2.1.2.1 Tâches de fouille de données

La fouille de données est l'étape de l'ECD qui vise à extraire des régularités (ou des irrégularités) de l'ensemble de données préparées. Les tâches primaires du data mining sont:

- **Concept Description: Caractérisation et Comparaison :** La description de données fournit une vue générale d'une classe de données ou une distinction d'un ensemble de classes comparatives. Elle permet de décrire les ensembles de données à différents niveaux de granularité et selon différents angles.
La *caractérisation* fournit un regroupement «summarization» de la collection de données, cependant la *comparaison* de classe ou de concept (aussi connue comme la *discrimination*) fournit des descriptions comparant des collections de données.
- **Classification et Prédiction:** Il s'agit de trouver des modèles ou fonctions qui décrivent et distinguent les classes de données ou concepts. Le modèle trouvé est basé sur l'analyse d'un ensemble de données « training data ». Il s'agit simplement de trouver un modèle en utilisant les objets pour lesquels les classes sont connues dans un but de prévoir les classes des objets pour lesquels leurs classes sont inconnues. Le modèle trouvé peut être représenté sous différentes formes comme règle de classification «IF-THEN-ELSE», arbre de décision, formule mathématique ou réseaux de neurones. Le cas où les valeurs à prévoir sont des données numériques la tâche est appelée prédiction.
- **Le regroupement «Clustering »:** Le regroupement est le processus qui conduit à générer des classes à partir d'analyse des objets de base. Les objets sont groupés en se basant sur la maximisation de similarités intra classes et la minimisation de similarité extra classes. Le regroupement peut faciliter la taxonomie (hiérarchies de classes). A l'encontre de la classification, le clustering est une tâche descriptive.
- **Analyse d'associations:** Elle consiste à trouver des modèles décrivant des dépendances significatives entre les variables. L'analyse d'association est largement utilisée dans le «market basket» et l'analyse des transactions de données. La découverte des relations peuvent être représentées sous la forme de règles d'association ou des ensembles de motifs fréquents.
- **Analyse d'évolution et de déviation:** L'analyse d'évolution des données décrit et modélise les régularités des objets pour lesquels leurs comportements changent dans le temps (évolution générale) ainsi que leurs déviations (la différence entre la valeur mesurée et sa correspondance de référence telle que la valeur précédente ou la valeur normative). Un exemple de cette analyse est la prédiction des tendances boursières, ou encore l'étude des changements climatiques.

2.1.2.2 Classification des méthodes de fouille de données

Les méthodes de fouille de données peuvent être classifiées selon

1. L'objectif de data mining prédiction et description [FPSS96].
 - Les méthodes prédictives utilisent quelque attributs ou variables dans la base de données pour prévoir des valeurs futures d'autres variables.
 - Les méthodes descriptives caractérisent les propriétés générales des données dans la base de données et met l'accent sur la compréhension et l'interprétation de ces dernières.
2. Selon le type de données qu'elles manipulent:
 - Méthodes de fouille numériques : comprennent, entre autres, les chaînes de Markov, les réseaux de neurones, les K-plus proches voisins, l'analyse en composante principale (ACP), les réseaux bayésiens, les algorithmes génétiques.
 - Méthodes de fouille symboliques : comprennent, entre autres, l'extraction de motifs fréquents, la recherche de règles d'associations, l'Analyse de Concepts Formels (ACF).
3. Selon l'information à priori :
 - Les méthodes dites supervisées et celles dites non supervisées. Une méthode supervisée va proposer une classification des objets en s'appuyant sur un modèle préétabli d'exemples ou d'échantillons sélectionnés au hasard.
 - Les méthodes dites non supervisées, une méthode non supervisée va produire un modèle sans information à priori.

2.1.3. Interprétation des résultats

L'étape d'interprétation du processus d'ECD est également appelée le post traitement. Elle consiste en la prise en charge des résultats bruts de la fouille de données, en leur transformation pour leur interprétation et validation par l'analyste en unités de connaissance. Cette étape demande une implication de l'analyste qui doit interpréter des résultats de fouille potentiellement volumineux. La forme des résultats est différente selon la méthode de fouille utilisée : motif fréquent, concept formel, règle d'association, cluster par exemple.

Afin de faciliter l'interprétation, les résultats sont transformés pour faire l'objet d'une visualisation graphique. Les connaissances attendues par l'analyste orientent l'interprétation. L'analyste peut alors être amené à filtrer parmi les connaissances extraites celles qu'il juge triviales, redondantes, sans intérêt, fausses en comparaison de ce qu'il souhaite trouver.

2.2 Modèle CRISP-DM

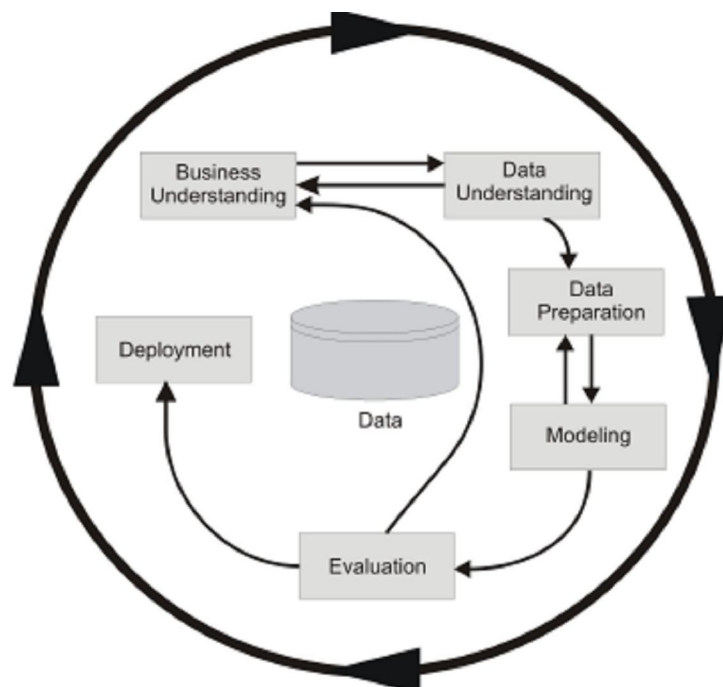


Figure 1-2 Modèle CRISP-DM [CCKK99]

Conçu en 1996, CRISP-DM (Cross Industry Standard Process for Data Mining) est un modèle de processus indépendant de l'industrie et d'outils d'extraction de données. CRISP-DM a évolué comme la norme, avec un processus standardisé qui consiste à définir un cadre pour la conduite de projets de data mining, ce qui permet d'orienter les activités de data mining :

- En tant que méthodologie, CRISP-DM comprend des descriptions des phases typiques d'un projet et des tâches comprises dans chaque phase, et une explication des relations entre ces tâches.
- En tant que modèle de processus, CRISP-DM offre un aperçu du cycle de vie du Data mining. Le modèle définit 6 phases pour accomplir un projet de data mining.

Le modèle de cycle de vie comporte six phases :

2.3.1. Compréhension du domaine « Business Understanding »

C'est la phase initiale du modèle qui se concentre sur les objectifs et les besoins d'un projet selon une perspective d'affaire. Après une analyse réelle au sein de la compagnie, la connaissance acquise est convertie en une définition de problème de data mining qui constitue un plan pour l'atteinte des objectifs du projet.

2.3.2. Compréhension des données « Data Understanding »

La deuxième phase commence par une collection de toutes les données disponibles qui pourraient être pertinentes pour le projet. Pour se familiariser avec les données, des activités de description et d'exploration sont nécessaires. Une autre tâche importante est de vérifier la qualité des données.

2.3.3. Préparation des données « Data Préparation »

Dans la phase de préparation de données, les données pertinentes sont choisies parmi l'ensemble général de données découvertes dans la phase précédente. Dans une deuxième étape, les données initiales doivent être nettoyées, intégrées et formatées. Toutes ces activités visent à construire l'ensemble final de données adéquat pour le data mining. Les tâches dans cette phase seront probablement exécutées plusieurs fois afin de fournir un ensemble de données de qualité.

2.3.4. Modélisation « Modeling »

La phase de modélisation consiste à la sélection des méthodes et des algorithmes possibles pour le data mining et ajuster les valeurs optimales de leurs paramètres pour la tâche spécifique du data mining. Plusieurs méthodes du data mining pourraient être adéquates pour l'exploration et devraient être évaluées à cette étape. Quelques algorithmes ont des exigences spécifiques pour les données, dans ce cas un retour à la phase de préparation est nécessaire. Les modèles découverts doivent être mesurés et évalués selon le but du data mining.

2.3.5. Evaluation « Evaluation »

Après la construction des modèles pour le data mining, leurs degré de réalisation des objectifs d'affaires doit être mesuré. Un modèle peut être d'une haute qualité selon une perspective d'analyse de données, mais pourrait être très pauvre dans la satisfaction des besoins d'affaires. La phase finale se concrétise avec une décision sur l'utilisation des résultats de data mining.

2.3.6. Déploiement « Deployment »

Le processus ne se termine pas avec la création d'un modèle correct. La connaissance trouvée doit être organisée et présentée selon des formes compréhensibles et utilisables par l'utilisateur. En plus, un modèle doit être suivi et maintenu pour permettre une utilisation future. Des besoins d'ajustement des modèles sont parfois nécessaires du fait que les besoins de l'utilisateur changent dans le temps (Un modèle valide ne pourrait pas être valide tout le temps). La

complexité de la phase de déploiement est très dépendante des besoins définis au début du projet. La fin du processus est marqué par la génération d'un rapport final.

3. Règles d'associations

La méthode de recherche de règles d'association est née pour analyser les articles fréquemment achetés ensemble dans les supermarchés. Chaque sortie de caisse correspond à une transaction où plusieurs items ont été achetés simultanément.

Une règle d'association de la forme $T1 \rightarrow T2$, où $T1$ et $T2$ sont des motifs. $T1$ est appelé la prémisse ou l'antécédent de la règle, et $T2$ est la conclusion ou le conséquent de la règle où $T1 \cap T2 = \emptyset$. Une règle repose sur les notions de support et de confiance. Le support est le nombre ou le pourcentage de transactions qui contiennent les items de la règle (le support du motif $T1 \cup T2$). La confiance est le pourcentage de transactions qui contiennent les items du conséquent parmi celles qui contiennent l'antécédent. Une règle est dite valide si sa confiance est supérieure ou égale à un seuil de confiance arbitraire appelé confiance minimum, et si son support est supérieur ou égal au support minimum. Ainsi toute règle valide $T1 \rightarrow T2$ est basée sur un motif fréquent $T1 \cup T2$.

Les algorithmes de recherche de règles d'association, tel que l'algorithme fondateur Apriori [AS94], procèdent en deux étapes. La première est l'extraction des ensembles des motifs fréquents dont le support est supérieur à un seuil fixé par l'utilisateur.

Seconde étape est la génération des règles dont la confiance est jugée suffisante par l'utilisateur. Le nombre de règles extraites étant souvent important. Pour sélectionner les règles les plus intéressantes, il est utile de les classer par ordre décroissant de leur intérêt au sens d'un indice de pertinence.

3.1 L'extraction des motifs fréquents

L'extraction des motifs fréquents permet d'isoler, depuis une base de données, des ensembles d'attributs appelés motifs en accord avec un certain support. Ce support correspond au nombre d'objets qui partagent les attributs d'un motif et celui-ci doit être supérieur à un certain seuil, le support minimum, pour que le motif soit fréquent.

La recherche de motifs fréquents peut se faire en *largeur* ou en *profondeur* selon la façon d'exploration de l'espace des motifs. L'ordre (selon le sens du

parcours, de gauche à droite ou du haut en bas) dans lequel les motifs sont considérés.

Dans un but d'illustration, supposons que l'ensemble $\{a,b,c,d\}$ contient la totalité des items présents dans l'ensemble de la base de transactions. Les relations d'inclusion entre les sous-ensembles de $\{a,b,c,d\}$ (donc de l'ensemble des motifs) peuvent être représentées par la figure suivante.

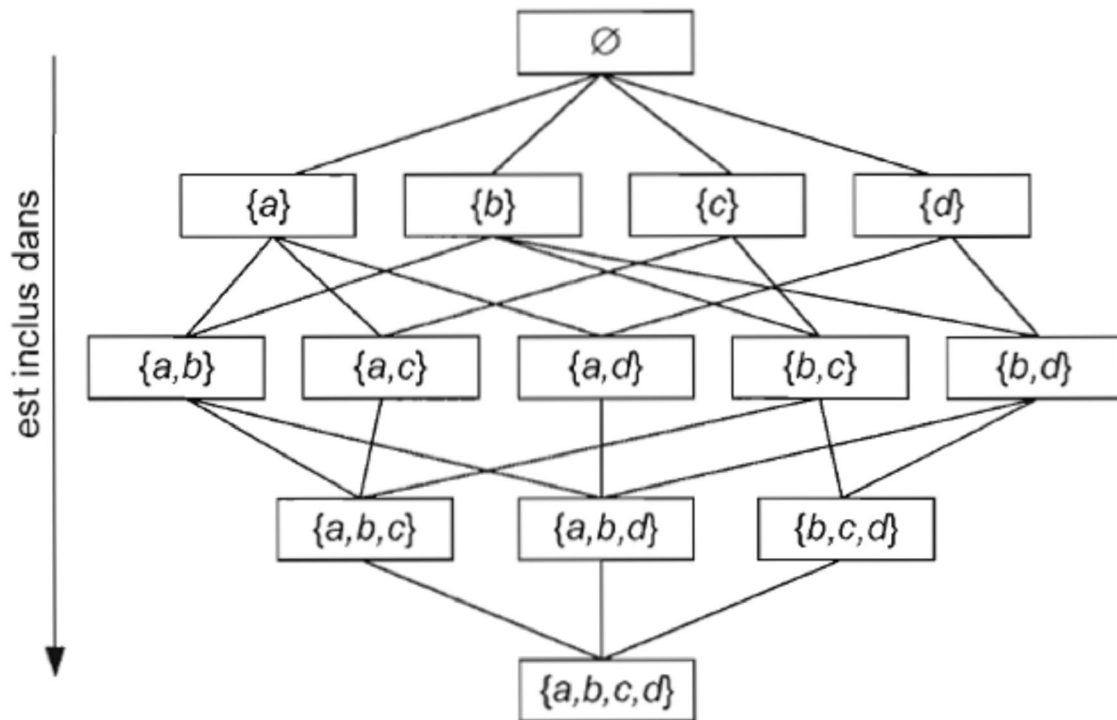


Figure 1-3 Présentation des sous ensembles de l'ensemble $\{a, b, c, d\}$.

Un algorithme de parcours en profondeur considèrera d'abord les sous-ensembles ayant les mêmes préfixes (par exemple $\{a\}$, $\{a,b\}$, $\{a,b,c\}$, $\{a,b,c,d\}$) alors qu'un algorithme de parcours en largeur (aussi appelé « horizontal » ou « par niveaux ») considèrera d'abord les sous ensembles de même taille.

Parmi les algorithmes de parcours en profondeur on trouve FP-growth [HPY00]. L'algorithme Apriori [AS94] est un algorithme de parcours en largeur.

3.2. Génération des règles d'associations

Sur la base des motifs fréquents, il est possible de construire des règles d'associations, de forme générale $A \rightarrow B$ qui associe un sous-ensemble d'attributs A avec un second sous-ensemble d'attributs B. La règle peut alors être interprétée

comme le fait que l'ensemble des objets avec les attributs de A présente également les attributs de B selon un certain support et une certaine confiance.

La construction des règles d'association valides depuis un motif fréquent (de longueur supérieure ou égale à deux) se fait de façon similaire à l'extraction de motifs fréquents. A partir d'un motif fréquent, la construction des règles débute par les règles dont la conclusion est de longueur 1, notées $P-\{i\} \rightarrow \{i\}$ où $\{i\}$ est un attribut seul et $P-\{i\}$ représente le motif P sans l'attribut $\{i\}$. Une fois ces règles construites, leur conclusions sont combinées pour donner de nouvelles règles candidates dont la conclusion est de longueur 2, notées $P-\{ij\} \rightarrow \{ij\}$. Ces nouvelles règles sont testées et le processus continu tant qu'il est possible de construire de nouvelles règles candidates.

3.3. L'algorithme d'extraction des règles 'Apriori'

L'extraction de motifs fréquents selon l'algorithme Apriori s'appuie sur deux principes fondamentaux:

- 1- *Propriété des sous-ensembles fréquents :*
- 2- *Tout super-motif d'un motif non fréquent est non fréquent.*

L'extraction des motifs fréquents commence par la recherche des motifs de longueur 1, ensuite les motifs fréquents sont enregistrés et combinés entre eux pour former des motifs candidats de longueur supérieure. Les motifs non fréquents en 1 sont éliminés, et par conséquent aucun de leur super-motif n'est considéré. La fréquence des motifs candidats est testée pour constituer un nouvel ensemble de motifs fréquents et l'algorithme continu tant que de nouveaux candidats peuvent être formés.

Algorithme Apriori

L_k est l'ensemble constitué des sous-ensembles d'items fréquents de longueur k.

C_k est un ensemble constitué des sous-ensembles d'items candidats de longueur k, notons bien que $L_k \in C_k$

Calculer L_1

$K \leftarrow 2$

Tantque $L_{k-1} \neq \emptyset$ **Faire**

$C_k \leftarrow \text{apriori-gen}(L_{k-1})$

Tantque $t \in D$ **Faire**

$C_t = \text{sousensemble}(C_k, t)$

Tantque $c \in C_t$ **Faire**

$c.\text{Count} \leftarrow c.\text{Count} + 1$

Fin Tanque

Fin Tanque

$L_k = \{c \in C_k \mid c.\text{count} \geq \text{minSup}\}$
 $k \leftarrow k+1$

Fin Tanque

Retourner $\bigcup_k L_k$

L'algorithme Apriori fait appel à deux algorithmes [AO02]:

- apriori-gen qui est constitué de deux phases la première phase nommée Joindre qui permet de trouver tous les candidats possibles de longueur k à partir de l'ensemble L_{k-1} , La deuxième phase qui permet d'effacer de C_k les éléments qui ne vérifient pas la propriété des sous ensemble fréquents.

Soient $p, q \in L_{k-1}$

1. insert into C_k

2. select $p[1], p[2], \dots, p[k-1], q[k-1]$

3. from p, q

4. Where $p[1] = q[1] \dots p[k-2] = q[k-2], p[k-1] < q[k-1]$

- sous-ensemble L'algorithme sous-ensemble calcule le sous-ensemble $C_t \subset C_k$ qui correspond à des sous-ensembles présents dans les transactions contenues dans D . Par exemple si $L_3 = \{123\}; \{124\}; \{134\}; \{135\}; \{234\}$, la phase joindre donne comme résultat $C_4 = \{ \{1234\}; \{1345\} \}$ ensuite la phase effacer donne le résultat : $C_4 = \{1234\}$ car l'élément $\{145\}$ n'est pas dans L_3 et donc $\{1345\}$ est effacé.

3.4. Mesures d'intérêts des règles d'associations

Les algorithmes de règles d'associations génèrent un nombre important de règles ce qui rend la tâche de leur interprétation par l'analyste très fastidieuse. Pour cette raison des méthodes de filtrages ou de classement sont nécessaires pour sélectionner les règles les plus intéressantes. Le classement des règles selon des mesures d'intérêt est L'une des approches les plus populaires pour remédier à ce problème [ST96].

Les mesures l'intérêt visent à capturer le degré d'intérêt q'un modèle peut évoquer durant son analyse. Les modèles intéressants sont censés à susciter une forte attention de la part des utilisateurs. Un utilisateur peut trouver un modèle intéressant à cause de diverses raisons - dont certaines peuvent être difficile à exprimer. Il a été constaté que l'intérêt est un concept vague [ST96], [UFS91] consistant de nombreux aspects qui sont difficiles à mettre en pratique, et donc difficiles à capturer. Dans certains cas, un comportement particulier dans un domaine pourrait être intéressant. Le même comportement exposé dans un autre domaine peut ne pas être intéressant. Ainsi, l'intérêt est dépendant du domaine et de

l'utilisateur. Dans certains autres cas, les mêmes caractéristiques peuvent être indépendantes du domaine et de l'utilisateur.

La capture de toutes les fonctionnalités de l'intérêt en une seule fois dans une mesure est très difficile. Par conséquent, les chercheurs portent généralement sur la capture des caractéristiques qui peuvent être importantes et relèvent pertinentes pour une application particulière. Des scores peuvent être affectés aux motifs découverts, et ceux qui sont évalués avec un rang haut par rapport à ces mesures peuvent être présentés à l'utilisateur. L'interaction entre les différents aspects tels que la pertinence, la nouveauté, l'inattendu, surprise et les connaissances utilisateurs déterminent l'intérêt d'une règle d'association. Selon l'implication de l'utilisateur, les mesures d'intérêt peuvent être classées en deux catégories - les mesures objectives et subjectives [ST96], [Fre99].

Selon [PSM94], l'intérêt d'une règle est défini par des facteurs tels que la nouveauté, l'utilité, la pertinence et la signification statistique. Le concept d'intérêt est donc un amalgame de deux aspects : l'aspect objectif et l'aspect subjectif [Fre99]. Les trois premiers sont du domaine du subjectif, alors que le dernier est du domaine objectif. Ces deux aspects sont captés à l'aide de deux types de mesures : les mesures objectives et les mesures subjectives.

3.4.1. Mesures d'intérêt objectives

Les mesures objectives d'un modèle expriment l'intérêt en fonction de la structure et les données sous-jacentes utilisées dans le processus de découverte. Elles sont fortement indépendantes de l'utilisateur et du domaine et révèlent des caractéristiques des données qui ne sont pas liés aux définitions spécifiques du domaine ou de utilisateur ; cela augmente leur applicabilité dans des situations différentes. Cependant, cette propriété d'indépendance peut limiter leur pouvoir de discrimination. Puisque une mesure objective peut être appliquée dans tous les domaines, elle considère l'intérêt à partir d'une perspective limitée qui est commun à tous les domaines. Par conséquent, les mesures objectives ne peuvent pas saisir toutes les complexités du processus de découverte [ST96]. Plusieurs mesures objectives sont fondées sur la force des relations de dépendance entre les éléments et l'intérêt est considéré comme étant directement proportionnel à cette force. Toutefois, ce point de vue peut conduire à des résultats erronés. Prendre «support» comme un exemple. S'il est utile de mesurer la signification statistique des règles générales, les règles qui sont les plus évidentes pour l'utilisateur ont des valeurs élevées de support. Ainsi, en mettant en évidence des règles basées sur des valeurs de support peuvent ne pas ajouter des connaissances supplémentaires. De même,

d'autres mesures objectives ont leurs propres limites. Les mesures objectives sont utilisées comme des filtres initiaux pour supprimer définitivement les règles qui ne sont pas intéressantes ou non rentable. Les règles qui sont statistiquement insignifiantes peuvent être supprimées car elles ne méritent pas plus d'attention. Le tableau suivant représente une liste des mesures objectives, ainsi que leurs définitions [AZÉ03A].

mesure	expression
Support	$P(AB)$
Confiance	$P(B A)$
Rappel	$P(A B)$
confiance centrée	$P(B A) - P(B)$
Lift	$\frac{P(AB)}{P(A)P(B)}$
Moindre contradiction	$\frac{P(AB) - P(A)P(B)}{P(B)}$
Piatetsky-Shapiro	$nP(A)(P(B A) - P(B))$
Loevinger	$\frac{P(B A) - P(B)}{P(B)}$
Corrélation	$\frac{P(AB) - P(A)P(B)}{\sqrt{P(A)P(A)P(B)P(B)}}$
Indice d'implication	$q_3(A, \bar{B}) = \sqrt{n} \frac{P(AB) - P(A)P(B)}{\sqrt{P(A)P(B)}}$
Pearl	$P(A) * P(B A) - P(B) $
Nouveauté	$P(AB) - P(A)P(B)$
Satisfaction	$\frac{P(\bar{B}) - P(\bar{B} A)}{P(\bar{B})}$
Spécificité	$P(A \bar{B})$
Fiabilité négative	$P(\bar{B} A)$
J-mesure	$P(AB) \log \frac{P(AB)}{P(A)P(B)} + P(A\bar{B}) \log \frac{P(A\bar{B})}{P(A)P(\bar{B})}$
Sebag-Schœnauer	$\frac{P(AB)}{P(A\bar{B})}$
Conviction	$\frac{P(A)P(B)}{P(A\bar{B})}$
Intensité d'implication	$\varphi(A, B) = 1 - \Phi(q_3(A, \bar{B}))$
Intensité d'implication entropique	$\Psi(A, B) = \sqrt{\varphi(A, B) \times \tau(A, B)}$

Tableau 1-1 Liste des mesures objectives [AZÉ03A]

Dans ce qui suit, on va définir quelques mesures extraites Tableau 1-1.

3.4.1.1 Support

Le support [AIS93b] d'une règle $A \rightarrow B$ indique la proportion d'objets vérifiant à la fois la prémisse et la conclusion de la règle.

Cette mesure est symétrique, i.e. $\text{support}(A \rightarrow B) = \text{support}(B \rightarrow A)$.

$$\text{support}(A \rightarrow B) = P(AB)$$

- **Confiance**

La confiance [AIS93b] indique la proportion d'objets vérifiant la conclusion parmi ceux vérifiant la prémisse. Cette mesure est non symétrique et insensible à la

taille des données. Du fait que la confiance est un rapport elle peut conduire à considérer un bruit ou une règle triviale comme des règles très intéressantes. La confiance centrée résout ce dernier problème.

$$\text{confiance}(A \rightarrow B) = P(B|A) = P(A \wedge B) / P(A)$$

$$\text{confiancecentree}(A \rightarrow B) = P(B|A) - P(B)$$

3.4.1.3 Confiance centrée

Elle permet donc de relativiser la confiance d'une règle par rapport à la taille de sa conclusion. De plus, l'introduction de $P(B)$ dans cette mesure permet de la rendre sensible à la taille des données.

3.4.1.4 Rappel

Le rappel permet d'évaluer la proportion d'objets vérifiant la prémisse de la règle parmi ceux vérifiant la conclusion.

$$\text{rappel}(A \rightarrow B) = P(A|B)$$

3.4.1.5 Lift

Le lift [IBM96] représente le rapport à l'indépendance de la règle $A \rightarrow B$. L'indépendance entre les attributs A et B est égale à $P(A) * P(B)$. L'indice brut d'association réellement observé est lui égal à $P(AB)$. Le lift permet donc d'apprécier simplement, pour une règle $A \rightarrow B$, sa distance à l'indépendance. Par exemple, une règle $A \rightarrow B$ ayant un lift égal à 2 indique que les individus ayant la propriété A ont deux fois plus de chances d'avoir la propriété B que les individus en général. Cette mesure est symétrique et ne permet donc pas de distinguer les règles $A \rightarrow B$ et $B \rightarrow A$.

$$\text{lift}(A \rightarrow B) = P(AB)/(P(A) * P(B))$$

3.4.1.6 Moindre contradiction

La moindre contradiction [AK02, Azé03b] mesure la différence entre le nombre d'exemples et le nombre de contre-exemples de la règle, cette différence est normalisée par le nombre d'objets vérifiant la conclusion de $A \rightarrow B$. Cette mesure permet de sélectionner les règles ayant plus d'exemples que de contre exemples.

$$\text{contramin}(A \rightarrow B) = (P(AB) - P(A\bar{B}))/P(B)$$

3.4.1.7 Mesure de Sebag-Schoenauer

Cette mesure calcule simplement le rapport entre le nombre d'exemples de la règle et son nombre de contre-exemples.

$$\text{Seb}(A \rightarrow B) = P(AB)/P(A\bar{B})$$

3.4.1.8 Conviction

La conviction [BMS97] peut s'interpréter en fonction des contre-exemples de $A \rightarrow B$. En effet, une conviction élevée indique que le nombre de contre-exemples vérifiant la règle est inférieur à celui attendu sous l'hypothèse d'indépendance entre les attributs de la règle. Donc lorsque le nombre de contre-exemples augmente, la conviction diminue pour atteindre son minimum lorsque la règle ne possède aucun exemple. La conviction représentant l'écart à l'indépendance de la règle $A \rightarrow B$ est interprétable de la manière suivante : une règle ayant une conviction égale à 2 indique qu'il est deux fois plus convaincant de considérer les attributs A et B reliés par la règle $A \rightarrow B$ que de les considérer indépendants (A et B sont indépendants, $P(AB) = P(A)P(B)$).

$$\text{conviction}(A \rightarrow B) = P(A)P(\bar{B}) / P(A\bar{B})$$

3.4.2. Mesures d'intérêt subjectives

Les experts du domaine (les utilisateurs) jouent un rôle important dans l'interprétation et l'application des résultats de l'extraction de données. D'où la nécessité d'intégrer les vues de l'utilisateur. En général, les utilisateurs diffèrent dans leurs croyances et leurs intérêts et peuvent être intéressés par des aspects différents du domaine en fonction de leur domaine de travail. En outre, ils peuvent avoir également des objectifs différents et des opinions différentes quant à l'applicabilité et l'utilité aux résultats d'ECD. Cette variation dans l'intérêt renforce l'importance de l'injection de la subjectivité dans l'évaluation de l'intérêt [ST96]. «Actionability» et «unexpectedness» sont deux aspects qui déterminent l'intérêt subjectif [ST96]. Les règles sont intéressantes si elles sont inattendues (surprenantes pour l'utilisateur) ou d'une action (si l'utilisateur peut agir avantageusement). Une autre caractéristique de l'intérêt se rapporte à la connaissance préalable de l'utilisateur. La connaissance préalable augmente l'intérêt d'un sujet. En outre, les connaissances actuelles sur un sujet deviennent intéressantes si ces connaissances sont directement pertinentes pour les buts de l'utilisateur [Ram90]. En conséquence, l'intérêt peut être défini comme une heuristique qui mesure la pertinence de la contribution aux connaissances relatives aux objectifs d'une personne. Si une information est pertinente pour "knowledge goals" ou "Connaissance des objectifs", l'intérêt envers elle augmente.

Les mesures subjectives dépendent essentiellement des buts, des connaissances et des croyances de l'utilisateur qui doivent être préalablement recueillis. Elles sont associées à des algorithmes permettant de n'extraire que les

règles conformes ou en contradiction avec les croyances de l'utilisateur, et ainsi d'orienter la notion d'intérêt vers la nouveauté ou l'inattendu (unexpectedness). Ces mesures intègrent explicitement les connaissances du décideur. Les approches intégrant des mesures d'intérêt subjectives les plus discutées dans la littérature sont basées sur les patrons, les convictions, les attentes, les impressions générales.

3.4.2.1 Les patrons

Un des premiers travaux consacrés au sujet est celui de [KMR+94] et utilise des patrons « templates » qui imposent une liste d'items devant se retrouver dans les règles. Les patrons décrivent un ensemble de règles spécifiant les attributs apparaissant dans l'antécédent et un attribut prédictif dans le conséquent. Chacun des attributs appartient également à une hiérarchie de concepts.

3.4.2.2 Les convictions

Avec l'approche [PT98], les convictions sont exprimées de la même forme que les règles, et les algorithmes se basent sur la propriété monotonie des croyances « monotonicity of beliefs ». Cette propriété stipule que si une croyance de la forme $A \rightarrow B$ existe pour une base de données avec un certain degré de croyance, alors cette même croyance devrait être applicable pour un sous-ensemble de la base de données ayant une taille significative [PT98].

Avec l'approche [ST95], les convictions sont définies comme des expressions logiques de premier ordre auxquels on associe un degré de conviction. Les convictions sont classées en convictions fortes et convictions faibles. Les convictions fortes restent intactes quelque que soit les règles ou connaissances générées. Les convictions faibles ce sont des convictions que l'utilisateur est prêt à les changer avec la présence de nouvelles évidences.

3.4.2.3 Les attentes

La méthode de [LH96] consiste à extraire les motifs qui vont correspondre aux attentes de l'utilisateur. Les attentes ont la même syntaxe que les règles qui sont considérées comme des règles floues qui sont définies avec des variables floues. La technique est basée sur l'appariement flou des patterns découverts avec un ensemble de patterns spécifiés par l'utilisateur. Les niveaux d'appariement sont alors utilisés pour classer les patterns découverts. L'approche présentée dans [LH96] capture une mesure de règle "distance" basée sur une comparaison syntaxique entre une règle et une conviction. Une règle et une conviction sont "différentes" si les conséquences de la règle et de la conviction sont "semblables" mais les antécédents sont "éloignés" ou les conséquences sont "éloignés", mais les antécédents sont

"semblables", où "la similarité" et "la différence" sont définies syntaxiquement basées exclusivement sur la structure des règles.

3.4.2.4 Les impressions générales

[LH96] utilise 2 modèles de représentation des croyances de l'utilisateur : General Impressions (GI) et Reasonably Precise Knowledge (RPK).

Avec les impressions générales, l'utilisateur n'a pas de concept détaillée mais juste des vagues sentiments (Par exemple, dans un domaine de prêt de logement, l'utilisateur peut estimer que la possession d'un haut salaire mensuel augmente sa chance d'obtenir un prêt). Avec les connaissances relativement précises (PRK) l'utilisateur a une idée très précise (Par exemple, dans le même domaine de prêt, l'utilisateur peut croire que si son salaire mensuel est plus de 5.000\$, on accordera un prêt. Évidemment, l'utilisateur ne peut pas être sûr si que ce sont exactement 5.000\$. Il y a un flou entourant la valeur 5000\$ dans son/son esprit.)

Le problème de PRK c'est un problème d'attente pour sélection et classification des règles sur la base d'une comparaison syntaxique.

3.5 L'intérêt de classement des règles d'associations

L'utilisation d'un algorithme d'extraction des règles d'association de type Apriori permet d'extraire toutes les règles d'association au dessus de seuils de fréquence et de confiance spécifiés par l'utilisateur. La plupart des algorithmes de data mining génèrent un grand nombre de règles qui sont valides mais évidents ou peu intéressants. Le grand nombre des règles générées rendent l'inspection manuelle très délicate et les tâches d'interprétation des résultats par les experts, très fastidieuses et coûteuse. Pour cette raison des méthodes de filtrages et de classement de motifs sont nécessaires pour sélectionner les motifs les plus intéressants, afin de faciliter la phase de post-traitement. Un moyen de sélectionner les règles intéressantes parmi l'ensemble des résultats est de les classer selon une mesure d'évaluation. Le classement des règles par ordre d'intérêt, aide dans le choix des règles les plus pertinentes et permet à l'utilisateur de concentrer son analyse sur ces règles.

L'approche proposée a pour but de faciliter la tâche d'analyse des résultats de data mining, en mettant les règles les plus intéressantes selon une mesure d'évaluation subjective (distance sémantique relative à une ontologie). L'ontologie modélise les connaissances a priori du domaine qui vont servir au calcul des distances et par conséquent au classement des résultats. Les règles d'associations

de rang élevé (haut de liste) issues de classement à base de mesures subjectives (sur la base d'inattendu) sont indicatives des connaissances nouvelles et inconnues.

Le classement des modèles sur la base de mesures l'intérêt est une approche intuitive à la résolution des problèmes de qualité et quantité des règles.

4. Conclusion

La fouille de donnée «data mining » est l'exploration et l'analyse de grandes quantités de données afin d'y découvrir des motifs. Le processus du data mining est un processus qui est à la fois itératif et interactif, incluant le nettoyage de données, l'intégration de données, la sélection de données, la transformation de données, l'exploration de données, l'évaluation des motifs et la présentation de connaissances. Le nombre de règles générées à partir d'un processus de data mining est très important. La définition de mesures s'avère nécessaire pour sélectionner et filtrer les règles les plus intéressantes. Certaines de ces mesures sont le fruit de calculs de probabilités comme par exemple l'intérêt, la conviction, le support ou la confiance. Une autre catégorie de mesures utilise des connaissances du domaine. Ces méthodes sont alors dites subjectives, par opposition aux premières qualifiées d'objectives.

Les approches objectives sont fondées sur la force des relations de dépendance entre les éléments où l'intérêt est considéré comme étant directement proportionnel à cette force. Elles utilisent la propriété du support pour filtrer les règles candidates, puis à partir des règles obtenues, une ou plusieurs mesures de qualité objectives sont appliquées pour sélectionner les meilleures règles.

Les approches subjectives utilisent des modèles pour représenter la connaissance de l'utilisateur qui sera utiliser pour un appariement avec la connaissance extraite par les algorithmes de data mining dans un but de classement ou filtrage.

Chapitre 2

Représentation de la connaissance et ontologie

1. Introduction
2. Représentation de la connaissance
 - 2.1. Langages de représentation des connaissances
3. Ontologie
 - 3.1. Composantes d'une ontologie
 - 3.2. Le cycle de vie des ontologies
 - 3.3. Construction d'ontologies
 - 3.4. Typologie des ontologies
 - 3.5. Langages de représentation des ontologies
 - 3.6. Ontologie et représentation des connaissances
 - 3.7. Ontologie floue
4. Similarités entre concepts
 - 4.1 Mesures reposant sur la distance
 - 4.2 Mesures reposant sur le contenu en information
 - 4.3 Mesures Mixtes
5. Logique floue
 - 5.1. Variables linguistiques floues
 - 5.2. Les ensembles flous et la fonction d'appartenance
 - 5.3. Opérations logiques sur les ensembles flous
6. Conclusion

1. Introduction

Un système utilisant la connaissance doit avoir accès non seulement aux termes utilisés par l'être humain mais également à la sémantique qui leur est associée. Les ontologies visent à représenter cette connaissance en étant à la fois interprétables par l'homme et par la machine.

Une ontologie est une conceptualisation d'un domaine d'intérêt, elle vise à représenter les connaissances dans un but d'interprétation aussi bien par l'homme que par la machine. L'utilisation d'ontologies en informatique vise à intégrer une couche de connaissances aux systèmes afin de permettre des traitements élaborés de l'information qu'ils manipulent. Cependant, le formalisme conceptuel soutenu par l'ontologie peut ne pas être suffisant pour représenter des connaissances imprécises qui se trouvent dans beaucoup de domaines d'application en raison du manque de limites bien définies entre les concepts des domaines (le sens de concepts flous). La connaissance par sa nature est lexicalement imprécise et non catégorique, elle contient l'ambiguïté et le flou. Les ensembles flous ont été introduits pour représenter et manipuler des connaissances avec des incertitudes en utilisant les fonctions d'appartenance à des ensembles flous. Elle a été introduite par L.A. Zadeh dans la fin des années 60 comme extension de la logique booléenne pour prendre en charge le concept de la vérité partielle « partial truth ».

Les ontologies sont utilisées dans l'intelligence artificielle, le Web sémantique, le génie logiciel, informatique biomédicale, de la bibliothèque des sciences et de l'architecture de l'information comme une forme de représentation des connaissances sur le monde ou une partie de celui-ci.

2. Représentation de la connaissance

Il convient tout d'abord de caractériser la connaissance par rapport aux termes de donnée, information, processus et connaissance.

La donnée est la moins porteuse de sens de tous ces termes. Les données ne sont ni vraies, ni fausses, ni significatives à moins d'être récupérées, représentées et réinterprétées. Les données deviennent **informations** quand elles prennent un sens soit pour le système soit pour l'utilisateur. L'information, constituée de données, devient **connaissance** à partir du moment où elle sert de fondement à une inférence, au déclenchement d'un processus [Lam02].

Représenter la connaissance a pour objectif de modéliser la connaissance en omettant certains détails non significatifs pour en permettre une meilleure manipulation [Kay97]. La représentation ne correspond pas à l'entité dans son intégralité. Prenons par exemple une carte routière, son intérêt est de représenter une région afin de pouvoir prévoir un déplacement. Une carte à taille réelle n'aurait aucun intérêt. Une représentation est une structure composée de symboles construits à partir d'un ensemble de règles de formation [Kay97]. L'ensemble des règles de formation est défini par le langage de représentation choisi. Un ordinateur gère des symboles, il est médiateur de la connaissance, comme l'est un livre. L'utilisateur de l'ordinateur accède, lui, à la sémantique associée à la représentation [Bac99].

La représentation des connaissances est un domaine de l'intelligence artificielle qui est préoccupé par la manière d'utiliser un système de symboles pour représenter «un domaine de discours» - ce qui peut être parlé, avec des fonctions qui peuvent être ou pas dans le domaine des discours qui permettent l'inférence (raisonnement formalisé) sur les objets dans le domaine du discours à se produire.

2.1. Langages de représentation des connaissances

L'élément de tout système basé sur la connaissance est une base de connaissances. Une base de connaissances est capable de stocker des données mais est également capable de leur associer une représentation formelle, i.e. associée à une sémantique clairement définie et conçue pour être interprétée par des programmes. La représentation formelle de connaissances a pour objectif de rendre celles-ci interprétables aussi bien par une machine que par un être humain.

Les connaissances peuvent ainsi spécifier des relations et des contraintes sur les données de telle sorte que les programmes puissent raisonner sur ces données

pour en déduire de nouvelles connaissances. Les bases de connaissances s'appuyant sur des langages de représentation des connaissances afin non seulement de fournir une structure appropriée pour stocker les données mais surtout pour leur associer une interprétation du domaine considéré.

Dans ce qui suit on va présenter les grandes familles de langages de représentation de connaissances.

2.1.1. Réseau sémantique [Qui68] : est une représentation graphique d'une conceptualisation d'une (ou plusieurs) connaissance humaine [Qui68]. Il est représenté sous la forme d'un graphe étiqueté et orienté. Un arc lie un noeud de départ à un noeud d'arrivée. De nombreuses études [Woo75], [Bra77] ont montré que ce type de graphe manque de précision sémantique et mène à des confusions entre les relations et aussi entre les classes et individus. Elles ont mené à la définition de nouveaux formalismes tels que les frames, les logiques de description et les graphes conceptuels.

2.1.2. Les frames [Min75] sont présentés comme étant une structure de données capable de représenter des objets structurés. Un frame représente une classe ou un objet. Les frames sont organisés dans une hiérarchie suivant un lien de spécification. Les composants du frame sont appelés « slots », ils sont considérés comme des attributs de la structure. L'information peut être partagée entre plusieurs frames grâce à l'héritage.

2.1.3. Les graphes conceptuels [Sow84] : ont été présentés par Sowa en 1984 [Sow84] et utilisent une notation à base de graphes. Ils ont été définis comme un langage pivot entre le langage naturel et la logique du premier ordre. Ils visent à formaliser les relations entre prédicats et arguments dans une phrase.

2.1.4. Les logiques de description (LD) [BH91] : issues des frames reposent sur trois notions de base : les concepts représentant des classes (ensemble d'objets), les rôles (relations liant deux objets) et les individus. Pour décrire ces éléments, deux structures sont utilisées : la T-BOX et la A-BOX. La T-BOX (boîte terminologique) comprend la description des concepts et des rôles. Cette description est structurée à l'aide du lien hiérarchique sorteDe. Deux concepts particuliers figurent au minimum dans la T-BOX : le concept le plus générique (anything) et le concept le plus spécifique (nothing). La A-BOX (boîte assertionnelle) est constituée des individus, de leur description et des règles qui leur sont attachés. Les inférences reposent sur la reconnaissance d'instances de concepts à partir de leur définition, la détection des

concepts plus généraux ou plus spécifiques, et la classification ordonnant les concepts dans la hiérarchie. Les logiques de description reposent sur une sémantique et une syntaxe rigoureuses [BH91].

4. Ontologies

« Ontologie » est un terme emprunté à la philosophie qui implique une branche de la philosophie qui traite la nature et l'organisation de la réalité. Plusieurs définitions d'ontologies sont données dans la dernière décade, mais celle qui caractérise l'essentiel d'une ontologie est fondée sur la définition reliée dans [Gru93a] : « Une ontologie est une *spécification formelle*, et *explicite* d'une *conceptualisation partagée* ». *Conceptualisation* réfère à un modèle abstrait de certains phénomènes dans le monde qui identifie les concepts appropriés de ce phénomène. *Explicite* signifie que le type de concepts utilisés et les contraintes sur leur utilisation doivent être explicitement définis. *Formelle* réfère au fait qu'une ontologie doit être compréhensible par la machine, c'est-à-dire cette dernière soit capable d'interpréter la sémantique de l'information fournie. *Partagée* indique que l'ontologie supporte la connaissance consensuelle, et elle n'est pas restreinte à certains individus mais acceptée par un groupe [BKD+01].

Qu'est-ce qui fait partie d'une ontologie de domaine est limité par la capacité de représentation de méta-modèle (le langage utilisé pour construire le modèle). Chaque représentation des connaissances dans le langage diffère de sa manière et sa capacité d'expression. En général, une ontologie se compose de trois parties: définitions concept, définitions des rôles et de définitions de l'inférence.

Typiquement, une ontologie contient une description hiérarchique des concepts importants d'un domaine et décrit les propriétés de chaque concept à travers un mécanisme d'attributs valeurs. Des relations entre les concepts peuvent être décrites au moyen de phrases logiques ou axiomes. Selon le langage utilisé pour définir l'ontologie, certaines notions peuvent être introduites pour la modélisation du domaine et du mécanisme d'inférence.

L'ontologie peut jouer un rôle important dans les domaines de *collaboration*, *interopérabilité*, *éducation* et *modélisation* [Fik98].

3.1. Composantes d'une ontologie

Les connaissances traduites par une ontologie sont à véhiculer à l'aide des éléments suivants [GP99] :

- **Les concepts**, aussi appelés termes ou classes de l'ontologie, correspondent aux abstractions pertinentes d'un segment de la réalité (le domaine du problème), retenues en fonction des objectifs qu'on se donne et de l'application envisagée pour l'ontologie. Selon [GP99] ces concepts peuvent être classifiés selon plusieurs dimensions : 1) niveau d'abstraction (concret ou abstrait) ; 2) atomicité (élémentaire ou composée) ; 3) niveau de réalité (réel ou fictif). Un concept représente pour un objet matériel, une notion ou une idée [UK95]. Il est composé de trois parties : un ou plusieurs termes, une notion et un ensemble d'objets. La notion correspond à la sémantique du concept, elle est définie à travers ses propriétés et ses attributs. La notion est appelée intention du concept. L'ensemble d'objets correspond aux objets définis par le concept, il est appelé extension du concept ; les objets sont les instances du concept. Le ou les termes permettent de désigner le concept. Ces termes sont aussi appelés labels de concept. Un concept est défini à partir d'une sémantique référentielle (due à son extension) et une sémantique différentielle (due à son intention). Un concept peut avoir une extension vide, c'est le cas des concepts génériques ou abstraits comme par exemple « la vérité ». Deux concepts peuvent avoir la même extension et des intentions différentes. Il est considéré par certains auteurs que l'intention d'un concept permet à elle seule de définir le sens d'un concept [GCG94] ; d'autres auteurs considèrent que le sens dépend de l'intention et de l'extension du concept [KP99].
- **Les relations** traduisent les associations (pertinentes) existant entre les concepts présents dans le segment analysé de la réalité. Ces relations incluent les associations suivantes: 1) Sous-classe-de (généralisation – spécialisation) ; 2) Partie-de (agrégation ou composition) ; 3) Associée-à ; 4) Instancede, etc. Ces relations nous permettent d'apercevoir la structuration et l'interrelation des concepts, les uns par rapport aux autres.
- **Les fonctions** constituent des cas particuliers de relations, dans laquelle un élément de la relation, le nième (extrant) est défini en fonction des n-1 éléments précédents (intrants).
- **Les axiomes** sont des expressions qui sont toujours vraies. Leur inclusion dans une ontologie peut avoir plusieurs objectifs : définir la signification des composants, définir des restrictions sur la valeur des attributs, définir les arguments d'une relation, vérifier la validité des informations spécifiées ou en déduire de nouvelles.

- **Les instances** constituent la définition extensionnelle de l'ontologie ; ces objets véhiculent les connaissances (statiques, factuelles) à propos du domaine du problème.

3.2. Le cycle de vie des ontologies

Un cycle de vie inspiré du génie logiciel est proposé dans [DIE01]. Il comprend une étape initiale **d'évaluation des besoins**, une étape de **construction ou conception**, une étape de **diffusion**, et une étape **d'utilisation**. Après chaque utilisation significative, l'ontologie et les besoins sont **réévalués** et l'ontologie peut être étendue et, si nécessaire, en partie reconstruite.

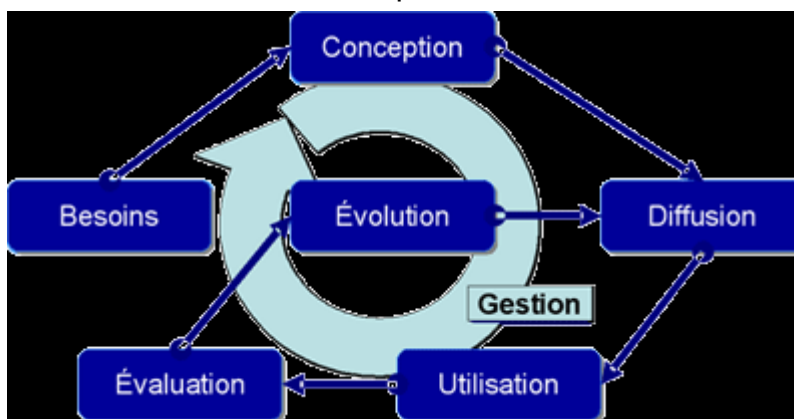


Figure 2-1 Le cycle de vie d'une ontologie [URL05]

3.3. Construction d'ontologies

Plusieurs méthodologies ont été définies pour cadrer le développement d'ontologies de domaine (TOVE [GF95], ENTREPRISE [UK95], METHONTOLOGY [FER97]).

Quelque soit la méthodologie adoptée, les objectifs de construction de l'ontologie et les besoins qui en découlent doivent être clairement définis [FRÉ07], le domaine de connaissance délimité et si nécessaire découpé en termes de connaissances du domaine, connaissances de raisonnement et connaissances de haut niveau, le degré de formalisme de l'ontologie et sa granularité précisée [UK95]. Le processus de construction de l'ontologie est constituée de trois phases, en conceptualisation, formalisation et implémentation.

3.3.1. Conceptualisation

La conceptualisation consiste à identifier, les connaissances du domaine. Il s'agit des concepts, relations, propriétés des concepts et relations, règles, contraintes, etc. La découverte des connaissances d'un domaine peut s'appuyer à la

fois sur l'analyse de documents et sur l'interview d'experts du domaine. Une fois les concepts et relations identifiés par leurs termes, il faut en décrire la sémantique en indiquant, à priori en langage naturel, leurs instances connues, les liens qu'ils entretiennent entre eux, leurs propriétés. Le résultat de la conceptualisation est un modèle conceptuel d'ontologie « *Ontology Conceptual Model* ».

3.3.2. Ontologisation ou Formalisation

C'est la traduction de la conceptualisation dans un formalisme de représentation des connaissances, par exemple DL. L'ontologisation consiste en la structuration sémantique et la formalisation des connaissances. L'ontologisation doit mener à la construction de hiérarchies de concepts, de relations, et attributs des concepts. Une fois le modèle conceptuel structuré, il faut le traduire dans un langage semi-formel de représentation d'ontologies par exemple DL (description logic).

3.3.3. Opérationnalisation ou Implémentation

Il s'agit de coder l'ontologie formalisée en un langage de représentation des connaissances, par exemple OWL. L'opérationnalisation est la transcription de l'ontologie dans un langage formel et opérationnel de représentation de connaissances pour permettre à une machine, de manipuler ces connaissances du domaine via cette ontologie.

3.4. Typologie des ontologies

Le critère pour la classification des ontologies est le contenu de la connaissance qu'elles représentent, c'est-à-dire le sujet de la conceptualisation [Gua98].

- **Les ontologies de haut niveaux** : appelées aussi Modèles d'ontologies ou ontologies TOP, décrivent des concepts très généraux qui sont indépendant d'un domaine particulier .Il s'agit par exemple des concepts de temps, d'espace, d'événement (exp. CYC)
- **Les ontologies de domaine et les ontologies dédiées à une tâche** : En utilisant les termes des ontologies de haut niveau, décrivent le vocabulaire pour une tâche ou un domaine générique. Le vocabulaire est généralement lié à un domaine de connaissances générique comme la médecine ou le droit. Les concepts d'une ontologie de domaine sont souvent définis comme une spécialisation des concepts des ontologies de haut niveau (exp. Entreprise).

- **Les ontologies d'application** : C'est une spécialisation des ontologies du domaine et de tâche pour une activité particulière.

Basé sur la fonction de leur usage, [GD06] classifie les ontologies aux cinq catégories : génériques, de domaine, d'application, de représentation et de méthodes.

- **L'ontologie générique** : elle décrit des concepts génériques et indépendants d'un domaine ou d'un problème particulier. WordNet par exemple est une ontologie dont le but est de représenter la langue naturelle anglaise [NH05].
- **L'ontologie de domaine** : Cette catégorie spécifie le point de vue sur un domaine cible. Les concepts d'une ontologie sont liés à un domaine de connaissances générique comme l'informatique, le commerce, la loi, etc. Une ontologie de domaine constituée de
 - o Une description en extension du vocabulaire du domaine
 - o Une typologie
 - o Une classe hiérarchie
- **L'ontologie d'application** : définit la structure des connaissances nécessaire à la réalisation d'une tâche particulière.
- **L'ontologie de représentation** : Une ontologie de représentation nous fournit une structure de représentation et des primitives pour décrire les concepts des ontologies de domaine et des ontologies génériques. On les désigne également comme ontologies abstraites ou de haut niveau parce qu'elles permettent de définir des concepts abstraits et peuvent être re-utilisées pour définir des concepts spécifiques [NH05].
- **L'ontologie de méthode** : décrit le processus de raisonnement d'une façon d'indépendante d'un domaine. Elle nous fournit les concepts et les relations pour spécifier un processus de raisonnement.

3.5. Langages de représentation des ontologies

Différents langages de spécification d'ontologies issus des formalismes précédemment présentés sont apparus à partir des années 1990, tels que CycL et KIF [GF94], LOOM [Mac91], F-Logic [KLW95], OCML et des langages orientés Web Sémantique OIL (Ontology Inference Layer), XOL (XML based Ontology Exchange Language), SHOE (Simple HTML Ontology Extensions), TOPIC MAPS, RDF et RDF

Schéma, OWL (Ontologie Web Language) [NH05]. Dans cette section on va présenter deux langages recommandés par le W3C à savoir RDF/RDF Schéma et OWL.

3.5.1. RDF et RDF Schéma

RDF (Resource Description Framework) est une recommandation du W3C pour décrire des ressources. C'est un modèle de graphe pour décrire les méta-données en permettant leur traitement automatisé. A l'origine, il a été défini pour décrire des ressources du Web telles que les pages Web ; cependant une ressource peut être toute chose ayant une identité (objet physique, concept abstrait, etc.). Le RDF est basé sur la notion de triplet (sujet, prédicat, objet). Le sujet et l'objet sont des ressources liées par le prédicat. RDF utilise la syntaxe XML, mais il ne donne aucune signification spécifique pour le vocabulaire comme (sous-classe ou type). Les primitives de modélisation offertes par RDF sont très basiques.

RDF Schéma [GB04] fournit un ensemble de primitives simples, mais puissantes, pour la structuration de la connaissance d'un domaine en classes et sous-classes, propriétés et sous-propriétés avec la possibilité de restreindre leur domaine d'origine (`rdf :domain`) et leur domaine d'arrivée (`rdf :range`). Il est reconnu comme un langage d'ontologie qui définit des classes, des propriétés, des sous-classes, des super-classes, des sous-propriétés, des super-propriétés, le domaine de définition et le domaine image des propriétés.

3.5.2. OWL

Développé par le groupe de travail sur le Web Sémantique du W3C, OWL peut être utilisé pour représenter explicitement les sens des termes des vocabulaires et les relations entre ces termes. OWL vise également à rendre les ressources sur le Web aisément accessibles aux processus automatisés [VHM04], d'une part en les structurant d'une façon compréhensible et standardisée, et d'autre part en leur ajoutant des méta informations. Pour cela, OWL a des moyens plus puissants pour exprimer la signification et la sémantique que XML, RDF, et RDF-S [GB04]. De plus, OWL tient compte de l'aspect diffus des sources de connaissances et permet à l'information d'être recueillie à partir de sources distribuées, notamment en permettant la mise en relation des ontologies et l'importation des informations provenant explicitement d'autres ontologies [SWM04].

OWL a trois sous langages de plus en plus expressifs: OWL Lite, OWL DL, et OWL Full :

- **OWL Lite** : supporte les utilisateurs ayant besoin principalement d'une hiérarchie de classification et des contraintes simples (un ensemble est limité à 0 ou 1 élément, par exemple). Il a une complexité formelle inférieure à celle de OWL DL. OWL Lite supporte seulement un sous-ensemble de constructions du langage OWL.
- **OWL DL** : D'après son nom OWL DL utilise la logique de description DL [BCM+03]. Il supporte les utilisateurs qui réclament l'expressivité maximale tout en retenant la complétude informatique (toutes les conclusions sont garanties d'être calculables), et la possibilité de décision (les calculs finiront en un temps fini). Il inclut toutes les constructions du langage OWL, qui ne peuvent être utilisées que sous certaines restrictions.
- **OWL Full** : a été défini pour les utilisateurs qui veulent une expressivité maximale et une liberté syntaxique de RDF sans des garanties informatiques. OWL Full permet à une ontologie d'augmenter la signification du vocabulaire prédéfini (RDF ou OWL). Il est peu probable que n'importe quel logiciel de raisonnement soit capable de supporter le raisonnement complet de chaque caractéristique de OWL Full. Autrement dit, en utilisant OWL Full en comparaison avec OWL DL, le support de raisonnement est moins prévisible puisque l'implémentation complète de OWL Full n'existe pas actuellement.

3.6. Ontologie et représentation des connaissances

Le problème posé par la représentation des connaissances est de formaliser dans un langage formel de représentation les connaissances permettant de traiter le problème à résoudre. Traditionnellement, un langage formel se définit à partir d'un alphabet comprenant des symboles de variables, des symboles de fonction et des symboles de prédicat et des symboles d'opérateurs logiques (connecteurs), des quantificateurs, etc.

L'utilisation de langages de représentation des connaissances permet d'associer aux concepts et aux relations une description formelle qui fait référence à une sémantique clairement définie comme dans le cas de LD. Cela permet d'associer aux concepts et relations de l'ontologie une interprétation unique qui puisse ainsi être comprise de la même façon par les humains ou les machines. Le fait qu'une ontologie soit associée à une seule interprétation ne veut pas dire que pour un domaine il n'existe qu'une seule conceptualisation et qu'une seule interprétation.

La connaissance contenue dans l'ontologie peut représenter plusieurs structures [NH05].

- Les ontologies terminologiques ou linguistiques spécifient les termes utilisés pour représenter la connaissance d'un domaine. Un exemple de ce type d'ontologie est le réseau sémantique UMLS (Unified Medical Language System).
- Les ontologies de l'information spécifient la structure des enregistrements d'une base de données. Les schémas de base de données en sont un exemple. Elles proposent un cadre de représentation de la connaissance stockée mais ne spécifient pas de détails sur la sémantique des champs.
- Les ontologies pour la modélisation de la connaissance spécifient la conceptualisation de la connaissance. Ces ontologies ont une structure beaucoup plus riche que celle des deux autres types. Elles sont généralement conçues en fonction de l'utilisation prévue de la connaissance qu'elles contiennent.

3.7. Ontologie floue

L'ontologie est une conceptualisation d'un domaine dans un format compréhensible par l'homme, directement exploitable par la machine se composant d'entités, attributs, relations et axiomes [FvHH+01]. Cependant, le formalisme conceptuel soutenu par l'ontologie peut ne pas être suffisant pour représenter des connaissances incertaines qui se trouvent dans beaucoup de domaines d'applications en raison du manque de limites bien définies entre les concepts des domaines. De plus, la connaissance floue joue un rôle important dans beaucoup de domaines qui font face à une énorme quantité de connaissances imprécises et vagues, tels que text mining, multimedia, information system, medical informatics et le traitement du langage naturel [ZK92]. Pour manipuler l'incertitude des connaissances, une solution possible est d'incorporer la théorie floue dans l'ontologie. Alors nous pouvons produire des ontologies floues, qui contiennent des concepts flous et des appartenances floues. Les ontologies floues sont capables de s'occuper de la connaissance floue [WY01], elles sont basées sur le concept que chaque objet est en relation avec les autres objets dans l'ontologie, avec un degré d'appartenance μ affectée à cette relation. La valeur d'appartenance μ ($0 \leq \mu \leq 1$).

D'une manière formelle, une ontologie (O) organise la connaissance de

domaine en termes de concepts (C), propriétés (P), relations (R) et axiomes (A) et peut être formellement défini comme 4-tuple $O = (C, P, R, A)$ [JZ+07], où :

- C est un ensemble de concepts définis pour le domaine. Un concept est souvent considéré comme une classe dans une ontologie.
- P est un ensemble de propriétés de concept. Une propriété $p \in P$ est définie comme un cas d'une relation ternaire de la forme $p(c, v, f)$, où $c \in C$ est un concept d'ontologie, v est une valeur de propriété associée à c et f définit des facettes de restriction sur v . Quelques facettes de restriction sont – le type (t_f), cardinality (c_f) et le range (r_f). La facette de type (t_f) peut être un des types de données standard soutenus par les rédacteurs d'ontologie, c'est-à-dire $(t_f) \in \{\text{boolean, integer, float, string, symbol, instance, class, ...}\}$. La facette cardinality (c_f) définit les limites supérieures et inférieures sur le nombre de valeurs pour la propriété. La facette de range (r_f) spécifie un interval de valeurs qui peuvent être assignées à la propriété.
- $R = \{r \mid r \subseteq C \times C \times R_t\}$ est un ensemble de relations sémantiques binaires définies entre les concepts dans C. $R_t = \{\text{one-to-one, one-to-many, many-to-many}\}$ est l'ensemble du type de relation. Un ensemble de relations fondamentales ou de base est défini comme $\{\text{synonym of, kind of, part of, instance of, property of}\} \subseteq R$.
- A est un ensemble d'axiomes. Un axiome est un fait réel ou une règle d'inférence.

L'ontologie floue est créée comme une extension à l'ontologie standard de 4-tuple $O_F = (C, P_F, R_F, A_F)$ où :

- C est un ensemble de concepts. A la différence de la définition précédente, chaque concept a ici quelques propriétés dont la valeur est concept floue ou ensemble floue.
- P_F est un ensemble de propriétés. Une propriété $p_F \in P_F$ est définie comme une 5-tuple de la forme $p_F(c, v_F, q_F, f, U)$, où $c \in C$ est un concept d'ontologie, v_F représente des valeurs de propriété, q_F les modélise les qualificatifs linguistiques, qui peuvent contrôler ou changer la force d'une propriété de valeur v_F , f est les facettes de restriction sur v_F et U est l'univers de discours. v_F et q_F sont les

concepts flous sur U , mais q_F change le degré flou de v_F . Par exemple, "le prix" est une propriété du concept "fruit". La valeur "prix" peut être le concept flou "bon marché" ou le nombre flou "environ 50" et les qualificatifs linguistiques peuvent être "très", "peu", "près" etc. Donc, la valeur finale "de prix" peut être "très bon marché" ou "peu chère".

- R_F est un ensemble de relations d'inter-concept entre les concepts. Comme les propriétés de concept flou, $r_F \in R_F$ est défini comme une 5-tuple de la forme $r_F(c_1, c_2, t, s_F, U)$ ou $c_1, c_2 \in C$ sont des concepts d'ontologie, t représente le type de relation, U est l'univers de discours et s_F les forces de relation de modèles qui est un concept flou sur U , qui peut représenter la force d'association entre les paires du concept $\langle c_1, c_2 \rangle$.
- A_F est un ensemble de règles floues. Dans un système flou l'ensemble des règles floues est utilisé comme base de connaissances « knowledge base ».

4. Similarités entre concepts

L'évaluation du lien sémantique entre deux concepts dans une ontologie est un problème de longue date dans le domaine de l'intelligence artificielle et de la psychologie [NH05]. La similarité sémantique est une évaluation du lien sémantique entre deux concepts dont le but est d'estimer le degré par lequel les concepts sont proches dans leur sens [Res99]. La similarité entre deux concepts est liée aux caractéristiques qu'ils ont en commun. Rada et al. [RMBB89] ont suggéré que la similarité dans un réseau sémantique peut être calculée en se basant sur les liens taxonomiques « is-a ». Plus généralement, le calcul de la similarité entre concepts peut être basé sur les liens hiérarchiques de spécialisation/généralisation.

Deux catégories principales d'approches pour calculer la distance sémantique entre les termes organisés dans une structure hiérarchique ont été proposées dans la littérature [JC97]: les approches basées sur la distance et les approches basées sur le contenu d'information. L'idée générale derrière les approches basées sur la distance [LC98, RMBB89, WP94] est de trouver le chemin le plus court entre deux concepts en nombre d'arcs et ensuite, traduire cette distance en une distance sémantique. Les approches basées sur le contenu d'informations [JC97, RMBB89] sont inspirées par la perception que les paires de concepts qui partagent beaucoup de contextes communs sont sémantiquement rattachées.

4.1. Mesures reposant sur la distance

Les mesures reposant sur la distance considèrent que la similarité entre deux concepts peut être calculée à partir du nombre de liens qui les séparent. La mesure du "edge counting" [RMBB89] évalue la distance sémantique à partir du nombre de branches séparant les concepts, par le plus court chemin dans la hiérarchie.

La similarité sémantique entre deux concepts correspond à l'inverse de la distance entre deux concepts.

Dans ce qui suit on va présenter 3 mesures basées sur la distance

(1) La mesure de « Edge Counting »

La distance sémantique est définie comme le nombre d'arcs (les noeuds) le long du chemin le plus court entre n'importe quels deux termes.

(2) La mesure de « Leacock & Chodorow »

La mesure de distance proposée par Leacock & Chodorow compte aussi sur le chemin le plus court entre deux termes, t_1 et t_2 . On calcule la similarité entre deux termes t_1 et t_2 comme suit [LC98] :

$$relatedness(t_1, t_2) = -\log(len(t_1, t_2) / 2D) \quad \text{où}$$

$relatedness(t_1, t_2)$ est la similarité des termes t_1 et t_2 ; $len(t_1, t_2)$ est la longueur du chemin le plus court entre deux termes (nombre d'arcs); D est la profondeur maximum de la structure. L'utilisation de la fonction $-\log$ permet de normaliser la similarité entre $[0, 1]$ (1 signifiant que les concepts sont totalement similaires).

La distance sémantique est l'inverse de $relatedness(t_1, t_2)$, elle est égale à

$$\frac{1}{relatedness(t_1, t_2)}$$

(3) La mesure de Wu & Palmer

Cette mesure prend en compte à la fois la profondeur des concepts dans la hiérarchie de concepts et la structure de la hiérarchie de concepts [WP94].

$$Sim_{wu}(c_1, c_2) = \frac{2 * depth(c)}{depth(c_1) + depth(c_2)}$$

où $depth(c_i)$ correspond au niveau de profondeur du concept c_i et c représente le concept le plus spécifique qui généralise c_1 et c_2 .

La valeur de la similarité est comprise entre 0 et 1 (1 signifiant que les concepts sont totalement similaires).

La distance sémantique est l'inverse de similarité

$$\text{Dist}_{WU}(c1,c2)=1/\text{Sim}_{WU}(c1,c2).$$

Pour comparer des groupes de concepts, Rada et autres [Res93] définissent une version étendue de mesure de "distance" d'inter concepts:

$$\text{Distance}(X_1 \wedge \dots \wedge X_k, Y_1 \wedge \dots \wedge Y_m) = \frac{1}{km} \sum_{i=1}^k \sum_{j=1}^m \text{Distance}(X_i, Y_j)$$

La mesure étendue de Rada donne la distance sémantique entre deux "conjonctivité concepts" ou les groupes de concepts qui contiennent k X_i et m Y_j de concepts atomiques respectivement.

Une autre méthode considère seulement le meilleur appariement sémantique parmi les concepts de groupe B pour chaque concept dans le groupe A. Cette méthode donne une mesure asymétrique exprimant la contribution sémantique de concepts par rapport à B. en échangeant A et B, nous pouvons combiner les deux mesures asymétriques dans une mesure symétrique. Azuaje et al dans [JC97] utilisent une telle mesure :

$$\text{Dist}_{azu}(A, B) = \frac{1}{k+m} \left(\sum_{i=1}^k \min_j (\text{Distance}(X_i, Y_j)) + \sum_{j=1}^m \min_i (\text{Distance}(X_i, Y_j)) \right)$$

Où A et B sont deux groupes de k et m concepts respectivement, la Distance est la distance sémantique d'inter concept de Rada.

La distance retenue pour notre approche est la distance de Hausdorff, qui définit la distance entre 2 ensembles $A=\{a_1, \dots, a_p\}$ et $B=\{b_1, \dots, b_q\}$ comme:

$$\text{Distance}(A, B) = \max(h(A, B), h(B, A)) \quad \text{où} \quad h(A, B) = \max_{a \in A} \min_{b \in B} \|a - b\|$$

La fonction $h(A, B)$ est appelé la «distance» orientée de Hausdorff (directed husdroff 'distance') de A à B (cette fonction n'est pas symétrique et n'est donc pas une vraie distance). Elle identifie le point $a_i \in A$ qui est le plus éloigné de tout point de B, et mesure la distance de a_i à son plus proche voisin dans B. La distance de Hausdorff, $\text{Distance}(A, B)$, mesure le degré de décalage entre les deux ensembles, puisque elle reflète de la distance du point de A qui est le plus éloigné de tout point de B et vice versa [HAR93].

4.2. Mesures reposant sur le contenu en information

Les approches reposant sur le contenu en information supposent que l'information détenue par les concepts puisse être quantifiée. La similarité est alors calculée à partir de l'information partagée par les concepts. Les différentes mesures proposées pour évaluer la similarité entre concepts reposent sur ce calcul.

Afin de calculer l'information contenue par les concepts, un corpus de référence est choisi. Les concepts sont alors pondérés par une fonction correspondant à l'information portée par un concept dans le corpus. Le contenu en information du concept est défini par ses occurrences dans le corpus ainsi que celles des concepts qu'il subsume. Il a pour objectif d'utiliser la probabilité d'obtenir un concept dans un corpus. Le contenu en information d'un concept c se calcule de la façon suivante $CI(c) = -\log(p(c))$ [Res95].

Plus deux concepts partagent l'information plus ils sont semblable. L'information partagée par deux concepts est indiquée par le contenu d'information des concepts qui les subsument dans la taxonomie. Formellement

$$sim(c_1, c_2) = \max_{c \in S(c_1, c_2)} [-\log(c)]$$

Où $S(c_1, c_2)$ est l'ensemble des concepts qui subsument c_1 et c_2 à la fois. Une classe qui accomplit la valeur maximum dans l'Équation précédente sera appelée subsumer le plus informatif; le plus souvent il y a un seul subsumer qui est le plus informatif. Le maximum (en ce qui concerne le contenu d'information) correspond à la prise de la première intersection de réseau sémantique ou du chemin le plus court.

En pratique, on a besoin souvent de mesurer la similarité de mot, plutôt que la similarité de concept. L'utilisation $s(w)$ pour représenter l'ensemble des concepts dans la taxonomie qui sont des sens (synonyms) de mot w .

$$sim(w_1, w_2) = \max_{c_1, c_2} [sim(c_1, c_2)] \text{ Où } c_1 \text{ varie sur } s(w_1) \text{ et } c_2 \text{ sur } s(w_2). \text{ C'est en}$$

harmonie avec Rada et. al (1989) le traitement des concepts disjonctives avec l'approche des arcs.

4.3. Mesures Mixtes

Le principe des mesures mixtes est de considérer le plus court chemin reliant deux concepts dans l'ontologie et de pondérer ces liens à partir de leur poids sémantique. Le poids sémantique des liens prend notamment en compte le contenu en information des concepts [RS95] [JC97].

5. La Logique floue

La logique floue a été introduite par L.A. Zadeh dans la fin des années 60 comme extension de la logique booléenne pour prendre en charge le concept de la vérité partielle « partial truth ». Cette logique consiste à être imprécis dans les affirmations, et de répondre à des propositions vagues. Par exemple, en logique classique, la question : Est-ce que cette personne est grande? On ne peut répondre que par vrai, si c'est le cas, ou faux dans le cas contraire. Avec la logique floue, on peut représenter les cas où la personne est «*très petite, moyennement petite, normale, pas très grande, grande, etc.* ».

Une variable comme "taille" peut être exprimée par différentes valeurs. Si on interprète cette variable comme une *variable linguistique*, les valeurs correspondantes seraient des *valeurs linguistiques* « *pas très grand, moyennement petit, assez grand, etc.* ». Les variables linguistiques sont appelées **variables floues**, et les valeurs linguistiques sont des *labels* pour des **ensembles flous**, qui sont définies sur un univers de discours.

5.1. Variables linguistiques floues

Les variables dans les mathématiques prennent d'habitude des valeurs numériques, cependant dans les applications logiques floues, les variables linguistiques «non numériques» sont souvent utilisées pour faciliter l'expression des règles et des faits. La grande utilité de variables linguistiques est qu'elles peuvent être modifiées via les opérations linguistiques sur les termes primaires. Par exemple, si jeune est associé à la valeur 0.7 alors très jeune est automatiquement déduit comme le fait d'avoir la valeur $0.7 * 0.7 = 0.49$ et non très jeune reçoit la valeur $(1 - 0.49)$, c'est-à-dire 0.51 tel que l'opérateur très(x) a été définie comme $x*x$. En général ces opérateurs peuvent être définis avec flexibilité pour correspondre à l'application, ce qui donne un pouvoir supplémentaire d'expression des règles et des faits flous.

5.2. Les ensembles flous et la fonction d'appartenance

Un ensemble flou est une extension d'un ensemble classique. Les ensembles classiques permettent seulement pleine appartenance ou aucune appartenance du tout, alors que les ensembles flous permettent l'appartenance partielle. Dans un ensemble classique, une appartenance ou une non appartenance d'élément x dans l'ensemble A est décrite par une fonction $\mu_A(x)$ tel que $\mu_A(x)=1$ si $x \in A$ et $\mu_A(x)=0$ si $x \notin A$

La théorie des ensembles flous étend ce concept en définissant l'appartenance partielle. Un ensemble flou sur un univers de discours U est caractérisé par une fonction d'appartenance $\mu_A(x)$ qui prend des valeurs dans l'intervalle $[0,1]$.

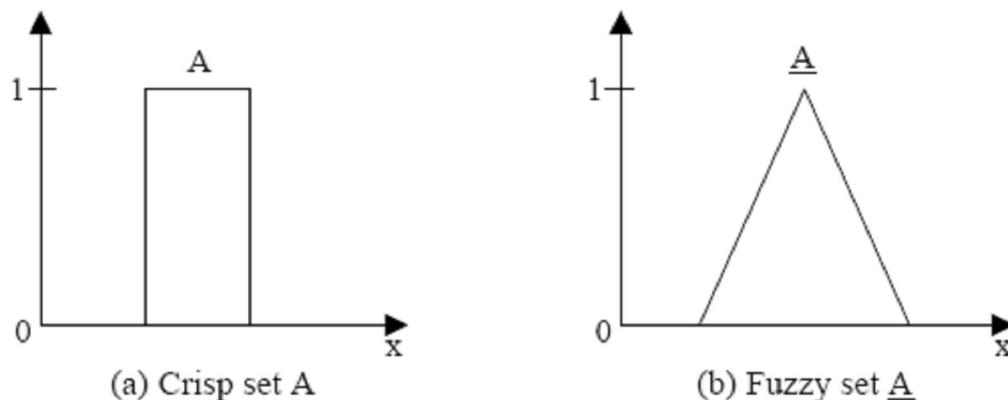


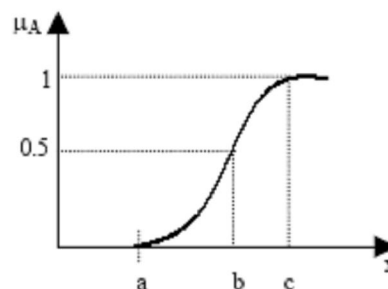
Figure 2-2 La fonction d'appartenance pour un ensemble ordinaire et flou [MH02]

Les ensembles flous représentent les étiquettes linguistiques du bon sens comme lent, rapide, petit, grand, lourd, bas, moyen, haut, grand, etc. Un élément donné peut être un membre de plus qu'un ensemble flou à la fois. Un ensemble flou dans U peut être représenté comme un ensemble de paires ordonnées. Chaque paire se compose d'un élément générique x et de son degré de fonction d'appartenance; c'est-à-dire $A = \{(x, \mu_A(x)) | x \in U\}$ x est appelée la valeur support si $\mu_A(x) > 0$.

Une fonction d'appartenance est essentiellement une courbe qui définit comment chaque point de l'espace des entrées, lui correspond une valeur d'appartenance (ou le degré d'appartenance) entre 0 et 1. Différents types de fonctions d'appartenance sont utilisés, en incluant les courbes triangulaires, trapézoïdales, la cloche généralisée formée, les courbes Gaussiens, les courbes polynomiales et les fonctions de sigmoïdal.

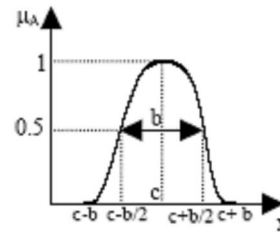
- *S – membership function :*

$$S(x; a, b, c) = \begin{cases} 0 & \text{for } x < a \\ 2[(x-a)/(c-a)] & \text{for } a \leq x \leq b \\ 1-2[(x-c)/(c-a)] & \text{for } b \leq x \leq c \\ 1 & \text{for } x > c \end{cases}$$



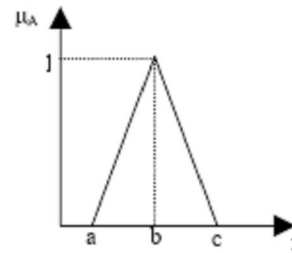
- *Phi – membership function :*

$$\pi(x; b, c) = \begin{cases} S(x; c - b, c - b/2, c) & \text{for } x \leq c \\ 1 - S(x; c, c + b/2, c + b) & \text{for } x \geq c \end{cases}$$



- *Triangular membership function :*

$$T(x; a, b, c) = \begin{cases} 0 & \text{for } x < a \\ (x - a)/(b - a) & \text{for } a \leq x \leq b \\ (c - x)/(c - b) & \text{for } b \leq x \leq c \\ 0 & \text{for } x > c \end{cases}$$



- *Trapezoidal membership function :*

$$\Pi(x; \alpha, \beta, \gamma, \delta) = \begin{cases} 0 & \text{for } x < \alpha \\ (x - \alpha)/(\beta - \alpha) & \text{for } \alpha \leq x \leq \beta \\ 1 & \text{for } \beta \leq x \leq \gamma \\ (\delta - x)/(\delta - \gamma) & \text{for } \gamma \leq x \leq \delta \\ 0 & \text{for } x > \delta \end{cases}$$

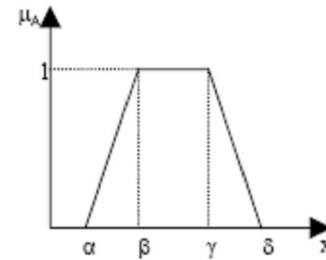


Figure 2-3 Les fonctions d'appartenance [MH02]

5.3. Opérations logiques sur les ensembles flous

Les opérations des ensembles flous ressemblent aux opérations des ensembles classiques. Pour définir les opérateurs logiques sur les ensembles flous, Considérons d'abord les opérateurs sur les ensembles classiques. Les opérations des ensembles classiques les plus élémentaires sont l'union, l'intersection et le complément, qui correspondent essentiellement aux opérateurs OR AND et NOT, respectivement.

Soit A et B deux sous-ensembles de U. L'union de A et de B, dénoté, $A \cup B$ contient tous les éléments soit en A soit en B tel que : $\mu_{A \cup B}(x) = 1$ si $x \in A$ ou $x \in B$.

L'intersection de A et B, dénoté, $A \cap B$ contient tous les éléments qui sont simultanément dans A et B; tel que $\mu_{A \cap B}(x) = 1$ si $x \in A$ et $x \in B$

Le complément de A est dénoté par $\tilde{A}$ et cela contient tous les éléments qui ne sont pas dans A; tel que $\mu_{\tilde{A}}(x) = 1$ si $x \notin A$ et $\mu_{\tilde{A}}(x) = 0$ si $x \in A$.

En logique floue, la vérité de n'importe quelle déclaration est une question de degré. Pour définir des opérateurs de logique floue, nous devons trouver les opérateurs correspondants qui préservent les résultats d'utilisation des opérateurs AND, OR et NOT. La réponse est les opérations min, max et complément. Ces opérateurs sont définis, respectivement, comme

$$\mu_{A \cup B}(x) = \max [\mu_A(x), \mu_B(x)]$$

$$\mu_{A \cap B}(x) = \min [\mu_A(x), \mu_B(x)]$$

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x)$$

Les formules pour les opérateurs AND, OR et NOT dans l'équation précédente sont utiles pour prouver d'autres propriétés mathématiques des ensembles; cependant, le min et max ne sont pas les seules voies de décrire l'intersection et l'union de deux ensembles. [Zad65] a défini l'union floue et l'intersection floue comme

$$\mu_{A \cup B}(x) = \mu_A(x) + \mu_B(x) - \mu_A(x) \mu_B(x)$$

$$\mu_{A \cap B}(x) = \mu_A(x) \mu_B(x)$$

En terme général, l'intersection de deux ensembles flous A et B est spécifié par une correspondance binaire T suivante:

$$\mu_{A \cap B}(x) = T(\mu_A(x), \mu_B(x))$$

Par exemple, l'opérateur binaire T peut représenter la multiplication de $\mu_A(x)$, $\mu_B(x)$.

6. Conclusion

Les ontologies permettent une représentation formelle d'un domaine de connaissance sous la forme d'une terminologie hiérarchisée munie de relations sémantiques. Les approches de la représentation de connaissances sont quelquefois divisées grossièrement dans deux catégories : les formalismes basés sur la logique, qui ont évolués de l'intuition que le calcul de prédicat pourrait être utilisé sans ambiguïté pour capturer des faits du monde; et d'autres, représentations à base non logique.

Une ontologie est une spécification formelle, et explicite d'une conceptualisation partagée [Gru93a]. Typiquement, une ontologie contient une description hiérarchique des concepts d'un domaine et décrit les propriétés de chaque concept à travers un mécanisme d'attributs valeurs. Des relations entre les concepts peuvent être décrites au moyen de phrases logiques ou axiomes.

. La connaissance par sa nature est lexicalement imprécise et non catégorique, elle contient l'ambiguïté et le flou. Les approches conventionnelles à la représentation de connaissances manquent de moyens pour représenter le sens de concepts flous. L'ontologie floue a été introduite pour représenter la connaissance dans tous les domaines dans lesquels les concepts à représenter ont une définition imprécise. Ainsi une instance possède un degré d'appartenance à un concept au lieu d'appartenir ou non, complètement.

Les méthodes de calcul de similarité existantes peuvent être classées en deux catégories, à savoir les approches basées sur la structure et les approches basées sur le contenu d'informations. Les approches basées sur la structure calculent la dissemblance entre concepts en utilisant le plus court chemin entre deux concepts. Les approches basées sur le contenu d'informations utilisent conjointement l'ontologie et le corpus. Le contenu informationnel d'un concept traduit la pertinence d'un concept dans le corpus en tenant compte de sa spécificité ou généralité.

Chapitre 3

L'utilisation des ontologies dans le processus de fouille de données

1. Introduction
2. Ontologies en fouille de données
 - 2.1. Les ontologies dans le cycle de vie CRISP-DM
 - 2.2. Travaux existants sur l'intégration des ontologies dans l'ECD
3. Travaux existants sur les mesures d'intérêt et évaluation
4. Conclusion

1. Introduction

La découverte de connaissance dans les bases de données (data mining) a été défini dans [FPSS96] comme étant "le processus non trivial d'identification, à partir de données, de motifs valides, nouveaux, utiles et compréhensibles". Le processus de data mining est un processus qui est à la fois itératif et interactif.

La connaissance à priori du processus de fouille de données ou du domaine peut aider à choisir les informations appropriées, réduire l'espace des hypothèses, représentent les résultats d'une manière plus compréhensible et d'améliorer l'ensemble du processus.

Une ontologie est une conceptualisation d'un domaine d'intérêt, elle vise à représenter les connaissances dans un but d'interprétation aussi bien par l'homme que par la machine. Principalement, les ontologies sont utilisées pour la communication (entre les machines et / ou humains), raisonnement automatique, et de la représentation et la réutilisation des connaissances (Cimiano et al., 2004).

2. Ontologies en fouille de données

Les ontologies ont été appliquées comme connaissance du domaine en fouille de données pour améliorer le processus de data mining. Les trois principales opérations d'ECD peuvent profiter du domaine de connaissances représentées par des ontologies.

(1) Au cours de l'étape de préparation des données, les ontologies peuvent faciliter l'intégration de données hétérogènes et de guider la sélection de données pertinentes pour être exploités.

(2) Durant le traitement, les connaissances du domaine permettent la spécification des contraintes pour guider les algorithmes d'extraction de données, par exemple la réduction de l'espace de recherche.

(3) Au cours de l'étape d'interprétation, des experts du domaine des connaissances permet de visualiser et de valider les unités d'extraction.

Les ontologies peuvent être appliquées pour la compréhension du domaine d'application, préparation et sélection des données, génération des règles de haut niveau et l'analyse d'intérêt des motifs ou des règles. La figure 3-1 met en claire la relation entre le domaine de connaissance et le processus de data mining selon CRISP-DM où Les flèches supérieures indiquent des boucles de retour « feed-back » et les flèches pointillées indiquent des flux (potentiels) de connaissance de domaine. La connaissance de domaine conduit non seulement la phase initiale de data mining (business understanding) mais elle pourrait aussi être appliqué pour préparer et extraire les données et confronter les résultats [Poh03]. En outre, la connaissance du domaine pourrait à son tour aussi être affectée par les résultats produits, du fait que ces résultats peuvent être en contradiction avec la connaissance initiale et exigeant ainsi des mécanismes de feed-back pour l'actualiser.

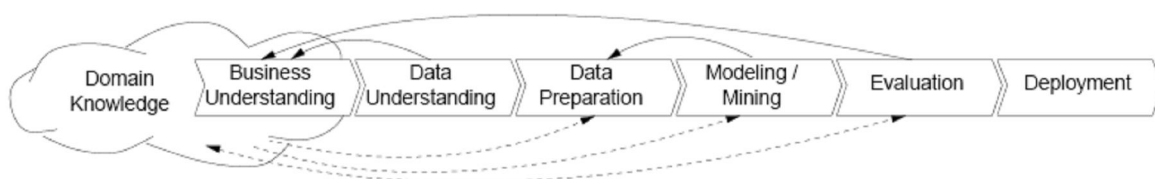


Figure 3-1 Domaine de connaissance et CRISP-DM [Poh03]

2.1. Les ontologies dans le cycle de vie CRISP-DM

Dans [CST04] Cespivova évoque les rôles généraux joués par les ontologies dans un CRISP-DM

- **Compréhension du domaine** : Le rôle d'ontologies dans la Compréhension du domaine n'est pas particulier à l'ECD. Les ontologies semi-formelles peuvent aider un nouveau venu à devenir familier avec les concepts les plus importants et leurs relations, cependant les ontologies formelles permettent d'identifier des hypothèses contradictoires qui ne pourraient pas être évidentes à première vue.
- **Compréhension de Données** : Pour la Compréhension de Données, les éléments d'une ontologie doivent être associés aux schémas de données et vice versa. Cela permet l'identification d'attributs manquants ou redondants.
- **Préparation de Données** : La phase de Préparation de Données est étroitement liée avec la phase de modélisation. L'utilisation réelle d'une ontologie de domaine peut contribuer à l'identification des groupes d'attributs ou de valeurs en exploitant les relations sémantiques.
- **Modélisation** : Dans la phase de modélisation, les ontologies peuvent être utilisées par les algorithmes pour générer des modèles tout en prenant la sémantique de données en considération.
- **Évaluation** : Dans la phase d'évaluation, les modèles découverts sont interprétés selon les concepts de l'ontologie (concepts qui correspondent aux attributs) et confrontés aux connaissances à priori.
- **Déploiement** : Dans la phase de déploiement, le domaine de connaissance peut être alimenté par la connaissance extraite.

2.2. Travaux existants sur l'intégration des ontologies dans l'ECD

On va présenter dans cette section quelques travaux qui ont été réalisés dans le domaine de l'intégration des connaissances expertes dans le processus d'ECD.

[ES04] utilisent une ontologie pour définir une représentation conceptuelle des données pour la phase compréhension des données. Ils proposent un niveau d'abstraction conceptuel au dessus du niveau logique de base de données pour

faciliter l'utilisation et le développement du processus de ECD en utilisant un utilitaire graphique. Cet utilitaire permet la modélisation de la connaissance experte du domaine ainsi que la définition de séquence d'opérations de prétraitement sur la base de la description conceptuelle des données seulement [Bri08],

[BER05] définissent un outil (Intelligent Discovery Assistant) basé sur une ontologie de data mining pour assister dans le processus de ECD. Il a comme rôle d'énumérer les différents modèles valides de data mining et aide dans l'exploration et le choix de l'espace des modèles possibles et valides.

Selon une synthèse des travaux sur les règles d'association de [EYB06], [CZSG03], utilisent une ontologie pendant la phase de prétraitement où les données sont levées aux concepts plus généralisés selon l'ontologie, ensuite le processus de data mining est exécuté selon une règle d'association conventionnelle avec un algorithme comme Apriori [AS94]. Les auteurs considèrent que la généralisation de données précédentes rend possible de considérer des sous catégories dans le calcul de support, et cela permet de générer des règles avec un meilleur support. En outre, les règles obtenues peuvent être plus faciles à interpréter, du fait qu'elles contiennent des concepts de haut niveau qui représentent des renseignements plus riches que les termes spécifiques dans la base de données.

Dans [HGSY05], La connaissance de domaine est utilisée pour généraliser des règles découvertes. Les règles d'associations généralisées constituent une approche différente puisque l'extraction des règles exploite le fait que les termes appartiennent à différents niveaux d'une ontologie. Si l'on connaît les ancêtres d'un terme, alors un critère est appliqué afin de contraindre le processus d'extraction (bloquer les règles qui introduisent à la fois un terme et son ancêtre, par exemple).

Dans [Poh03], les ontologies sont employées pour déterminer l'intérêt de la règle en vérifiant si les règles découvertes confirment, contredisent ou révèlent de nouvelles connaissances quand elles sont comparées à la connaissance disponible dans l'ontologie. En outre, l'auteur propose aussi des mécanismes de feed-back pour actualiser la connaissance de domaine à partir règles produites.

D'autres approches, comme ExCIS [LMP05], utilisent la connaissance de domaine dans les deux phases (prétraitement et post-traitement). Dans leurs travail, la phase de prétraitement utilise une ontologie pour guider la construction de data sets spécifiques pour les tâches de data mining particulières. L'étape suivante est l'application d'un algorithme de data mining standard qui extrait des motifs de ces

data sets. Finalement, dans la phase de post-traitement, les règles générées peuvent être interprétées et/ou filtrées, comme leurs concepts sont généralisés selon une ontologie.

3. Travaux existants sur les mesures d'intérêt et évaluation

Les techniques subjectives généralement comparent les croyances de l'utilisateur contre les règles découvertes par les algorithmes d'extraction de données. Les mesures subjectives déterminent si un modèle est utile ou inattendu. L'utilité « Actionability » se réfère à la capacité des organisations de faire quelque chose d'utile avec le motif découvert. Par conséquent, un motif est dit qu'il est intéressant s'il est à la fois inattendu et utile. Il s'agit clairement d'une vue très subjective, car une l'utilité dépend non seulement du domaine, mais aussi des objectifs de l'utilisateur à un moment donné dans le temps [ST96].

Les études sur les mesures d'intérêt peuvent être classées en deux catégories: des mesures objectives et des mesures subjectives. Les mesures objectives ont été introduites et utilisées pour l'évaluation des propriétés statistiques des modèles extraits (la force statistique du pattern). Cependant l'intérêt d'une règle est essentiellement subjectif [KMR+94, LH96, ST96]. Des mesures d'intérêts dites subjectives comme l'inattendu "unexpectedness" d'un motif dépendent du décideur et pas seulement de la force statistique du motif. Les mesures d'intérêt subjectives confrontent les modèles extraits à des connaissances exprimées sur le domaine.

[KMR+94] propose un modèle basé sur les patrons 'Template'. Dans cette approche, l'utilisateur spécifie les règles intéressante et non intéressante en utilisant des patrons. Un Patron décrit un ensemble de règles en terme d'attributs apparaissant dans l'antécédent et le conséquent. Chacun des attributs appartient à une hiérarchie de concepts. Le système récupère les règles qui match le patron de l'ensemble des règles.

([NLH+98] et [SRV+97]) proposent des algorithmes d'extractions de règles d'associations qui peut prendre les contraintes spécifiées par l'utilisateur dans le processus de génération de règles en sorte que seule les règles qui satisfont les contraintes sont générées. L'idée d'utiliser les contraintes dans le processus d'exploration des règle est permet d'éviter la génération des règles non pertinentes.

[LH96] utilise un algorithme de filtrage et classement floue de règles où les utilisateurs sont supposés décrire leurs connaissances en termes d'ensembles flous et variables linguistiques. La technique est caractérisée en demandant à l'utilisateur

de spécifier un ensemble de modèles en fonction de ses connaissances antérieures. Cette série de motifs spécifiés est utilisée par un algorithme de classement flou.

Par exemple, un algorithme d'extraction génère les motifs suivants d'une base de données d'accidents :

Les motifs générés:

1. If P_Age > 50, Loc = straight then Class = slight
2. If P_Age > 65, Loc = bend, Speed > 50 then Class = killed
3. If P_Age > 50, Loc = T-junct then Class = slight

Le motif spécifié par l'utilisateur est :

If P_Age = OLD, Loc = BAD_VISIBILITY then Class = BAD_ACCIDENT

Après avoir spécifié le sens sémantique de "OLD", "BAD_VISIBILITY" et "BAD_ACCIDENT", un algorithme est ensuite exécuté pour déterminer le degré d'appareillement entre les motifs découverts et les motifs spécifiés par l'utilisateur. Différents algorithmes de classement sont utilisés à des fins différentes. Si l'objectif est de confirmer une hypothèse, l'algorithme classe les motifs découverts de sorte que les motifs avec le plus grand degré d'appareillement sont classés en premier. Les résultats d'un tel classement est comme suit:

Résultats conforme à l'attente

- A1. If P_Age > 65, Loc = bend, Speed > 50 then Class = killed
- A2. If P_Age > 50, Loc = T-junct then Class = slight
- A3. If P_Age > 50, Loc = straight then Class = slight

D'autre part, si le but est de trouver des motifs inattendus dans le sens d'une conséquence inattendue, alors un autre résultat de classement est produit comme indiqué ci-dessous:

Résultats contre les attentes

- B1. If P_Age > 50, Loc = T-junct then Class = slight
- B2. If P_Age > 65, Loc = bend, Speed > 50 then Class = killed
- B3. If P_Age > 50, Loc = straight then Class = slight

Cela montre que le motif B1 est à l'encontre de l'attente de l'utilisateur car au lieu d'une blessure grave, la personne âgée souffre d'une légère blessure.

[PT98], Les auteurs ont mis au point un algorithme capable d'extraire une collection minimale de règles étonnantes par rapport à un ensemble de croyances. La définition de L'inattendu est exprimée sous la forme d'une règles logique tels que : Pour une règle $A \rightarrow B$ à l'égard de la croyance $X \rightarrow Y$ doit satisfaire les conditions suivantes :

- (a) B et Y= FALSE.
- (b) A et X =TRUE sur un échantillon statistiquement large.
- (c) La règle A, $X \rightarrow B$ est TRUE.

Puisque la condition en (a) fait que B et Y sont contradictoires, il s'ensuit logiquement que la règle A, $X \rightarrow \neg Y$ est VRAIS.

[ST96], contrairement à l'approche en [PT98] qui consiste à modéliser les connaissances par un ensemble d'implications logiques puis d'utiliser ces connaissances, les auteurs proposent d'utiliser les probabilités de croyances et la révision des croyances et des convictions en tant que cadre pour la description de l'intérêt subjectif. Concrètement, un système de croyance est utilisé pour définir l'inattendu. Les convictions sont définies comme des expressions logiques de premier ordre auxquels on associe un degré de conviction. Deux types de convictions sont présentés, les convictions fortes «hard» et les croyances faibles «soft». La croyance faible pourrait être révisée et mise à jour à la lumière de nouveaux éléments de preuve. Toutefois, les nouveaux motifs qui sont en contradiction avec les croyances dures, ne déboucherait pas sur la modification des croyances, mais aurait tendance à indiquer que ces motifs sont intéressants. Plusieurs approches ont été proposées pour évaluer le degré de conviction (Approche bayésienne, Approche fréquentielle,...).

[Sah99] a adopté une approche différente à la découverte des motifs intéressants en éliminant les règles d'association non intéressantes. Plutôt que les utilisateurs font définir l'ensemble de leurs connaissances d'un domaine, ils sont invités à identifier plusieurs des règles de non-intéressantes, générée par l'algorithme Apriori. Le système d'extraction des règles, semi-automatiquement supprime les règles qui semblent être similaires aux choix identifiés. Cette approche permet aux experts du domaine d'éviter le processus long et difficile à définir les raisons pour laquelle une règle est intéressante. L'efficacité de cette approche réside dans sa capacité à éliminer rapidement les grandes familles de règles qui ne sont pas intéressantes, tout en limitant l'utilisateur à quelques simples interactions.

La plupart des approches pour la détermination de l'intérêt subjective en utilisant l'inattendu, ont adoptées le schéma suivant :

- Obtenir des vues de l'utilisateur.
- Représenter les vues de l'utilisateur sous une forme adaptée pour le calcul.
- Exécuter le processus de data mining pour en extraire des règles sur le domaine.
- Comparer les règles avec les vues de l'utilisateur afin de déterminer le degré de conflit.
- Présenter les règles qui sont en conflit avec les vues de l'utilisateur comme les règles les plus intéressantes.

Toutefois, ces méthodes diffèrent dans les détails tels que le schéma de représentation, la méthode de comparaison et les mesures d'intérêts. ([PT99], [LH96], [LHC97], [PT99] et [LH96]).

Selon [LH96], [ST96] n'est qu'une proposition et aucun système n'a été mis en œuvre qui utilise cette approche.

L'approche présentée dans [LH96] est basée sur une comparaison entre la syntaxe d'une règle et une conviction. Dans [LH96], une règle et une croyance sont «différents» si le conséquents et la croyance sont «semblables», mais les antécédents sont "éloignés" ou les conséquents sont "éloignés", mais les antécédents sont «similaires». La similarité et «différence» sont définies sur la base exclusive de la structure syntaxique des règles. Par exemple, considérons la conviction : homme (X) $\rightarrow$ humain (X) et la règle femme (X) $\rightarrow$ humain (X). Selon [LH96], cette règle est différente de la croyance. Cette règle, même si elle est différente de la croyance, elle n'est pas surprenante puisque l'un ne contredit pas l'autre en aucune façon [ST96].

Les méthodes ([KMR+94], [NLH+98] et [SRV+97]) sont des méthodes orientées requêtes pour rechercher les règles intéressantes. Ils ont l'inconvénient de chercher ce que l'utilisateur demande, ce qui est difficile de trouver les vraies règles inattendues car l'utilisateur ne connaît pas a priori les règles intéressantes pour les demander d'une part, et que les règles non retournées par le système peuvent être de grand intérêt aussi.

Une limitation à l'égard de l'utilisation des croyances comme base pour l'intérêt est relative à *l'acquisition des connaissances* nécessaires pour l'évaluation.

Obtenir des points de vue des utilisateurs est difficile en pratique, et les connaissances acquises peuvent être aussi incomplètes. Les utilisateurs peuvent ne pas être en mesure de spécifier complètement, chaque et toutes les croyances au sujet d'un domaine. Spécification partielle des croyances de l'utilisateur peut conduire à des résultats incorrects dans l'attribution des scores d'intérêts à un grand nombre de règles. Bon nombre de ces règles marquées incorrectement se rapportent à des attributs, des relations et de croyances, que l'utilisateur a omis de préciser. En outre, l'utilisateur peut également modifier ses croyances avec le temps. Maintenir une base de connaissances (croyances de l'utilisateur) est une question importante qui mérite l'attention.

Dans notre approche, on propose une mesure d'intérêt en fonction de la distance sémantique de la règle qui est basée sur les connaissances préalables représentées par une ontologie. Les mesures d'intérêts peuvent jouer un rôle important dans l'identification des patterns nouvelles, pertinentes, implicites et compréhensibles de l'ensemble des patterns issus du processus de data mining. Les connaissances ne sont pas exprimées en tant que règles, ni en terme spécifique à une approche mais en terme de hiérarchie de concepts. Les ontologies permettent de partager les connaissances, ce qui augmente considérablement le potentiel de réutilisation des connaissances déjà disponibles par les ontologies de domaine comme "ONTODerm" pour la dermatologie, la "BIO-ONT" pour biomédecine ", ASFA, oneFish, FIGIS, AGROVOC" pour l'alimentation...etc. Notre approche permet la possibilité de pondération des concepts de l'ontologie, ce qui donne une flexibilité dans l'expression des connaissances de domaine. En outre, pour s'occuper de la connaissance imprécise, nous utilisons une ontologie floue pour la représentation des connaissances non catégoriques. Notre travail rentre dans la phase post traitement du processus du data mining, ce qui permet à notre approche de s'intégrer à tout système d'exploration des règles indépendamment de l'algorithme de génération.

4. Conclusion

Les ontologies permettent une représentation formelle d'un domaine de connaissance, sous la forme d'une terminologie hiérarchisée munie de relations sémantiques.

L'intégration des connaissances du domaine dans le processus du data mining aide à optimiser le nombre d'itérations, réduire le temps de modélisation et à évaluer la pertinence des motifs résultats découverts et leurs expressions dans des termes concis et à de différents niveaux d'abstraction. Les ontologies ont été appliquées comme connaissances de domaine pour la compréhension du domaine d'application, préparation et sélection des données, génération des règles de haut niveau et l'analyse d'intérêt des motifs ou des règles afin d'améliorer le processus de data mining.

Dans ce chapitre on a présenté quelque travaux relatifs à l'utilisation des ontologies dans le processus de data mining, et on a fait une synthèse des travaux qui sont reliés à l'utilisation des critères subjectives dans le filtrage et le classement des règles. La plupart des approches utilisent des connaissances durant la phase d'extraction ou en post-traitement. Les connaissances sont exprimées soit en termes d'attentes, de croyances ou simplement des requêtes. Une limitation à l'égard de ces approches est relative à l'acquisition des connaissances, de leurs maintenances et à leurs partages.

Notre contribution qui fera l'objet du prochain chapitres consiste à présenter une nouvelle approche qui se base sur la connaissance de domaine et définit une nouvelle mesure d'intérêt « distance sémantique de la règle » qui sera utilise dans le classement des patterns.

Partie 2 : Contribution

Chapitre 4

Utilisation des ontologies floues dans le classement des règles intéressantes

1. Introduction
2. Présentation de l'approche
 - 2.1. Principe de l'approche
 - 2.2. Description de l'approche
 - 2.3. Notre approche dans le modèle CRISP-DM
3. Présentation du système
 - 3.1. Processus de classement des règles d'association
 - 3.2. Représentation des connaissances du système
 - 3.2.1. Représentation des règles d'associations
 - 3.2.2. Représentation d'une ontologie floue
4. Présentation de l'algorithme de classement des règles
 - 4.1. Calcul de la distance sémantique entre concepts
 - 4.2. L'ontologie floue et le calcul de distance
 - 4.3. Pondération des relations taxonomique
 - 4.4. Calcul de la distance d'une règle d'association
 - 4.5. Présentation des résultats
 - 4.6. Algorithme Pseudo code
5. Conclusion

1. Introduction

La fouille de données vise à extraire de manière la plus efficace possible des règles pertinentes à partir de bases de données volumineuses. Il a été observé que la plupart des algorithmes de data mining génèrent un grand nombre de motifs qui sont valides mais évidents ou peu intéressants. Pour cette raison des méthodes de filtrages sont nécessaires pour sélectionner les motifs les plus intéressants. Les motifs sont pertinents selon des mesures d'intérêt objectives qui évaluent les propriétés statistiques des modèles extraits ou selon des mesures d'intérêt subjectives qui confrontent les modèles extraits à des connaissances exprimées sur le domaine. L'intégration des connaissances du domaine dans le processus du data mining aide à évaluer la pertinence des motifs résultats découverts et leurs expressions dans des termes concis et à de différents niveaux d'abstraction.

Dans ce chapitre on va présenter notre approche pour le classement des règles d'association selon leur degré d'intérêt. Une présentation générale de l'approche et de son emplacement dans un processus de data mining sera décrite, ainsi qu'une présentation détaillée de l'algorithme de calcul des distances sémantiques.

2. Présentation de l'approche

2.1. Principe de l'approche

Nous souhaitons classer les règles d'association qui sont présentées à l'analyste en les classant par qualité décroissante en fonction des connaissances disponibles sur le domaine des données. Notre objectif est de faciliter la tâche de l'analyste en définissant une méthodologie de sélection des règles de qualité. Notre modèle de connaissances est une ontologie caractérisée par un réseau de concepts structurés hiérarchiquement par une relation de généralisation appelée 'EST-UN'. La relation 'EST-UN' a la sémantique classique de l'intension des ensembles d'objets dénotés par les concepts.

L'idée de base derrière notre approche est que plus un pattern est constitué d'éléments conceptuellement distants plus il est surprenant et que tout pattern surprenant est inattendu et donc intéressant. Les concepts hautement reliés sont regroupés ensembles dans une hiérarchie. Plus les concepts sont éloignés plus ils sont moins reliés. Notre idée consiste à exploiter les relations taxonomiques de l'ontologie dans un but de sélectionner les motifs les plus intéressants d'un ensemble de modèles existants. Un classement des patterns par ordre décroissant de distance conceptuelle facilite l'accès aux patterns les plus intéressants. L'algorithme de classement suppose que les règles issues d'un algorithme de data mining comme «a-priori» répondent aux mesures objectives de l'utilisateur.

2.2. Description de l'approche

Notre approche intègre la connaissance experte dans la phase post traitement dans un but de classement des modèles en utilisant la similarité inter-concepts d'une ontologie. La Similarités entre concepts dans une ontologie est une évaluation du lien sémantique entre deux concepts dont le but est d'estimer le degré par lequel les concepts sont proches dans leur sens [Res99].

Étant donné un ensemble de règles, notre approche consiste à classer les règles les plus intéressantes en fonction du degré de similarité entre l'antécédent et le conséquent de la règle en se basant sur la structure hiérarchique des concepts formant l'antécédent et le conséquent. L'antécédent aussi bien que le conséquent sont des conjonctions de concepts en vocabulaire de l'ontologie.

La majorité des travaux portant sur le calcul de similarité dans une ontologie considèrent que la similarité peut être évaluée uniquement à partir des liens

taxonomiques (ou lien « IS-A ») [RMBB89] [Res95] [WP94] [JC97]. On adoptant cette approche, notre contribution à la phase post traitement du processus de data mining consiste à évaluer la distance à partir des liens taxonomiques d'une ontologie.

Deux catégories principales d'algorithmes pour calculer la distance sémantique entre les termes organisés dans une structure hiérarchique ont été proposées dans la littérature : les approches basées sur la distance et les approches basées sur le contenu d'information.

Notre choix a été porté sur la catégorie des approches basées sur la distance pour des raisons de simplicité, moins de calcul et facilité de maintenance.

Notre approche assure le classement des règles selon leurs degrés d'intérêt en fonction de la distance sémantique qui mesure l'éloignement conceptuel de l'antécédent et le conséquent de la règle.

2.3. Notre approche dans le modèle CRISP-DM

Les ontologies ont été appliquées comme connaissance du domaine en fouille de données pour améliorer le processus de data mining. Elles peuvent être appliquées pour la compréhension du domaine d'application, préparation et sélection des données, génération des règles de haut niveau et l'analyse d'intérêt des motifs ou des règles.

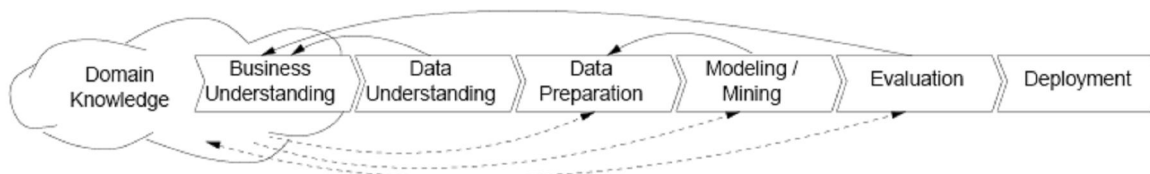


Figure 3-1 *Domaine de connaissance et CRISP-DM [Poh03]*

Notre approche s'inscrit dans le cadre de la phase d'évaluation du processus CRISP-DM. C'est dans cette phase où l'intérêt des règles est mesuré et une décision sur l'utilisation des résultats de data mining est prise.

La connaissance de l'utilisateur dans notre cas à une hiérarchie de concepts avec un support de l'information imprécise ou floue.

3. Présentation du système

Le système de classement des règles d'association (Figure 4-4) est composé principalement d'un processus de classement, d'une base de connaissance, et une base de règles d'association. Les règles d'associations sont représentées sous forme PMML, alors que les connaissances des utilisateurs sont représentées sous forme RDF avec prise en charge de l'information floue. (Voir section « Représentation des connaissances du système »). Protégé est utilisé comme outil d'acquisition de connaissances.

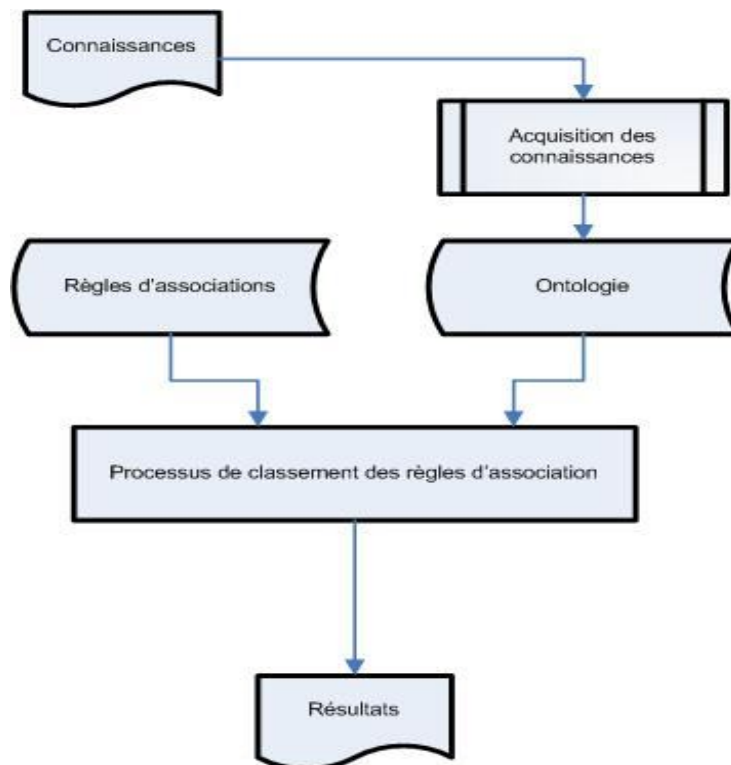


Figure 4-1 Système de classement des règles d'associations

3.1. Processus de classement des règles d'association

Le processus de classement des règles d'association est présenté sur la Figure 4-5 et il est décrit comme suit :

3.1.1 Construction de la matrice des relations pondérées:

Construction de la matrice des relations pondérées: Il s'agit de la lecture du fichier représentant l'ontologie et l'application à chaque relation lue la fonction de pondération

$$\text{pondération } f(\mu, \omega) = \begin{cases} \omega + \omega(1 - \mu) & \mu \neq 0 \\ 0 & \mu = 0 \end{cases}$$

La matrice est de dimension NxN, avec N le nombre des nœuds dans la hiérarchie. Chaque élément (i, j) de la matrice représente le poids de relation entre le concept (i) et le concept (j).

3.1.2 Construction de la matrice du plus court chemin :

En se basant sur la matrice des relations pondérées comme entrée, construire la matrice du plus court chemin de dimension $N \times N$ en appliquant l'algorithme de Dijkstra [URL06], où chaque élément (i,j) de la matrice représente la distance sémantique entre le concept (i) en ligne et le concept (j) en colonne.

3.1.3 Calcul des distances des règles :

En utilisant le fichier des règles d'association issue du processus de data mining, et de la matrice des relations pondérées issue de l'étape précédente, calculer la distance sémantique de chaque règle.

3.1.3 Classement des règles: Classement des règles par ordre décroissant de distance de règle.

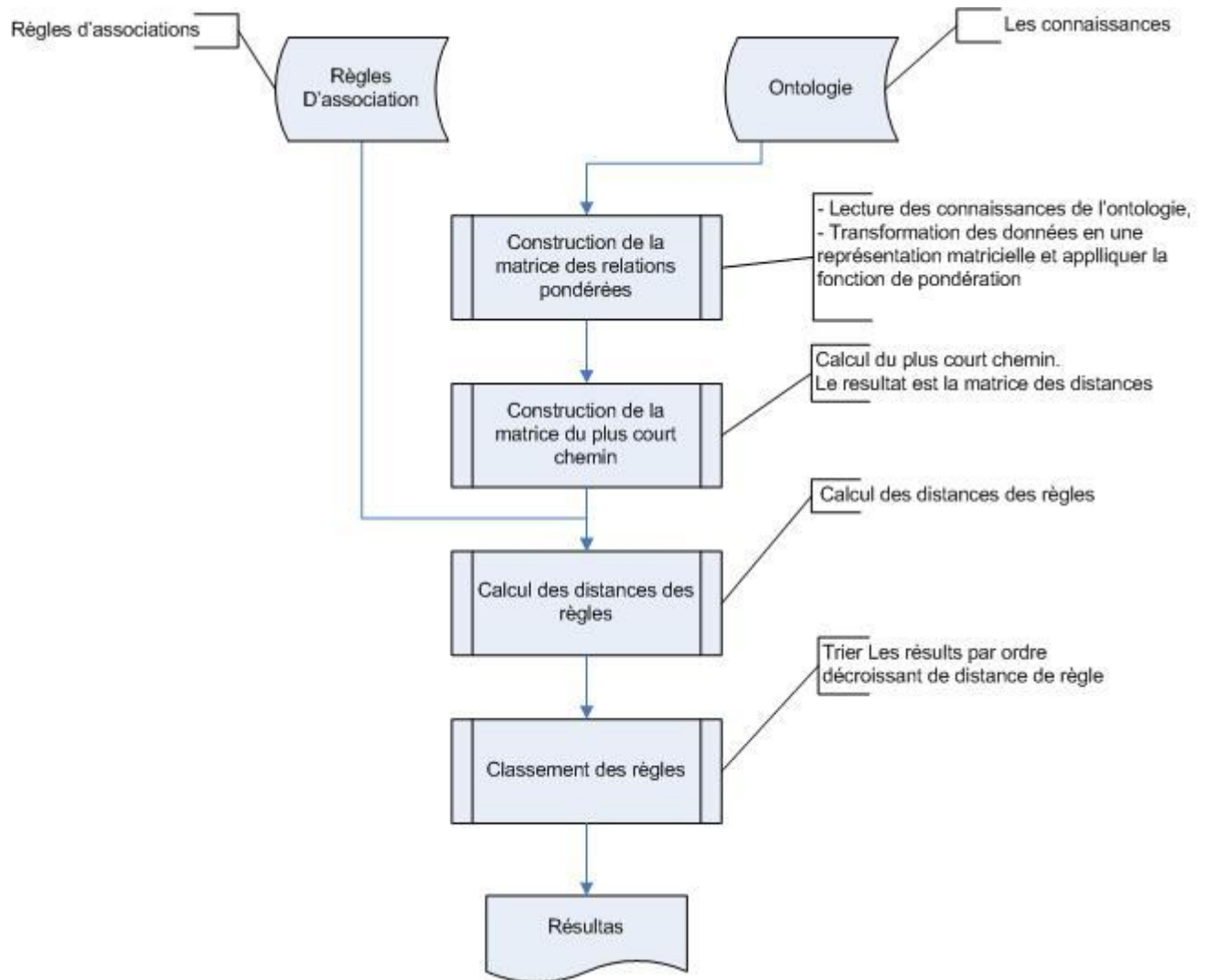


Figure 4-2 Processus de classement des règles d'associations

3.2. Représentation des connaissances du système

3.2.1. Représentation des règles d'associations

Les modèles issus d'un processus de data mining peuvent être exprimés en un langage standard qui permet l'échange et la communication entre applications. Notre approche consomme un modèle de règle d'association décrit en PMML. PMML (Predictive Model Markup Language) est une langue basée sur XML pour définir des modèles de data mining. En d'autres termes, PMML joue le rôle de format standard pour partager, entre les différentes applications, des modèles complexes de données issues de fouille de données. PMML est basé sur DTD précis qui définit la structure XML pour une variété de modèles de données tel que les modèles basés sur les arbres, les modèles de régression, les modèles de cluster, les règles d'association, les réseaux de neurones ...etc. Dans ce qui suit un extrait d'un document PMML.

Extrait d'un Document PMML

```
<PMML version="2.0" >
  <Header copyright="www.dmg.org" description="example model for association rules"/>
  .....
  <!-- We have five items in our input data -->
    <Item id="1" value="A" />
    <Item id="2" value="B" />
    <Item id="3" value="C" />
  ....
  <!-- three frequent itemsets with a single item -->
    <Itemset id="1" support="0.833" numberOfItems="1">
      <ItemRef itemRef="1" />
    </Itemset>
  .....
  <!-- three frequent itemsets with two items. -->
    <Itemset id="4" support="0.667" numberOfItems="2">
      <ItemRef itemRef="1" />
      <ItemRef itemRef="2" />
    </Itemset>
    <Itemset id="5" support="0.5" numberOfItems="2">
      <ItemRef itemRef="1" />
      <ItemRef itemRef="3" />
    </Itemset>
  ....
  <!-- one frequent itemset with three items. -->
    <Itemset id="7" support="0.5" numberOfItems="3">
      <ItemRef itemRef="1" />
      <ItemRef itemRef="2" />
      <ItemRef itemRef="3" />
    </Itemset>
```

```

<!-- Three rules satisfy the requirements -->
<AssociationRule support="0.667" confidence="0.8"
    antecedent="1" consequent="2" />
...
<AssociationRule support="0.5" confidence="1.0"
    antecedent="5" consequent="2" />
</AssociationModel>
</PMML>

```

Figure 4-3 Extrait d'un Document PMML

3.2.2. Représentation d'une ontologie floue

En dépit du caractère expressif de RDFS/OWL, ils ne sont pas assez puissants pour s'occuper des connaissances vagues et incertaines qui sont communément trouvés dans beaucoup d'applications de monde réel. Pour stocker la connaissance floue en RDF sans être obligé à suivre aucune extension, nous devons fournir une façon de sérialiser la connaissance floue dans les triples RDF pour contourner cette insuffisance [SST+08]. Les triples RDF sont représentés avec un prédicat binaire. Parmi les techniques les plus répandues, la réification et les nœuds blancs.

Utilisation de la réification

RDF possède en plus d'être un langage standardisé, avec représentation graphique, la caractéristique originale comme réification [URL02]. Le modèle proposé est un graphique, dans lequel chaque nœud représente un concept et chaque lien entre deux nœuds forme une relation entre les concepts. En plus, chaque relation possède un niveau associé de confiance. La description du niveau supplémentaire de confiance est réalisée en utilisant une réification manuelle, c'est-à-dire en faisant un RDF "statement" sur le triple RDF en lui ajoutant l'information qui contient le degré de vérité de la relation floue. Dans l'exemple « Vegetable has tomato avec un degré d'appartenance de 0.70 » le concept "Vegetable" est rattaché au concept "Tomatto" avec une relation floue "hasTomatto" et un degré de vérité égale à 0.70.



Figure 4-4 Exemple de relation avec degré de confiance

Cette phrase peut être représentée en RDF comme :

```
<rdf:Description rdf:about="#s1">
<rdf:subject rdf:resource="Vegetable"/>
<rdf:predicate rdf:resource=" hasTomatto "/>
<rdf:object>rdf:resource="Tomatto"</rdf:object>
<rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Statement"/>
< hasTomatto rdf:datatype="http://www.w3.org/2001/XMLSchema#float"> 0.70</hasTomatto>
</rdf:Description>
```

En suivant les principes de représentation précédents notre modèle de connaissance est capable d'utiliser n'importe quel type de relations floues entre les concepts du monde réel.

Utilisation des nœuds blancs

Dans les langues telles que RDF et le OWL, une propriété est une relation binaire utilisée pour relier deux individus ou un individu et une valeur [URL01]. Cependant, dans certains cas, il est nécessaire d'utiliser des relations pour relier un individu à plus qu'un individu pour représenter certains concepts. Par exemple, on veut représenter des propriétés d'une relation, telles que la certitude, la sévérité, pertinence d'une relation, ...etc.

Dans l'exemple (vegetable has tomatto avec un degré 0.7), Il y a une relation binaire entre vegetable et Tomatto avec une valeur de degré quantitative décrivant cette relation (0.7).

La solution consiste à créer un individu qui représente la relation elle-même, avec les liens du sujet de la relation (vegetable) à cette instance (VT_Tomatto) et avec les liens de cette instance à tous les participants qui représentent des renseignements supplémentaires sur cette instance. Nous utilisons des nœuds en blanc dans RDF pour représenter des instances d'une relation.

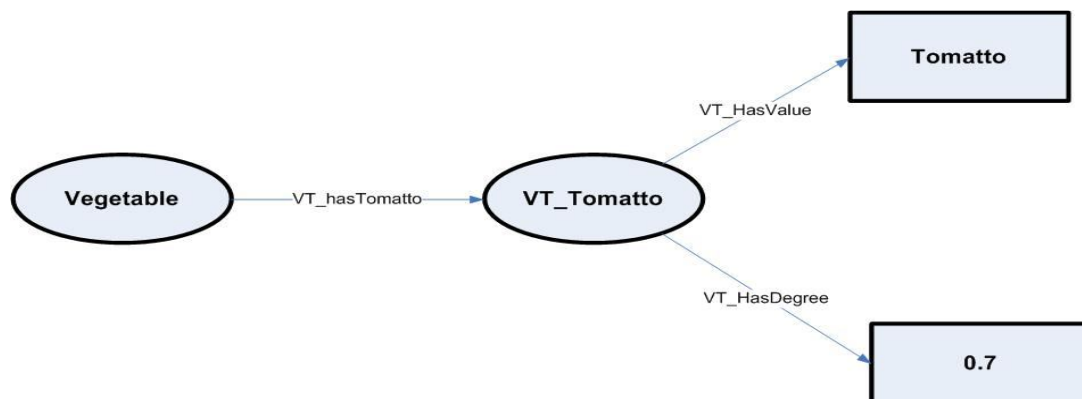


Figure 4-5 Exemple de nœud blanc

On adopte cette technique dans la représentation de la connaissance floue pour notre approche pour son support avec les outils RDF tel que protégé. Cette solution est considérée comme « best practice » par w3c [NR06].

4. Présentation de l'algorithme

Dans cette section nous présentons un algorithme pour classer les règles inattendues. Les règles que nous considérons sont de la forme « body→head » ou body et head sont des conjonctions de concepts en vocabulaire de l'ontologie.

On suppose qu'autre technique a exécuté la tâche de découverte des règles et a éliminé les patterns les moins intéressants de point de vue de critères objectifs.

Avec de tel classement, un utilisateur peut vérifier simplement peu de patterns sur le haut de la liste pour confirmer.

4.1. Calcul de la distance sémantique entre concepts

Les mesures reposant sur la distance, considèrent que la similarité entre deux concepts peut être calculée à partir du nombre de liens qui les séparent. La mesure du «edge counting » [RMBB89] évalue la distance sémantique à partir du nombre de branches séparant les concepts par le plus court chemin dans la hiérarchie. Il y a beaucoup d'algorithmes dans la littérature pour le calcul de la distance entre concept tel que « Edge Counting », « The Leacock & Chodorow », « Wu & Palmer », et autres. On a retenu la forme la plus simple de calcul de distance à savoir « Edge Counting » qui stipule que la distance entre deux nœuds de concept élémentaires, A et B, est le chemin le plus court qui relie A et B, c'est-à-dire le nombre minimal d'arcs qui séparent A et B [RMBB89] ou la somme des poids des arcs le long de ce chemin le plus court [RS95].

Exemple

Soit L'exemple de taxonomie suivante :

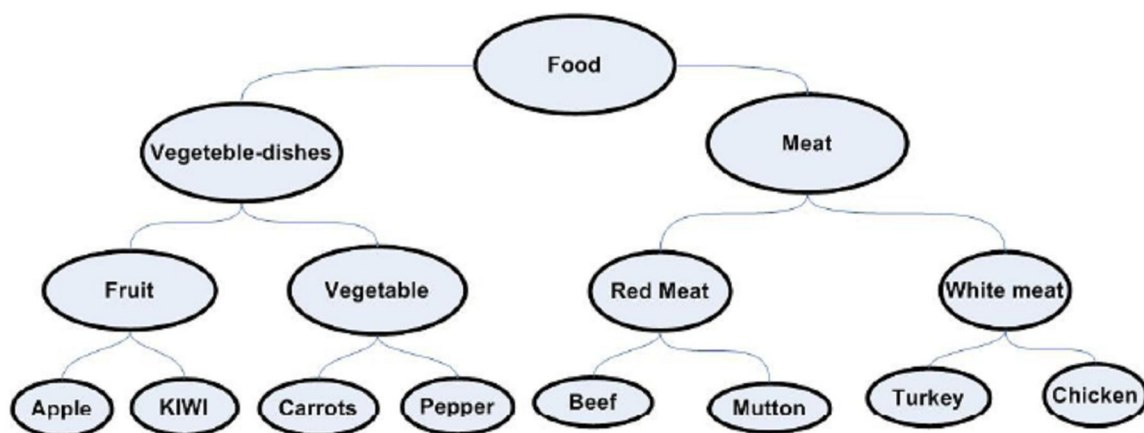


Figure 4-6 Exemple de taxonomie

Un exemple de distance entre 2 noeuds du graphe est :

$\text{Dist}(\text{Apple}, \text{Kiwi}) = 2$ $\text{Dist}(\text{Carrots}, \text{Pepper}) = 2$

$\text{Dist}(\text{Apple}, \text{Meat}) = 4$ $\text{Dist}(\text{Fruit}, \text{Red Meat}) = 4$

4.2. L'ontologie floue et le calcul de distance

Les structures taxonomiques floues peuvent refléter l'appartenance partielle d'un élément à un autre, et donc rendre possible la capture sémantique plus riche que les connaissances traditionnelles du domaine des représentations [Chen00]. Un concept peut appartenir à un ou plusieurs concepts avec un degré d'appartenance μ_i ($0 \leq \mu_i \leq 1$) dont $\sum \mu_i = 1$. La somme à 1 de l'ensemble des relations d'un concept est une normalisation et permet de ne pas favoriser les concepts avec un grand nombre de relations sur ceux avec un nombre réduit. La distance entre A et B est la somme des poids des arcs le long du chemin le plus court entre A et B [RS95]. Afin de calculer cette distance dans le cas d'une taxonomie floue, Il est nécessaire d'associer aux branches du graphe dans la taxonomie une pondération en fonction du degré d'association μ . La section suivante décrit ce processus.

4.3. Pondération des relations taxonomiques

Un problème largement reconnu à l'approche de « edge-counting », est qu'elle repose sur la notion que les liens dans la taxonomie représentent des distances uniforme. Pour en remédier, un poids (ω) doit être associé à chaque concept qui tient compte de la densité de la hiérarchie pour ce concept et de sa profondeur dans l'hiérarchie.

La pondération des relations taxonomiques est réalisée comme suit :

Si μ représente le degré d'appartenance de noeud X_i dans X_j et ω le poids initial associé à X_i , le poids w_{ij} associé à l'arc reliant le noeud X_i au noeud X_j est égale à $w_{ij} = \omega + (1 - \mu) \omega$. En effet, la distance entre 2 noeuds adjacents, connectés X_i et X_j noté w_{ij} avec un degré d'appartenance $\mu = 1$ est égale à 1 dans un graphe non pondéré, cependant cette distance est égale à ω dans le cas d'un graphe pondéré; en outre si on accepte que la distance est égale à 1 entre un noeud X_i qui appartient intégralement à un noeud X_j , on peut déduire intuitivement que la distance d'un noeud X_i qui appartient un à un noeud X_j avec une degré d'appartenance $0 < \mu < 1$, doit être supérieur à 1 et en fonction du complément à μ , d'où :

$$w = f(\mu, \omega) = \begin{cases} \omega + \omega(1 - \mu) & \mu \neq 0 \\ 0 & \mu = 0 \end{cases}$$

- w est le poids associé à l'arc reliant le nœud X_i au nœud X_j .
- μ ($0 \leq \mu \leq 1$) est le degré d'appartenance d'un concept d'appartenance du nœud X_i dans X_j .
- ω le poids initial associé à X_i .

4.4. Calcul de la distance d'une règle d'association

On définit la distance d'une règle comme étant la distance entre son antécédent et son conséquent. Pour une règle d'association de la forme $X \rightarrow Y$ ou $X = X_1 \wedge \dots \wedge X_k$ et $Y = Y_1 \wedge \dots \wedge Y_m$, la distance de la règle est $\text{Distance}(X, Y)$. On considérant les deux cotés de la règle c.-à-d. l'antécédent et le conséquent comme deux conjonctions de concepts. Pour calculer la distance entre deux ensembles de concepts à savoir l'antécédent et le conséquent, on a utilisé la distance de Hausdorff. La distance de Hausdorff entre 2 ensembles $A = \{a_1, \dots, a_p\}$ et $B = \{b_1, \dots, b_q\}$ est:

$$\text{Distance}(A, B) = \max(h(A, B), h(B, A)) \quad \text{où} \quad h(A, B) = \max_{a \in A} \min_{b \in B} \|a - b\|$$

La distance de Hausdorff, $\text{Distance}(A, B)$, mesure le degré de décalage entre les deux ensembles, puisque elle reflète de la distance du point de A qui est le plus éloigné de tout point de B et vice versa [DAN93]. Le choix de la distance de Hausdorff est justifié par l'éloignement maximal entre l'ensemble A et l'ensemble B qui représente le degré de surprise pour notre approche.

4.5. Présentation des résultats

Les résultats présentés à l'utilisateur issus de l'algorithme, est une liste de règles ordonnée par ordre décroissant par rapport à leurs distances sémantiques.

4.6. Algorithme

// Cette procédure appelée par l'algorithme transforme le degré d'appartenance à //un poids pour chaque arc dans le graphe et calcul la distance du chemin allant de X_i à X_j .

Procédure weight(ShorestPath(X_i, X_j))

Début

Pour $X_k \leftarrow X_i$ **à** X_j **Pas 2**

 //Calculer le poid de l'arc.

$w(X_k, X_{k+1}) \leftarrow (\omega + \omega_{X_k} (1 - \mu));$

 //Additionner le poid

$\text{Total_Weight} \leftarrow \text{Total_Weight} + w(X_k, X_{k+1})$

X_k **Suivant**

Retourner (Total_Weight)

Fin

Soit G un graphe de hiérarchie de concept.

Soit ND le nombre de nœuds du graphe.

Soit R l'ensemble des règles $R = \{R_i / R_i = \text{Body} \rightarrow \text{Head}\}$ où $i \in [1, N]$

N : Nombre de règles

X_i, X_j : Concepts Atomiques; $i \in [1, k]$; $j \in [1, m]$

Body $= X_1 \wedge \dots \wedge X_k$,

Head $= Y_1 \wedge \dots \wedge Y_m$

ShortestPath(X_i, X_j) est Le plus court chemin entre X_i, X_j

// Création d'une matrice carrée des distances des nœuds du graphe

Début

Pour $i \leftarrow 1$ à ND

Pour $j \leftarrow 1$ à ND/2

 //ShortestPath(X_i, X_j) est le plus court chemin entre X_i, X_j ;

 // la sommes des poids de path de nœud X_i au nœud X_j

 Distance(X_i, X_j) $\leftarrow$ weight(ShortestPath(X_i, X_j))

j Suivant

i Suivant

 // Creation du tableau de distance des règles

Pour $i \leftarrow 1$ à N

 Distance(R_i) $\leftarrow$ Distance($X_{1i} \wedge \dots \wedge X_{ki}, Y_{1i} \wedge \dots \wedge Y_{mi}$)

i Suivant

 Trier Distance(R) selon un ordre décroissant.

Fin

5. Conclusion

Dans ce chapitre on a présenté une nouvelle approche pour le classement des règles d'associations par ordre d'intérêt défini en fonction du degré de similarité entre l'antécédent et le conséquent de la règle en se basant sur la structure hiérarchique des concepts formant l'antécédent et le conséquent. L'antécédent aussi bien que le conséquent sont des conjonctions de concepts en vocabulaire de l'ontologie. La distance de Hausdorff a été choisie comme mesure entre l'antécédent et le conséquent d'une règle d'association.

On a adopté PMML comme langage de standard pour la représentation des modèles de data mining. Cela permet à notre approche de communiquer avec n'importe quel outil qui adopte ce standard.

Certaines techniques pour s'occuper des connaissances floues avec les langages standards tels que RDF/RDFS sans recourir à des extensions propriétaires ont été présentées. L'utilisation des nœuds blancs a été retenue comme une solution de choix.

Chapitre 5

Etude de cas

1. Introduction
2. Présentation du cas d'étude théorique
 - 2.1. Application de l'algorithme de classement des règles
 - 2.1.1. Construction de la matrice des relations pondérées
 - 2.1.2. Construction de la matrice du plus court chemin
 - 2.1.3. Calcul des distances des règles
 - 2.1.4. Présentation des résultats
 - 2.2. Interprétation des résultats
3. Présentation du cas d'étude réel
 - 3.1. Extraction des règles d'associations
 - 3.2 Application de l'algorithme de classement des règles
 - 3.2.1. Cas d'ontologie ordinaire
 - 3.2.2. Cas d'ontologie Floue
4. Conclusion

1. Introduction

Dans ce chapitre on va présenter 2 cas d'études. Le premier cas d'étude est théorique avec un nombre réduit de règles et de concepts (6 règles et 16 concepts), ce qui va nous permettre de suivre le déroulement de notre algorithme étape par étape pour valider les résultats de notre approche. L'ontologie floue de nourriture utilisée durant le premier cas d'étude est basée sur l'ontologie de [Chen00], où un concept peut appartenir à un ou plusieurs concepts. Après une simple présentation du cas d'étude, on va procéder à l'application de notre algorithme à savoir la construction de la matrice des relations pondérées, la construction de la matrice des plus courts chemins, le calcul des distances des règles et classement des résultats.

Le deuxième cas d'étude porte sur une base de données réelle de recensement des revenus (census-income) [URL4] avec un nombre de règles (2.225 règles) et de concepts (81 concepts) de l'ontologie. Les règles ont été extraites en utilisant une implémentation de l'algorithme 'Apriori'. Ce deuxième cas va nous permettre de valider notre approche sur un nombre acceptable de règles et de concepts.

A la fin de chaque étude de cas, une interprétation est donnée. On termine ce chapitre par une conclusion.

2. Présentation du cas d'étude théorique

Soit un ensemble de règles d'associations $R=\{(a) \text{ Apple} \rightarrow \text{Kiwi} , (b) \text{ Apple} \rightarrow \text{Carrots}, (c) \text{ Pepper, Carrots} \rightarrow \text{Turkey, Chicken}, (d) \text{ Kiwi} \rightarrow \text{Tomato}, (e) \text{ Tomato} \rightarrow \text{Pepper} , (f) \text{ Tomato, Pepper} \rightarrow \text{Turkey, Chicken}\}$ issus d'un processus de data mining et une ontologie de domaine (Figure 5-1). Le problème consiste à classer les règles d'associations par leurs ordres d'intérêt afin de faciliter à l'utilisateur l'accès à l'information la plus pertinente. L'ontologie floue de nourriture (Figure 5-1), illustre une hiérarchie des articles alimentaires et le degré d'appartenance entre les articles et leurs ancêtres. Dans cette hiérarchie, un article peut avoir plus qu'un parent et donc appartenir partiellement à une ou plusieurs catégories avec un certain degré; Par exemple « Tomato » est à 0.7 « Vegetable » et à 0.3 « Fruits », par contre « Kiwi » est 1 à « Fruit ».

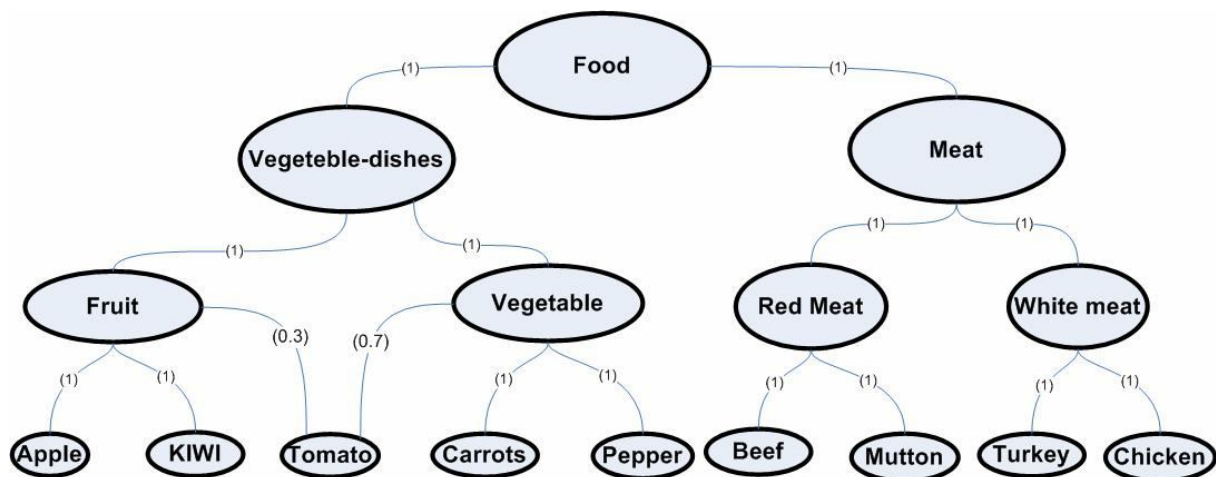


Figure 5-1 Ontologie floue de nourriture

2.1. Application de l'algorithme de classement des règles

Afin de classer les règles, l'algorithme se base sur les concepts de l'ontologie du domaine. Pour procéder, un numéro séquentiel est attribué à chaque concept (Figure 5-2). Ce numéro devient l'identificateur du concept dans notre système. On a identifié 16 concepts pour ce cas d'étude.

```

Reading and mapping label from file ...
0 Apple
1 Kiwi
2 Tomato
3 Carrots
4 Pepper
5 Beef
6 Mutton
7 Turkey
8 Chicken
9 Fruit
10 Vegetable
11 Red Meat
12 White Meat
13 Vegetable-dishes
14 Meat
15 Food

```

Figure 5-2 Identification des concepts de l'ontologie

Une fois les concepts identifiés et numérotés, On procède à la lecture des relations de l'ontologie Figure 5-3. Chaque ligne de relation dans l'ontologie (lien entre deux nœuds adjacents), a été représentée sous le format (noeud-fils, noeud-parent, degré-appartenance, poids), par exemple La 3eme ligne (2, 9, 0.3, 1) de la (Figure 5-3) représente le lien entre le concept '2' qui est 'Tomato' et le concept '9' qui est 'fruit' avec un degré d'appartenance égal à '0.3' et un poids égal à '1'.

```

Reading ontology from file ...
0,9,1,1
1,9,1,1
2,9,0.3,1
2,10,0.7,1
3,10,1,1
4,10,1,1
5,11,1,1
6,11,1,1
7,12,1,1
8,12,1,1
9,13,1,1
10,13,1,1
11,14,1,1
12,14,1,1
13,15,1,1
14,15,1,1

```

Figure 5-3 Relations de l'ontologie

2.1.1. Construction de la matrice des relations pondérées

Pour chaque relation de l'ontologie appliquer la fonction de pondération.

$$f(\mu, \omega) = \begin{cases} \omega + \omega(1 - \mu) & \mu \neq 0 \\ 0 & \mu = 0 \end{cases}$$

Remarquons que le poids ω est égal à 1 pour toutes les relations, ce qui réduit la fonction de pondération à :

$$f(\mu, 1) = \begin{cases} 1 + 1(1 - \mu) & \mu \neq 0 \\ 0 & \mu = 0 \end{cases}$$

Soit $w(i, j)$, $\mu(i, j)$ le poids et le degré d'appartenance entre le concept 'i' et le concept 'j', respectivement.

Le poids pondéré de la relation (i, j) est $w(i, j) = 1 + (1 - \mu(i, j))$

Une simple application donne les résultats suivants :

$w(i,j) = (1+(1-\mu)) = 1$ pour toute relation ou $\mu(i,j)=1$

$w(\text{tomat}, \text{fruit}) = 1+(1-\mu(\text{tomat}, \text{fruit})) = 1+(1-0.3) = 1.7$

$w(\text{tomat}, \text{Vegetable}) = 1+(1-\mu(\text{tomat}, \text{Vegetable})) = 1+(1-0.7) = 1.3$

(Figure 5-4) représente le résultat de pondération du graphe (Figure 5-1),

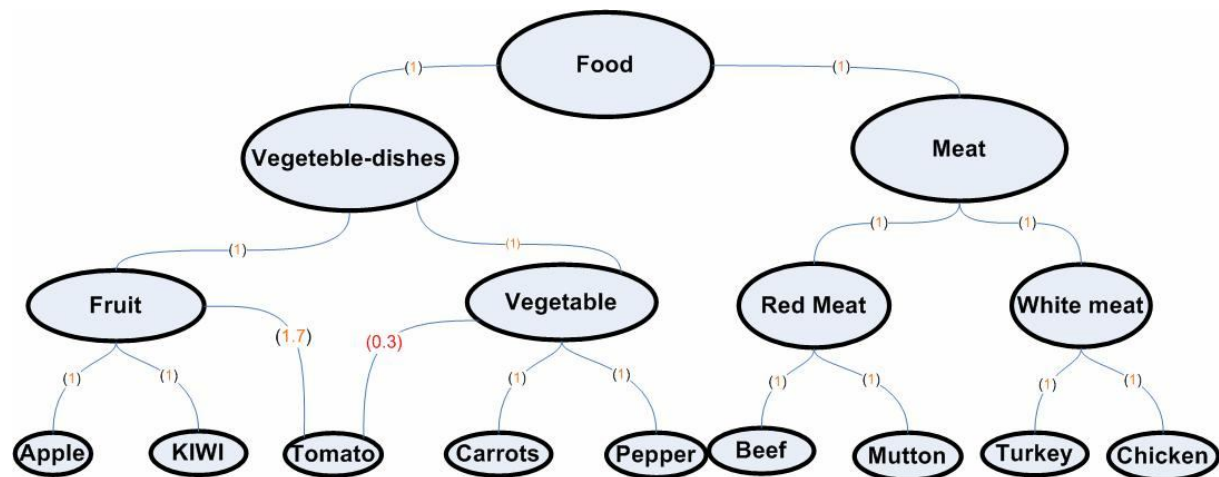


Figure 5-4 Graphe pondéré de nourriture

La figure 5-5 représente la matrice des relations pondérées où chaque élément de la matrice représente le poids entre le concept en ligne 'i' et le concept en colonne 'j'. S'il n'y a pas de relation directe entre 2 concepts, cette valeur est égale à '-1' qui est la valeur par défaut. Remarquons que $w(2,9)=7$ et $w(0,9)=1$ ce qui correspond au $w(\text{tomat}, \text{fruit})$ et $w(\text{Apple}, \text{fruit})$ respectivement.

building Matrix from ontology File...

-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	1.7	1.3	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	1
1	1	1.7	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	1	-1	-1
-1	-1	1.3	1	1	-1	-1	-1	-1	-1	-1	-1	-1	-1	1	-1	-1
-1	-1	-1	-1	-1	1	1	-1	-1	-1	-1	-1	-1	-1	-1	1	-1
-1	-1	-1	-1	-1	-1	-1	1	1	-1	-1	-1	-1	-1	-1	1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1	1	1	-1	-1	-1	-1	-1	1
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	1	1	-1	-1	-1	1
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	1	1	-1	1
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	1	1

Figure 5-5 Matrice des relations pondérées

2.1.2. Construction de la matrice du plus court chemin

La distance sémantique entre 2 nœuds du graphe, est la somme des poids du plus court chemin qui les sépare. Le résultat du calcul du plus court chemin issu de

notre implémentation de l'algorithme de 'Dijkstra' est présenté dans le Figure 5-6 où chaque élément contient la distance entre la ligne et la colonne correspondante.

Shortest path matrix ...

-1	2	2.7	4	4	6	6	6	6	6	1	3	5	5	2	4	3
2	-1	2.7	4	4	6	6	6	6	6	1	3	5	5	2	4	3
2.7	2.7	-1	2.3	2.3	6.3	6.3	6.3	6.3	6.3	1.7	1.3	5.3	5.3	2.3	4.3	3.3
4	4	2.3	-1	2	6	6	6	6	6	3	1	5	5	2	4	3
4	4	2.3	2	-1	6	6	6	6	6	3	1	5	5	2	4	3
6	6	6.3	6	6	-1	2	4	4	4	5	5	1	3	4	2	3
6	6	6.3	6	6	2	-1	4	4	4	5	5	1	3	4	2	3
6	6	6.3	6	6	4	4	-1	2	2	5	5	3	1	4	2	3
6	6	6.3	6	6	4	4	2	-1	5	5	3	1	4	2	3	3
1	1	1.7	3	3	5	5	5	5	-1	2	4	4	4	1	3	2
3	3	1.3	1	1	5	5	5	5	2	-1	4	4	4	1	3	2
5	5	5.3	5	5	1	1	3	3	4	4	-1	2	2	3	1	2
5	5	5.3	5	5	3	3	1	1	4	4	2	-1	3	1	2	2
2	2	2.3	2	2	4	4	4	4	1	1	3	3	-1	2	1	1
4	4	4.3	4	4	2	2	2	2	3	3	1	1	2	-1	1	1
3	3	3.3	3	3	3	3	3	3	2	2	2	2	2	1	1	-1

Figure 5-6 Matrice du plus court chemin

Figure 5-6 représente la distance sémantique inter concepts de l'ontologie.

Soit $\text{Dist}(i, j)$ la distance sémantique entre le concept i et le concept j .

$\text{Dist}(0, 1) = \text{Dist}(\text{Apple}, \text{Kiwi}) = w(\text{Apple}, \text{Fruit}) + w(\text{Fruit}, \text{Kiwi}) = 2$

$\text{Dist}(0, 2) = \text{Dist}(\text{Apple}, \text{Tomato}) = w(\text{Apple}, \text{Fruit}) + w(\text{Fruit}, \text{Tomato}) = 2.7$

2.1.3. Calcul des distances des règles

La distance de Hausdorff est utilisée pour le calcul de la distance sémantique de chaque règle. L'ensemble des règles en entrée à classer est représenté en Figure 5-7.

```
Reading rules from file ...
1 <- 0 <28.5, 29.6>
3 <- 0 <22.7, 27.7>
7 8 <- 4 3 <22.7, 28.3>
2 <- 1 <24.3, 28.4>
4 <- 2 <24.8, 28.2>
7 8 <- 4 2 <24.1, 20.1>
```

Figure 5-7 Les règles en entrées

Figure 5-7 représente la liste des règles où chaque ligne est sous le format 'Conséquent <-Antécédent (support, Confiance)'. En pratique cette liste est issue d'un algorithme d'extraction de règles d'associations comme 'Apriori'. Les valeurs de support et de confiance sont ignorées par notre algorithme.

Le correspondant mnémonique de l'ensemble des règles de la Figure 5-6 est

$R = \{(a), (b), (c), (d), (e), (f)\}$ où :

(a) Apple → Kiwi

(b) Apple → Carrots

(c) Pepper, Carrots → Turkey, Chicken

(d) Kiwi → Tomato

(e) Tomato → Pepper

(f) Tomato, Pepper → Turkey, Chicken

Application du calcul de la distance

La distance de Hausdroff est utilisée pour le calcul de la distance entre 2 groupes de concepts.

$$\text{Distance } (A, B) = \max (h (A, B), h (B, A)) \quad \text{où} \quad h(A, B) = \max_{a \in A} \min_{b \in B} \|a - b\|$$

Le calcul de la distance entre 2 concepts ou groupe de concepts notée Dist est comme suit :

(a) $\text{Dist}(\text{Apple}, \text{Kiwi})=2$

(b) $\text{Dist}(\text{Apple}, \text{Carrots})=4$

(c) $\text{Dist}(\text{Pepper} \wedge \text{Carrots}, \text{Turkey} \wedge \text{Chicken})= \max (h (A, B), h (B, A))$

$$\begin{aligned} h (A, B) &= \text{Max} (\\ &\quad \text{Min}([\text{Dist}(\text{Pepper}, \text{Turkey}), \text{Dist}(\text{Pepper}, \text{Chicken})]), \\ &\quad \text{Min}([\text{Dist}(\text{Carrots}, \text{Turkey}), \text{Dist}(\text{Carrots}, \text{Chicken})])) \\ &= \max(\min(6, 6), \min(6, 6)) = 6 \\ h(B, A) &= \text{Max} (\\ &\quad \text{Min}([\text{Dist}(\text{Turkey}, \text{Pepper}), \text{Dist}(\text{Turkey}, \text{Carrots})]), \\ &\quad \text{Min}([\text{Dist}(\text{Chicken}, \text{Pepper}), \text{Dist}(\text{Chicken}, \text{Carrots})])) \\ &= \max(\min(6, 6), \min(6, 6)) = 6 \end{aligned}$$

(c) $\text{Dist}(\text{Pepper} \wedge \text{Carrots}, \text{Turkey} \wedge \text{Chicken})= \max (h (A, B), h (B, A)) = \max(6, 6) = 6$

(d) $\text{Dist}(\text{Kiwi}, \text{Tomato})=2.7$

(e) $\text{Dist}(\text{Tomato}, \text{Pepper})=2.3$

(f) $\text{Dist}(\text{Tomato} \wedge \text{Pepper}, \text{Turkey} \wedge \text{Chicken})= \max (h (A, B), h (B, A))$

$$\begin{aligned} h (A, B) &= \text{Max} (\\ &\quad \text{Min}(\text{Tomato}, \text{Turkey}), \text{Dist}(\text{Tomato}, \text{Chicken})), \\ &\quad \text{Min}(\text{Dist}(\text{Pepper}, \text{Turkey}) + \text{Dist}(\text{Pepper}, \text{Chicken}))) \\ &= \max(\min(6.3, 6.3), \min(6, 6)) = 6.3 \\ h(B, A) &= \text{Max} (\\ &\quad \text{Min}([\text{Dist}(\text{Turkey}, \text{Tomato}), \text{Dist}(\text{Turkey}, \text{Pepper})]), \\ &\quad \text{Min}([\text{Dist}(\text{Chicken}, \text{Tomato}), \text{Dist}(\text{Chicken}, \text{Pepper})])) \\ &= \max(\min(6.3, 6), \min(6.3, 6)) = 6 \end{aligned}$$

(f) $\text{Dist}(\text{Tomato} \wedge \text{Pepper}, \text{Turkey} \wedge \text{Chicken})= \max (h (A, B), h (B, A)) = \max(6, 6.3) = 6.3$

Le résultat du calcul manuel est présenté au tableau 5-2

Label	Règle	Distance
(a)	Apple → Kiwi	2.00
(b)	Apple → Carrots	4.00
(c)	Pepper, Carrots → Turkey, Chicken	6.00
(d)	Kiwi → Tomato	2.70
(e)	Tomato → Pepper	2.30
(f)	Tomato, Pepper → Turkey, Chicken	6.3

Tableau 5-1 Tableau des distances des règles

Figure 5-8 représente le classement de la liste des règles avec leurs distances correspondantes.

```

Sorting rules based on husrdoff distance ...
[Distance=6.3] [7, 8] <- [4, 2]
[Distance=6.3] [Turkey, Chicken] <- [Pepper, Tomato]
[Distance=6] [7, 8] <- [4, 3]
[Distance=6] [Turkey, Chicken] <- [Pepper, Carrots]
[Distance=4] [3] <- [0]
[Distance=4] [Carrots] <- [Apple]
[Distance=2.7] [2] <- [1]
[Distance=2.7] [Tomato] <- [Kiwi]
[Distance=2.3] [4] <- [2]
[Distance=2.3] [Pepper] <- [Tomato]
[Distance=2] [1] <- [0]
[Distance=2] [Kiwi] <- [Apple]

```

Figure 5-8 Classement des règles et leurs distances

2.1.4. Présentation des résultats

Le classement des règles en fonction de leurs distances se fait par ordre décroissant. Le résultat de cet ordre est : (f), (c), (b), (d), (e), et (a) comme le montre la Figure 5-8.

2.2. Interprétation des résultats

De point de vue de système de décision les règles (f), (c) appartiennent à un niveau plus élevé (Food) que les règle (b) et (d) qui appartiennent au niveau (vegetable-dishes). La règle (e) appartient à un niveau restreint (Vegetable), cependant la règle (a) appartient au niveau (Fruit). Plus on remonte dans la hiérarchie plus la décision est importante et donc la règle découverte est plus intéressante. Il s'agit ici d'une relation nouvelle (règles (f) et (c)) entre le croisement de domaines (vegetables-dishes, Meat) qui sont plus loin que les domaines (vegetables, Fruits) de la règle (b,d). Les règles (a) et (e) concernent les domaines de bas niveau Fruit et Vegetable, respectivement est donc elles sont moins intéressantes. Notons que (d) est plus surprenante que la règle (e), même si 'tomat' est à la fois Fruit et Vegetable avec des degrés différents. Le fait que tomat est plus proche de Vegetable que Fruit, la règle (d) est plus intéressante que la règle (e)

3. Présentation du cas d'étude réel

Dans un but de validation de notre approche, notre cas d'étude porte sur une base de données réelle de recensement des revenus (census-income) [URL4] qui contient le recensement des revenus de 48.842 personnes. Cette base de données est composée de 14 attributs dont 6 attributs sont à valeurs continue. On a retenu 10 attributs dont 4 continus ont été ignorés et 2 attributs ont été discrétisés. Tableau 1-1 représente la liste des attributs bruts de la base d'étude.

age: continuous.
workclass: Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked.
fnlwgt: continuous.
education: Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool.
education-num: continuous.
marital-status: Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse.
occupation: Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty, Handlers-cleaners, Machine-op-inspct, Adm-clerical, Farming-fishing, Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces.
relationship: Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried.
race: White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black.
sex: Female, Male.
capital-gain: continuous.
capital-loss: continuous.
hours-per-week: continuous.
native-country: United-States, Cambodia, England, Puerto-Rico, Canada, Germany, Outlying-US(Guam-USVI-etc), India, Japan, Greece, South, China, Cuba, Iran, Honduras, Philippines, Italy, Poland, Jamaica, Vietnam, Mexico, Portugal, Ireland, France, Dominican-Republic, Laos, Ecuador, Taiwan, Haiti, Columbia, Hungary, Guatemala, Nicaragua, Scotland, Thailand, Yugoslavia, El-Salvador, Trinidad&Tobago, Peru, Hong, Holand-Netherlands.
class: >50K, <=50K

Tableau 5-2 Les attributs de la base de données recensement du revenu

Avant de procéder à l'extraction des règles en utilisant une implémentation de l'algorithme 'Apriori' on a conduit les opérations de préparations suivantes :

1. Opérations sur les données

- Ignorer les attributs capital-gain, capital-loss , hours-per-week , fnlwgt
- Discrétisation de la variables continue 'Age' aux valeurs suivantes (age<=20, 20<age <=30, 30<age <=40, 40<age <=50 , age>50)
- Discrétisation de variables continue 'education_num' aux valeurs suivantes (num<=9 , 9< num<=13, 13<num<=15,num>15)
- Généralisation à Europe les valeurs pays suivantes (England,Greece, Italy, Poland, Portugal, Ireland, France, Hungary, Scotland, Holand-Netherlands,Yugoslavia, Germany)
- Généralisation à Asia les valeurs pays suivantes (India, Japan, China, Iran, Vietnam, Taiwan, Thailand,Hong,Cambodia)
- Généralisation à SouthAmerica les valeurs pays suivantes (South, Cuba, Honduras, Philippines, Jamaica, Mexico, Dominican-Republic)
- Les lignes avec des valeurs manquantes 'NULL' ont été éliminées.

2. Construction d'une ontologie

Après analyse de notre base données, une ontologie de 81 concepts a été crée (tableau 5-3). Les noms des rubriques ont été associés 'mapped' à des concepts atomiques où le nom de la rubrique dans la base de données désigne le nom du concept dans l'ontologie. En outre 4 nouveaux concepts ont été créés qui généralisent les concepts atomiques. Le concept (census-income) à la racine de la hiérarchie (niveau 0), et 3 autres concepts au niveau 1. (work, education et personal)

Le concept 'work' regroupe (workcalss,occupation et salary-class),

Le concept 'Education' regroupe (educationLevel,EducationNum et salary-class)

Le concept 'Porsonal' regroupe (race,sex,native-country,age et material_status).

Le concept 'census-income' regroupe (work, Education et Education)

Tout les concepts ont été numérotés avec un code de 0 à 81.

3. Numérotation des concepts

Associer un numéro identificateur à chaque concept de l'ontologie. Le résultat de toute ces opérations est présentée au tableau 5-3 où chaque concept est identifie avec un code et associe à son parent et un poids de mesure.

Level 0	Level 1	Level 2	Level 3	Code	Poids-test1	Poids-test2
Census-Income				81	1	1
	Work			78	1	1
		workclass:		68	1	1
			Private,	0	1	1
			Self-emp-not-inc,	1	1	2
			Self-emp-inc,	2	1	2
			Federal-gov,	3	1	3
			Local-gov,	4	1	3
			State-gov,	5	1	5
			Without-pay,	6	1	5
			Never-worked.	7	1	7
		occupation:		71	1	1
			Tech-support,	30	1	3
			Craft-repair,	31	1	3
			Other-service,	32	1	3
			Sales,	33	1	1
			Exec-managerial,	34	1	1
			Prof-specialty,	35	1	2
			Handlers-cleaners,	36	1	2
			Machine-op-inspct,	37	1	2
			Adm-clerical,	38	1	2
			Farming-fishing,	39	1	2
			Transport-moving,	40	1	2
			Priv-house-serv,	41	1	2
			Protective-serv,	42	1	2
			Armed-Forces.	43	1	2
		salary_class:		75	1	1
			>50K,	55	1	0
			<=50K	56	1	0
	EDUCATION			79	1	1
		education_Level		69	1	1
			Bachelors,	8	1	7
			Some-college,	9	1	4
			11th,	10	1	4
			HS-grad,	11	1	4
			Prof-school,	12	1	4
			Assoc-acdm,	13	1	4
			Assoc-voc,	14	1	4
			9th,	15	1	5
			7th-8th,	16	1	5
			Masters,	17	1	5
			1st-4th,	18	1	5
			10th,	19	1	6
			Doctorate,	20	1	6
			5th-6th,	21	1	6
			Preschool.	22	1	6
			12th,	65	1	0
		Education_num		76	1	1
			<=9	57	1	0
			9< Education<=13	58	1	0
			13<num<=15	59	1	0

			>15	60	1	0
	Personnel			80	1	1
		race:		72	1	1
			White,	44	1	1
			Asian-Pac-Islander,	45	1	0
			Amer-Indian-Eskimo,	46	1	0
			Other,	47	1	0
			Black.	66	1	0
		sex:		73	1	1
			Female,	48	1	0
			Male.	49	1	0
		native-country:		74	1	1
			Europe	50	1	0
			Asia	51	1	0
			SouthAmerica	52	1	0
			United-States,	53	1	0
			Canada,	54	1	0
		age:		77	1	1
			age <=20	67	1	0
			20<age <=30	61	1	0
			30<age <=40	62	1	0
			40<age <=50	63	1	0
			>50	64	1	0
		marital-status:		70	1	1
			Married-civ-spouse,	23	1	1
			Divorced,	24	1	1
			Never-married,	25	1	1
			Separated,	26	1	1
			Widowed,	27	1	1
			Married-spouse-absent,	28	1	1
			Married-AF-spouse.	29	1	3

Tableau 5-3 Ontologie de recensement de revenu

3.1. Extraction des règles d'associations

Pour générer les règles d'associations, nous avons utilisés une implémentation de l'algorithme Apriori [URL03] avec un minimum de support égal à 0,2 et un minimum de confiance égal à 0,2. Le nombre des règles générées est égal à 2.225.

Un extrait du fichier en entrée, écran d'exécution et fichier en sortie est présenté à la (Figure 5-9). Le fichier en sortie est un fichier texte où chaque ligne représente une règle sous la forme conséquent $\leftarrow$ Antécédent (support, confiance).

Extrait du fichier du census-income en entrée

62	5	8	58	25	38	44	49	53	56
63	1	8	58	23	34	44	49	53	56
62	0	11	57	24	36	44	49	53	56
64	0	10	57	23	36	66	49	53	56
61	0	8	58	23	35	66	48	52	56
62	0	17	59	23	34	44	48	53	56
63	0	15	57	28	32	66	48	52	56
64	1	11	57	23	34	44	49	53	55
62	0	17	59	25	35	44	48	53	55
63	0	8	58	23	34	44	49	53	55

Ecran d'exécution de génération des règles

```
D:\Mining>apriori -c20 -s20 -tr to_mine.txt rules.txt
apriori - find frequent item sets with the apriori algorithm
version 5.4 (2008.12.06) (c) 1996-2008 Christian Borgelt
reading to_mine.txt ... [66 item(s), 28150 transaction(s)] done [0.27s].
filtering, sorting and recoding items ... [35 item(s)] done [0.01s].
reducing transactions ... [8034/28150 transaction(s)] done [0.04s].
building transaction tree ... [15524 node(s)] done [0.01s].
checking subsets of size 1 2 3 4 5 6 done [0.07s].
writing rules.txt ... [2225 rule(s)] done [0.03s].

D:\Mining>
D:\Mining>
D:\Mining>
D:\Mining>
```

Extrait du fichier des règles en sortie

```
38 <- 48 (31.3, 26.1)
31 <- 11 (32.9, 20.0)
31 <- 57 (43.6, 20.2)
...
34 <- 55 (26.7, 25.8)
35 <- 55 (26.7, 24.1)
24 <- 63 (23.8, 22.0)
24 <- 48 (31.3, 28.5)
```

Figure 5-9 Extrait des résultats du processus d'extraction des règles

3.2. Application de l'algorithme de classement des règles

L'algorithme de classement prend en charge les cas de l'ontologie ordinaire et l'ontologie floue. On va présenter dans cette section l'application de notre algorithme pour un cas de l'ontologie ordinaire en Tableau 5-3, et une application pour le cas d'ontologie floue en Tableau 5-4.

3.2.1 Cas d'ontologie Ordinaire

Pour classer les règles nous avons utilisé l'implémentation de notre algorithme de classement avec l'ontologie en Tableau 5-3.

Deux jeux de tests ont été utilisés.

- Le premier jeu correspond à un poids égal à 1 pour tous les concepts (Colonne 'Poids-test1' de la Tableau 5-3)
- Le deuxième jeu correspond à poids différents pour les concepts (voir concepts (Colonne 'Poids-test2' de la Tableau 5-3).

Un extrait des résultats du test1 est présenté en (Figure 5-10), cependant celui du second test sont présentés en (Figure 5-11).

Les extraits en (Figure 5-10) et (Figure 5-11) représentent les deux premières lignes à l'intérieur de chaque valeur de distance.

```
[Distance=6]      [Adm-clerical] <- [Female]
[Distance=6]      [Craft-repair] <- [HS-grad]
[Distance=6]      [Craft-repair] <- [EducationNum<=9]
[Distance=4]      [Exec-managerial] <- [>50K]
[Distance=4]      [Prof-specialty] <- [>50K]
```

Figure 5-10 Extrait du résultat du premier test

```

[Distance=12] [Bachelors](7) <- [Never-married](1)
[Distance=12] [Bachelors] <- [>50K, Married-civ-spouse]
[Distance=11] [Craft-repair] <- [HS-grad]
[Distance=11] [Bachelors] <- [>50K]
[Distance=9] [Bachelors] <- [9<EducationNum<=13]
[Distance=9] [Never-married] <- [Some-college]
[Distance=8] [HS-grad] <- [>50]
[Distance=8] [>50] <- [HS-grad]
[Distance=7] [Craft-repair] <- [EducationNum<=9]
[Distance=7] [Craft-repair] <- [30<age<=40, Male]
[Distance=6] [Adm-clerical] <- [Female]
[Distance=6] [9<EducationNum<=13] <- [Some-college]
[Distance=5] [Private] <- [age>50]
[Distance=5] [Private] <- [40<age<=50]
[Distance=4] [Exec-managerial] <- [>50K]
[Distance=4] [Prof-specialty] <- [>50K]
[Distance=3] [Divorced] <- [40<age<=50]
[Distance=3] [Divorced] <- [Female]

```

Figure 5-11 Extrait du résultat du second test

3.2.1.1. Interprétation des résultats

Notre approche est fondée sur une hiérarchie du (Tableau 5-3) qui guide nos résultats. La hiérarchie a une profondeur maximale de 3 qui est la même que la profondeur minimale.

Le premier essai a été effectué avec un poids égal à 1, pour tous les concepts; Dans ce cas, tous les concepts ont le même degré d'intérêt pour l'utilisateur. Notre algorithme de classement sélectionne les règles qui ont le concept 'subsumer' le plus haut en premier. Le concept subsumer commun des règles ((1), (2) et (3) de la Figure 5-10) est le concept le plus haut «revenu de recensement», cependant le concept subsumer commun des règles ((4) et (5) de la Figure 5-11) est le concept «work». Règle (1) concerne le sexe 'sex' et la profession 'occupation' cependant, les règles (2), (3) concernent la relation entre la «profession» et «l'éducation». Les 2 dernières règles (4) et (5) expriment la relation entre la «profession» et «classe de salaire». Nous pensons, que la règle (1), (2) ou (3) est plus intéressante car elle nous donne des informations entre l'éducation 'Education' et les informations personnelles 'Personal' et que le preneur de décision se situe à un niveau supérieur que celui qui s'occupe de 'l'occupation' et 'salaire' (La paie par exemple) qui se situe à niveau inférieur au premier.

Le deuxième essai a été réalisé avec un poids de concept 'bachelors' égal à 7 (voir tableau 3 pour les poids des autres concepts). Dans ce cas L'utilisateur veut mettre davantage l'accent sur ce concept 'bachelors' en lui associant un poids élevé

ou parce qu'il est vraiment important dans le domaine de l'étude. L'algorithme de classement des règles sélectionne ceux aux poids les plus élevés. Le concept subsumer commun des règles (1) et (2) de la (Figure 5-11) est le concept 'census-income' «recensement de revenu», mais dans ce cas avec le concept 'Bachelor' en tant que membre de la règle. Dans ce cas, l'utilisateur accentue son étude sur les personnes ayant un niveau d'éducation 'bachelors' et de leurs relations avec les informations personnelles 'Personal'. Le concept subsumer commun des 2 dernières règles (Figure 5-11) est le concept «Personal». Ces règles expriment la relation entre les concepts «sexe» 'sex', «âge» 'age' et le statut 'martial-status'. Bien que la mesure d'intérêt est subjective (ce qui peut être très intéressant pour l'un, peut être d'intérêt moindre ou pas d'intérêt pour les autres), Nous croyons que plus nous remontons dans la hiérarchie, plus la décision est importante et la vision du décideur est plus large, stratégique et importante, et de même la règle découverte est plus intéressante. Notre approche suit cette vision.

3.2.2. Cas d'ontologie Floue

Level_rank(0)	Education Level (1)	Low_Level(2)	High_Level(3)
0	Preschool.	1.00	0.00
1	1st-4th	0.93	0.07
2	5th-6th}	0.87	0.13
3	7th-8th	0.80	0.20
4	9th	0.73	0.27
5	10th	0.67	0.33
6	11th	0.60	0.40
7	12th	0.53	0.47
8	HS-grad	0.47	0.53
9	Some-college	0.40	0.60
10	Assoc-acdm	0.33	0.67
11	Assoc-voc	0.27	0.73
12	Bachelors	0.20	0.80
13	Prof-school	0.13	0.87
14	Masters	0.07	0.93
15	Doctorate	0.00	1.00

Tableau 5-4 Concepts flous de l'ontologie de recensement

Afin d'étudier le cas flou on a défini 2 concepts 'Low_Level' et 'High_Level'. éducation comme indiqué au Figure 5-5. Tout niveau d'éducation peut s'exprimer en fonction de ces deux concepts avec un degré μ . Une personne peut avoir un High_Level avec un certain degré et avoir et un Low_Level avec un autre degré d'appartenance. On a défini une fonction d'appartenance $\mu_{\text{Hight}}(\text{Level_Rank}) = \text{Level_Rank}/15$ et son complément $\mu_{\text{Low}}(\text{Level_Rank}) = 1 - \text{Level_Rank}/15$ où 'Level_Rank' est un numéro séquentiel allant de 0 a 15 avec 0 représente 'Preschool' et 15 représente 'Docotrate' (Figure 5-4 colonne (0)). Le résultat de la fonction d'appartenance est indique à la (Figure5-4 colonne (3) et (4)) pour les concepts flous 'Low_Level' et 'Hight_Level' respectivement. Par exemple le concept 'Masters' (Figure 5-4 ligne (14)) appartient avec un degré $\mu=0.93$ au 'High_level' concept colonne (3), et avec 0.07 au concept 'Low_Level' colonne (2).

Le sous-ensemble floue 'High_Level' est définit mathématiquement comme

High_Level={

Preschool/0.00, 1st-4th/0.07, 5th-6th/0.13, 7th-8th/0.20, 9th/0.27,
10th/0.33, 11th/0.40, 12th/0.47, HS-grad /0.53, Some-college/0.60,
Assoc-acdm /0.67, Assoc-voc /0.73, Bachelors /0.80,
Prof-school /0.87, Masters /0.93, Doctorate/1.00
}

Et le sous-ensemble floue Low_Level est définit mathématiquement comme:

```

Low_Level={
    Preschool/1.00, 1st-4th/0.93, 5th-6th/0.87, 7th-8th/0.80, 9th/0.73,
    10th/0.67, 11th/0.60, 12th/0.53, HS-grad /0.47, Some-college/0.40,
    Assoc-acdm /0.33, Assoc-voc /0.27, Bachelors /0.20,
    Prof-school /0.13, Masters /0.07, Doctorate/0.00
}

```

Le tableau 5-5 représente l'ontologie floue où le poids représente le poids attribué au concept qui est une connaissance experte qui montre l'importance du concept dans le domaine d'application ; μ désigne le degré d'appartenance du concept à son parent (Level-1). Par exemple HS-grade appartient avec un degré de 0.53 au 'High_Level' et de 0.47 au 'Low_Level'.

Level 0	Level 1	Level 2	Level 3	Poids	μ
Census-Income				1	1.00
	Work			1	1.00
		workclass:		1	1.00
			Private,	1	1.00
			Self-emp-not-inc,	1	1.00
			Self-emp-inc,	1	1.00
			Federal-gov,	1	1.00
			Local-gov,	1	1.00
			State-gov,	1	1.00
			Without-pay,	1	1.00
			Never-worked.	1	1.00
		occupation:		2	1.00
			Tech-support,	2	1.00
			Craft-repair,	1	1.00
			Other-service,	2	1.00
			Sales,	2	1.00
			Exec-managerial,	2	1.00
			Prof-specialty,	2	1.00
			Handlers-cleaners,	2	1.00
			Machine-op-inspct,	2	1.00
			Adm-clerical,	2	1.00
			Farming-fishing,	2	1.00
			Transport-moving,	2	1.00
			Priv-house-serv,	2	1.00
			Protective-serv,	2	1.00
			Armed-Forces.	2	1.00
		salary_class:		1	1.00
			>50K,	1	1.00
			<=50K	1	1.00
	EDUCATION			1	1.00
		High_Level		1	1.00
			Bachelors,	1	0.80
			Some-college,	1	0.60
			11th,	1	0.40
			HS-grad,	1	0.53
			Prof-school,	1	0.87
			Assoc-acdm,	1	0.67
			Assoc-voc,	1	0.73
			9th,	1	0.27
			7th-8th,	1	0.20
			Masters,	1	0.93
			1st-4th,	1	0.07
			10th,	1	0.33
			Doctorate,	1	1.00
			5th-6th,	1	0.13
			Preschool.	1	0.00
			12th,	1	0.47
		Low_Level		1	1.00
			Bachelors,	1	0.20
			Some-college,	1	0.40
			11th,	1	0.60

			HS-grad,	1	0.47
			Prof-school,	1	0.13
			Assoc-acdm,	1	0.33
			Assoc-voc,	1	0.27
			9th,	1	0.73
			7th-8th,	1	0.80
			Masters,	1	0.07
			1st-4th,	1	0.93
			10th,	1	0.67
			Doctorate,	1	0.00
			5th-6th,	1	0.87
			Preschool.	1	1.00
			12th,	1	0.53
		Education_num		1	1.00
			<=9	1	1.00
			9< Education<=13	1	1.00
			13<num<=15	1	1.00
			>15	1	1.00
	Personnel			1	1.00
		race:		1	1.00
			White,	2	1.00
			Asian-Pac-Islander,	2	1.00
			Amer-Indian-Eskimo,	2	1.00
			Other,	2	1.00
			Black.	2	1.00
		sex:		1	1.00
			Female,	2	1.00
			Male.	1	1.00
		native-country:		1	1.00
			Europe	1	1.00
			Asia	1	1.00
			SouthAmerica	1	1.00
			United-States,	1	1.00
			Canada,	1	1.00
		age:		1	1.00
			age <=20	1	1.00
			20<age <=30	1	1.00
			30<age <=40	1	1.00
			40<age <=50	1	1.00
			>50	1	1.00
		marital-status:		1	1.00
			Married-civ-spouse,	2	1.00
			Divorced,	2	1.00
			Never-married,	2	1.00
			Separated,	2	1.00
			Widowed,	2	1.00
			Married-spouse-absent,	2	1.00
			Married-AF-spouse.	2	1.00

Tableau 5-5 Ontologie floue de recensement de revenu

Un extrait des résultats de notre test est présenté en (Figure 5-12), avec les deux premières lignes à l'intérieur de chaque valeur de distance.

```
[Distance=15.29] [Craft-repair] <- [HS-grad]
[Distance=15.29] [HS-grad] <- [Never-married]
[Distance=14.8] [Never-married] <- [Some-college]
[Distance=14.8] [Some-college] <- [Never-married]
[Distance=14.29] [HS-grad] <- [age>50]
[Distance=14.29] [age>50] <- [HS-grad]
[Distance=13.8] [40<age<=50] <- [Some-college]
[Distance=13.8] [Some-college] <- [40<age<=50]
[Distance=13.4] [Bachelors] <- [Never-married]
[Distance=13.4] [Bachelors] <- [>50K, Married-civ-spouse]
[Distance=13.29] [EducationNum<=9] <- [HS-grad]
[Distance=13.29] [HS-grad] <- [EducationNum<=9]
[Distance=12.8] [9<EducationNum<=13] <- [Some-college]
[Distance=12.8] [Some-college] <- [9<EducationNum<=13]
[Distance=12.4] [Bachelors] <- [>50K]
[Distance=12.4] [Bachelors] <- [>50K, Male]
[Distance=8] [Adm-clerical] <- [Female]
[Distance=8] [Adm-clerical] <- [Female, Private]
[Distance=7] [Married-civ-spouse] <- [>50K]
[Distance=7] [>50K] <- [Married-civ-spouse]
[Distance=6] [Craft-repair] <- [EducationNum<=9]
[Distance=6] [Divorced] <- [Female]
[Distance=5] [Exec-managerial] <- [>50K]
[Distance=5] [Prof-specialty] <- [>50K]
[Distance=4] [Male] <- [age>50]
[Distance=4] [age>50] <- [Male]
```

Figure 5-12 Extrait du résultat de classement 'Ontologie floue'

3.2.2.2. Interprétation des résultats

Dans ce cas, notre approche est fondée sur une hiérarchie du (Tableau 5-5) qui guide nos résultats. Elle a deux variables floues High_Level et Low_Level.

L'essai a été conduit avec un poids égal à 7 pour tous les concepts relatifs à l'éducation. (Voir tableau 5-5). Dans ce cas L'utilisateur veut mettre davantage l'accent sur ces concepts en leurs associant un poids élevé ou parce qu'ils sont vraiment importants dans le domaine de l'étude. L'algorithme de classement des règles sélectionne ceux aux poids les plus élevés. Le concept subsumer commun des règles (1) et (2) de la (Figure 5-12) est le concept 'census-income' «recensement de revenu», mais dans ce cas avec le concept 'HS-Grade' en tant que membre de la règle. 'HS-Grade' a un poids de 7, mais un degré d'appartenance de 0.53. De même pour les règles (3) et (4) de la (Figure 5-12) ou le concept 'some-college' est membre de la règle, cependant le degré d'appartenance est 0.60. Comme précédemment on peut conclure que plus nous remontons dans la hiérarchie, plus la décision est importante et la vision du décideur est plus large, stratégique et importante, et de même pour la règle et plus le degré d'appartenance du concept figurant dans la règle est moins plus la règle est plus intéressante.

4. Conclusion

Dans ce chapitre, nous avons présentés 2 cas d'études. Le premier cas d'étude sur un ensemble réduit de règles et une ontologie de nourriture avec un nombre faible de concepts ce qui nous a permis de suivre le déroulement de notre algorithme étape par étape.

Afin de valider notre approche sur un cas réel, un deuxième cas d'étude a été présenté où nous avons utilisés une base de données réelle de recensement des revenus. On a conduit la génération des règles, (nombre de règles 2.225 règles), création de l'ontologie, (81 concepts) et l'application de notre algorithme de classement des règles.

Les résultats issus des deux cas d'études ont reflétés nos attentes de résultats.

Dans ce qui suit, nous présentons une conclusion de notre travail et des perspectives pour des travaux futurs.

Conclusion Générale

La fouille de données vise à extraire des règles pertinentes à partir de bases de données volumineuses. Le processus du data mining est un processus qui est à la fois itératif et interactif, incluant le nettoyage de données, l'intégration de données, la sélection de données, la transformation de données, l'exploration de données, l'évaluation des motifs et la présentation de connaissances.

Il a été observé que la plupart des algorithmes de data mining génèrent un grand nombre de motifs qui sont valides, mais évidents ou peu intéressants. Pour cette raison des méthodes de filtrage sont nécessaires pour sélectionner les motifs les plus intéressants. Les motifs sont pertinents selon des mesures d'intérêt objectives qui évaluent les propriétés statistiques des modèles extraits, ou selon des mesures d'intérêt subjectives qui confrontent les modèles extraits à des connaissances exprimées sur le domaine.

L'utilisation de la connaissance préalable du domaine ou du processus dans le domaine de data mining, peut aider à choisir les informations appropriés (pré traitement), diminuer l'espace d'hypothèse (traitement), représenter les résultats d'une façon plus compréhensible et améliorer le processus (post traitement). Dans la phase (post traitement), l'intégration des connaissances du domaine dans le processus du data mining, aide à évaluer la pertinence des motifs et de les exprimer dans des termes concis et à des différents niveaux d'abstraction. Les ontologies permettent une représentation formelle d'un domaine de connaissance, sous la forme d'une terminologie hiérarchisée munie de relations sémantiques [Gru93a]. L'ontologie représente la connaissance avec les rapports entre les concepts. Elle est organisée comme une hiérarchie DAG (Directed Acyclic Graph). Les concepts hautement reliés sont regroupés ensembles dans la hiérarchie. Plus les concepts sont éloignées plus ils sont moins reliés.

La connaissance par sa nature est lexicalement imprécise et non catégorique, elle contient l'ambiguïté et le flou. L'ontologie floue a été introduite pour représenter la connaissance dans tous les domaines dans lesquels les concepts à représenter ont une définition imprécise. Au lieu d'appartenir ou non complètement à un concept flou, une instance possède un degré d'appartenance. Les ontologies floues sont capables de s'occuper de la connaissance floue [WY01] où les concepts sont en relation entre eux dans l'ontologie, avec un degré d'appartenance μ ($0 \leq \mu \leq 1$).

Dans ce document on a présenté une nouvelle approche, pour classer les motifs inattendus basée sur la distance conceptuelle. Dans un réseau sémantique «IS-A», la forme la plus simple de déterminer la distance entre deux nœuds de concept élémentaires, A et B, est le chemin le plus court qui relie A et B, c'est-à-dire le nombre minimal d'arcs qui séparent A et B [RMBB89] ou la somme des poids des arcs le long de ce chemin le plus court [RS95]. L'algorithme de classement proposé utilise une ontologie floue pour calculer la distance des règles d'associations, sur laquelle se base le classement. L'idée derrière notre approche est plus la règle est décrite avec des concepts distants, plus la règle représente un degré d'intérêt élevé. Ce travail constitue une contribution pour aider l'utilisateur à identifier les motifs les plus intéressants.

Une étude de cas théorique d'ontologie floue de nourriture a été présentée pour l'application de l'implémentation de notre approche, ainsi qu'une étude de cas réel de recensement du revenu.

De nombreuses perspectives s'offrent à la suite de nos travaux à savoir :

- L'intégration de notre approche dans l'algorithme de calcul des patterns comme l'algorithme a- priori en définissant un seuil de distance. Cela permet de réduire considérablement le nombre de règles.
- La prise en compte d'autres types de relation dans le calcul de la distance inter-concept et dans la pondération des arcs.
- Utilisation d'autre type de distance autre que « edge counting » et d'autre catégorie de mesure tel que les mesures basées sur le contenu d'information.
- L'étude de notre approche avec d'autre type de motifs, à savoir les motifs séquentiels.

Références

[AIS93a] R. Agrawal, T. Imielinski, and A. Swami. Database mining: A performance perspective. *IEEE Transactions on Knowledge and Data Engineering*, 5(6):914–925, December 1993.

[AIS93b] R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In *19 ACM SIGMOD Conf. on the Management of Data*, Washington, DC, May 1993.

[AK02] Jérôme Azé and Yves Kodratoff. Evaluation de la résistance au bruit de quelques mesures d'extraction de règles d'association. In Danièle Hérim and Djamel A. Zighed, editors, *EGC*, volume 1 of *Extraction des Connaissances et Apprentissage*, pages 143–154. Hermes Science Publications, 2002.

[AS94] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules in large databases. In *International Conference On Very Large Data Bases (VLDB '94)*, pages 487–499, San Francisco, Ca., USA, September 1994. Morgan Kaufmann Publishers, Inc.

[Azé03a] Jérôme Azé. *Extraction de connaissances dans des données numériques et textuelles*, thèse de doctorat, Université Paris-Sud, December 2003.

[Azé03b] Jérôme Azé. Une nouvelle mesure de qualité pour l'extraction de pépites de connaissances. In Mohand-Said Hacid, Yves Kodratoff, and Danielle Boulanger, editors, *EGC*, volume 17 of *Revue des Sciences et Technologies de l'Information - série RIA ECA*, pages 171–182. Hermes Science Publications, 2003.

[Bac99] B. Bachimont, « L'intelligence artificielle comme écriture dynamique : de la raison graphique à la raison computationnelle. », dans J. Petitot & P. Fabbri (eds.), *Au nom du sens* (pp. 290-319), Grasset.

[BCM+03] Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.

[BH91] F. Baader and B. Hollunder. A terminological knowledge representation system with complete inference algorithms. In H. Boley and M. M. Richter, editors, *Proceedings of the International Workshop on Processing Declarative Knowledge (PDK'91)*, volume 567 of *LNAI*, pages 67–86, Kaiserslautern, FRG, July 1991. Springer Verlag.

[BKD+01] Jeen Broekstra, Michel C. A. Klein, Stefan Decker, Dieter Fensel, Frank van Harmelen, and Ian Horrocks. Enabling knowledge representation on the web by extending RDF schema. In *WWW*, pages 467–478, 2001.

[BLA98] BLAZQUEZ M., FERNANDEZ M., GARCIA-PINAR J. M. & GOMEZ-PEREZ A., Building Ontologies at the Knowledge Level using the Ontology Design Environment, in *Proceedings of the Banff Workshop on Knowledge Acquisition for Knowledge-based Systems*, 1998.

- [BMS97] Sergey Brin, Rajeev Motwani, and Craig Silverstein. Beyond market baskets: Generalizing association rules to correlations. SIGMOD Record (ACM Special Interest Group on Management of Data), 26(2):265–276, June 1997.
- [BPSMM00] T. Bray, J. Paoli, C. M. Sperberg-McQueen, and E. Maler. extensible markup language (XML) 1.0 (second edition). <http://www.w3.org/TR/REC-xml>, 2000.
- [Bra77] Ronald J. Brachman. What’s in a concept: Structural Foundations for Semantic Networks. International Journal of Man-Machine Studies, 9:127–152, March 1977.
- [Bri08] Laurent Brisson. Intégration de connaissances expertes dans le processus de fouille de données pour l’extraction d’informations pertinentes. PhD thesis, HAL - CCSD, January 23 2008.
- [BT97] Gideon Berger and Alexander Tuzhilin. Discovering unexpected patterns in temporal data using temporal logic. In Temporal Databases, Dagstuhl, pages 281–309, 1997.
- [CCKK99] Chapman, Pete; Clinton, Julian; Kerber, Randy; Khabaza, Thomas; Reinartz, Thomas; Shearer, Colin; Wirth, Rüdiger: CRISP-DM 1.0: Step-by-step data mining guide. CRISP-DM consortium, 1999.
- [Chen00] Chen, G., Wei, Q., and Kerre, E. E. (2000). Fuzzy Data Mining: Discovery of Fuzzy Generalized Association Rules. In Bordogna, G. and Pasi, G., editors, Recent Issues on Fuzzy Databases, pages 45–66. Physica-Verlag.
- [CSD98] S. Chakrabarti, S. Sarawagi, and B. Dom. Mining surprising patterns using temporal description length. In Ashish Gupta, Oded Shmueli, and Jennifer Widom, editors, Twenty-Fourth International Conference on Very Large databases VLDB’98, pages 606–617, New York, NY, 1998. Morgan Kaufmann.
- [CST04] Cespivova H., Rauch J., Svatek V., Kejkula M., Tomeckova M.: Roles of Medical Ontology in Association Mining CRISP-DM Cycle. In: ECML/PKDD04 Workshop on Knowledge Discovery and Ontologies, Pisa, Italy, 2004.
- [CZSG03] Xiaoming Chen, Xuan Zhou, Richard B. Scherl, and James Geller. Using an interest ontology for improved support in rule mining. In Yahiko Kambayashi, Mukesh K. Mohania, and Wolfram Wöß, editors, DaWaK, volume 2737 of Lecture Notes in Computer Science, pages 320–329. Springer, 2003.
- [HAR93] Daniel P. Huttenlocher, Gregory A. Kl, and William J. Rucklidge. Comparing images using the hausdorff distance. IEEE Transactions on Pattern Analysis and Machine Intelligence, 15:850–863, 1993.
- [DIE01] DIENG R., CORBY O., GANDON F., GIBOIN A., GOLEBIEWSKA J., MATTA N. & RIBIÈRE M., Méthodes et outils pour la gestion des connaissances : une approche pluridisciplinaire du knowledge management (2ième édition), Dunod Edition Informatiques Séries Systèmes d’Information, 2001.

[dLMFHS02] Deborah L. McGuinness, Richard Fikes, James Hendler, and Lynn Andrea Stein. DAML+OIL: An ontology languages for the semantic web. IEEE Intelligent Systems, 2002.

[DT93] V. Dhar and A. Tuzhilin. Abstract-driven pattern discovery in databases. IEEE Transactions on Knowledge and Data Engineering, 5(6):926–938, December 1993.

[ES04] Euler Timm and Scholz Martin. Using Ontologies in a KDD Workbench. In Buitelaar, P. and Franke, J. and Grobelnik, M. and Paa, G. and Svatek, V. (editors), Workshop on Knowledge Discovery and Ontologies at ECML/PKDD '04, pages 103--108, Pisa, Italy, 2004.

[EYB06] Eduardo L. G. Escovar, Cristiane A. Yaguinuma, and Mauro Biajiz. Using fuzzy ontologies to extend semantically similar data mining. In Mario A. Nascimento, editor, SBBD, pages 16–30. UFSC, 2006.

[FER97] Fernández-López M, Gómez-Pérez A, Juristo N (1997) METHONTOLOGY: From Ontological Art Towards Ontological Engineering. Spring Symposium on Ontological Engineering of AAAI. Stanford University, California, pp 33–40

[Fik98] Fikes, R. Multi-use Ontologies. Stanford University. [Online] 1998. <http://www.ksl.stanford.edu/people/fikes/cd222/1998/Ontologies/tsld001.htm>

[FPSS96] Usama M. Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. From data mining to knowledge discovery: An overview. In Advances in Knowledge Discovery and Data Mining, pages 1–34. 1996.

[FRÉ07] Frédéric Fürst. L'ingénierie ontologique. Nantes : Institut de recherche en informatique, 2002. Rapport de recherche n° 02-07.

[Fre99] Alex Alves Freitas. On rule interestingness measures. Knowl.-Based Syst, 12(5-6):309–315, 1999. [FvHH+01] D. Fensel, F. van Harmelen, I. Horrocks, D. McGuinness, and P. F. Patel-Schneider. OIL: An ontology infrastructure for the semantic web. IEEE Intelligent Systems, 16(2):38–45, 2001.

[GB04] Ramanathan V. Guha and Dan Brickley. RDF vocabulary description language 1.0: RDF schema. W3C recommendation, W3C, February 2004. <http://www.w3.org/TR/2004/REC-rdf-schema-20040210/>.

[GCG94] N. Guarino, M. Carrara, and P. Giaretta. Formalizing ontological commitment. In Proceedings of the 12th National Conference on Artificial Intelligence, volume 1, pages 560–568, Seattle, Washington, July–August 1994. AAAI Press.

[GD06] Gayo Diallo, Une Architecture à Base d'Ontologies pour la Gestion Unifiée des Données Structurées et non Structurées, Thèse, Université Joseph Fourier – Grenoble I, 2006, page 93-107.

[GF94] M.R. Genesereth, R.E. Fikes, Knowledge Interchange Format, Version 3.0 Reference Manual. Technical Report Logic-92-1, Computer Science Department, Stanford University, 1994.

- [GF95] M. Gruninger and M. Fox. Methodology for the Design and Evaluation of Ontologies. In Proceedings of the Workshop on Basic Ontological Issues in Knowledge Sharing at the 14th International Joint Conference on Artificial Intelligence IJCAI1995, 1995.
- [GF96] M. Gruninger and M.S. Fox. "The Logic of Enterprise Modelling", Modelling and Methodologies for Enterprise Integration, P. Bernus & L. Nemes (Eds.), Cornwall, Great Britain: Chapman and Hall, 1996.
- [GP99] Gómez-Pérez A. "Ontological engineering: a state of the art", Expert Update – Knowledge Based Systems and Applied Artificial Intelligence, Vol. 2 No.3, pp.33-43, 1999.
- [Gru93a] T. R. Gruber. Towards principles for the design of ontologies used for knowledge sharing. In N. Guarino and R. Poli, editors, Formal Ontology in Conceptual Analysis and Knowledge Representation, Dordrecht, The Netherlands, 1993. Kluwer Academic Publishers.
- [Gru93b] Thomas R. Gruber. A translation approach to portable ontology specifications. Knowledge Acquisition, 5(2):199–220, 1993.
- [Gua98] Nicola Guarino, editor. Formal Ontology in Information Systems: Proceedings of the First International Conference. IOS Press, Amsterdam, 1998.
- [Her05] Nathalie Hernandez. Ontologies de domaine pour la modélisation du contexte en Recherche d'Information. Thèse de doctorat, Université Paul Sabatier, Toulouse, France, décembre 2005.
- [HGSY05] Xiangdan Hou, Junhua Gu, Xueqin Shen, and Weili Yan. Application of data mining in fault diagnosis based on ontology. In ICITA, pages 260–263. IEEE Computer Society, 2005.
- [HM09a] S. Hamani, R. Maamri, "Fuzzy Ontology-Driven Method for Ranking Unexpected Rules", CIA09, 3-4 May 2009, Saida, Algeria.
- [HM09b] S. Hamani, R. Maamri, "Ontology-Driven Method for Ranking Unexpected Rules", iCIT409, 2-3 June 2009, Amman, Jordanie.
- [HPY00] Jiawei Han, Jian Pei, and Yiwen Yin. Mining frequent patterns without candidate generation. SIGMOD Record (ACM Special Interest Group on Management of Data), 29(2):1–??, 2000.
- [IBM96] IBM (1996). IBM Intelligent Miner User's Guide. Version 1 Release 1, SH12-6213-00 edition.
- [JC97] Jay J. Jiang and David W. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. CoRR, cmp-lg/9709008, 1997. informal publication.
- [JZ+07] Jun Zhai Lixin Shen Zhou Zhou Yan Liang, Fuzzy Ontology Model for Knowledge Management, School of Economics and Management, Dalian Maritime University, Dalian 116026, P. R. China

- [Kay97] D. Kayser, *La représentation des connaissances*, Hermes, ISBN 2-86601-647-5, 1997.
- [KLW95] M. Kifer, G. Lausen, and J.Wu. Logical foundations of object-oriented and frame-based languages. *Journal of the ACM*, 42(4):741–843, May 1995.
- [KMR+94] Mika Klemettinen, Heikki Mannila, Pirjo Ronkainen, Hannu Toivonen, and A. Inkeri Verkamo. Finding interesting rules from large sets of discovered association rules. In *CIKM*, pages 401–407. ACM, 1994.
- [KP99] G. Kassel, S. Perpette, Cooperative ontology construction needs to carefully articulate terms, notions and objects, In *Proceedings of the International Workshop on Ontology Engineering on the Global Information Infrastructure*, 1999.
- [Lam02] G. Lame, *Construction d’ontologie à partir de texte, une ontologie du droit dédiée à la recherche d’information sur le Web*, Thèse de doctorat, Ecole des Mines de Paris, 2002.
- [LC98] Claudia Leacock and Martin Chodorow. Combining local context and WordNet similarity for word sense identification. In Christaine Fellbaum, editor, *WordNet: An Electronic Lexical Database*, pages 265–283. The MIT Press, Cambridge, Massachusetts, 1998.
- [LH96] Bing Liu and Wynne Hsu. Post-analysis of learned rules. In *AAAI/IAAI*, Vol. 1, pages 828–834, 1996.
- [LHC97] Bing Liu, Wynne Hsu, and Shu Chen. Using general impressions to analyze discovered classification rules. In *KDD*, pages 31–36, 1997.
- [Lin98] D. Lin, An information-theoretic definition of similarity, In *Proceedings of the 15th international conference on Machine Learning*, pp 296-304, 1998.
- [LMP05]Brisson, L., Collard, M., and Pasquier, N. (2005). Improving Knowledge Discovery Process Using Ontologies. In *International ICDM Workshop on Mining Complex Data*, Houston, Texas, USA.
- [LS99] Ora Lassila and Ralph R. Swick. Resource description framework (RDF) model and syntax specification. World Wide Web Consortium, Recommendation REC-rdf-syntax-19990222, February 1999.
- [Mac91] Robert M. Macgregor. “Using a Description Classifier to Enhance Deductive Inference”, *Proc. Seventh IEEE Conference on AI Applications*, Miami, Florida, February, 1991, pp 141-147.
- [Min75] M. Minsky. A framework for representing knowledge. In P. (ed.) Winston, editor, *The Psychology of Computer Vision*. McGraw-Hill, 1975.
- [MH02] Ir. Henry Nasution, M.T, Mechanical Engineering Faculty of Industrial Technology, Bung Hatta University : 2002.

- [MM95] John A. Major and John J. Mangano. Selecting among rules induced from a hurricane database. *J. Intell. Inf. Syst.*, 4(1):39–52, 1995.
- [MS93] Michael Sussna. Word sense disambiguation for free-text indexing using a massive semantic network. In *Proceedings of the Second International Conference on Information and Knowledge Management*, pages 67–74, Washington, D.C., November 1993.
- [MvH04] D. McGuinness and F. van Harmelen. 2004 OWL web ontology language overview. W3C recommendation, World-Wide-Web Consortium, <http://www.w3.org/TR/owl-features/>, 2004.
- [NH05] Nathalie HERNANDEZ, ONTOLOGIES DE DOMAINE POUR LA MODELISATION DU CONTEXTE EN RECHERCHE D'INFORMATION, THESE 2005.
- [NFM00] N. Fridman Noy, R. Ferguson, and M. Musen. The knowledge model of prot?g?-2000: Combining interoperability and flexibility. In *Proceedings of EKAW 2000*, LNCS 1937, page 17?32. Springer, 2000.
- [NLH+98]. Ng. R. T., Lakshmanan, L. V. S., Han, J., and Pang, A. “Exploratory mining and pruning optimizations of constrained associatino rules.” *SIGMOD-98*, 1998.
- [NR06] Natasha Noy and Alan Rector. Defining N-ary relations on the semantic web. W3C note, W3C, April 2006. <http://www.w3.org/TR/2006/NOTE-swbp-n-aryRelations-20060412/>.
- [Par04] David Parry. A fuzzy ontology for medical document retrieval. In Martin Purvis, editor, *Australasian Workshop on Data Mining and Web Intelligence (DMWI2004)*, volume 32 of *CRPIT*, pages 121–126, Dunedin, New Zealand, 2004. ACS.
- [Poh03] Carsten Pohle. Integrating and updating domain knowledge with data mining. In Marc H. Scholl and Torsten Grust, editors, *VLDB PhD Workshop*, volume 76 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2003.
- [PS91] Gregory Piatetsky-Shapiro. Discovery, analysis, and presentation of strong rules. In *Knowledge Discovery in Databases*, pages 229–248. AAAI/MIT Press, 1991.
- [PSM94] Gregory Piatetsky-Shapiro and Christopher J. Matheus. The interingness of deviations. In *KDD Workshop*, pages 25–36, 1994.
- [PT98] Balaji Padmanabhan and Alexander Tuzhilin. A belief-driven method for discovering unexpected patterns. In *KDD*, pages 94–100, 1998.
- [PT99] Balaji Padmanabhan and Alexander Tuzhilin. Unexpectedness as a measure of interestingness in knowledge discovery, January 09 1999.
- [P99] Padmanabhan, B., 1999. *Discovering Unexpected Patterns in Data Mining Applications*. Doctoral Dissertation, New York University, May 1999.
- [Qui68] M. R. Quillian. Semantic memory. In M. Minsky, editor, *Semantic information processing*. MIT Press, 1968.

[Ram90] Ram A 1990 Knowledge goals: A theory of interestingness. Proc. 12th Annual Conf. of the Cognitive Science Society, Cambridge, MA

[RBG02] Nick Routledge, Linda Bird, and Andrew Goodchild. UML and XML schema. In Xiaofang Zhou, editor, Thirteenth Australasian Database Conference (ADC2002), volume 5 of CRPIT, pages 157–166, Melbourne, Australia, 2002. ACS.

[Res93] Philip Resnik. Semantic classes and syntactic ambiguity. ARPA Workshop on Human Language Technology, March 1993. Resnik, P. Princeton.

[Res95] P. Resnik, Using information content to evaluate similarity in a taxonomy, In Proceedings of the 14th joint conference in Artificial Intelligence, 1995.

[Res99] P. Resnik, Semantic similarity in a taxonomy: an information based measure and its application to problems of ambiguity in natural language, Journal of Artificial Intelligence Research, volume 11, pp 95-130, 1999.

[RMBB89] R. Rada, H. Mili, E. Bicknell, and M. Blettner. Development and application of a metric on semantic nets. IEEE Transactions on Systems, Man, and Cybernetics, 19(1):17–30, January-February 1989.

[RS95] Ray Richardson and Alan F. Smeaton. Using WordNet in a knowledge-based approach to information retrieval. Technical Report CA-0395, School of Computer Applications, Dublin City University, Dublin, Ireland, 1995.

[SA95] R. Srikant and R. Agrawal. Mining generalized association rules. In Proceedings of the 21st International Conference on Very Large Data Bases (VLDB '95), pages 407–419, San Francisco, Ca., USA, September 1995. Morgan Kaufmann Publishers, Inc.

[Sah99] Sahar S 1999 Interestingness via what is not interesting. Proc. ACM Conf. on Data Mining (KDD-99) (San Diego, CA: ACM Press) pp 332–336

[Sow84] John Sowa. Conceptual structures. Information processing in man and machine. Addison-Wesley, Reading, US, 1984.

[Sow98] J. F. Sowa. Conceptual graph standard and extensions. Lecture Notes in Computer Science, 1453:3–??, 1998.

[SRV+97]. Srikant, R., Vu, Q. and Agrawal, R. “Mining association rules with item constraints.” KDD-97, 1997, pp. 67-73.

[SST+08] Nikos Simou, Giorgos Stoilos, Vassilis Tzouvaras, Giorgos B. Stamou, and Stefanos D. Kollias. Storing and querying fuzzy knowledge in the semantic web. In Fernando Bobillo, Paulo Cesar G. da Costa, Claudia d’Amato, Nicola Fanizzi, Kathryn B. Laskey, Kenneth J. Laskey, Thomas Lukasiewicz, Trevor P. Martin, Matthias Nickles, Michael Pool, and Pavel Smrz, editors, URSW, volume 423 of CEUR Workshop Proceedings. CEUR-WS.org, 2008.

[ST95] Abraham Silberschatz and Alexander Tuzhilin. On subjective measures of interestingness in knowledge discovery. In KDD, pages 275–281, 1995.

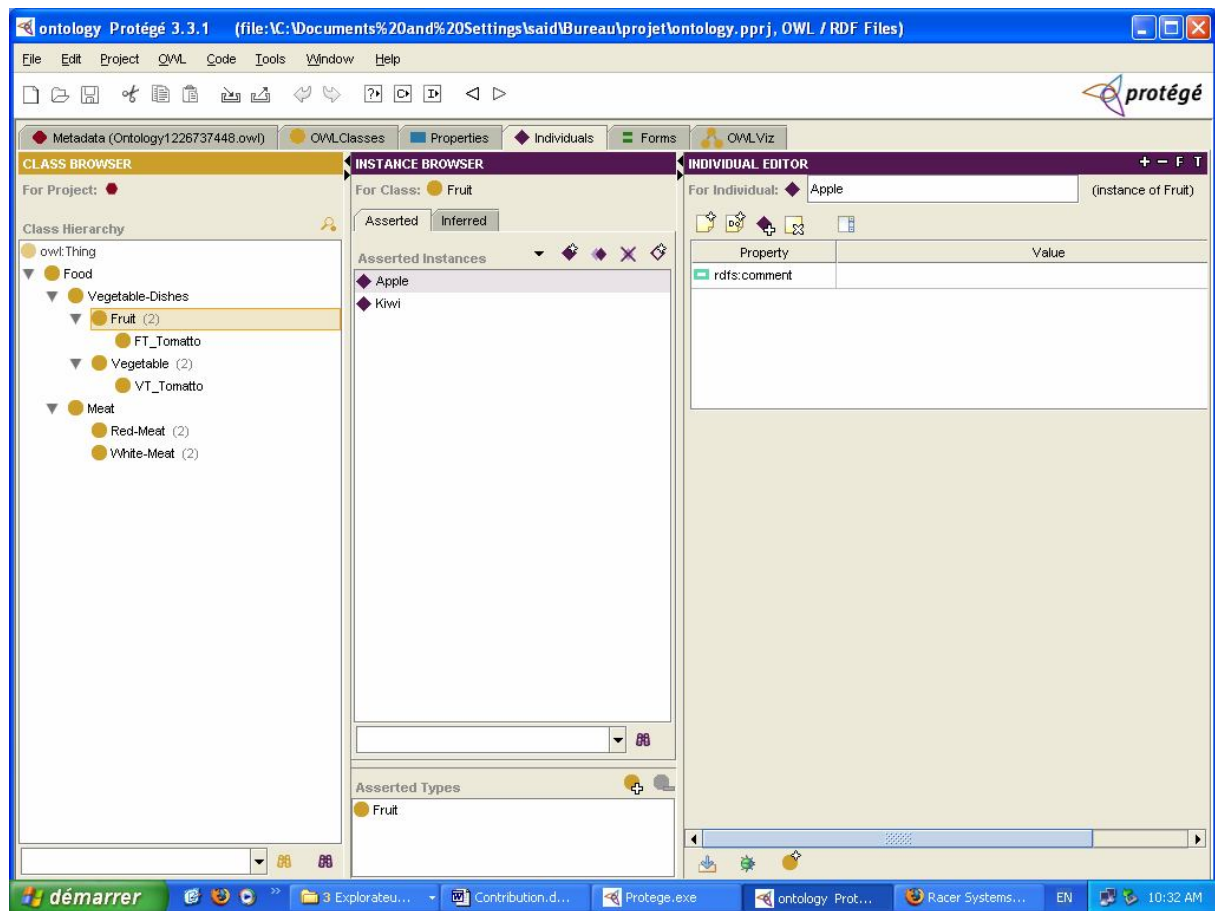
- [ST96] Abraham Silberschatz and Alexander Tuzhilin. What makes patterns interesting in knowledge discovery systems. *IEEE Transactions on Knowledge and Data Engineering*, 8(6):970–974, December 1996.
- [Sub98] Ramesh Subramonian. Defining diff as a data mining primitive. In *KDD*, pages 334–338, 1998.
- [Suz97] Einoshin Suzuki. “Autonomous Discovery of Reliable Exception Rules”, *Proc. Third Int’l Conf. Knowledge Discovery and Data Mining (KDD)*, AAAI Press, Menlo Park, Calif., pp. 259–262, 1997.
- [SWM04] M. K. Smith, C. Welty, and D. McGuinness. *OWL web ontology language guide, W3C recommendation*. World Wide Web Consortium, 2004.
- [TBMM01] H. S. Thompson, D. Beech, M. Maloney, and N. Mendelsohn (Eds). “XML Schema Part 1: Structures”. W3C Recommendation, May 2001. <http://www.w3.org/TR/xmlschema-1/>.
- [UFS91] Ramasamy Uthurusamy, Usama M. Fayyad, and W. Scott Spangler. Learning useful rules from inconclusive data. In *Knowledge Discovery in Databases*, pages 141–158. 1991.
- [UK95] M. Uschold and M. King. Towards a methodology for building ontologies. In *Workshop on Basic Ontological Issues in Knowledge Sharing*, held in conjunction with *IJCAI-95*, Montreal, Canada, 1995.
- [URL01] <http://www.w3.org/TR/swbp-n-aryRelations/#pattern1>
- [URL02] 3C Recommendation, RDF Reification, http://www.w3.org/TR/rdf-schema/#ch_reificationvocab.
- [URL03] Chistian Borgelt. <http://www.borgelt.net/software.html>.
- [URL04] <ftp://ftp.ics.uci.edu/pub/machine-learning-databases/census-income/>
- [URL05] http://interstices.info/jcms/c_17672/ontologies-informatiques?part=3
- [URL06] http://en.wikipedia.org/wiki/Dijkstra%27s_algorithm.
- [vHM04] Frank van Harmelen and Deborah L. McGuinness. *OWL web ontology language overview*. W3C recommendation, W3C, February 2004. <http://www.w3.org/TR/2004/REC-owl-features-20040210/>.
- [Woo75] William A. Woods. What’s in a link: Foundations for semantic networks. In Ronald J. Brachman and Hector J. Levesque, editors, *Readings in Knowledge Representation*. Morgan Kaufmann, Los Altos, CA, 1975.
- [WP94] Zhibiao Wu and Martha Stone Palmer. Verb semantics and lexical selection. In *ACL*, pages 133–138, 1994.

- [WY01] Dwi H. Widyantoro and John Yen. A fuzzy ontology-based abstract search engine and its user studies. In FUZZ-IEEE, pages 1291–1294, 2001.
- [Zad65] L. A. Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965.
- [Zim91] Hans-Jürgen Zimmermann. *Fuzzy Set Theory—And Its Applications*. Kluwer, Dordrecht, second revised edition, 1991.
- [ZK92] L. A. Zadeh and J. Kacprzyk. *Fuzzy Logic for the Management of Uncertainty*. JohnWiley & Sons, Inc., 1992.

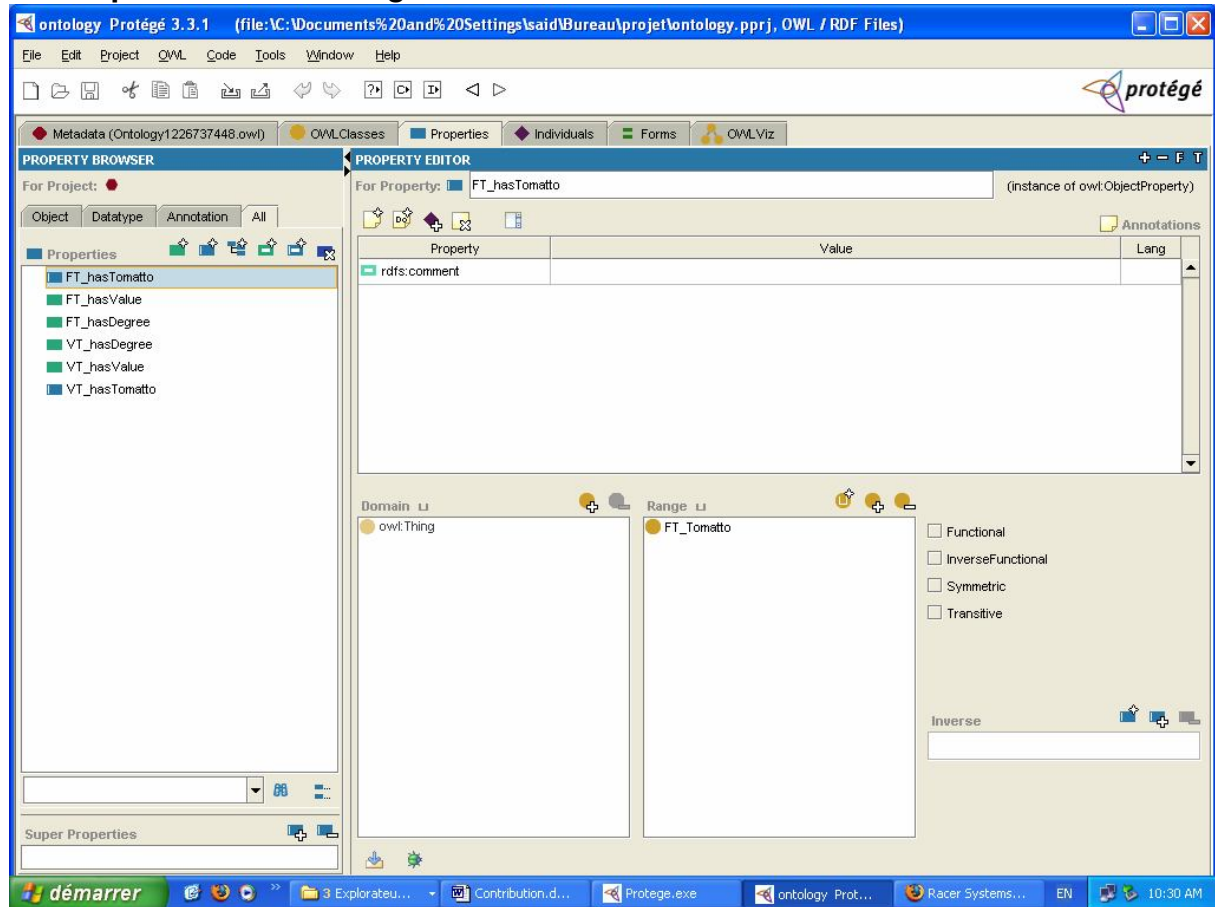
ANNEXE

L'ontologie de nourriture sous protégé

1. Hiérarchie de concepts sous protégé



2. Propriétés de l'ontologie nourriture



« Tomatto » a été représentée comme un nœud blanc avec « FT_Tomatto » sous « Fruit » et « VT_Tomatto » sous « Vegetable ».

3. Extrait de l'ontologie de nourriture

```
<?xml version="1.0"?>
<rdf:RDF xmlns="http://www.owl-ontologies.com/Ontology1226737448.owl#"
  xml:base="http://www.owl-ontologies.com/Ontology1226737448.owl"
  xmlns:p1="http://www.owl-ontologies.com/assert.owl#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:owl="http://www.w3.org/2002/07/owl#">
  <owl:Ontology rdf:about=""/>
  <Fruit rdf:ID="Apple"/>
  <Red-Meat rdf:ID="Beef"/>
  <Vegetable rdf:ID="Carrots"/>
  <White-Meat rdf:ID="Chicken"/>
  <owl:Class rdf:ID="Food"/>
  <owl:Class rdf:ID="Fruit">
    <rdfs:subClassOf rdf:resource="#Vegetable-Dishes"/>
  </owl:Class>
  <owl:DatatypeProperty rdf:ID="FT_hasDegree">
    <rdfs:domain rdf:resource="#FT_Tomatto"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
    <FT_hasDegree rdf:datatype="http://www.w3.org/2001/XMLSchema#float">0.3</FT_hasDegree>
  </owl:DatatypeProperty>
  <owl:ObjectProperty rdf:ID="FT_hasTomatto">
    <rdfs:range rdf:resource="#FT_Tomatto"/>
  </owl:ObjectProperty>
  <owl:DatatypeProperty rdf:ID="FT_hasValue">
    <rdfs:domain rdf:resource="#FT_Tomatto"/>
    <FT_hasValue rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Tomatto</FT_hasValue>
  </owl:DatatypeProperty>
  <owl:Class rdf:ID="FT_Tomatto">
    <rdfs:subClassOf rdf:resource="#Fruit"/>
  </owl:Class>
  <Fruit rdf:ID="Kiwi"/>
  <owl:Class rdf:ID="Meat">
    <rdfs:subClassOf rdf:resource="#Food"/>
  </owl:Class>
  <Red-Meat rdf:ID="Mutton"/>
  <Vegetable rdf:ID="Pepper"/>
  <owl:Class rdf:ID="Red-Meat">
    <rdfs:subClassOf rdf:resource="#Meat"/>
  </owl:Class>
  <White-Meat rdf:ID="Turkey"/>
  <owl:Class rdf:ID="Vegetable">
    <rdfs:subClassOf rdf:resource="#Vegetable-Dishes"/>
  </owl:Class>
  <owl:Class rdf:ID="Vegetable-Dishes">
    <rdfs:subClassOf rdf:resource="#Food"/>
  </owl:Class>
  ....
  <owl:Class rdf:ID="White-Meat">
    <rdfs:subClassOf rdf:resource="#Meat"/>
  </owl:Class>
</rdf:RDF>
```