REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE FARHAT ABBAS – SETIF
UFAS (ALGERIE)

MEMOIRE

Présenté à la Faculté des Sciences de l'Ingénieur
Département d'Informatique
Pour l'Obtention du Diplôme de

MAGISTER

ÉCOLE DOCTORALE(STIC)

Option : Ingénierie des Systèmes Informatique

Par :

Mr : Abdelouahab ATTIA

THEME

Extraction des connaissances par les règles d'association : Extension au cas flous

Soutenu le : 20/04/2010. Devant : un jury composé de :

Dr. TOUAHRIA Mohamed	MC à l'université (UFAS) Sétif	Président
Dr. MOUSSAOUI Abdelouahab	MC à l'université (UFAS) Sétif	Rapporteur
Dr. KHABABA Abdallah	MC à l'université (UFAS) Sétif	Examinateur

Remerciements

Je remercie tout d'abord le bon dieu pour m'avoir donnée le courage et la santé pour accomplir ce travail.

Je tiens à remercier mon encadreur, Monsieur : **Dr. MOUSSAOUI ABDELOUAHAB**, pour l'attention, le soutien, les précieux conseils et recommandations qu'il m'a accordé. Je le remercie surtout pour son suivi continuel durant la réalisation de ce mémoire.

Mes sincères remerciements s'adressent aussi à Monsieur le président de jury pour m'avoir fait l'honneur de présider le jury.

Je tiens également à remercier les examinateurs de ce mémoire pour m'avoir honoré de leur présence et accepté de juger ce travail.

Mes remerciements vont également à tous les enseignants et les responsables de l'école doctorale STIC/ISI de l'UFA Sétif pour l'effort qu'ils ont consacré pour la réussite de l'école.

Je n'oublierais pas à la fin, Monsieur : DRALI AHCEN, cadre dans la société SNTF/B.B.Arréridj qui n'a cessé de m'encourager.

Finalement, je remercie toute ma famille, tous mes amis et tous mes collègues.

L'étudiant
ATTIA ABDLOUAHAB

Sommaire

Liste des figures

Liste des tableaux

Introduction générale

Introduction

Depuis les années 90s, le monde suit avec optimisme l'évolution des technologies de l'information, particulièrement dans les réseaux de télécommunication, comme « *Internet* », permettant ainsi aux entreprises le contact facile et rapide avec leurs clients respectifs. Cette évolution rapide des moyens de communication a multiplié le rendement des entreprises et en contrario les a obligé à *collecter*, *enregistrer*, *maintenir*, *traiter, extraire* et *diffuser* rapidement toute les informations utiles relatives aux leurs activités essentielles

Aujourd'hui les données collectées sont stockées dans différentes bases de données, elles comportent un très grand nombre d'attributs en ayant plus d'un million d'enregistrements, parfois. Il est nécessaire donc, de développer et/ou d'automatiser des outils efficaces pour explorer ces informations volumineuses afin de récolter des *connaissances utiles*.

Cependant, l'émergence de la fouille de données (*Data Mining* en anglais) a fortement contribuée à l'évolution des techniques de prise de décision. La fouille de données, à l'époque, était considérée comme étant un domaine d'extraction de connaissance à partir des données ou ECD (en anglais *KDD pour knowledge Discovery in Databases*). Maintenant, ces deux concepts désignent deux domaines complètement différents. Il ne faut pas confondre la fouille de données avec l'extraction de connaissance à partir de donnée. La fouille de données est un sous-processus de processus ECD, englobant plusieurs algorithmes et techniques qui permettent de mieux analyser ces bases de données gigantesques. Principalement, les outils de la fouille de données sont dérivés des domaines tels que : statistique classique, visualisation, reconnaissance des formes, et de l'intelligence artificielle. Leur but est de réaliser des modèles à partir des bases de données. Ces modèles seront utilisés, par la suite, dans différentes tâches menant à la découverte des nouvelles connaissances.

Les règles d'association sont l'une des techniques les plus utilisées dans le processus de fouille de données. Introduite par *Agrawal et al.* [AGRA 1993]. Cette technique est d'un grand intérêt pour la communauté de la fouille de données où plusieurs recherches ont été menées afin de développer de nouveaux algorithmes permettant, à la fois, de découvrir et d'extraire de nouvelles relations entre un grand nombre d'attributs [AGRA 1994][SAVA 1995]. La première fois, les règles d'association étaient conçues pour résoudre des problèmes

purement économiques (*analyse du panier de la ménagère*) [AGRA 1993]. Actuellement, elles sont considérées comme l'une des techniques les plus intéressantes dans le domaine de la fouille de données dans différentes disciplines, comme le traitement des indices boursiers, l'analyse des données web, l'extraction d'informations à partir d'images médicales, etc. Il y a lieu de préciser que l'utilisation des règles d'association (RA) a contribué fortement à l'émergence de nouveaux domaines, propre à la fouille de données, comme le *web mining*, *text mining*, *image mining*, *graphe mining*, *ADN mining*, etc.

Au début, les règles d'association étaient utilisées pour tirer des relations à partir de données binaires. La fonction qui nous renseigne sur la présence ou non de l'attribut, dans une transaction, est définie dans l'ensemble {0,1} (le 1 signifie que l'attribut apparait dans la transaction, et le 0 désigne le cas contraire).

Les données, en générale, sont de deux types : *quantitatif* et *qualitatif*, conduisant à la définition d'autre type de règles basées sur les règles d'association binaire, nous citons :

- ***Les règles d'association quantitatives***, où les attributs quantitatifs sont divisés en intervalles et où les éléments sont soit des membres ou non de ces intervalles. Avec cette approche, une base de données quantitative peut être transformée en une base de données binaire. Lors de processus de traitement en risque de négliger les éléments situe aux extrémités des intervalles, cela forcément conduit à un problème qui ce nommé « *problème d'extrémité d'intervalle* », en anglais « *the sharp boundary problem* ».
- ***Les règles d'association généralisées***, où l'attribut peut avoir un ou plusieurs ancêtres comme « *imprimante laser* », « *imprimante jet d'encre* » ; leurs ancêtre est « *imprimante* ». Avec un ou plusieurs taxinomies (structure en arbre), cet attribut permet d'étendre la table de transaction et augmenter les chances ainsi pour trouver des relations intéressantes.

La théorie des sous-ensembles flous a un impact sur les techniques de la fouille de données et sur les règles d'association, en particulier, où l'on trouve dans la littérature plusieurs approches basées sur les règles d'associations classiques en intégrant les sous-ensembles flous, nous citons : l'approche *quantitative*, *structure taxonomies floues*, et la *recherche des motifs séquentiels flous* [MANN 1995][KUOK 1998][GYEN 2001][FIOT 2004].

Dans ce travail, on s'intéresse de près aux règles d'association et aux règles d'association floues où la théorie des sous-ensembles flous ainsi que la théorie de la logique floue, qui en découle, sont utilisées comme modèles de représentation des connaissances dans

le processus d'extraction des règles d'associations floues. En outre, nous proposons une nouvelle approche pour l'extraction des règles d'associations floues basée sur l'algorithme *Apriori* initial [AGRA 1994] étendu aux cas de données floues.

Ce mémoire est subdivisé en quatre chapitres :

1. Le premier chapitre « ***Extractions des connaissances à partir des données*** » : Ce chapitre contient les concepts de base, d'extraction des connaissances, ainsi qu'un survol sur le processus « fouille de données ».
2. le deuxième chapitre « ***les règles d'association classique »***: où l'on décrit les concepts de bases des règles d'associations classiques ainsi que les différents algorithmes permettant leurs extractions. En fin de ce chapitre, nous présentons les règles d'associations quantitatives et généralisées.
3. Le troisième chapitre « ***généralité sur la logique floues*** » : ce chapitre est une brève introduction à la théorie des sous ensembles flous qui constitue la base dans l'extraction des règles d'associations floues. Les différents opérateurs flous font également office de ce même troisième chapitre.
4. le quatrième chapitre « ***les règles d'association floues*** » : ici, nous décrivons les différentes approches permettant à la fois l'extraction des règles d'association floues et le calcul de support. Comme contribution, nous avons opté pour une extension de l'algorithme initial *Apriori* aux donnés flous.

Chapitre I

Extraction de Connaissances à Partir de Données (ECD)

I.1. Introduction

Aujourd'hui, l'informatique est devenu un outil indispensable dans des domaines différents tel que: l'industrie, la biologie, le marketing, la médecine, etc. et dont le but majeur est de stocker les données, ainsi que leurs traitements.

Jours après jours, les données collectées augmentent d'une façon exponentielle atteignant des volumes de l'ordre de Téraoctets [STEP 2005]. Le stockage de ces données ne pose pas de réelles difficultés du point de vue informatique, mais le besoin d'interpréter ou de trouver de nouvelles relations entre les éléments stockés dans ces bases a suscité beaucoup d'intérêts.

En effet, ces données volumineuses causent beaucoup de problèmes dans leur analyse comme : la représentation, la recherche, l'exploration, etc. Aussi les besoins des organismes ne restent pas dans un cadre classique ou l'application des requêtes traditionnelles. Les systèmes classiques de gestion de bases de données ne fournissent pas des fonctionnalités nécessaires pour la bonne exploitation de ces données gigantesques et du coup ils ne satisfirent plus les exigences des organismes, qui deviennent de plus en plus exigeants. En effet, ces derniers ont besoin de systèmes d'aide à l'analyse et à la prise de décision de plus en plus performants. Par conséquent, le développement technologique a eu un fort impact sur tous les secteurs, ce qui a poussé les organismes à viser d'autres pistes pour la compréhension des phénomènes complexes. Les chercheurs ont visé d'autres techniques pour une meilleure exploitation de ces grandes mass de données [FRAW 1992] [JIAW 2000][DUVA 2000].

Dans ce chapitre nous définissons et nous illustrons les différentes notions du processus d'extractions des connaissances à partir des données. Nous commençons par définir les concepts de base de la fouille de données avant de présenter les méthodes et les techniques relatives à ces dernières.

I.2 Historique d'évolutions des systèmes décisionnels

Le développement de l'outil informatique, plus particulièrement dans le logiciel (*Soft*), à simplifier beaucoup de tâches dans la vie quotidienne de l'homme et surtout dans la prise des décisions dans des domaines différents. La figure (FIG.1) illustre l'évolution de ces systèmes.

Dans les années (70), les décideurs recevaient des piles de listings avec des tableaux chiffrés, à cette époque la décision se faisait par des rapports qui sont les résultats d'une analyse de données spécifiques et dont lesquelles l'utilisateur (décideur) a un apport primordial.

Au début des années (80), l'apparition des langages des requêtes comme SQL (*Select Query Language*), a permis de mieux cibler l'information à extraite. La décision se fait par une analyse de données spécifiques en fonction de leur contexte [GARD 2001].

L'évolution s'est poursuivie par les systèmes de requêtes, on a assisté à l'émergence d'autres types d'analyses comme celui de l'analyse décisionnelle : comme «*On Line Analytical Processing (OLAP)* », qui opèrent sur des entrepôts de données (appelés « *Datawerhosse*» en anglais). Ce type d'analyse, qui à débuté dans les années (90), s'est encore dotée d'outils plus sophistiqués, comme les méthodes de navigation dans les données: « *drill-down/drill-up, rotate, slicing, scoping* ». Ces opérations facilitent encore plus de tâches pour les décideurs afin de prendre les bonnes décisions.

Vers le milieu des années (90), des architectures plus complètes et plus complexes, ont commencé à apparaître dans les systèmes d'extraction de connaissances à partir de données, c'est l'émergence de « *Data Mining* ». Dans le début du deuxième millénaire, les portes se sont ouvertes sur l'extraction et à la gestion des connaissances. Ces systèmes visent à fournir une exploration très souple et une représentation synthétiques des données et la déduction de nouvelles connaissances.

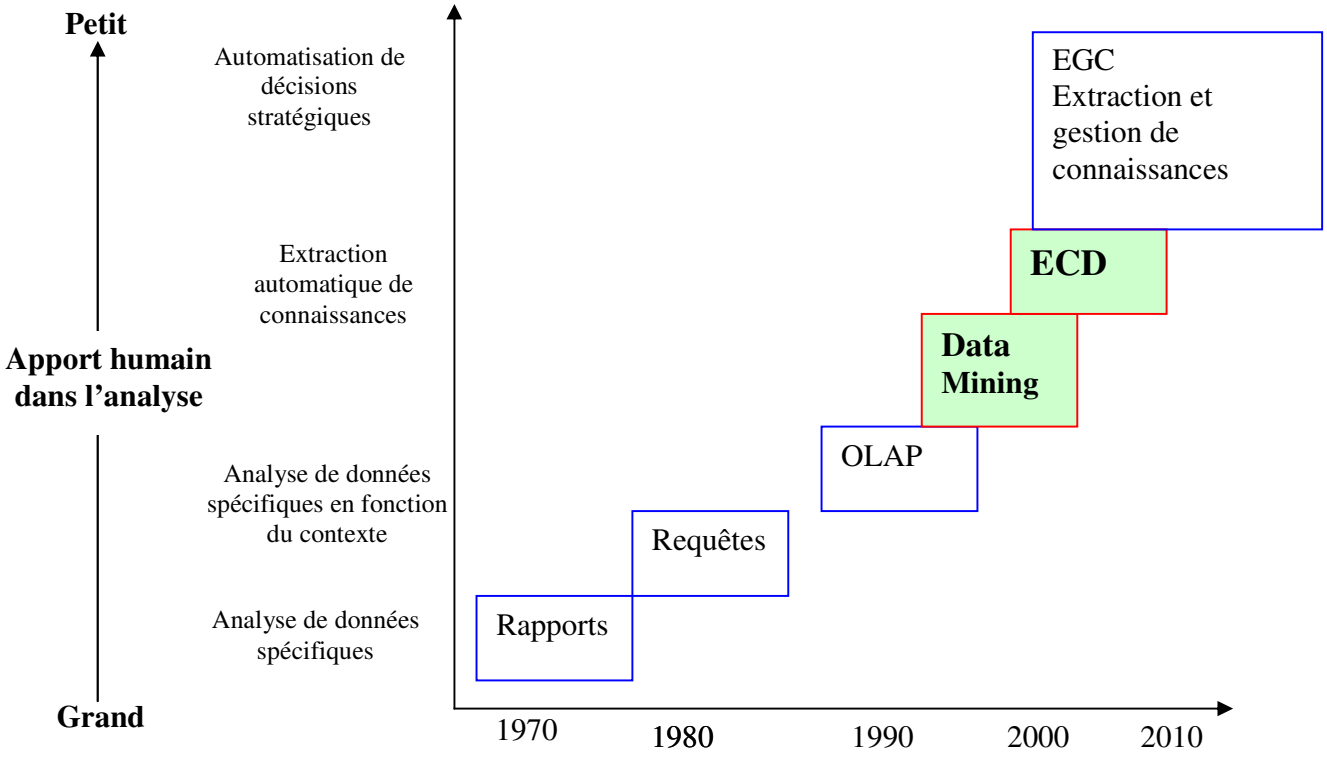


FIG.1. EVOLUTION DES SYSTEMES DECISIONNELS [ZIGH 2003]

I.3 L'extraction des connaissances à partir de donnée (ECD)

I.3.1. définition d'ECD

Le terme ECD désignant l'extraction des connaissances à partir des données (*Knowledge Discovery in Databases : KDD*) est né des travaux croisés des chercheurs en statistique, en intelligence artificielle, en reconnaissance des formes, et de visualisation [JIAW 2000]. Ces travaux ont pour objectif l'automatisation de processus de découverte des connaissances pertinentes nouvelles et utiles dans les bases de données gigantesques. On trouve dans la littérature plusieurs définitions de l'ECD. *Frawley et al*. [FRAW1992] indiquer que la découverte des connaissances est l'extraction d'informations non triviale implicite, auparavant inconnues, potentiellement utiles à partir des données. Afin d'obtenir ces informations, nous essayons de crées des modèles, parmi ceux erronés et d'autres utiles. Pour choisir les modèles intéressants, il faut l'examiner dans un programme. Les modèles intéressants sont appelés *connaissances*, et le résultat du programme est appelé *découverte de connaissance* [FRAW1992].

Fayyad et al. [FAYY1996], ont défini l'ECD comme *«le processus non trivial d'identification, à partir de données, de patterns valides, nouveaux, utiles et compréhensibles»*. D'après cette définition, on conclut que l'ECD est un processus qui comporte plusieurs étapes débutant par parcourir toute la base de données ou une partie de celle-là afin de décrire des modèles ou patterns ou découvrir des relations entre les données. Le but est de rendre les informations plus lisibles et compréhensibles, pour découvrir des nouvelles connaissances qui sont utiles à la prise de décisions.

Une autre définition dans [ZIGH 2003] annonce que : *« l'ECD vise à transformer des données (volumineuses, multiformes, stockées sous différents formats sur des supports pouvant être distribués) en connaissances. Ces connaissances peuvent s'exprimer sous forme d'un concept général qui enrichit le champ sémantique de l'usager par rapport à une question qui le préoccupe. Elles peuvent prendre la forme d'un rapport ou d'un graphique. Elles peuvent s'exprimer comme un modèle mathématique ou logique pour la prise de décision. Les modèles explicites quelle que soit leur forme, peuvent alimenter un système à base de connaissances ou un système expert ».* Cette définition est plus détaillée que la précédente et où elle introduit un nouveau concept qui est celui de l'utilisation des connaissances extraites, par un système de gestion des connaissances.

I.3.2. Processus de L'ECD

L'ECD est un processus itératif et interactif qui exige l'intervention de l'utilisateur dont le but est de découvrir des connaissances de qualités. Selon le schéma de [FAYY 1996], illustré dans la figure (FIG.2). Le processus d'ECD comporte essentiellement cinq étapes comme explicitées ci-dessous:

1. ***La sélection des données :*** L'objectif de l'extraction de connaissance est de déduire des nouvelles connaissances valides et utiles. Cet aspect est très important, mais on ne peut pas appliquer le processus d'ECD sur toutes les données qu'on a, donc le besoin est exprimé par l'utilisateur. Ce dernier fait la sélection des données selon l'objectif visé. Par exemple, si on veut extraire ou trouver des relations entre les produits vendus par une entreprise commerciale, il est inutile de consulter les données du personnel qui travaillent dans l'entreprise, ou encore de télécharger des pages Web qui parlent de Marketing; il s'agit d'explorer les données transactionnelles archivées, concernant les achats des clients.

Cette étape concerne donc le filtrage des données qui comprend deux opérations nécessaires :

a) ***La réduction de la dimensionnalité des données*** : l'élimination d'attributs sans intérêt, ou ayant beaucoup de valeurs erronées ou manquantes

b) ***La réduction de la taille des données*** : l'application des techniques du *Data Mining* est très coûteuse en terme de temps CPU et d'espace mémoire, c'est pour cela qu l'on ne peut pas l'appliquer sur la totalité des données [STEP 2005], comme les algorithmes génétiques, par exemple. Ces derniers explorent uniquement une partie de l'espace de solution selon une fonction *fitness* propre au processus de sélection ou guidé par l'utilisateur [JESU 2009]. La réduction des données peut également être faite par des techniques statistiques d'échantillonnage [LUDO 2006].

2. ***Le prétraitement des données :*** Le rôle de cette étape est de préparer les données afin qu'ils soient de meilleurs qualités afin d'arriver à des résultats de qualité.

Le prétraitement des données concerne, entre autres, le nettoyage des données, c'est-à-dire l'élimination du bruit, ainsi que le traitement des valeurs manquantes, ou erronées. Il faudrait alors définir les méthodes à utiliser pour le remplacement de ces valeurs [JIAW 2000]. De nombreuses solutions existent pour ce problème. On peut, par exemple, remplacer les valeurs manquantes par la valeur la plus fréquente de l'attribut en question, ou l'on estime ces valeurs à partir des enregistrements complets à travers la régression ou les réseaux de neurones [JIAW 2000].

Par exemple dans [ZIGH 2003] une valeur est considérée comme erronée si elle s'écarte de la moyenne de deux fois l'écart type.

3. ***La transformation de données :*** Cette étape consiste à préparer les données brutes, et à les convertir en données appropriées. La transformation se fait par attribut, c'est-à-dire toutes les valeurs d'un attribut doivent être transformées en un format unique. Formellement, un attribut ***A*** est transformé en ***A'*** qui serait utilisable par la tâche de la fouille de données choisie.

Un exemple connu de transformation est la discrétisation de variable continue. 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. Il y a beaucoup de méthodes de discrétisation dans la littérature [STEP 2005].

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.

4. ***Fouille de données (Data Mining) :*** Dans cette étape, des méthodes intelligentes sont utilisées afin d'extraire des modèles ou *patterns*. Cette étape est aussi désignée comme l'étape ***cœur du processus d'ECD*** (Voir en détail dans la (**section I.4.)**. Il est clair que les étapes qui précèdent la fouille de données sont très importantes, car la qualité des modèles ou patterns extraites, ainsi que leur coût d'extraction sont liés directement à ces étapes.

5. ***Evaluation et interprétation des connaissances :*** Les modèles ou patterns extraits ne sont pas dans la plupart du temps exploitables. En effet, il est difficile d'avoir directement des connaissances valides et utiles, à ce point là. Il existe, cependant, des méthodes d'évaluation des modèles extraits. Ces méthodes peuvent aussi aider à corriger les modèles, et à les ajuster aux données [JIAW 2000].

Les connaissances obtenues devraient être interprétables, *nouvelles*, *valides* et *utiles* aux utilisateurs. Ces derniers peuvent les utiliser directement, ou les incorporer dans un système de gestion de connaissances. Enfin, cette étape identifie les modèles intéressants qui représentent les connaissances ne se basant pas seulement sur des mesures d'intérêt ou des résultats affichées mais aussi sur l'avis de l'expert.

Ces principales étapes de processus (ECD) visent à partir de données volumineuses l'extraction des connaissances, qui peuvent être exprimés sous forme d'un concept général qui enrichit le champ sémantique de l'usager par rapport à une question qui le préoccupe. Elles peuvent prendre la forme

d'un rapport ou d'un graphique. Elles peuvent aussi s'exprimer comme des modèles mathématiques ou logiques. Toutes ces formes des connaissances ont comme but, commun et le plus intéressant, l'aide dans la prise de décision.

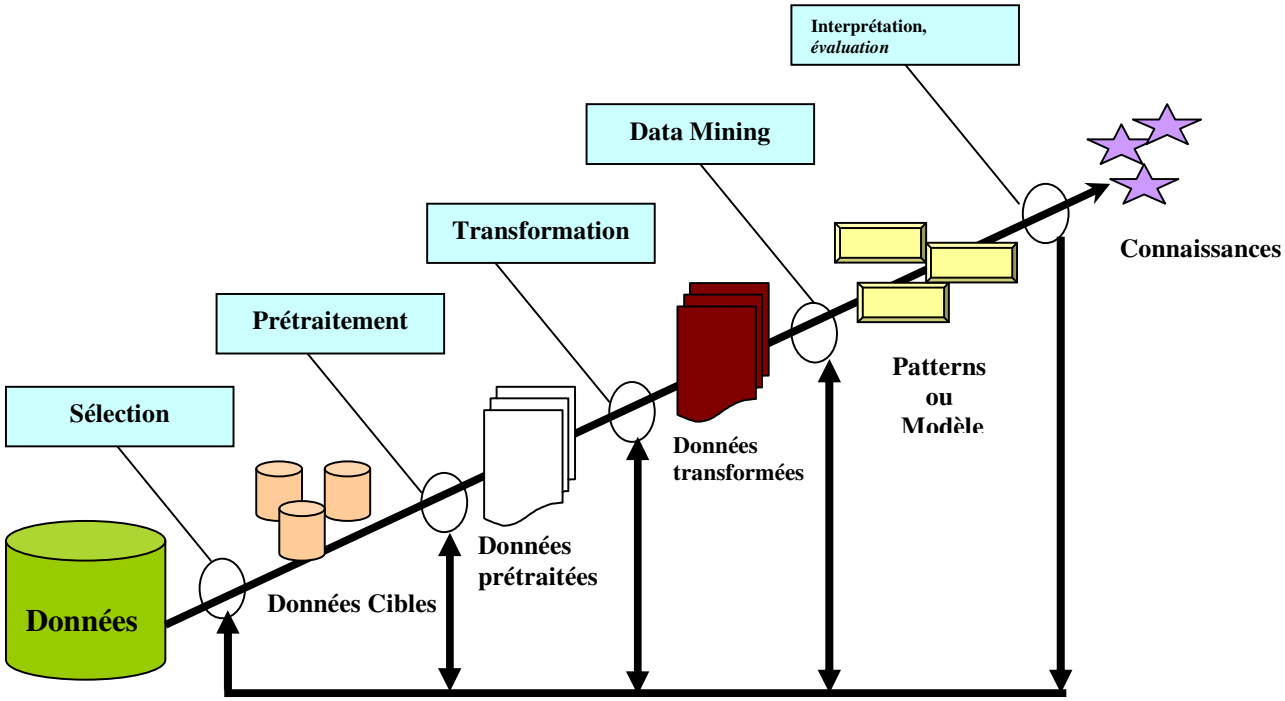


FIG.2 : PROCESSUS D'ECD [FAYY1996]

Selon le schéma de [FAYY 1996],il est claire que le processus est **itératif,** car il fait des retours dans n'importe quelle étape, pour des besoins de correction et de validation afin d'obtenir des connaissances de qualité. Ces retours sont déclenchés par l'utilisateur, ce qui montre l'aspect **interactif** du processus.

I.3.3. caractéristique de processus ECD

Selon *Frawley et al*. [FRAW1992], l'ECD expose les principales caractéristiques suivantes:

- ***Langage de haut niveau***: La découverte des connaissances est représentée par un langage qui ne doit pas nécessairement être utilisé par les humains, mais son expression doit être compréhensible.
- ***Précision***: La mesure de certitude implique de savoir que les modèles découvraient qu'ils sont représenté correctement le contenu d'une base de données ou non.

- ***La notion d'intérêt*** : la découverte des connaissances est considérée comme intéressante si elle remplit les normes prédéfinies en désignant un modèle intéressant si elle est nouvelle, potentiellement utile et non trivial.
- ***Efficacité***: le temps d'exécution de l'algorithme est acceptable quelque soit le type de traitement.

I.4. Fouille de donnée (Data Mining)

L'expression de « *Data Mining* » est apparue en 1960, à cette époque, elle a un sens péjoratif. Le développement des moyens informatiques et du calcul a permis le stockage d'une grande masse d'informations. Le traitement et l'analyse de celle-ci nécessitant des méthodes sophistiquées, ce qui à conduit les chercheurs d'utiliser des méthodes d'autres domaines, tel que : les statistiques, les mathématiques, l'intelligence artificielle, etc. afin de mieux exploiter ces gros volumes de données. Jusqu'au milieu des années 90, le « *Data Mining* », trouve sa place dans le champ scientifique.

I.4. 1. Définition de la fouille de données

La fouille de données est l'ensemble des techniques et des méthodes intelligentes permettant l'extraction des connaissances, auparavant inconnues, à partir d'une grande masse de données. Il s'agit de recherche aux profonds de ces données visant à découvrir des informations cachées, afin de rendre ces quantités de données sous forme des modèles (une connaissance qui concerne la totalité des données et on peut l'appliquer à chaque nouvelle donnée) ou patterns (une connaissance qui concerne une partie des données et on ne peut pas l'appliquer à des nouvelles données) [MEHM 2001]. Ainsi on trouve dans la littérature plusieurs définitions de la fouille de données, parmi elles [VALE 2006] :

- C'est un processus de recherche d'information dans des bases de données gigantesque
- Exploration est analyse des données par des moyens automatique et semi-automatique pour la découverte de modèles de données ou la découverte des connaissances.
- Un processus itératif et interactif de découverte des modèles valides, nouveaux, utiles, est compréhensible dans une grande base de données.
- Est l'étape coeur dans le processus (ECD).

Le terme *Data Mining* est plus populaire que le terme ECD (extraction des connaissances à partir des données). Beaucoup, font la confusion entre le concept de Data Mining et celui de l'ECD, ils

les considèrent comme synonymes. Le Data Mining est l'élément essentiel dans le processus d'extraction des connaissances à partir des données [DUVA 2000].

I.4.2. Les méthodes de fouille de données

Comme nous l'avons indiqué dans la section précédente le *Data Mining* dans son sens restreint est au cœur du processus d'ECD. La fouille de données englobe plusieurs méthodes, qui permettent de créer des modèles ou patterns afin de les utiliser pour la découverte de connaissances. Nous nous concentrerons sur les méthodes d'explications et de prédictions. Ces méthodes peuvent être classifiées en trois catégories [RIAD 2007].

I.4.2.1. Les méthodes de visualisation et de description

Les méthodes de visualisation et de description, sont issues de la statistique descriptive et de l'analyse des données, ainsi que de la visualisation graphique. Ces méthodes permettant la visualisation des données, et ciblent donc l'analyse exploratoire des données. L'objectif demeure toujours le même; le dégagement de patterns, de structures, de synthèses, des habitudes de la population etc. Cette analyse de données s'appuie essentiellement sur la puissance de l'oeil et du cerveau humain.

En effet, les méthodes de visualisation et de description sont fondées sur des graphiques, qui facilitent l'interprétation à l'utilisateur. La visualisation d'un graphique sert principalement à explorer les données, ou à confirmer des hypothèses. Les graphiques incluent les statistiques élémentaires (tel que la moyenne, l'écart type, la variance, le mode, la médiane), les histogrammes pour un affichage uni-variable, les nuages de points (*scatterplot*), les courbes de niveaux (contour plot) pour un affichage bi-variables, la matrice de nuages de points (*scatterplot matrix*), les graphiques de treillis (*treillis plot*) et les graphiques de visualisation radiale (*radial visualization*) pour un affichage multi-variables [HAND2001].

I.4.2.2. Les méthodes de classification et de structuration

Les méthodes de classification et de structuration connus sous le nom *classification automatique ou apprentissage non supervisé*. Ces méthodes proviennent de l'analyse des données, de la reconnaissance des formes, de l'apprentissage automatique et du connexionnisme [RIAD 2007].

I.4.2.3. Les méthodes d'explication et de prédictions

Les méthodes d'explication et de prédictions ont pour objectif de relier un phénomène à expliquer à un phénomène explicatif, elles sont utilisées pour prévoir un comportement, ou bien pour

classer de nouveaux cas dans des catégories prédéfinies. Ces méthodes sont issues de la statistique, de l'économétrie, de la reconnaissance de formes, de l'apprentissage automatique et du connexionnisme. Parmi ces méthodes, on site :

1. ***Les Réseaux de Neurones :*** Les réseaux de neurones ciblent la prédiction. Ils permettent de construire un modèle qui prédit la valeur d'une variable à partir d'autres variables connues appelées variables prédictrices. Si la variable à prédire est discrète (qualitative), alors il s'agit d'une ***classification***, si elle est continue (quantitative), alors il s'agit de ***régression***.

Un réseau neuronal est composé de groupes de noeuds (neurones). Chaque groupe de noeuds correspond à une couche. Un réseau neuronal est formé par au moins trois couches ; *entrée*, *intermédiaire* et *sortie*. Dans la couche entrée, chaque noeud correspond à une *variable prédictrice*. La couche sortie contient un ou plusieurs noeuds ; la ou les variables à prédire. Le réseau peut avoir plusieurs couches intermédiaires (mais une seule couche entrée et une seule couche de sortie). Les couches intermédiaires, appelées aussi couches cachées.

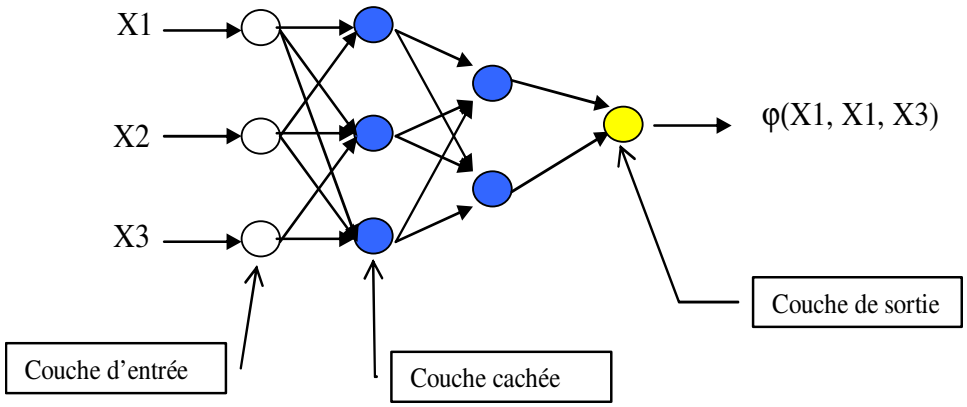


FIG.3 RESEAU DE NEURONE[RIAD 2007].

2. ***Les Réseaux Bayésiens :*** Les Réseaux Bayésiens correspondent à une modélisation descriptive des données. Egalement appelés réseau de croyance, réseau bayésien est un réseau probabilistique. Un réseau bayésien est un graphe dirigé, acyclique où chaque noeud représente une variable continue ou discrète, et chaque arc représente une dépendance probabilistique. Si un arc relie un noeud Y à un noeud Z, alors Y est le parent de Z, et Z est le descendant de Y. Chaque variable est indépendante des variables aux quelles elle n'est pas reliée. Les variables peuvent être continues ou discrètes. Chaque lien entre deux variables est pondéré par la valeur de la dépendance probabilistique, ainsi la valeur que porte l'arc reliant Y à Z est en fait P(Z/Y) [JIAW 2000].

3. ***Les méthodes statistiques de prédiction :*** englobe les méthode ci-après :

- ***La régression linéaire* :** C'est une technique statistique qui vise la prédiction de la valeur d'une variable continue. L'objectif est de déterminer le meilleur modèle qui relie une variable quantitative de sortie à plusieurs variables prédictrices d'entrées. Cette opération s'appelle ajustement du modèle aux données. Les modèles linéaires sont les plus fréquemment utilisés. C'est ce qu'on appelle la régression linéaire. La relation qui relie une variable à prédire à plusieurs autres variables prédictrices est une équation de régression, souvent sous cette forme: $Y = a_0 + a_1 X_1 + a_2 X_2 + ... + a_p X_p$ Il s'agit là de prédire la variable Y à partir de p variables X .

- ***La régression logistique* :** Les modèles de régression logistique produisent la probabilité qu'une variable ait une valeur donnée. Par exemple, au lieu de prédire s'il est risqué ou non d'accorder un crédit à un client X, cette méthode essaye d'estimer la probabilité pour que la décision soit favorable ou non. La régression logistique est utilisée seulement si la variable à prédire est de type binaire (ne peut prendre que deux valeurs), elle cible donc la classification.

- ***L'analyse discriminante linéaire* :** C'est une vieille méthode (publiée en 1936 par R.A Fisher) qui vise la classification. Elle essaye de dégager des hyperplans qui séparent des classes (des lignes pour deux dimensions, des plans pour trois dimensions, etc.). Le modèle résultant est facile à interpréter car les utilisateurs n'ont qu'à déterminer de quel côté l'enregistrement se situe (donc on le classifie).

Cette technique n'est pas populaire en Data Mining pour trois raisons. Premièrement, elle suppose que toutes les variables prédictrices ont une distribution normale ce qui n'est pas toujours le cas. Deuxièmement, les variables prédictrices doivent être ordonnables (une variable *couleur_des_yeux* nous poserait des problèmes). Finalement, les frontières qui séparent les classes sont linéaires (lignes, plans,…), ce qui n'est pas réel.

Ceci dit, de nouvelles versions de techniques d'analyse discriminante essayent de résoudre ces trois problèmes [TOW 1999].

4. ***Les Arbres de Décision :*** Les arbres de décision ciblent la classification (prédiction de variables discrètes). Comme son nom l'indique, cette méthode consiste à construire un arbre. Un enregistrement (qu'on veut classifier) entre par le noeud racine, et passe d'un noeud père à un noeud fils s'il satisfait une condition posée. Le noeud feuille auquel il arrivera est sa classe. Un arbre de décision peut donc être aperçu comme étant un ensemble de règles qui mènent à une classe.

La figure (FIG.4) montre un arbre de décision utilisé pour la prise de décision de l'approbation ou non de crédits bancaires :

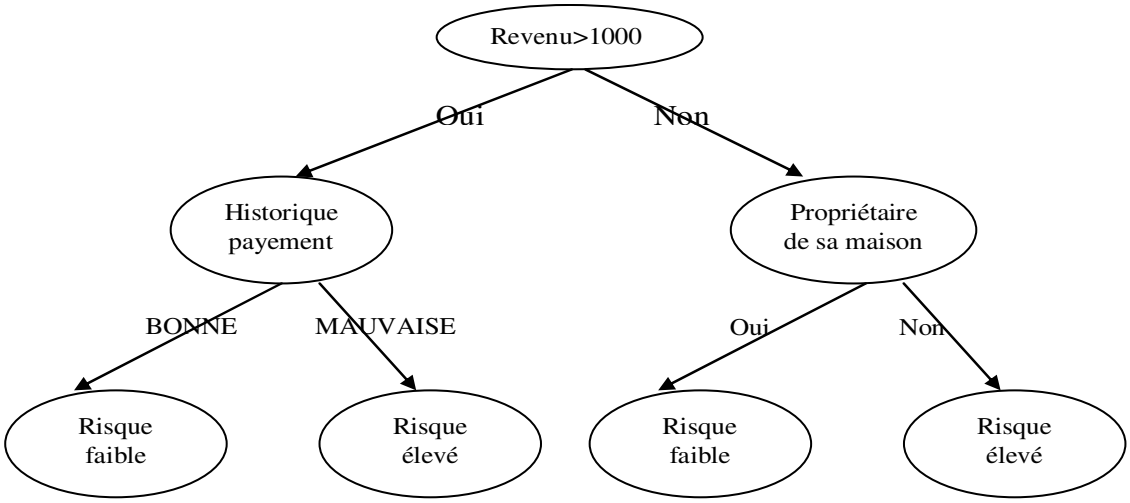


FIG.4 EXEMPLE D'ARBRE DE DECISION

Un dossier candidat (enregistrement) pointe sur le noeud racine. Selon la condition (information que porte chaque noeud) mentionnée, il pointe sur l'un des noeuds fils. Cette action sera exécutée itérativement jusqu'à ce que le dossier candidat pointe sur un nœud feuille. Ce noeud feuille porte le nom de la classe à laquelle il appartient. Le dossier candidat ne peut emprunter qu'un seul chemin. La construction de l'arbre se fait par un algorithme approprié. Plusieurs algorithmes existent dans la littérature, les plus connus sont CART (*Classification And Regression Trees*) [BREI 1984], C4.5 [QUIN1986] et CHAID [HART 1975] (*Chi-squared Automatic Interaction Detection*). Un arbre de petite taille risque de mal prédire les données, et de donner des résultats inexacts. Un arbre complexe risque d'être sur-ajusté, il prédit très bien les cas à partir lesquelles il fut construit, mais pas les nouveaux cas.

5. ***Les règles d'association*** : les règles d'association sont parmi les méthodes de recherche des implications (relation) entre les attributs d'une base de données. Les règles d'association ont été introduites par Agrawal et al. [AGRA 1993]. L'étymologie des règles d'associations, aussi connues sous le nom de l'analyse du panier de la ménagère (*Market Basket Analysis*), vient des travaux qui ont été réalisés à partir des données provenant des supermarchés. L'objectif de ces travaux est d'identifier les items ou les groupes d'items (Itemset), fréquemment achetés ensembles par un client lors d'une même transaction où son support est supérieur ou égal à un seuil minimum, *Minsupp*, fixé par l'utilisateur [AGRA 1993].

Ces règles sont de la forme: $A \rightarrow B$ ***[support% confiance%],*** Où $A \cap B = \emptyset$, A et B qui sont les attributs (ou ensembles d'attributs). Cette règle veut dire que lorsque ***A*** est présenté dans une transaction, ***B*** est susceptible d'être présent aussi. L'ensemble d'items *A* est appelé *Antécédent* ou *Condition* et l'ensemble d'items *B* est appelé *Conséquent* ou *Résultat*. Pour exprimer la qualité d'une règle d'association, on lui associe, deux pourcentages appelés respectivement le support et la confiance. Le support est le nombre de transactions dans lesquelles A et B apparaissent tous les deux, alors que la confiance est le rapport de *support(A,B)* sur le *support(A).*

Chacune de ces méthodes comporte plusieurs techniques et algorithmes, et chacune d'elles a, ses points forts et ses limites Ainsi chaque technique ou algorithme est appliquée à des types de données bien appropriées, certaines sont meilleures pour les données quantitatives et d'autres sont bien pour les données qualitatives.

Par conséquent, à tout jeu des données et à tout problème, correspond une ou plusieurs méthodes. Le choix des méthodes se fera en fonction de la tâche à résoudre, de la nature et de la disponibilité des données, des connaissances et des compétences disponibles, et de la finalité du modèle construit. Parmi les critères de choix des méthodes, on trouve : la complexité de la construction du modèle, complexité de son utilisation, ses performances, sa pérennité, et plus principalement, de l'environnement de l'entreprise.

I.4.3. Les taches du Data Mining

Les techniques de Data Mining, sont appliquées sur les bases de données pour but de construire des modèles d'une base de données afin de décrire le comportement actuel et de prédire les comportements futurs [MEHM 2001]. On peut classer ces modèles, comme des modèles de calculs ou des modèles logiques [MEHM 2001], ces modèles sont utilisés pour différentes tâches et pour des traitements différents, menant à la découverte de connaissances, ces tâches sont [VALE 2006] :

- ***la classification*** *: affecter une classe à chaque individu parfois utilisée pour la prédiction.*
- ***Le clustering ou classification non supervisée*** *: c'est une tâche descriptive qui permet d'identifier des groupes d'individus*
- ***La découverte de règles d'associations*** *: c'est une tâche descriptive pour la recherche des implications entre attributs.*
- ***La découverte de séquences*** *: similaire à la recherche des règles d'association avec insertion d'une notion de temps.*

- ***La détection de déviation ou la détection d'écart*** *: pour identifier les valeurs exceptionnelles.*
- ***La recherche des similitudes*** *: pour identifier des séquences communes entre instances. Très utilisée dans le domaine bioinformatique* **[SITE 01].**

II.4.4. Architecture d'un système de Data Mining

Pour présenter l'architecture du système de *Data Mining*, nous allons nous baser sur le point de vue de Han et Kamber) **[JIAW 2000],** qui est une vue plus large sur les fonctionnalités du Data Mining. Les auteurs considèrent ka fouille de données comme un processus de découverte des connaissances intéressantes à partir d'une large taille de données stockées que ce soit dans des bases de données, des entrepôts de données (*Data warehouse*), ou encore dans d'autres dépôts d'information. Les principaux composants de l'architecture d'un système de *Data Mining* sont schématisés par la figure (FIG.5) :

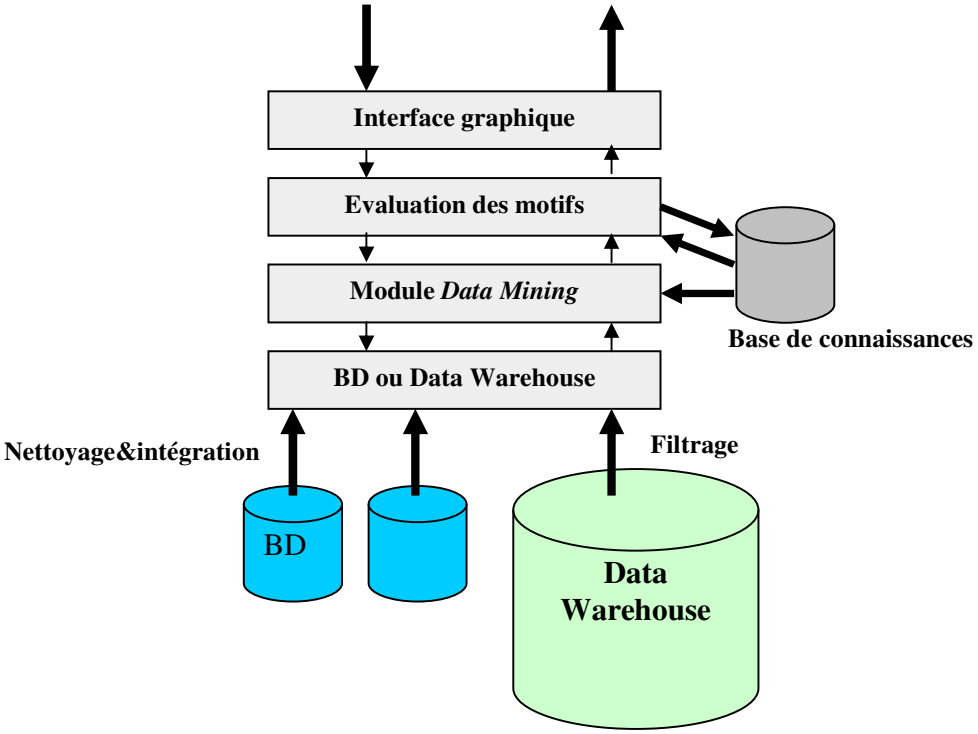


FIG.5: ARCHITECTURE D'UN SYSTEME DE DATA MINING *[JIAW 2000]*

1. ***Bases de données, Data warehouse, ou autre dépôt d'informations*** : des techniques de nettoyage et de prétraitement vont être appliquées à ces entrepôts de données.
2. ***Serveur de base de données ou de Data warehouse*** **:** permet de répondre aux demandes des utilisateurs pour la recherche des données appropriées à leurs requêtes.

3. ***Base de connaissances*** **:** c'est le domaine de connaissances utilisé pour guider la recherche ou l'évaluation des patterns trouvés. Une telle base peut contenir des concepts hiérarchiques permettant d'organiser les champs ou leurs valeurs au cours des différents niveaux d'abstraction.
4. ***Module Data Mining:*** il inclut les modules fonctionnels concernant les tâches de caractérisation, d'association, de classification automatique, etc.
5. ***Module d'évaluation des motifs*** : c'est un module qui est mis en interaction avec le module de *Data Mining* afin de focaliser la recherche des motifs intéressants. Une fois bien intégré dans le processus d'extraction, ce type de filtrage permet d'augmenter l'efficacité du système de *Data Mining* en affinant la recherche et se concentrant seulement sur les motifs intéressants.
6. ***Interface graphique avec l'utilisateur*** : c'est un module qui assure la communication entre les utilisateurs et le système de *Data Mining*. Il permet aux utilisateurs :
   - d'interagir avec le système en spécifiant les requêtes ou les tâches d'extraction.
   - de parcourir les schémas et les structures de la base de donnée du *Data warehouse*.
   - d'évaluer les motifs trouvés et les représenter sous différentes formes.

I.4.5. Domaine d'application

Le *Data Mining* est utilisé dans de nombreux domaines tel que :

- Le domaine des assurances pour analyser des risques par (exemple : caractérisation des clients à hauts risques), automatisation du traitement des demandes (exemple : le diagnostic des dégâts) et détermination automatique du montant des indemnités.
- Le domaine de finance pour faire des consentements, des prêts automatisés, support à la décision de crédit, détection des fraudes.
- Les grandes distributions pour faire les profils des consommateurs et les modèles d'achats, constitution des rayonnages, marketing (ciblé certaine clientèles).

La table (TAB.1) illustre quelque exemple d'application du Data Mining dans des secteurs d'activités différentes.

Secteurs d'activité	Exemples d'applications
Assurance	Analyse des clients à "haut risque", détection de fraudes
Industrie	Analyse des pannes, organisation des ateliers, contrôle qualité, prévision des ventes, gestion des stocks
Finances	Minimisation de risques financiers, suivis des clients, gestion de portefeuilles, marketing ciblé, attribution de crédit bancaire, détection de fraude…
Grande distribution VPC	Détermination du profil des consommateurs (habitudes d'achat, préférences par secteurs géographiques…), détection des produits à succès, constitution des rayons, marketing ciblé…
Télécommunication	Classification des clients, détection des fraudes, analyse des départs de clients, simulation de tarifs
Médecine, Laboratoires pharmaceutiques	Aide au diagnostic, prédiction de médication
Santé publique	Détection de facteurs de risque, analyse du génome, mise au point de médicaments
Internet	Profilage d'utilisateurs, amélioration des sites par la détermination des rubriques à fort impact, e-commerce, détection d'intrusion,

TAB. 1 LES PRINCIPALES APPLICATIONS DU DATA-MINING

I.5.Conclusion

L'ECD est un domaine qui a connu une émergence remarquable pendant ces dernières années, ce succès s'est réalisé sur le champ scientifique, et s'est prolongé au champ commercial. En effet, plusieurs éditeurs d'outils de Data Mining ont formé un marché riche de logiciels fiables, implantant pratiquement la totalité des méthodes existantes, et ciblant toutes les tâches de Data Mining.

Bien que l'ECD ait répondu à plusieurs questions, il reste un travail considérable à réaliser. Cet immense travail, couramment présenté dans la littérature, montre que l'ECD est un champ en cours d'émergence. Son caractère interactif nécessite l'intervention d'un humain et préférablement d'une personne ayant un minimum d'expertise dans son domaine d'application.

Il est évident que les techniques de Data Mining sont des processus complexe et compliqué, cette complexité se traduit au niveau de l'implantation, donc le Data Mining n'est pas une boîte noire qu'on utilise, ou un système totalement automatique avec lequel on gère les données pour obtenir des connaissances utiles.

Dans le prochain chapitre, nous exposons un état de l'art sur les règles d'association (RA).

Chapitre II

les règles d'association binaire

II.1. Introduction

L'étude des règles d'association entre attributs booléens est ancienne, elle est liée à l'analyse des tableaux croisés 2×2 [HAJEK 1999] L'une des premières méthodes de la recherche des règles d'association est la méthode ***GUHA*** (General Unary Hypotheses Automaton) initiée par Hajek, et al. en 1966 [HAJEK 1966], où apparaissent les notions de support et de confiance. D'autres recherches sont initiées par l'auteurs **Gras** en 1979 qui à définit des règles d'implication statistique pour aider les didacticiens à trouver des relations entre les acquisitions de notions élémentaires chez les élèves d'une classe. En 1986, *Guigues et Duquenne* [MART 2006] s'intéressaient plutôt à une représentation ordonnée de concepts avec les implications informatives. En 1993 *Agrawal et al***.** [AGRA 1993], ont privilégié l'extraction et l'optimisation des règles d'association dans les grandes bases de données.

L'émergence de la fouille de données, une étape qui est située au cœur du processus de découverte des connaissances à partir des données (ECD) [FAYY 1996], dont le but est de réaliser des modèles d'une base de données, afin de les utiliser pour différentes tâches menant à la prise des décisions, parmi lesquelles l'analyse des liens. Cette dernière est accomplie par plusieurs techniques, y compris les règles d'associations, devenue l'une des plus importantes applications dans le domaine du *Data Mining*. Appliquées pour première à des fins de marketing, les règles d'associations sont utilisées pour des explications comme :

- Quels sont les produits fréquemment vendus ou acheté ensemble ?
- Que mettons en vente ?
- Où mettons ce produit dans les rayonnages ?
- etc.

De nos jours elles sont introduites dans différents domaines comme outils d'aide à la prise de décision. Les types de données sont qualitative ou/et quantitative. C'est pour cette d'raison qu'il existe plusieurs types de règles d'associations, on trouve les règles d'associations classiques (dites parfois règles d'associations binaire) [AGRA 1993], les règles d'associations généraliser [SRIK1995], les règles d'associations quantitative appliquées aux données quantitative [SRIK 1996] et les règles d'associations flous appliquées aux données flous [KUOK 1998].

Dans ce chapitre, nous nous concentrons sur les règles d'association binaires, quantitatives et généralisées. Des définitions ainsi que certaines notions sont nécessaires afin de faciliter la compréhension de cette technique. En outre nous citons les algorithmes qui constituent la base de recherche de ce type de relation.

II.2. Les règles d'association classique (binaire)

L'explosion des volumes de données disponibles dans les domaines commerciaux, plus particulièrement dans le monde des grands magasins qui accueillent quotidiennement des centaines de clients pour faire leurs courses en achetant un ensemble d'articles et produits. Ces marchandises dont la plus pare sont étiquetées par un code-barres, qui est utilisé pour de nombreuses raisons, notamment pour connaître le prix dans l'opération d'achat nommée ici, *transaction* [AGRA 1993].

Les auteurs **Agrawal et al.** [AGRA 1993], analysèrent les transactions pour identifier les articles fréquents dans le panier de la ménagère, c'est-à-dire des groupes de produits fréquemment achetés ensemble, afin de découvrir des relations entre les produits vendus. Ces relations sont de la forme $(A \rightarrow B)$, dont le sens «*si les articles qui constituent A sont dans le panier, alors les articles qui constituent B sont aussi dedans* ». Ces relations jouent un rôle important dans la le processus de prise de décisions, étant donné que le même mécanisme peut être exploité dans la découverte des connaissances dans les bases de données en général [AGRA 1993].

II.2.1. Définition (item, itemset, itemset fréquent)

Dans les bases de données un **Item** est une paire (*attribut*, *valeur*), dont l'attribut à un nom, qui doit prendre des valeurs, ce dernier doit appartenir à un domaine bien défini.

Un ensemble d'items constitue un **Itemset** ; un **Itemset** est dit **fréquent** si et seulement si son support est supérieur ou égal à un seuil minimal, noté ***Minsupp***, fixé par l'utilisateur. Les itemsets n'ayant pas le support suffisant sont dites **non fréquents**.

Dans ce qui suit on utilise les notations suivantes :

- **D** : Ensemble des transactions, une transaction notée Ti (Ti est l'I^{ème} transaction)
- **Lk** : Ensemble des itemsets fréquents de taille k.
- **Ck** : Ensemble des itemsets candidats de taille k.
- **k-itemsets** : les itemsets de taille k.

II.2.2. Définition d'une règle d'association

Les règles d'associations aussi connues sous le nom de l'analyse du panier de la ménagère (*Market Basket Analysis*) et dont l'étymologie viennent des travaux qui ont été réalisés à partir des données provenant des supermarchés. L'objectif étant d'identifier les items ou les groupes d'items (Itemset), fréquemment achetés ensembles par un client [AGRA 1993].

D'une manière générale, une règle d'association est une expression de la forme :

$A \rightarrow B[Support\%, Confiance\%]$, Où $A \cap B = \emptyset$, A et B sont des attributs (ou ensembles d'attributs), cela veut dire que lorsque A est présent dans une transaction, B est susceptible d'être présent aussi. L'ensemble d'items A est appelé *Antécédent* ou *Condition* et l'ensemble d'items B est appelé *Conséquent* ou R*ésultat*.

Pour exprimer la qualité de la règle d'association, deux pourcentages sont associés, à la règle, appelés respectivement le ***support*** et la ***confiance*** ; dont le support représente le nombre de transactions dans lesquelles A et B apparaissent toute les deux en même temps dans la transaction, alors que la confiance est le rapport de *support (A, B)* sur le support de A .

Le support est une mesure dite d'utilité de la règle ; alors que la confiance est une mesure de précision de la règle. Enfin une règle d'association s'énonce comme suit : $A \rightarrow B[Support\%, Confiance\%]$. Ces mesures sont bien détaillées dans (***section II.5***).

II.2.3.Utilité des règles d'associations

Les règles d'associations sont exploitées pour la découverte des connaissances et sont aussi un outil efficace pour la prise de décision dans les bases de données en général. Elles sont appliquées dans plusieurs domaines. En marketing, par exemple, elles permettent d'identifier les produits ou articles qui sont achetés lors d'une même transaction ou par un même client et offrent donc la possibilité d'identifier des opportunités de ventes croisées. Ainsi on peut décider sur ce qui doit être mis en vente, et la façon de placer les marchandises sur les rayonnages, dans le but de maximiser les ventes.

Les règles d'associations sont aussi utilisées dans le monde de WEB (*Web Mining*), en analysant l'ordre dans lequel les utilisateurs accèdent aux pages d'un site WEB, afin d'apporter les modifications nécessaires pour rendre le site plus convivial, permettant ainsi aux internautes de trouver rapidement les informations recherchées [STEP 2005]. Elles sont également appliquées dans la fouille de texte (en anglais *Text Mining*) pour la recherche des mots clés dans un texte, par exemple [STEP 2005].

II.3. La nomenclature des règles d'association

Selon la nomenclature qui est été proposé par Lu et al. [LU 2000], les règles d'associations sont classées en quatre groupes (voir TAB.2). Cette classification est faite selon deux visions : la première faite selon les types de transactions, alors que la seconde est faite selon le nombre de dimensions de la règle d'association :

II.3.1. Selon le type de transaction

- *Une règle d'association est dite **intratransactionnelle** si elle cherche des relations à l'intérieur d'une même transaction, comme c'est le cas dans le panier d'achat.*
- *Une règle d'association est dite **intertransactionnelle** si elle cherche des relations d'occurrences entre des items répartis sur plusieurs transactions ; généralement ordonnées selon une hiérarchie temporelle.*

Ce dernier type de règles tente d'identifier les séquences d'items et par conséquent leur ordre d'apparition devient primordial.

En d'autres mots, les règles ***intratransactionnelles*** identifient quels sont les items ou événements qui occurrent simultanément, tandis que les règles ***intertransactionnelles*** identifient les items ou événements qui se succèdent fréquemment.

II.3.2. Selon la dimension

- *Une règle d'association est qualifiée **unidimensionnelle**, si elle comporte un seul Items dans l'**antécédent** et un autre dans **le conséquent***
- *Une règle d'association est qualifiée **multidimensionnelle** si elle comporte dans une partie soit **l'antécédent** ou **le conséquent** un ou plusieurs Items ou les deux en même temps*

La table (TAB.2) illustre cette classification avec des exemples :

Règle d'association	**Intratransactionnelle**	**Intertransactionnelle**
unidimensionnelle	***Définition***: Association entre des items de même dimension se retrouvant dans une même transaction ***Exemple***: imprimante → cartouche d'encre	***Définition***: Associations entre items de même dimension se retrouvant dans des transactions différentes, généralement ordonnées selon une hiérarchie temporelle. ***Exemple***: imprimante → cartouche d'encre Dans les 2 semaines qui suivent
multidimensionnelle	***Définition***: Association entre des items de même dimension et dimension distincte se retrouvant dans une même transaction ***Exemple***: Âge(20-35) ∧ salarie → achat (voiture)	***Définition:*** Associations entre items de même dimension et de dimension distincte se retrouvant dans des transactions différentes, généralement ordonnée selon une hiérarchie temporelle. ***Exemple:*** Âge (06-14) ∧malade(X) → prend Médicament(Y) Dans les 8 jours qui suivent

TAB.2 : NOMENCLATURE DES DIVERSES REGLES D'ASSOCIATION

II.4 Processus d'extraction des règles d'associations

Le processus d'extraction des règles d'association, en général, se déroule en quatre phases la figure (FIG.6) illustre l'enchaînement de ces phases.

1. ***Sélection et préparation des données :*** Les données utilisées par les algorithmes d'extraction de règles d'associations passent préalablement par deux étapes : la première permet de sélectionner des données à partir d'une base de données et dans certains cas en prenant en considération le choix de l'utilisateur, ainsi qu'une taille réduite des données traitées dans le but est d'assurer l'efficacité des algorithmes, alors que la deuxième sert à la transformation de ces données.

2. ***Découverte des Itemsets fréquents :*** Cette étape est la plus coûteuse en termes de temps d'exécution, car le nombre d'itemsets fréquents dépend exponentiellement du nombre d'items manipulés. Pour N items, on a 2^N itemsets potentiellement fréquents, (***voir en détails dans la section espace de recherche ou treilles des Itemset section II.4).***

3. ***Génération des règles d'association*** **:** Pour générer une règle d'association on choisisse un seuil minimal de support (***Minsup)*** *et* un seuil minimal de confiance (***Minconf)***. Seules les règles ayant un support et une confiance dépassant les seuils indiqués sont acceptées. Cette opération avec ces mesures est un problème qui dépend exponentiellement de la taille de l'ensemble des itemsets fréquents.

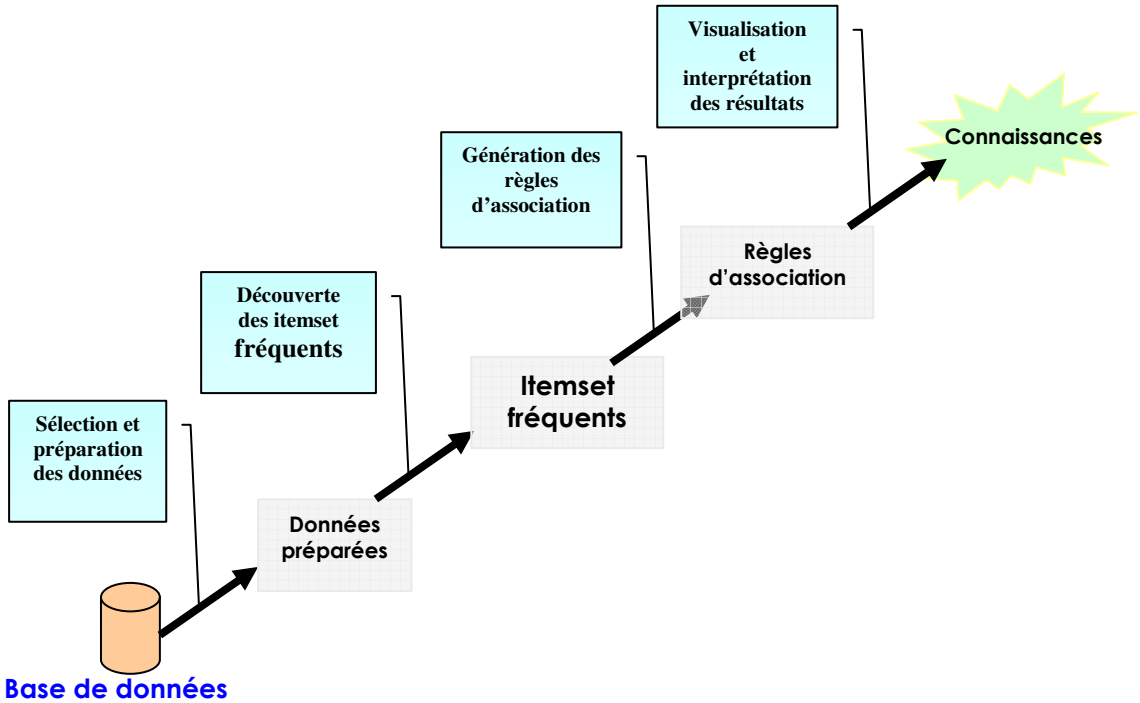


FIG. 6 LES ETAPES DE L'EXTRACTION DES REGLES D'ASSOCIATIONS

4. ***Visualisation et interprétation des règles d'association :*** C'est une étape de L'IHM (l'interface homme machine), la présentation des règles doit être exposé par des formulaires clairs et compréhensibles ainsi que par des graphiques et/ou sous forme de tableaux à l'aide des outils logiciels (comme ***WEEKA, SAS, ORANGE***, etc.). Il faut en considération la priorité des règles les unes par rapport aux autres, ainsi que les critères définis par les experts.

II.5. Espace de recherche ou le treillis des Itemsets

Dans la recherche des règles d'association les bases de données sont vues comme une matrice binaire, chaque ligne est un objet, alors que les colonnes représentent des propriétés binaires. Les colonnes de la matrice binaire sont aussi appelées items, et les lignes appelées transactions [THOM 2003]. Par exemple, dans le cas des opérations d'achats, une ligne est un passage à la caisse, et chaque colonne est un article.

La table (TAB.3) contient un ensemble des transactions des Items A, B, C, D. Le treillis des Itemsets contient « $2^{|items|}(Itemests)$ », on associe à chaque Itemset un indice pour indiquer son support. Le treillis comporte « $N = |Items| + 1$ » niveaux. Plus le nombre d'items est grand, plus l'espace de recherche est large, plus l'algorithme qui génère les Itemsets fréquent est défaillant [HIPP 2000].

Supposons que nous voulons extraire les Itemsets fréquents à partir de la base de données (voir TAB.3), en fixant le seuil minimum de support à 2. La figure (FIG.7) illustre le treillis des Itemsets. La ligne qui traverse le treillis est une frontière au dessus de lesquelles sont situées les Itemsets infréquents, et au dessous de laquelle sont situées les Itemsets fréquents. La majorité des algorithmes d'extraction de règles d'associations essaient d'avoisiner cette frontière pour élaguer le mieux possible l'espace de recherche inutile [HIPP 2000].

Idtran	Item A	Item B	Item C	Item D
01	1	1	0	0
02	1	1	1	0
03	1	0	1	1
04	1	1	0	1
05	0	1	0	1
06	0	0	1	1

TAB.3 L'ENSEMBLE DE TRANSACTION

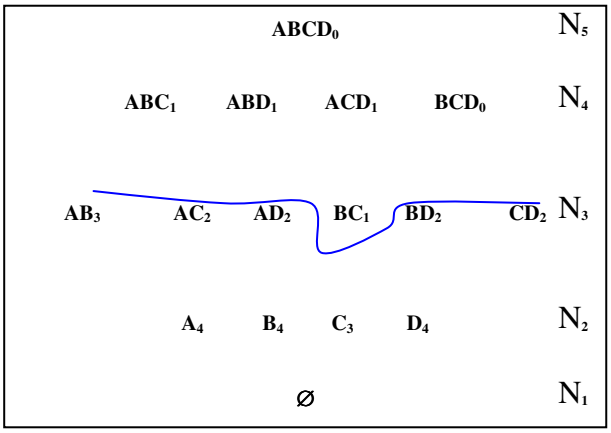


FIG. 7 TREILLIS DES ITEMS

Il y a deux façons de parcourir l'espace de recherche, en *largeur* ou en *profondeur*. Un algorithme de parcours en profondeur [HIPP 2000], considèrera d'abord les sous-ensembles ayant mêmes préfixes (par exemple {a}, {a, b}, {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. Ces deux ordres de parcours sont illustrés par la figure suivante (FIG.8).

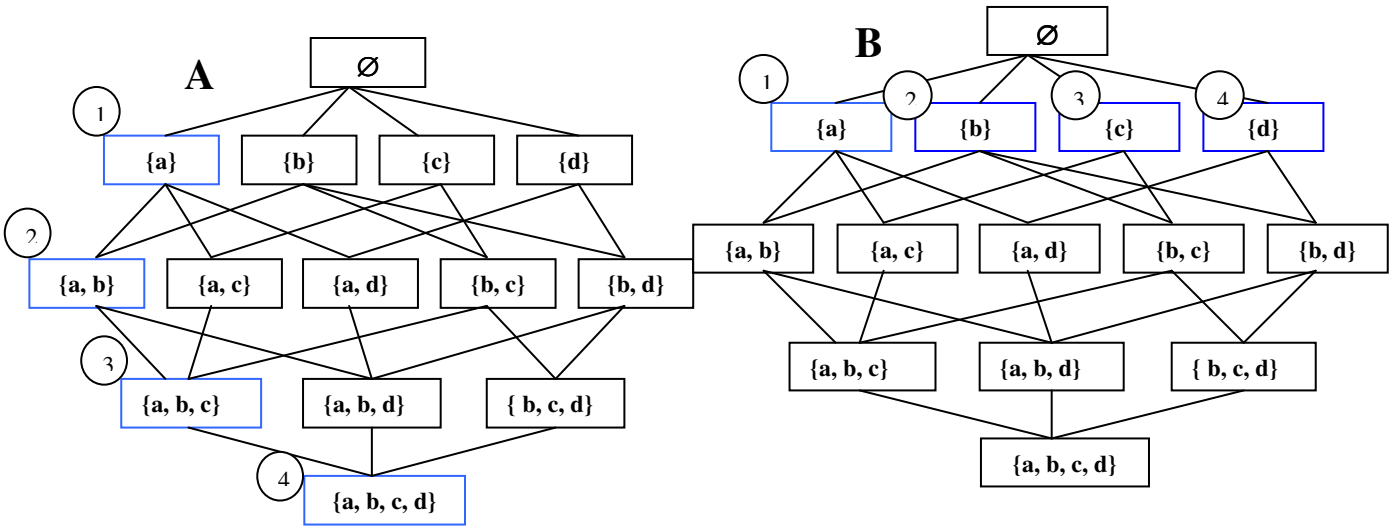


FIG. 8: ORDRE DE TRAITEMENT DES SOUS-ENSEMBLES DE L'ENSEMBLE {A,B,C,D} DANS UN ALGORITHME DE PARCOURS EN PROFONDEUR (A) ET EN LARGEUR (B). LES SOUS-ENSEMBLES SONT TRAITES DANS L'ORDRE 1, 2, 3, 4, ETC.

Dans le premier cas, les supports de tous les itemsets de **niveau (k-1)** du treillis doivent être calculés avant de générer les itemsets d'un **niveau k**. Le deuxième parcours consiste, en fait, à construire un arbre (une projection de la base de données) de manière récursive tout en scannant la base de données.

II.6. Mesure de qualité d'une règle d'association

L'objectif principal du processus ECD est d'extraire des connaissances non triviales et utiles, parmi lesquelles la production des règles d'association intéressantes.

Agrawal et al. Proposent deux mesures de qualité (***support, confiance***). Par définition une règle d'association est de la forme : $A \rightarrow B[Support\%, Confiance\%]$ [AGRA 1993].

Les quantités numériques (***support, confiance***) vont servir à valider l'intérêt d'une telle règle. Il existe deux visions pour exprimer le support et la confiance, ensembliste et probabiliste [PLAS 2005].

II.6.1. Support

Du point de vue ensembliste, le *support* d'une règle est la fréquence d'apparition simultanée des Items qui apparaissent dans la condition et dans le résultat dans la transaction. Le support est mesuré par le pourcentage de transactions présentant A et B :

$$Supp(A \rightarrow B) = \frac{m}{|T|} \quad \textbf{(1)}$$

$|T|$ *:* est le nombre total de transaction de la base de données.

m : le nombre de transaction où (A et B) apparaissent en même temps dans la même transaction

Du point de vue probabiliste, chaque sous-ensemble d'items se voit associé l'événement selon lequel la transaction contient les items de ce sous-ensemble. Le support s'exprime donc par la probabilité de réaliser simultanément les événements A et B :

$$Supp(A \rightarrow B) = P(A \cap B) = P(B / A) \times P(A) \quad \textbf{(2)}$$

Où A est l'événement « *la transaction contient tous les items de l'ensemble A* » et B est l'événement « *la transaction contient tous les items de l'ensemble B* ».

II.6.2. Confiance

Du point de vue ensembliste la *confiance* est le rapport entre le nombre de transactions où tous les articles figurant dans la règle sont apparaît dans la transaction notée (*m),* et le nombre de transactions où les articles de la partie condition apparaissent notée (c) :

$$conf = \frac{freq(condition \,\&\, résulta)}{freq(condition)} = \frac{m}{c} \quad \textbf{(3)}$$

Du point de vue probabiliste la confiance est égale à la probabilité de réalisation de l'événement *B* sachant que l'événement *A* est réalisé :

$$Conf(A \rightarrow B) = P(B / A) = \frac{P(A \cap B)}{P(A)} = \frac{Supp(A \rightarrow B)}{Supp(A \rightarrow B)} \quad \textbf{(4)}$$

Le support est une mesure pour indiquer la fiabilité de la règle d'association, ainsi il est utilisé pour éliminer les règles non intéressantes. C'est une propriété utilisée très souvent pour choisir les règles d'associations efficaces, dans un processus de découverte de ce type de relation ; alors que la confiance est une mesure qui sert à indiquer la précision de la règle.

II.6.3. Stratégie de Calcul du support

Plusieurs algorithmes d'extraction des Itemsets fréquents ont été élaborés, dont chacun utilise une stratégie de calcul de support déférente. Il existe, en fait, deux stratégies de calcul de support d'un Itemset, [HIPP 2000] :

- la première consiste à balayer la base de données afin de vérifier pour chaque transaction si l'Itemset est inclus. Le support de cet Itemset est le nombre de vérifications dont le résultat est positif. Cette stratégie coûteuse en termes de temps de calcul ou unités CPU.
- la deuxième stratégie est basée sur les identificateurs des transactions. Ces derniers sont notés par ***tidlist*** pour (***T**ransaction **Id**entifier **List***). Une ***tidlist*** est la liste des transactions dans lesquelles un Itemset est apparaît. Supposons qu'on a deux Itemset(I, J), X est l'union de I et J, et qu'on dispose de la liste des identificateurs des transactions contenant I ; et de même pour J. l'intersection de ces deux listes, est la liste des identificateurs des transactions contenant X. Le support de X est le cardinal de cette liste. Cette stratégie demande beaucoup d'espace mémoire [HIPP 2000].

II.6.4. Les mesures d'évaluation et de validation

La complexité des règles d'associations est en fonction du nombre plus ou moins élevé d'Itemset fréquents générés. Pour N items fréquents on peut avoir 2^N *Itemset fréquents*, ce qui conduit à produire un très grand nombre de règles. Il est clair qu'on a d'autres problèmes tel que le choix de meilleures règles. Parfois ce processus génère des règles d'association dite triviales ou inutiles. La première catégorie (triviales) représente les règles évidentes, c'est-à-dire celles qui sont déjà connu et qui n'apportent pas d'information de plus. Par exemple la règle : **SI** *achat d'une imprimante* **ALORS** *achat de cartouches d'encre*.

La deuxième catégorie de règles (inutiles) représente les règles difficilement interprétables. Tous ces problèmes ont conduits les chercheurs à proposer plusieurs critères pour mesurer la qualité des règles, visant à améliorer les performances du processus d'extraction des règles d'association [STEP 2004], et où on peut recenser un très grand nombre de mesures dans la littérature. *Benôıt Vaillant et a.l* [VAIL 2005], ont implémenté la plate-forme ***HERBS*** dans le laboratoire (ERIC - Université Lumière - Lyon 2) ils ont utilisé vingt mesures de qualité, basé sur les quatre cardinalités de la règle d'association ($A \rightarrow B$), ces cardinalités sont les fréquences d'apparitions de l'exemple et de contre exemple de la règle d'association, et on peut les exprimer par des probabilités de survenu des évènements (tables de contingence TAB.4(a), TAB.4(b)). Les fréquences sont schématisées par la figure (FIG.9).

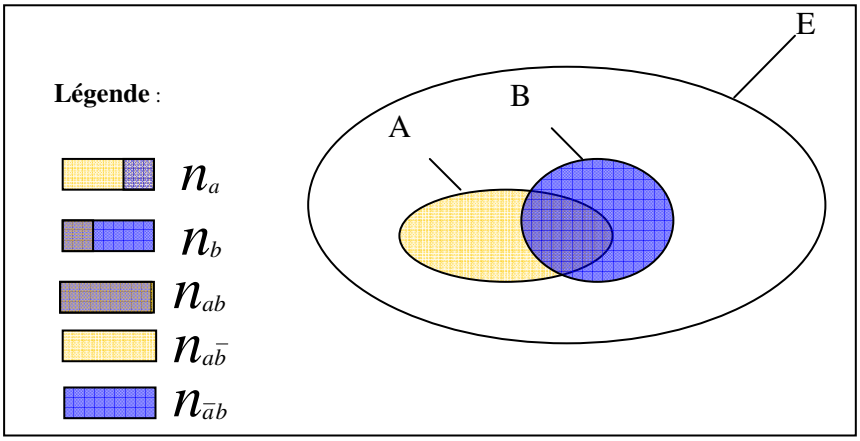


FIG. 9 LES FREQUENCE D'APPARITIONS DES ITEMSET A ET B

B / A	0	1	Total
0	$n_{\bar{a}\bar{b}}$	$n_{\bar{a}b}$	$n_{\bar{a}}$
1	$n_{a\bar{b}}$	n_{ab}	n_a
Total	$n_{\bar{b}}$	n_b	n

(a)

B / A	0	1	Total
0	$p_{\bar{a}\bar{b}}$	$p_{\bar{a}b}$	$p_{\bar{a}}$
1	$p_{a\bar{b}}$	p_{ab}	p_a
Total	$p_{\bar{b}}$	p_b	1

(b)

TAB.4 TABLES DE CONTINGENCE

Nous rappelons quelques mesures d'intérêts des règles d'association, et pour des détails consulte**,** exactement en présente les mesure (*Confiance centrée*, *Rappel*, *lift*, *pearl*, *Piatetsky-Shapiro* [VAIL 2005]).

II.6.4.1 Confiance centrée

La *confiance centrée* est calculée par $(\text{confiancecentrée}(A \to B) = P(B/A) - P(B))$ [LALL 2004]. Cette mesure permet de prendre en considération la taille de partie droite de la règle d'association ($A \to B$). Elle permet donc de relativiser la confiance d'une règle par rapport à la taille de sa conclusion.

II.6.4.2 Rappel

Le *rappel* est une mesure aussi connue sous le nom de *sensibilité*, calculée par : $rappel(A \to B) = P(A/B)$ [LAVR 1999]. Elle permet d'évaluer la proportion d'objets vérifiant la prémisse de la règle parmi ceux vérifiant la conclusion.

II.6.4.3 Lift

Le *lift* représente le rapport entre l'indice brute, $P(AB)$ de la règle par celle de l'indice de l'indépendance entre les parties qui constituant la règle d'association qui est égale au produit des deux probabilités $P(A)\times P(B)$. Le lift permet donc d'apprécier simplement, pour une règle ($A \to B$), sa «distance » à l'indépendance. Il est calculé par ($lift(A \to B) = P(AB) / P(A)\times P(B)$). Si le lift d'une telle règle est inférieur ou égale 1, la règle n'est pas intéressante. Par exemple, une règle ($A \to B$) 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 ($A \to B$) et ($B \to A$) [PAUL 2008].

II.6.4.4. Pearl

La mesure de *Pearl* [LALL 2004],($Pearl(A \to B) = P(A)\times\left|P(B/A) - P(B)\right|$) permet d'évaluer l'intérêt d'une règle $A \to B$ par rapport à l'hypothèse d'indépendance entre la prémisse et la conclusion de la règle. L'utilisation de cette mesure permet de vérifier que la règle apporte vraiment une connaissance nouvelle sur les données. En effet, si la mesure de Pearl de la règle est proche de 0, cela nous indique que les Items (*A* et *B) sont* liés par une relation intéressante.

II.6.4.5. Piatetsky-Shapiro

L'indice de *Piatetsky-Shapiro* formellement est donné par : $PS(A \to B) = n\times P(A)\times(P(B/A) - P(B))$. Cet indice est très proche de l'indice de *Pearl,* puisqu'il s'exprime de la manière suivante : $PS(A \to) = n\times Pearl(A \to B)$. La présence de la quantité *n* dans l'expression de cet indice rend celui ci moins sensible à la taille des données, contrairement à l'indice de Pearl. Donc l'indice de *Piatetsky-Shapiro* [PIAT 1991] évalue l'intérêt d'une règle par rapport à son écart à l'indépendance.

II.7. Inductions des règles d'association

L'extraction des règles d'association occupe une place importante dans le processus de la découverte des connaissances à partir d'une base de données (ECD), dont le but est de découvrir les relations significatives entre les attributs caractérisant les objets, pour cela les auteurs [AGRA 1993] [AGRA 1994][SAVA 1995][ZAKI 1997][PASQ 1999][PEI 2000][HAN 2000], ont proposé plusieurs algorithmes d'extractions des règles d'association parmi eux :

II.7.1. Algorithme C5.0

Sont principe est transformer un arbre de décision sous la forme de règles ***Si ... Alors ...*** donc à partir de la structure de l'arbre on va déduire des règles d'association, cette forme de modèle est préférée car les arbres de décision sont lisibles et compréhensibles [MEHM 2001].

II.7.2. Le GRI (Generalised Rule Induction)

Le GRI (*Generalised Rule Induction*) est un autre outil d'induction de règles qui engendre un ensemble de règles indépendantes, contrairement aux ensembles des règles générés par C5.0 qu'ils sont disjonctifs (du fait qu'ils s'appuient sur un arbre de décision). Chaque règle comporte une partie **« conditions »,** portant sur les champs disponibles et une partie **« conclusion »**, portant sur différents champs que l'utilisateur aura choisis au préalable. De plus, chaque règle se voit affecter un taux de couverture et une précision [MEHM 2001].

II.7.3 Algorithmes fondé sur le support et la confiance

Les algorithmes d'extractions des règles d'association fondées sur l'approche *support-confiance* procéder en deux phases [AGRA 1993] :

- ***Phase1*** : consiste à l'extraction des Itemsets fréquents.
- ***Phase2*** : la génération des règles d'association à partir des Itemsets généré en phase1.

La phase 2 est la plus simple que la première phase, car l'extraction des Itemsets fréquents se faits dans un espace de recherche de taille exponentiels, (voir ***section II.4***). Donc nous concentrerons notre étude sur la partie d'extraction des Itemsets fréquents. Ces algorithmes sont classés en quatre classes, selon les stratégies de parcours de treillis des Itemsets, et la manière de calcul de support, [HIPP 2000]. La table (TAB.5), illustre cette classification.

Parcours en largeur		**Parcours en profondeur**	
Parcours de la base	Intersection des *tidlists*	Parcours de la base	Intersection des *tidlists*
AIS et Apriori	***Partition***	***FP-Growth***	***Eclat***

TAB. 5 LA CLASSIFICATION DE [HIPP 2000]

II.7.3.1. L'algorithme AIS

AIS est l'abréviation des Auteurs (Agrawal, Imielinski, Swami) [AGRA 1993]. Cet algorithme génère les règle d'association de la forme $A \rightarrow B$ où A est un seul Item, et B est un

Itemset. L'algorithme AIS parcourir la base de donnée plusieurs fois afin de calculer ce qu'on appel *k-Itemset* fréquent. Pendant le premier parcours, on extrait les *1-itemsets* fréquents c'est-à-dire l'extraction des Items fréquents, qu'il en stock dans un ensemble noté $F1$. Pour extraire les *2-itemsets* fréquents, l'algorithme parcourt la base de données pour tester les supports des 2-itemsets appartenant à l'ensemble des *2-itemsets* candidats noté C_2. Cet ensemble est généré à partir de l'ensemble $F1$, (ensemble des 1-itemsets fréquents). Les 2-itemsets candidats sont générés en étendant les 1-itemsets fréquents par un item de la même transaction. *A* La $k^{ème}$ passe, l'algorithme parcourt deux fois la base de données pour générer les *k-itemsets* fréquents : la première fois sert à générer les *k-itemsets* candidats ; alors que la deuxième sert à calculer le support de chaque candidat. Uniquement les candidats ayant un support supérieur au seuil minimum sont insérés dans l'ensemble Fk.

II.7.3.2. L'algorithme Apriori

L'algorithme *Apriori* et paru en 1994 [AGRA 1994], cet algorithme est un fondement essentiel pour l'extraction des règles d'association. *Apriori* est plus efficace qu'AIS au niveau de la génération des *Itemsets* candidats, parce qu'il élague encore plus l'espace de recherche. Ceci résulte du théorème de l'anti-monotonicité. En effet, *Agrawal* et *Srikant* [AGRA 1994], ont montré, que si un Itemset est fréquent alors tous les sous-ensembles de l'Itemset sont aussi fréquents, et inversement. Le nom de l'algorithme vient du fait qu'il tient en compte de la connaissance antérieure des itemsets fréquents.

La génération des itemsets fréquents se fait sur deux étapes **jointure et élagage.** La première étape consiste à génères les Itemsets candidats. Cette opération exige plusieurs parcours de la base de données par l'algorithme. Le premier parcours sert à identifier les candidats Ck (un ensemble d'itemsets et leurs supports respectifs). Seuls les items fréquents, sont conservés afin de former Lk (l'ensemble des *k-itemsets* fréquents). Cet ensemble aboutit à la génération de l'ensemble de candidats C_{k+1}

L'ensemble C_{k+1}, regroupe les *(k+1)-itemsets*, est généré en liant Lk avec lui-même. Pour que deux *k-itemsets* puissent être liés, ils doivent posséder k-1 items en commun.

La deuxième étape est **l'élagage,** où tous les (k+1)-*itemsets* appartiennes à C_{k+1} dont le support ne dépasse pas le *Minsupp* sont supprimés de la liste des candidats. Tout candidat retiré à l'étape k n'est plus considéré dans l'étape k+1 grâce au théorème de l'anti-monotonicité. On observe que la génération des candidats est un processus complexe, et le calcul des supports des itemsets candidats nécessite plusieurs parcours de la base de données.

II.7.3.3. L'algorithme Partition

Partition est développé par les auteurs *Savasere*, A.,*Omiecinski*, E., *Navathe*, S [SAVA 1995]. Cet algorithme s'inspire de l'algorithme *Apriori*, mais utilise une stratégie différente pour calculer les supports des itemsets candidats. En effet, Partition mémorise avec chaque itemset fréquent A la liste de toutes les transactions qui le supportent, il stocke juste l'identifiant de la transaction qu'on a appellera ***tidlist (voir section II.6.3)***. La génération des itemsets candidats est identique à celle d'*Apriori*, dont le support de chaque *Itemset* candidat est calculé via l'intersection des ***tidlists*** de tous ses items.

Il faut noter que *Partition* divise la base de données de manière à ce que la taille de chaque tranche coïncide avec la taille de la mémoire physique.

Ceci dit, il est possible que les ***tidlists*** des *Itemsets* localement fréquents ne pourraient pas être toutes stockées dans la mémoire centrale. Un deuxième problème peut mener à ce que l'ensemble global de candidats soit volumineux, si jamais la base de données ne serait pas homogène, et que le nombre des *Itemsets* localement fréquents soit énorme. Cependant, si la taille de la base de données coïncide avec la taille de la mémoire physique, *Partition* serait beaucoup plus performant et efficient qu'*Apriori* [GOET 03].

II.7.3.4. L'algorithme Eclat

Eclat [ZAKI 1997] est un algorithme qui se base sur la notion de « classe d'équivalence » pour chercher les *Itemsets* fréquents en profondeur, et sur le format vertical de la base de données pour calculer les supports de ces *Itemsets*. Ce calcul s'effectue grâce aux intersections des ***tidlists***. Ceci réduit la taille de la base de données chargée dans la mémoire physique, car seules les transactions concernant un *Itemset* sont chargées et utilisées pour l'intersection.

Deux *k-itemsets* appartiennent à la même classe d'équivalence s'ils ont en commun un préfixe de taille k-1, par exemple les 4-itemsets ABCD et ABCE appartiennent à la même classe d'équivalence. *Eclat* traite chaque classe d'équivalence séparément en mémoire, ce qui permet de décomposer le treillis des *Itemsets* en sous-treillis, où chaque sous-treillis représente une classe d'équivalence.

Eclat ne connaît pas tous les *Itemsets* fréquents à un niveau donné (puisqu'il parcourt le treillis des *Itemsets* en profondeur) avant de considérer les candidats du niveau suivant. Il en découle que cet algorithme génère plus de candidats qu'il en faut car il n'applique pas la propriété d'anti-monotonicité.

En comparaison avec *Partition*, la taille des transactions gardées en mémoire est en moyenne inférieure. En effet pour Partition, tous les *Itemsets* de tailles k ainsi que leurs

transactions sont stockés en mémoire centrale. Alors que pour Eclat, à une certaine profondeur d, tous les *k-itemsets* qui ont la même classe d'équivalence (avec $k \leq d$) sont stockés en mémoire.

II.7.3.5. L'algorithme FP-Growth (Frequent-Pattern Growth)

FP-Growth [HAN 2000] est un algorithme d'extraction de patterns fréquents utilisé pour générer des règles d'association. En effet, il réalise la tâche après deux parcours de la base de données, en s'appuyant sur une structure d'arbre, ce processus se déroule en deux grandes étapes :

1) la construction de l'arbre appelée FP-Tree (*Frequent-Pattern-Tree*) qui est une structure de donnée compacte de la base de données).
2) la génération des patterns fréquents à partir de cette structure.

L'algorithme correspondant est le suivant :

Algorithme: FP-Growth

Input : L'arbre FP-Tree qu'on notera T, Seuil minimum de support ms

Output : Ensemble des itemsets Fréquents : F(D,ms)

Algorithme :

01 FP-Growth (T, α)
02 **Si** T contient un seul chemin **Alors**
03 **Pour** chaque combinaison β des noeuds du chemin faire
04 Générer le pattern β U α avec support=support minimum des noeuds dans β
05 **FinPour**
06 **Sinon**
07 **Pour** chaque ai de l'index de T faire
08 Générer le pattern β = ai U α avec support = ai.support
09 Construire le pattern de base de β
10 Construire le FP-Tree conditionnel de β et l'affecter à T
11 **FinPour**
12 **Si** T ≠ Ø Alors
13 FP-Growth (T,β)
14 **FinSi**

La racine de l'arbre ne porte aucune information ('null'), alors que les autres noeuds portent deux informations, **L'Item** qu'il représente et son **support**. Un index est associé à l'arbre FP-Tree contient tous les Items fréquents. A chaque Item est associé un pointeur vers le premier noeud de l'arbre qui le contient.

La première étape est consacrée à la construction de l'arbre FP-Tree, elle se déroule comme suit :

1- La base de données est parcourue pour calculer le support des Items. Les Items fréquents sont générés dans un ensemble qu'on notera **F1**.

2- La base de données est parcourue une deuxième fois, et chaque transaction (ensemble d'items) est triée selon le support des items de manière décroissante. Chaque transaction **ti** est représentée par sa tête **p** (p est l'item le plus fréquent) et par la queue **Q** (ensemble trié des items moins fréquents que **p**). Pendant le même parcours, pour chaque transaction, une fonction Insert(p,Q,T) procède ainsi : Si la racine **T** a un noeud fils **N** tel que **N.Item=t.p** alors son support est incrémenté, sinon un noeud fils **N** est créé avec un support égal à 1. Ensuite, on affecte récursivement à **p** le premier Item de **Q**, et à **Q** le reste des Items, on vérifie s'il existe un noeud **N'** fils de **N** tel que **N'.Item=t.p,** si oui on incrémente son support, sinon on créé un nouveau noeud **N'** (fils de N) avec un support égal à 1. Cette fonction est exécutée récursivement jusqu'à ce que **Q** soit vide.

Il faut noter que l'objectif de tri des items (dans l'ordre décroissant de leurs fréquences) est de réduire la taille de l'arbre, car ainsi les items les plus fréquents seraient partagés par les transactions.

La deuxième étape consiste à générer les patterns fréquents à partir de l'arbre FP-Tree.

II.8. L'algorithme *Apriori* d'Agrawal et Srikant

II.8.1 L'architecture de l'algorithme

En 1993 *Agrawal et al.*, publient un premier article traitant les bases des règles d'association [AGRA 1993]. Deux articles parus en 1994 et 1996 [AGRA 1994], [AGRA 1996] qui ont présenté L'algorithme *Apriori*, cet algorithme découvre efficacement les règles d'association, il procède de manière itérative pour identifier les Itemsets fréquents, ce processus se déroule en deux étapes :

- ***Etape 1*** : calcul des 1-itemsets fréquents c'est-à-dire sous-ensembles fréquents de I (I c'est l'ensemble des Items) comportant un Item;
- ***Etape k :*** calcul des k-itemsets fréquents à partir des (k-1)-itemsets fréquents c'est-à-dire les sous-ensembles fréquents de I comportant k Items.

L'architecture de l'algorithme est illustrée dans le schéma suivant (FIG.10):

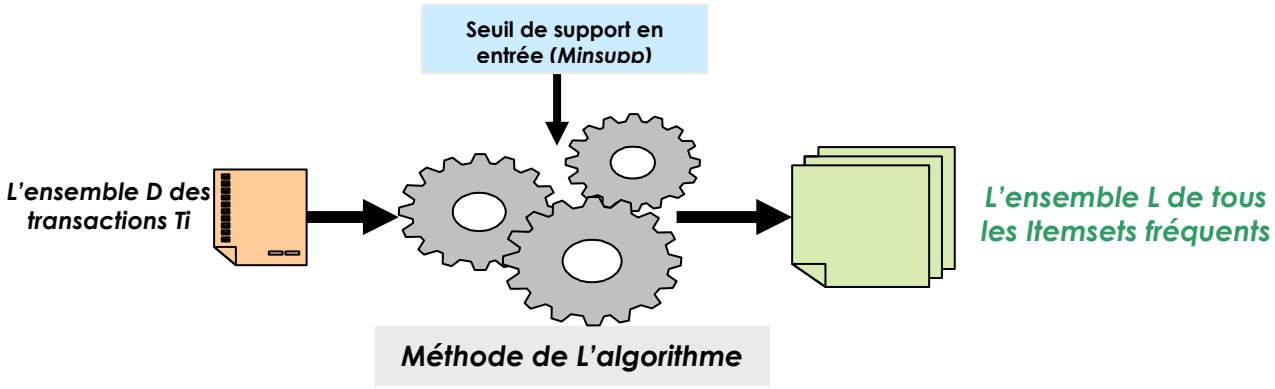


FIG. 10 L'ARCHITECTURE DE L'ALGORITHME *Apriori*

Avant de présenter l'algorithme ***Apriori*** il est nécessaire de donner quelques propriétés.

II.8.2. Quelque Propriété

II.8.2.1.Propriété sur les ensembles fréquents

- ***Propriété 1 :*** Support pour les sous-ensembles

  Si A $\subseteq$ B pour les itemsets A, B alors *supp*(A) >= *supp*(B) car toutes les transactions dans D qui supportent B supportent aussi nécessairement A.
- ***Propriété 2 :*** Les sous-ensembles d'ensembles fréquents sont fréquents.
- ***Propriété 3 :*** Les sous-ensembles d'ensembles non fréquents sont non fréquents

II.8.2.2. Propriété sur les règles d'association

- ***Propriété 4*** : Pas de composition des règles

Si X $\rightarrow$Z et Y $\rightarrow$Z sont vrais dans D, X $\cup$ Y $\rightarrow$Z n'est pas nécessairement vrai.

- ***Propriété* 5** : Décomposition des règles (FIG.11)

Si X $\cup$ Y $\rightarrow$Z convient, X $\rightarrow$Z et Y $\rightarrow$Z peut ne pas être vrai.

- ***Propriété* 6** : Pas de transitivité

Si X $\rightarrow$Y et Y $\rightarrow$Z, nous ne pouvons pas en déduire que X $\rightarrow$Z.

- ***Propriété* 7** : Déduire si une règle convient

Si A $\rightarrow$ (L-A) ne vérifie pas la confiance alors nous n'avons pas

B $\rightarrow$ (L-B) pour les itemsets L, A, B et B $\subseteq$A.

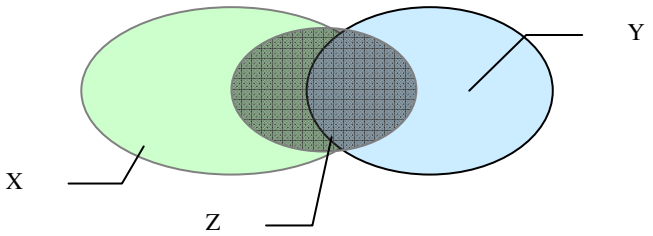


FIG.11 DECOMPOSITION D'UNE REGLE D'ASSOCIATION

II.8.3. Pseudo-code de l'algorithme APRIORI

L'algorithme reçoit l'ensemble **D** des transactions et un seuil minimum ***Minsup*** de support donné par l'utilisateur et à la fin du processus, on obtient L'ensemble **L** de tous les itemsets fréquents [AGRA 1994].

Algorithme APRIORI

Input **: D :** l'ensemble des transactions

Minsup : seuil minimum de support

Output **:**

L : les Itemsets fréquents.

Algorithme

1 L1 = {1-itemsets fréquents)
2 k=2 ;
3 ***Tant que*** L_{k-1} non vide ***faire***
4 C_k=Apriori-Gen (L_k) ;
5 ***Pour chaque*** t de B ***faire***
6 C_t = **Subset** (C_k, t) ; {les candidats contenus dans C_k}
7 ***Pour chaque*** c de C_t ***faire***
8 c.count++;
9 ***Fin pour***
10 ***Fin pour***
11 L_k = {c de C_t / c.count >= *minsup*};
12 k++ ;
13 ***Fin du tant que***
14 Return UL_k ;

Pour que l'algorithme découvre les Itemsets fréquents, il parcourt la base de données plusieurs fois. Dès la première passe, l'algorithme génère l'ensemble items fréquent noté par L_1. Pour générer les Itemsets fréquents à partir de k égal deux « les $k^{ième}$ passe où $k \geq 2$ », le processus se déroule en trois étapes:

- ***Etape1*** : Les *(k-1)-itemsets* fréquents trouvés lors de la (k-1) ^{ième} passe sont utilisés pour générer les itemsets candidats de taille k (algorithme **APRIORI-GEN**).

- ***Etape2*** : À chaque fois, la base de données est parcourue pour compter la fréquence des candidats par l'utilisation de fonction **Subset** qui fournit l'ensemble des itemsets C_k contenus dans une transaction.
- ***Etape3*** : À la fin, on sélectionne les **itemsets** qui ont une fréquence supérieure ou égale au **seuil *minsup*.**

ALG. APRIORI-GEN

Etape 1 :

Insert into C_k

Select $p.item_1$, $p.item_2$, ….., $p.item_{k-1}$, $q.item_{k-1}$

from L_{k-1} p, L_{k-1} q

where $p.item_1=q.item_1$ and $p.item_2=q.item_2$, …..,and $p.item_{k-2}=q.item_{k-2}$ and $p.item_{k-1}<q.item_{k-1}$

Etape 2:

Pour chaque **itemset** c de C_k ***faire***

Pour chaque s = **Subset** (k-1) de c ***faire***

Si $s \notin L_{k-1}$ ***alors***

Supprime c de C_k

Fin pour

Fin pour

Return C_k

Cet algorithme construit à partir de l'ensemble des *(k-1)-itemsets* fréquents L_{k-1} , l'ensemble des *k-itemsets* candidats C_k . Il se déroule en deux étapes :

- ***Etape1* :** cette étape connue par le nom « ***jointure*** », dans cette étape l'algorithme fusionne deux (k-1)-itemsets P et Q qui partagent leur *(k-2)-premiers* items.
- ***Etape2*** : cette étape connue par le nom « ***élagage*** », dans cette étape l'algorithme supprime de C_k tout *Itemset* c pour lequel au moins un sous-ensemble de longueur (*k-1)* de c n'appartient pas à L_{k-1}

II.8.4. Génération de règles d'associations

L'algorithme de génération des règles d'association a comme entrée un ensemble d'itemset fréquent et à la fin du déroulement du processus, un ensemble de règles d'association noté **(ER)** sera généré. Les règles qui, ayant une confiance supérieure ou égale à un seuil ***minconf*** seront acceptées alors que les règles où leur confiance ne satisfait pas le seuil ***minconf*** seront rejetées par l'algorithme.

Supposons qu'un itemset **X** qui se compose de deux Itemsets ***A*** et ***B*** illustre par la figure (FIG12. (a)), on renvoie une règle de la forme $A \rightarrow B$, si la confiance, c'est-à-dire le rapport ($support\ (A \cup B)\ /support\ (A)$) vaut au moins un seuil minimal de confiance ***minconf***. Pour la généralisation, nous donnons la **définition ci-après :**

Définition : Soit un ensemble $f_i \in F$ d'itemsets fréquents pour un seuil minimal de support ***minsup***, étant donné un seuil minimal de confiance ***minconf***, l'ensemble ***ER*** des règles d'association valides est donné par :

$$ER=\{r: I_j \rightarrow f_i - I_j / (I_j \in f_i) \wedge (F - I_j \cap I_j = \varnothing) \wedge (support\ (f_i)/support\ (I_j) >= minconf)\}$$

Il est claire que le support (I_j)>= support (f_i) (**propriété 1**). Cette propriété permet de minimiser le nombre de tests de l'algorithme ; par exemple soit F = {A, B, C, D} un **Itemset** fréquent et supposons que Ii = {A,C} donc, on a la règle d'association suivante:
(A,C)→ (B,D) et supposons que cette règle n'est pas valide. Donc tout itemset fréquent inclus dans {A, C} aura une confiance inférieure ou égale à la confiance de (A, C). On peut alors déduire directement que les règles : A→(B,C ,D) et C→(A, B ,D) ne sont pas valides non plus. Alors nous évitons de calculer les confiances de ces deux dernières règles. Réciproquement, si la règle (B, D)→(A ,C) est valide, alors les règles : (A, B, D)→C et (B,C,D)→ A sont également valides.

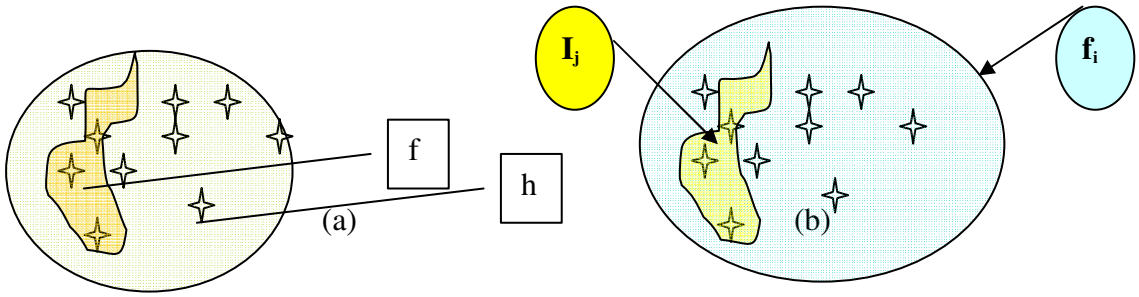


FIG.12 ITEMSETS QUI COMPOSE LA REGLE D'ASSOCIATION

Algorithme d'extraction de règles d'association

***Algorithme gen-rules* ;**

1 ***Pour*** chaque Itemset fréquent f_i ***faire***

2 Générés tous les sous-itemsets non vides I_j de f_i

3 ***Finpour***

4 ***Pour*** chaque sous-itemsets non vides I_j de f_i ***faire***

5 Produire la règle $(I_j \rightarrow (f_i - I_j))$ si $\frac{Supp(f_i)}{Supp(I_j)} \geq \textbf{\textit{Minconf}}$

6 ***Finpour***

II.8.5. Exemple d'utilisation de l'algorithme *Apriori*

II.8.5.1 Itemsets fréquent par Apriori

Afin de valider l'algorithme, nous allons l'appliquer sur une table de transactions. La table (TAB. 6) représente un ensemble de quatre transactions contenant cinq Items respectivement (A, B, C, D, E).

Ti	Items				
100	**A**	**B**	**C**	**D**	-
200	-	**B**	**C**	-	**E**
300	**A**	**B**	**C**	-	**E**
400	-	**B**	-	-	**E**

TAB. 6 ENSEMBLE DE TRANSACTION

Nous fixons le seuil minimum de support à 2. (*Minsup* =2) c'est-à-dire 50%. Le déroulement des étapes de l'algorithme est schématisé par la figure (FIG.13) comme ci-après :
La première étape L_1 généré les itemset fréquent on calcule les supports des items pour K =1

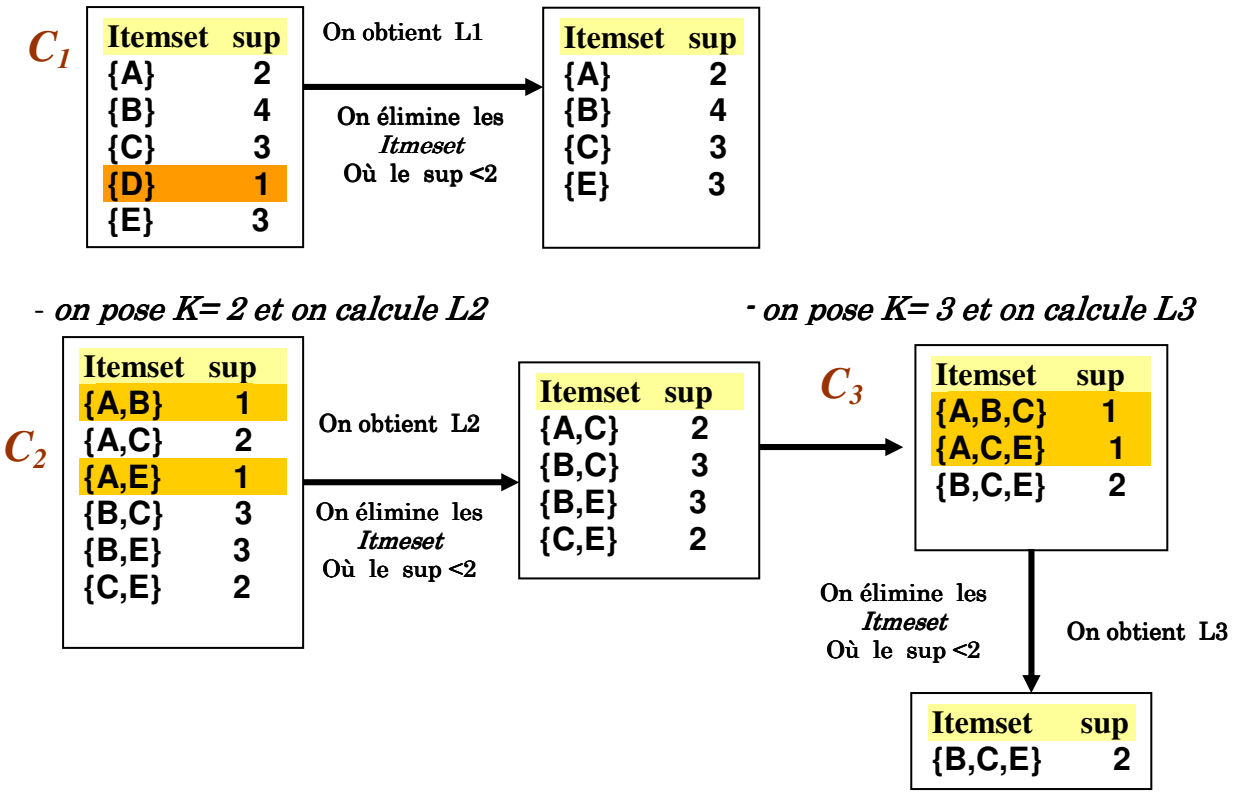


FIG.13 LES DEFFIRENTS ITEMSETS FREQUENTS GENERE PAR *APRIORI*

II.8.5.2 règles associative générés

Pour la génération des règles d'association, nous fixions le seuil minimum de la confiance par exemple à 50%, les règles extraites par Apriori sont décrites par la table (TAB.7)

règles	Support	Confiance	règles	Support	Confiance	règles	Support	Confiance
A→C	50%	100%	E→C	50%	66.67%	C,E→B	50%	100%
C→A	50%	66.67%	C→E	50%	66.67%	E→C,B	50%	66.67%
B→C	75%	75%	B→C,E	50%	50%			
C→B	75%	100%	B,C→E	50%	100%			
B→E	75%	75%	B,E→C	50%	66.67%			
E→B	75%	100%	C→B,E	50%	66.67%			

TAB.7 REGLES GENEREES PAR *APRIORI*

Par exemple en calculant le *lift* des règles générées décrites en tables (TAB.7) tables, afin de choisir des règles intéressantes. Dans cette exemple toutes les règles d'association ayant un *lift* ≤ 1 signifie que la règles n'est pas intéressante

règles	Confiance	Lift	règles	Confiance	Lift	règles	Confiance	Lift
A→C	**100%**	**2**	E→C	66.67%	0,88	C,E→B	100%	1
C→A	**66.67%**	**2**	C→E	66.67%	0,88	E→C,B	66.67%	0,88
B→C	75%	1	B→C,E	50%	1			
C→B	75%	1	B,C→E	100%	0,88			
B→E	75%	1	B,E→C	66.67%	0,88			
E→B	100%	1	C→B,E	66.67%	0,88			

TAB.8 LES REGLES INTERESSANTES

II.9. problématique de l'algorithme Apriori

L'objectif visé dans la recherche des relations entre attributs (règles d'association), est de générer toutes les règles d'association intéressantes c'est-à-dire les règles ayant un support et une confiance supérieure ou égal à des seuils minimaux (***Minsupp, Minconf***) respectivement fixé par l'utilisateur, aussi il faut donner une grande importance à l'étape de la représentation des règles d'association (affichage sur écran, impression. etc.), afin de faciliter la tâche aux utilisateurs pour qu'ils spécifient les règles qui conviennent.

Les auteurs ont proposé plusieurs algorithmes [AGRA 1993] [AGRA 1994][SAVA 1995] [ZAKI 1997][PASQ 1999][PEI 2000][HAN 2000], mais nous nous sommes concentré, essentiellement, sur l'algorithme ***Apriori*** [AGRA 1994], parce qu'il est le fondement des autres algorithmes.

Apriori reçoit en entrée une table des items qui contient un grand nombre d'enregistrements, dont le but est de générer touts les Itemsets fréquents qui dépassent un seuil (***Minsup).*** À partir de ces ensembles; l'algorithme génère les règles d'association qui ayant une confiance supérieure ou égale le seuil (***Minconf)***.

Généralement les algorithmes fondés sur ***Apriori*** *ont une complexité* exponentielle, car à chaque itération on obtient un nombre énorme de candidats. Par exemple, pour découvrir un Itemset fréquent de taille N, l'algorithme génère au total plus que 2^N candidats, ce qui nécessite d'utiliser un matériel puissant, dotés un grand espace de stockage et d'un processeur puissant. Cependant ***Apriori*** a deux inconvénients :

- Le grand nombre de candidats générés à chaque itération. Par exemple, pour un motif de longueur n on $(2^n - 2)$ règles d'association [MART 2006]
- L'algorithme nécessite répétitivement de balayer la base de données lors de chaque calcul de support.

II.10. Evaluation d'une règle d'association

En extraction de connaissance à partir de données, il est évident qu'une règle d'association doit être utilisée à la fin du processus d'extraction des règles dans un but décisionnel ou organisationnel, ce qui nécessite d'évaluer la qualité de cette connaissance [LALL 2004]. Pour cela, il faut suffisamment d'exemples vérifiant cette règle, ainsi qu'une quantité de contre-exemples qui ne porte pas préjudice au sens que prend cette règle dans son contexte d'extraction.

Le processus d'extraction des règles d'association peut générer des règles d'association triviales, ces règles sont évidentes et sont déjà connues et n'apportent pas d'information en plus (exemple : ***SI*** achat d'une imprimante ***ALORS*** achat de cartouches d'encre, etc.). Ainsi, on peut générer des règles inutiles qui sont difficiles à interpréter.

On trouve dans la littérature plusieurs critères d'évaluations des règles d'association qui sont divisées en deux catégories [LALL 2004], la première dite subjective (L'expert du domaine sait quels attributs qu'il souhaite avoir dans les règles d'association), et elle est rarement utilisée, car elle est contre l'objectif de l'ECD. La deuxième dite objective qui consistent à étudier le nombre d'exemples et de contre-exemples. Il faut que la règle obtenue ne soit pas trop générale ou évidente, car dans ce cas, il n'y a rien de nouveau! Elle ne doit pas non plus être trop spécifique puisque elle n'a aucune valeur si elle provient seulement des données aberrantes

On se limite dans ce travail à citer les critères pour les algorithmes de type *Apriori* qui sont basés sur le support et la confiance pour associer à chaque règle d'association de la forme : $A \rightarrow B$ des mesures permettant d'avoir une idée de sa qualité [LALL 2004]. Le support est une mesure dite d'utilité, la confiance est une mesure dite de précision.

A la fin on obtient les meilleures règles qui ont le support et la confiance supérieure ou égale à des seuils minimaux définis par un expert en fonction de ses objectifs et du type de données traitées.

II.11. Les extensions des règles d'association

Les algorithmes présentés dans les sections précédentes constituant les principes de base de la recherche des règles d'associations binaire. Plusieurs extensions ont été proposées. Ces extensions sont des améliorations des algorithmes classiques, notamment sur le point des fonctionnalités des algorithmes, afin de traiter d'autre type de règles d'association, telles que les règles d'association quantitatives, et les règles d'association généralisées qu'ont va éclaircir en bref, en restant toujours dans le cadre des données de types binaires.

Il existe aussi d'autres extensions comme par exemple, la recherche des motifs séquentiels, qui sont une autre forme de règle d'association intégrant la dimension temporelle.

II. 11.1. Les règles d'association quantitatives

En 1996, *Srikant et Agrawal* publient un article [SRIK 1996] sur les règles d'association **quantitatives.** Cette dernière est une nouvelle approche basée sur les règles d'association binaire. Ces relations sont sous la forme suivante : « **Si (Age est [18..35]) alors (salaire est 18000 DA) »**

L'idée est que les attributs quantitatifs peuvent être subdivisés en ensembles d'intervalles, de même que toutes les valeurs d'un attribut qualitatif peuvent être remplacées par des valeurs entières. Par exemple, la table personne (TAB.9) contient les attributs quantitatifs « *Age* et *NumCar»* et un attribut qualitatif « *Marrie »*

ID	Age	Marrie	NumCar
1000	21	No	0
2000	23	Yes	1
3000	24	No	1
4000	26	Yes	2
5000	29	Yes	2

TAB.9 TABLE : PERSONNE [Srik 1996]

Il est mentionné dans [SRIK 1996] que le problème des règles d'association quantitatives peut être transformé en un problème de règles d'association binaires, et n'importe quel algorithme pour la découverte des règles d'association binaire (classique) peut être utilisé pour trouver les règles d'association quantitatives.

Dans cet exemple, la table personne (TAB.9) est transformée en table (TAB.10), de données binaire, l'attribut « âge » est subdivisé en deux intervalles Age : [20..24] et Age : [25..29] et que l'attribut « Marrie » est remplacé par deux attributs « *Marrie 'Yes'* », « Marrie 'No' ». De même, l'attribut NumCar est subdivisé en trois attributs « NumCar 0 », « NumCar 1 », « NumCar 2 ».

ID	Age : [20..24]	Age : [25..29]	Marie : Yes	Marie : No	NumCar 0	NumCar 1	NumCar 2
1000	1	0	0	1	1	0	0
2000	1	0	1	0	0	1	0
3000	1	0	0	1	0	1	0
4000	0	1	1	0	0	0	1
5000	0	1	1	0	0	0	1

TAB.10 EXTENSION DE LA TABLE : PERSONNE [Srik 1996]

Cette transformation nous conduira à deux problèmes principaux :

- ***Premièrement :*** si les valeurs des intervalles sont trop larges, cela peut causer que le support de chaque intervalle peut être bas, c'est pour cela quelques règles de ces attributs ne peuvent pas être trouvées parce qu'elles ont un manque de support minimum, ce problème est appelé « ***le problème de min support »*** « ***Minsupp***».
- ***Deuxièmement*** : si le nombre de valeurs d'intervalles d'un attribut est petit, il y a une possibilité de perdre des informations, cela signifie que la confiance de certaines règles n'atteigne pas un seuil de confiance minimum, Ce problème est appelé ***« le problème de min confiance*** » « ***Minconf*** »

Nous sommes maintenant dans une situation de conflits : si les intervalles sont trop grands, nous ne pourrions pas atteindre le seuil minimum de confiance, et si ils sont trop petits, nous ne pourrions pas atteindre le seuil minimum de support, pour éviter ces problèmes, on fait toutes les rangés possibles des valeurs d'intervalles lors du processus de traitement. Cela signifie qu'en combinant les valeurs adjacentes d'intervalles, pour éviter le problème de « ***Minsupp***», et en augmentant le nombre d'intervalles pour éviter le problème de « ***Minconf*** ».

Cependant, cette approche conduit à deux problèmes majeurs, à savoir le temps d'exécution et le nombre de règles :

- **Temps d'exécution** : l'augmentation dans le nombre d'intervalle conduit à l'augmentation de temps d'exécution.
- **Nombre de règles** : si le problème de « ***Minsupp*** » est résolu, alors le nombre de règles générées est trop élevé, donc peut causer l'obtention des règles non intéressantes.

II.11.2.Les règles d'association généralisées

Les règles d'association généralisées ont été introduites par les auteurs (*Srikant et Agrawal*) en 1995, [SRIK 1995]. Ces relations sont similaires aux règles d'associations binaires, dont l'essentielle modification dans la méthode, consiste à prendre en compte une éventuelle structure hiérarchique appelée « **taxinomies »** sur l'ensemble des Items. L'idée de base de règle d'association généralisée est que les articles dans les grands magasins sont décrits par un code détaillé, par exemple des boots de taille 29 à 35 ont un code et des boots de taille 36 à 39 ont aussi un code différent que le premier. Si on cherche les règles d'associations ayant un support suffisant, cela diminuera les chances de les trouver. Mais si on arrive à indiquer aux algorithmes que les boots de taille 36-39 et les boots de taille 29-35, sont tous des boots donc le support de l'article ***boots,*** c'est la somme des supports des boots de différente taille, on augmente les chances de produire des règles d'associations sur les boots avec un support suffisant.

Table de transaction T :

Transaction	Items
100	Shirt
200	Jacket, Hiking Boots
300	Ski Pants, Hiking Boots
400	Shoes
500	Shoes
600	Jacket

(a)l'ensemble des transactions

Itemsets fréquents :

Itemset	Support
{Shirt}	1
{Jacket}	2
{Shoes}	2
{Hiking Boots}	2
{Ski Pants}	1
{Jacket, Hiking Boots}	1
{Ski Pants, Hiking Boots}	1

(b) l'ensemble des Itemsets fréquents

TAB.11 [Srik 1995]

Par exemple, si on prend un seuil minimum de supports supérieurs à 2 aucune règle ne sera produite. Une solution à ce problème consiste à l'extension de la table de transaction et une autre au niveau de l'algorithme.

Dans la table de transaction, remplacer chaque transaction T par une transaction T', où T', avec tous les items de T et en ajoutant l'ancêtre (Clothes est l'ancêtres de Shirt) de chaque items.

La méthode reçoit en entrée l'ensemble de transaction T et l'ensemble d'items I, et en plus de cela, une ou plusieurs hiérarchies sur l'ensemble I (taxinomie des items). La figure (FIG.14) montre une structure hiérarchique :

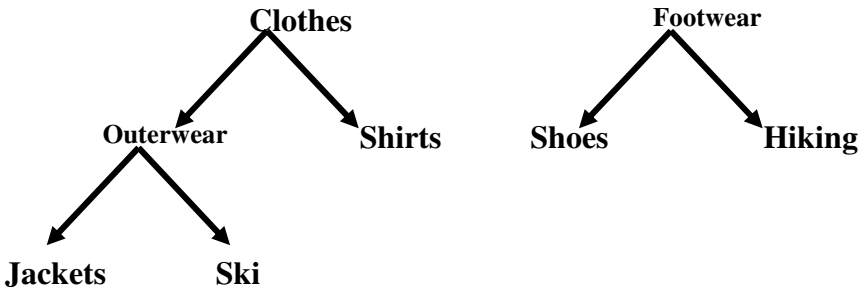


FIG.14. STRUCTURE TAXINOMIES [Srik 1995]

Transaction	Items
100	Shirt , Clothes
200	Jacket, Hiking Boots, Clothes, Outerwear , Footwear
300	Ski Pants, Hiking Boots, Clothes, Outerwear , Footwear
400	Shoes, Footwear
500	Shoes, Footwear
600	Jacket, Clothes, Outerwear

TAB.12 (A) EXTENSION DE LA TABLE T EN T'[Srik 1995]

Itemset	Support
{Jacket}	2
{Outerwear}	3
{Clothes}	4
{ Shoes }	2
{Hiking Boots}	2
{Footwear}	4
{Outerwear, Hiking Boots}	2
{clothes, Hiking Boots}	2
{Outerwear, Footwear}	2
{Clothes, Footwear}	2

TAB.12 (B) ITEMSETS FRÉQUENTS [Srik 1995]

Au niveau de l'algorithme, la construction des règles se passe en plusieurs étapes :

1. *Recherche des sous-ensembles fréquents de support supérieur à un seuil minimum noté* ***minsup.***

   Tout se passe comme si on complétait chaque transaction par tous les ancêtres des items de la transaction, les sous-ensembles fréquents contenant à la fois un Items et son ancêtre ne sont pas générés.

2. *Génération des règles à partir des sous-ensembles fréquents. La génération des règles d'association est similaire à la génération des règles d'association binaire (voir sectionII.7.4)*

3. *Elimination des règles non intéressantes* :

   Pour cela un critère d'intérêt des règles est posé pour ne pas déduire toutes les règles, en particulier si des règles ont comme partie antécédent des ancêtres de la partie antécédent d'autre règles. Les règles les plus générales sont privilégiées, sauf si les règles moins générales ont un support et une confiance significativement différente de la règle plus générale associe [Srik 1995].

II.12. Conclusion

Le domaine de l'extraction de connaissance à partir des données (ECD), permet la recherche et la découverte des règles associatives, qui demande une base théorique plus approfondie dans les *Mathématiques* et les *Statistiques* dans les bases de données gigantesque. Ce présent chapitre nous a permis de découvrir le concept des règles d'association et les publications liées à ce sujet. Nous avons vu que les premiers algorithmes traitant la découverte des règles d'association étaient développés, en premier lieu, dans un objectif de marketing. Actuellement, les règles d'associations trouvent tout leur intérêt dans tout processus de prise de décision, ce qui conduit à imposer certaines contraintes au niveau des critères d'évaluations.

Les règles d'associations en générale sont des outils efficaces pour identifier des relations entre attributs dans les bases de données. De même, ils peuvent faire découvrir aux analystes des associations inattendues. Ces relations peuvent ensuite être utilisées et intégrées dans les processus d'affaires de l'entreprise afin d'en améliorer les performances.

Sachant que les bases de données volumineuses demandent beaucoup de temps de calculs, qui devient un problème remarquable, les algorithmes discutés réussissent tout de même à identifier les règles d'associations présentes dans les données.

Comme nous l'avons déjà mentionné, les règles découvertes peuvent être intratransactionnelles ou intertransactionnelles. La recherche de règles, peut également s'effectuer sur une ou plusieurs dimensions, donnant à l'analyste beaucoup de flexibilité pour conduire sa recherche.

Plusieurs algorithmes existent pour l'extraction des règles d'association, on s'est concentré sur l'algorithme de base ***Apriori***, qui souffre de deux inconvénients majeurs : un nombre énorme de candidats et le parcours répétitive de la base de données lors de chaque calcul de support.

En effet, les algorithmes d'extraction des regèles d'association classique nécessitent le passage des données en mode binaire mais parfois on a besoin d'utiliser d'autres types données « **qualitatives** » (exemple : couleurs, type énuméré, etc.), et des données « **quantitatives** » (comme les mesures M, Kg, etc.). Dans l'extraction de connaissance, on a besoin de la précision des règles c'est pour cela qu'on propose de généraliser la technique des règle d'association aux données floues afin d'extraire des règles d'association floues, qui sont de type : "Si X est A alors Y est B", où X, Y sont des ensembles d'attributs et A, B sont des ensembles flous. Ce type de règles fera l'objet du quatrième chapitre de ce présent mémoire.

Chapitre III

Généralités sur la logique floue

III.1 Introduction

La logique floue est un modèle de représentation de l'information [CLAU 1998]. En effet, l'homme perçoit, imagine, raisonne, et prend des décisions à partir des modèles ou des représentations. Ainsi il est claire que la pensé humaine n'est pas une science exacte, en d'autre terme, elle n'est pas binaire, c'est-à-dire, l'homme ne prend pas des décisions sur des phénomènes courant par des valeurs vrai ou faux. Alors parfois, il utilise des termes, par exemple *moins cher*, *très cher*, etc. Il y a des siècles que l'homme recherche à maîtriser les incertitudes et les imperfections inhérentes à sa nature. À partir du XVII siècle l'homme arrive à formaliser la prise en compte des connaissances incertaines avec la théorie des probabilités. Cette dernière, a des limites et ne peuvent pas maîtriser les incertitudes psychologiques et linguistiques. Jusqu'à dans les années 50, l'homme est arrivé à résoudre le problème à l'aide de développements des théories de probabilité subjective avant l'émergence de la théorie des sous-ensembles flous à Berkeley dans le laboratoire de Lotfi Zadeh en 1965 [ZADE 1965]. et la théorie des possibilités en 1978 [ZADE 1978].. Ces deux dernières théories constituent aujourd'hui ce que l'on appelle Logique Floue.

La logique classique propose un mode de représentation binaire ce qui fait, qu'une proposition peut prendre l'un des valeurs, *vraie* ou *fausse*, mais le mode binaire n'est pas adaptable à la représentation de tous les problèmes rencontrés. Alors que la logique floue distingue une infinité de valeurs de véracité (entre 0 et 1) [BOUC 2003].

La logique floue offre des modes de raisonnement approximatifs. Ce dernier est le mode de raisonnement utilisé par les humains [IDRI 2003]. Cependant, la théorie des ensembles flous joue un rôle important dans la précision, notamment dans la représentation des phénomènes.

La logique floue trouve sa place dans les champs scientifiques, elle est introduite dans la plupart des domaines de recherche : industrielle, en biologie, en médecine, etc., ainsi que dans le domaine d'extraction des connaissances, car dans ce domaine en manipulant des données qui sont, dans la plupart des cas, qualitatives ou quantitatives. L'utilisation de la logique floue augmente les chances de tirer des connaissances plus précises et plus utiles.

La littérature sur la logique floue et ses applications est très riche [BOUC 2003]. Dans ce chapitre, nous nous allons nous intéresser aux concepts de base de la logique floue, qui sont nécessaires dans l'extraction des règles d'association floues.

III.2.Les ensembles flous

Les ensembles flous peuvent généralement être considérés comme une extension des ensembles classiques. Ils ont été introduits par Lofti A. Zadeh en 1965 [ZADE 1965]. Ce sont des ensembles permettant une adhésion graduelle de leurs éléments. Il s'agit d'un degré d'appartenance qui indique l'emplacement de l'élément dans l'ensemble flou. Ainsi, un élément peut appartenir aux plusieurs ensembles et cela est accompli par des degrés d'appartenances. Ces degrés sont des nombres réels dans l'intervalle [0..1] [GOTT 2006]. Un ensemble flou est défini par sa *fonction d'appartenance*, qui correspond à la notion de *fonction caractéristique* en logique classique [BOUC 2003].

Considérant A un sous-ensemble classique et X, un ensemble d'éléments. X_A, est la fonction caractéristique. Elle prend la valeur 0 pour les éléments X qui n'appartenant pas à A et la valeur 1 pour les éléments x qui appartenant à A. Elle peut être décrite par : $X_A :\rightarrow \{0,1\}$

$$X_A(x)=\begin{cases}1 & si \quad x\in A\\ 0 & \text{sin}on\end{cases}$$

Un sous-ensemble flou A est défini par une fonction d'appartenance qui associe à chaque élément x de X le degré d'appartenance $\mu_a(x)$ compris entre 0 et 1: $\mu_a(x) : X \rightarrow [0,1]$

$$X_A(x)=\begin{cases}\mu_a(x) & si \quad x\in A\\ 0 & \text{sin}on\end{cases}$$

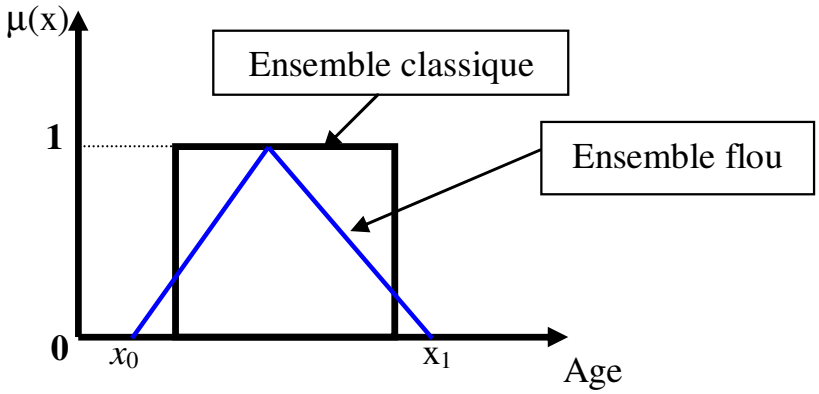


FIG.15 ENSEMBLE CLASSIQUE ET ENSEMBLE FLOU

Exemple : On prend l'exemple de la distance d'une ville à une autre. Pour représenter les distances entres les villes, nous utilisons les mots suivants **proche P**, **moyenne M** et **éloigné E.** La figure (FIG.15 (A)) montre un sous ensemble classique et la figure (FIG.15 (B) montre un sous ensemble flou. On

voit que la logique classique ne peut utiliser que le 0 et le 1, ainsi la distance est d'abord totalement proche dans l'intervalle [0, 150] puis, *moyenne* dans [150, 300] et enfin éloignée (plus de 300).

La figure(FIG.16 (A)) montre la représentation des fonctions *Proche, Moyenne* et *éloignée* dans le cas d'une logique binaire, alors que nous pouvons observé, dans la figure (FIG.16(B)) la représentation graphique des trois fonctions d'appartenance *Proche*, *Moyenne* et *Eloignée* dans la figure dans le cas flou.

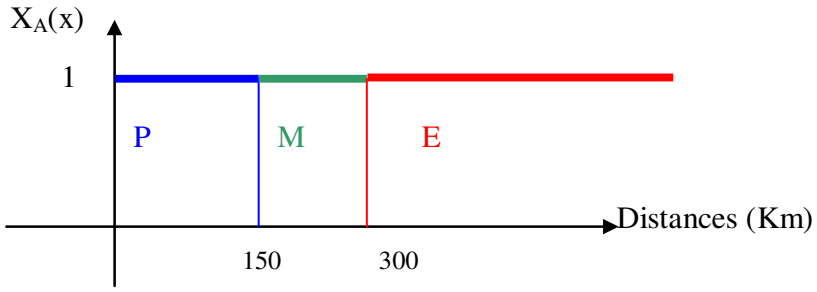


FIG.16 (A) (EN LOGIQUE CLASSIQUE)

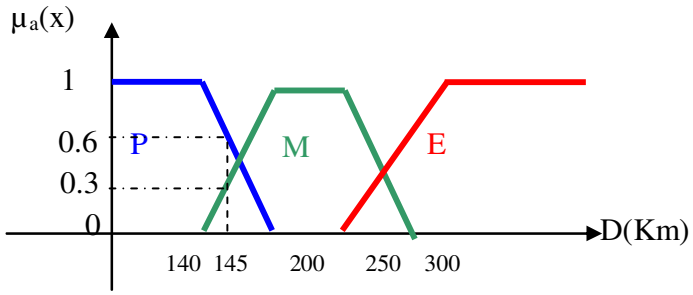


FIG.16 (B) (EN LOGIQUE FLOUE)

III.2.1 Univers du discours

La conception d'une application floue se déroule en plusieurs étapes, dont la première est de définir l'ensemble de référence. Ce dernier est appelé l'univers de discours [BOUC 1995].

Le terme «univers de discours » se réfère généralement à tout ensemble de termes utilisés dans un discours spécifique, c'est-à-dire une famille de termes linguistiques spécifiques au domaine concerné [JELE 1999]. Généralement l'univers de discours se conçoit par un expert dans le domaine, par exemple le concept de température : l'attribut « température » peut décrit par un certain nombre de

mots : ***chaud, froid, tiède, ou très chaud, assez chaud, tiède, assez froid, très froid.*** Chacun de ces termes est représenté par une fonction d'appartenance.

III.2.2. Variable linguistique

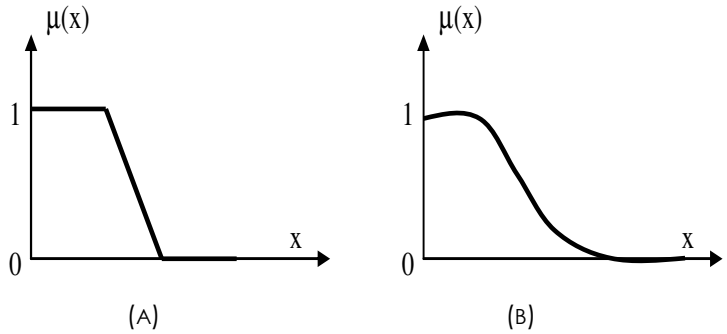
Une variable linguistique est représentée par un triplet (V, X_V, T_V) où V est la variable elle-même, X_V son domaine et T_V un ensemble de sous-ensemble flous de T_V utiliser pour caractériser V. Reprenons l'exemple de distance, cette variable définie sur l'ensemble des entiers positifs et caractérisé par les sous-ensembles flous (*Proche, Moyenne et Eloignée*). La variable distance est représentée par le triplet : (distance, E+, (*Proche, Moyenne et Eloignée*)).

III.2.3. Fonction d'appartenance

Généralement une fonction d'appartenance ($\mu_A(X)$) définis par des experts dans le domaine. Une fonction d'appartenance est une courbe. Cette dernière est utilisée pour définir comment chaque élément de l'univers de discours peut avoir une valeur d'appartenance, c'est-à-dire la valeur $\mu_A(X)$ mesure le degré avec lequel un élément x appartient à l'ensemble A. En effet, il n'y pas des règles précises pour définir une telle fonction d'appartenance. Chaque ensemble flou est représenté par sa propre fonction d'appartenance.

En générale une fonction d'appartenance peut avoir différentes formes [JELE 1999]:

- Monotones (croissante ou décroissante)
- Triangulaires
- Trapézoïdales
- En forme Gaussiennes



(A) (B)

FIG.17 EXEMPLE DES FONCTIONS D'APPARTENANCES MONOTONES DECROISSANTES

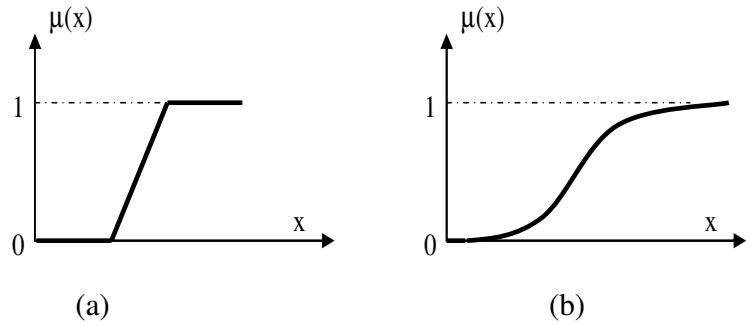


FIG.18 EXEMPLE DES FONCTIONS D'APPARTENANCES MONOTONES CROISSANTES

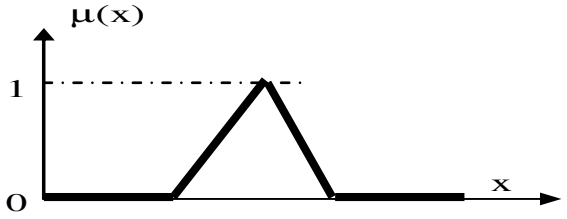


FIG.19 EXEMPLE DES FONCTIONS D'APPARTENANCES TRIANGULAIRES

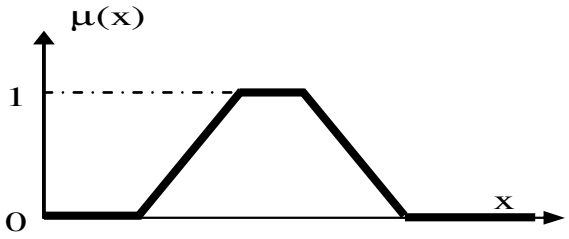


FIG.20 EXEMPLE DES FONCTIONS D'APPARTENANCES TRAPEZOÏDALES

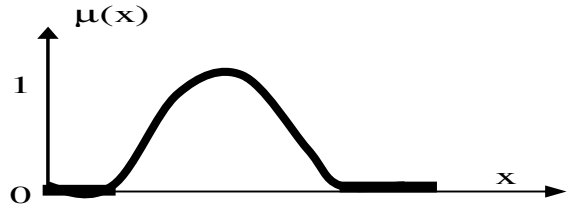


FIG.21 EXEMPLE DES FONCTIONS D'APPARTENANCES FORME GAUSSIENNE

III.2.4. Caractéristique d'un sous-ensemble flou

Un sous-ensemble flou est défini par sa fonction d'appartenance, et à partir de cette fonction on peu décrire plusieurs caractéristiques du sous-ensemble flou, qui sont présentés par la figure (FIG.21). Les concepts suivants sont importants **[BOUC 2003]**:

- ***Noyau* :** Le noyau d'un ensemble flous A de X noter par $Noy(A)$ est l'ensemble de tous les éléments qui lui appartiennent totalement, c'est-à-dire les éléments où leur degré d'appartenance vaut 1. Formellement : $Noy\ (A) = \{x \in\ X\ /\ \mu_A(x) = 1\}$
- ***Support* :** Le support d'un sous-ensemble flou A de X noté Supp(A) est l'ensemble de tous les éléments ayant un degré d'appartenance non nulle Formellement :
  $Supp\ (A) = \{x \in\ X\ /\ \mu_A(x) \neq 0\}$
- ***Hauteur* :** La hauteur d'un sous-ensemble flous A de X noté $h(A)$, est la valeur maximale prise par sa fonction d'appartenance. Formellement : $h(A) = \sup_{x \in X} \mu_A(x)$. ***Remarque*** : si $h(A) = 1$ on dira que le sous-ensemble flous est normalisé
- ***Cardinalité* :** La cardinalité d'un sous-ensemble flou A de X noté $|A|$ est la somme des degrés d'appartenance avec lequel des éléments de X appartiennent à A. Elle définie par : $|A| = \sum_{x \in X} \mu_A(x)$
- ***Coupe de niveau ou α–coupe* :** Une α-coupe d'un sous-ensemble flous A de X est un sous-ensemble des éléments ayant un degré d'appartenance supérieur ou égal à α, où α est le seuil d'appartenance. Formellement : $\alpha - coupe(A) = \{x \in X | \mu_A(x) \geq \alpha\}$

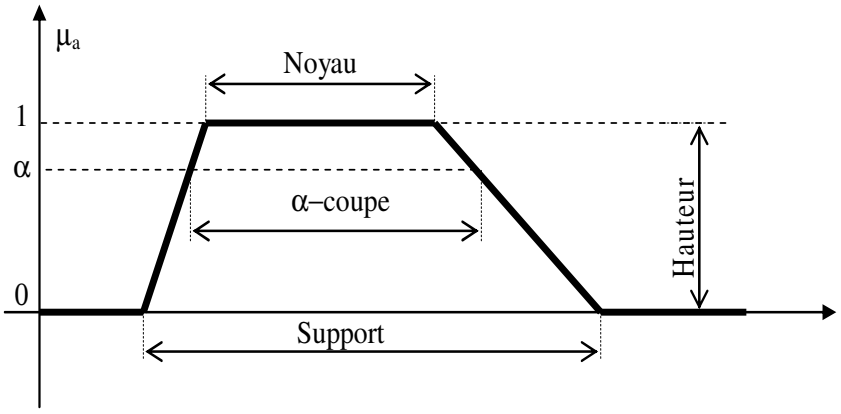


FIG.22 SUPPORT, NOYAU, HAUTEUR, α–COUPE D'UN SOUS-ENSEMBLE FLOU

III.3. Logique floue

La logique floue peut être vue comme une extension de la logique multivaluée. Cependant, leurs applications et ses objectifs sont très différents. Ainsi, le fait que la logique floue, traite les problèmes avec un mode de raisonnement approximatif plutôt que précise, ce qui implique, en général, les chaînes de raisonnement logique flou sont courte en longueur. En bref, en logique floue, tout y est inclus dans la vérité, tout simplement, il s'agit d'une question de degré d'appartenance. [BOUC 1995].

La puissance de la logique floue vient de la théorie des probabilités et la logique probabiliste. Cependant, une proposition en logique classique est soit vraie ou fausse. Alors que la logique floue permet une infinité de valeurs pour la véracité d'une proposition, c'est un nombre réel de l'intervalle [0,1], qui substituant sur des sous-ensembles floues.

III.3.1. les Opérateur flous

En général, les mêmes opérations qui sont effectuées sur les ensembles classiques, peuvent être appliquées sur les ensembles flous. Ces opérations sont les suivantes : le complément, l'union, et l'intersection. Selon les concepts de la logique floue, les opérations seront définies dans le reste de cette section.

Supposons A et B sont deux sous-ensembles flous et $\mu_A(x)$ et $\mu_B(x)$ leur fonction d'appartenance respective, on définit les opérations suivantes [BOUC 2003]:

1. ***Complément*** **:** la différence entre le complément des ensembles classiques et celui de floue est que le complément pour les ensembles flous peut avoir des valeurs différentes de 0. Il est clair que, dans les ensembles classiques le complément de 1 est le 0 et le complément de 0 est le 1. Nous supposons également que la fonction $\mu_{\bar{A}}$ (complément flou) est monotone, et non croissante, ce qui signifie pour tous les éléments a, b $\in [0,1]$ si $a < b$ alors $\mu_{\bar{A}}(x) \geq \mu_{\bar{B}}(x)$.

   Le complément flou est défini par la fonction d'appartenance $\mu_{\bar{A}}(x) = 1 - \mu_A(x)$

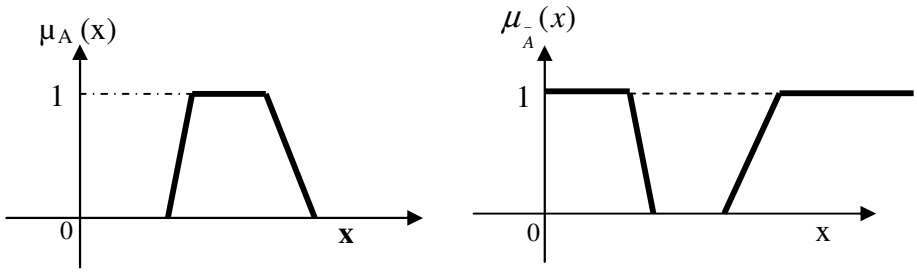


FIG.23 LE COMPLEMENT FLOU

2. ***Intersection*** **:** L'intersection de sous-ensembles flous A et B de X est un sous-ensemble flou C noté par : $A \cap B$, tel que : $\forall x \in X \mu_C(x) = Min(\mu_A(x), \mu_B(x))$

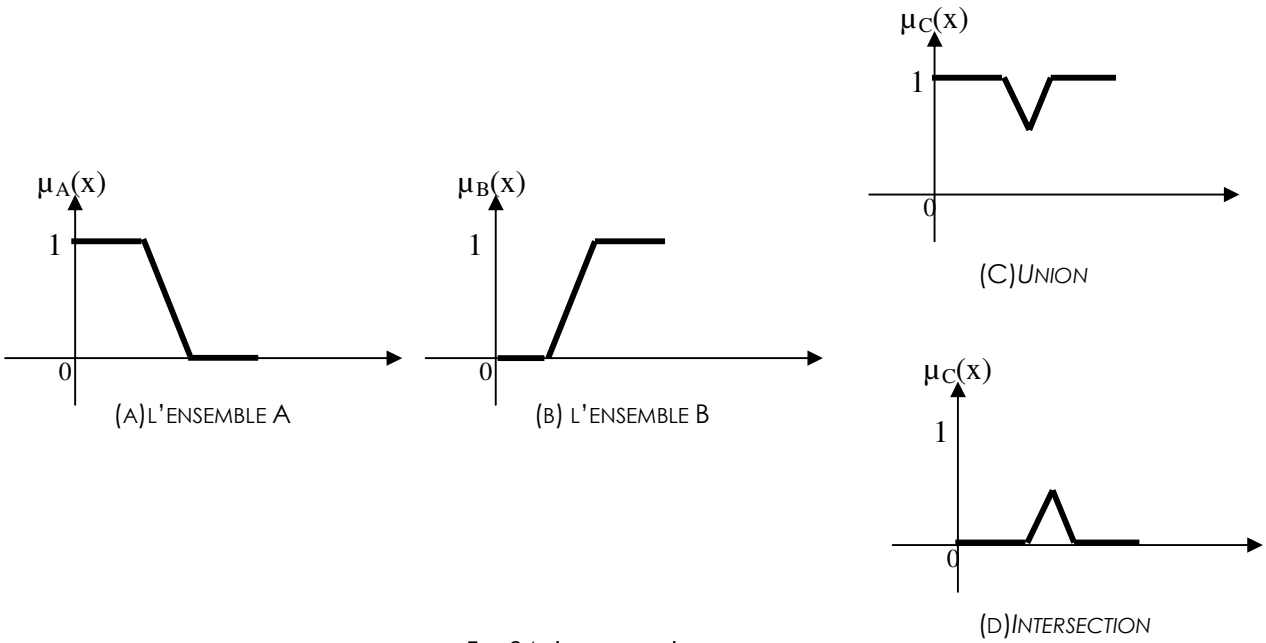
L'intersection floue donne les mêmes résultats que l'intersection pour les ensembles classiques. En effet $i(0,0)=0$, $i(0,1)=0$, $i(1,1)=1$, Alors le $Min(0,0)=0$, $Min(0,1)=0$, $Min(1,1)=1$,(i pour désigner l'intersection). Aussi il y a d'autres caractéristiques qui sont les suivant :

- *La commutativité* : $Min(a,b) = Min(b,a)$.
- *La monotonicité* : **Si** $(a \le a') \wedge (b \le b')$ **Alors** $Min(a,b) \le Min(a',b')$
- *L'associativité* : $Min(Min(a,b),c) = Min(a,Min(b,c))$.

3. ***Union*** **:** L'union de sous-ensembles flous A et B de X est un sous-ensemble flou C noté par : $A \cup B$, tel que : $\forall x \in X \mu_C(x) = Max(\mu_A(x), \mu_B(x))$

L'union floue donne les mêmes résultats, que l'union pour les ensembles classiques. En effet, $u(0,0)=0$, $u(0,1)=1$, $u(1,1)=1$, Alors le $Max(0,0)=0$, $Max(0,1)=1$, $Max(1,1)=1$,(u pour désigner l'union). Ainsi il y a d'autres caractéristiques qui sont les suivant :

- *La commutativité* : $Max(a,b) = Max(b,a)$.
- *La monotonicité* : **Si** $(a \le a') \wedge (b \le b')$ **Alors** $Max(a,b) \le Max(a',b')$.
- *L'associativité* : $Max(Max(a,b),c) = Max(a,Max(b,c))$



(A)L'ENSEMBLE A (B) L'ENSEMBLE B (C)UNION (D)INTERSECTION

FIG.24 L'UNION ET L'INTERSECTION FLOUE

III.3.2. Les normes triangulaires

Les normes triangulaire notées *(t-norme, t-conorme)* sont l'un des facteurs importants dans la recherche sur les règles d'association floues. Une norme triangulaire (*t-norme ou t-conorme)* sont des fonctions commutatives, associatives, dont 1 est l'élément neutre pour la *(t-norme),* et le 0 est l'élément neutre pour la *(t-conorme).*

Un exemple classique : L'opérateur *Max* satisfait les propriétés *de t-norme,* alors que, l'opérateur *Min* satisfait les propriétés *de t-conorme*, [MESI1999]. Les *(t-norme, t-conorme) les plus utilités sont décrite par la table(TAB.13) :*

t-norme	***t-conorme***
$T_M(x,y) = min(x,y)$	$S_M(x,y) = max(x,y)$
$T_P(x,y) = xy$	$S_P(x,y) = x+y-xy$
$T_W(x,y) = max\ (x+y-1,0)$	$S_W(x,y) = min(x+y,1)$

TAB.13 LES NORMES TRIANGULAIRES [MART 2003]

III.4. conclusion

Dans ce chapitre, nous avons présenté une brève introduction des concepts de base de la théorie des ensembles flous et de la logique floue. Ces notions nous semblent nécessaires pour le processus d'extraction des règles d'associations floues. L'utilisation des termes linguistiques rend les règles extraites plus compréhensibles. Cependant, nous pouvons dire que la logique floue est un outil de codification des connaissances.

Dans le chapitre qui suit nous présentons les règles d'association floue. Ainsi nous éclaircirons l'apport de la logique floue dans l'extraction des règles d'association floue.

Chapitre IV

les règles d'association floues

IV.1. Introduction

Le deuxième chapitre présente une vue générale sur les règles d'association classiques, connues sous le nom, *règles d'association binaires* [AGRA1993]. Dans la recherche des règles d'association, on considère que les attributs d'une telle base de données sont des données binaires, représentées par l'ensemble {0,1}, où le 1 signifie qu'un attribut apparaît dans un enregistrement, et le 0 pour le cas contraire. Cependant, dans la réalité, les données sont de différents types: numérique, catégoriel, entier, etc., et peuvent généralement, être des données **qualitatives** ou/et **quantitatives**.

La logique floue a un fort impact sur les techniques de la *fouille de données* et en particulièrement sur les règles d'associations. Plusieurs recherches ont été effectuées pour traiter les données quantitatives [KUOK 1998][DEGR 2001][GYEN 2001], ainsi que pour l'intégration de la théorie des sous-ensembles flous dans la recherche et l'extraction des règles d'association. Ces recherches ont donné naissance à une nouvelle approche nommée, *règles d'association floues.*

Les *règles d'association floues*, sont de nouvelles approches basées sur les règles d'association classiques. Ces règles sont de la forme: « *Si (X est A) alors (Y est B)* », où X, Y sont des d'attributs ou groupe d'attributs disjoints, alors que A, B sont des ensembles flous associés à ces attributs respectifs. Généralement A et B sont des termes linguistiques comme : *PEU, MOYEN, FAIBLE, FORT, VIEUX, AGÉ, etc*. Ces termes sont plus compréhensibles pour les utilisateurs.

En effet, de nombreuses recherches ont été menées pour pallier aux problèmes des règles d'association floues, comme dans [MANN 1997][KUOK 1998][FU 1998] [CHEN 2002][MART 2006] etc. Bien qu'il existe plusieurs approches pour extraire des règles d'association floues, nous nous sommes limités à citer les trois approches suivantes :

- ***L'approche quantitative***
- ***L'approche structures taxonomiques floues.***
- ***l'approche de découverte d'épisode et la recherche des motifs séquentiels flous.***

En outre, nous présentons notre contribution qui consiste en une nouvelle approche basée sur les règles d'association binaires, ce qui n'est rien d'autres qu'une une extension de l'algorithme *Apriori*, conduisant à l'extraction des règles d'association floues.

Dans la section IV.2, nous évoquons les définitions *d'Item flou, Itemset flous, Itemset flous fréquent*. Dans la section IV.3, nous introduisons les mesures de qualité d'un Itemset flous. La section IV.4 est consacrée aux différentes approches d'extraction des règles d'association floues. Enfin, nous terminons par notre contribution qui fait l'objet de la dernière section IV.5.

IV.2.Définition (Item flou, Itemset flous, Itemset flous fréquent)

- ***Définition 1 :*** Un item flou est un couple (Item, sous-ensemble flou). Par exemple, (chocolat, beaucoup) est un Item flou où *beaucoup* est un sous-ensemble flou défini par sa fonction d'appartenance [FIOT 2004].
- ***Définition 2 :*** Un Itemset flou est un ensemble d'items flous. Il peut être écrit sous la forme d'un couple de deux ensembles (ensemble d'items, ensemble des sous-ensembles flous associent à chaque item) ou sous la forme d'une liste d'items flous. Par exemple, ((chocolat, beaucoup) ; (lait, peu)) est un itemset flou où *beaucoup* et *peu* sont deux sous-ensembles flous [FIOT 2004].
- ***Définition 3 :*** Un Itemset flou est dit fréquent si son degré d'appartenance dépasse un seuil minimum *(Ω),* qui est entre 0 et 1, c'est-à-dire $(0 < \Omega \leq 1)$. Ce seuil sera fixé dans la plupart des cas par un expert.

En général, un Itemset flou est un couple (X, A) qui signifie que l'attribut X a la propriété A, où $X \subset I$ (I est l'ensemble des Items ou attributs). X se compose de plusieurs attributs c'est-à-dire $X = \{x_1, x_2 ....., x_p\}$, alors que $A \subset F$ (F est l'ensemble des sous ensembles flous). $A = \{a_1, a_2 ....., a_p\}$ où chacune de ces propriétés a_j, est associée à un attribut x_j respectivement. Finalement, l'item (x_j, a_j) est désigné sous le nom d'Item flou.

IV.3. Mesures de qualité des Itemsets flous

IV.3.1. Le degré d'un Itemset (X, A)

Un Item flou (x_j, a_j), est représenté dans la transaction par son degré d'appartenance noté par ($\mu(a_j)(t_i[x_j])$).Cette expression signifie que, le degré des Items (x_j) à l'I^{ème} transaction

est $\mu(a_j)$. On peut dire que l'item (x_j, a_j) est floue (c'est-à-dire x_j est a_j) lorsque $\mu(a_j)$ dépasse le seuil *(Ω)*.

Avant de calculer le support d'un Itemset (X, A), il faut définir tout d'abord, le degré d'appartenance de l'Itemset flous. Bien que, la détermination du degré d'appartenance d'un Itemset constitue un problème pour la recherche des Items flous fréquents. Plusieurs approches ont été proposées afin de bien traiter cette problématique [DEGR 2001] [ION 2006].

Des recherches, se sont basées sur les *t-norme* (T) et *t-conorme* ($\perp$) triangulaire, par exemple en utilisant le *Min*($\mu(a_j)(t_i[x_j])$) pour le *t-norme,* et le *Max* ($\mu(a_j)(t_i[x_j])$) pour le *t-conorme* avec, $j = \{1,2,3,.....p\}$ [FIOT 2004].

Cependant, d'autres recherches se sont basées sur le produit des degrés d'appartenances c'est-à-dire le degré d'un Itemset (X, A) égale $\prod_{j=1..p}\{\mu(a_j)(t_i[x_j])\}$ [KUOK 1998]. La table (TAB.14) montre un exemple de ces trois approches.

Ti	x1, a1	x2, a2	x3, a3	x4, a4	***t-norme Min***	***t-conorme Max***	***Produit***
T1	0.53	0.75	0.56	0.85	0.53	0.85	0.19
T2	0.75	0.65	0.60	0.80	0.60	0.80	0.23
T3	0.50	0.70	0.55	0.75	0.50	0.75	0.14
T4	0.60	0.80	0.55	0.75	0.55	0.80	0.20
T5	0.80	0.55	0.65	0.55	0.55	0.80	0.16

TAB.14 EXEMPLE DE CALCUL DES DEGRES D'APPARTENANCE

IV.3.2 Support d'un Itemset (X,A)

Le support d'un Itemset flou noté $Fsupp(X, A)$, est en général le pourcentage du nombre de transactions supportant l'Itemset flous (X, A) par rapport au nombre total des transactions [GYEN 2001].

Les degrés d'appartenances ($\mu(a_j)$), étant des quantités numériques, ils offrent plusieurs possibilités de calcul. Le comptage binaire, est l'un de ces calculs. Il est le plus simple, car il

ressemble à celui du cas des règles d'associations binaires, c'est-à-dire quand le degré d'appartenance supérieur à 0 ($\mu(a_j)>0$), il est considéré comme un 1 et sera pris en compte lors de calcul du support.

Fiot et al. ,dans [FIOT 2004] exposent trois approches différentes pour le calcul du support, en s'appuyant sur la cardinalité d'un Itemset flou. Cette cardinalité joue un rôle important dans le calcul du support d'une règle d'association flou, ces différents comptages sont les suivants :

- ***Comptage seuil*** : on compte toutes les transactions contenant l'item avec un degré d'appartenance qui dépasse un seuil fixé au début par un expert.
- ***Somme comptage***:on prend en considération toutes les transactions contenant l'item en additionnant tous les degrés d'appartenances de l'Item.
- ***Somme comptage seuil***: on prend en considération toutes les transactions contenant l'item en additionnant tous les degrés d'appartenances d'un Item, qui dépassent un seuil fixé au début par un expert.

Ces comptages permettent de calculer plusieurs supports, la table (TAB.15) illustre un exemple de calcul des ces cardinalités.

Transaction	X est A	X est A	X est A	X est A
T1	0.8	0.8	0.8	0.8
T2	0.3	0.3	0.3	0.3
T3	1	1	1	1
T4	0	0	0	0
Les différents comptages	Comptage binaire	Comptage seuil	Somme de comp.	Somme de comp. Seuil
	3	2 (ω=0.49)	2.1	1.8 (ω=0.49)

TAB.15 CALCUL DES CARDINALITES

Dans le cas du *Comptage binaire* et *Comptage seuil*, le support est formellement calculé par :

$$Fsupp(X,A)=\frac{nb(X,A)}{nbT} \quad \text{où} \quad nb(X,A)=\sum_{t\in T} nbt_i \text{ dont } \mu(a_j)(t_i[x_j])>0 \qquad (1)$$

Dans cet exemple le *Fsupp*(X, A) = 3/4 est 75%

Dans le cas du *comptage seuil*, on tient en compte les degrés d'appartenances qui dépassent le seuil ω fixé par l'expert, c'est-à-dire $\mu(a_j)(t_i[x_j]) >\omega$

Dans cet exemple le *Fsupp*(X,A) = 2/4 est 50%

Dans le cas des sommes des comptages, le support est calculé par les équations :

$$Fsupp(X,A) = \frac{\sum_{t \in T} T(\mu(a_j)(t_i[x_j]))}{nbT} \quad ou \ j = 1..p \tag{2}$$

$$Fsupp(X,A) = \frac{\sum_{t \in T} \perp(\mu(a_j)(t_i[x_j]))}{nbT} \quad ou \ j = 1..p \tag{3}$$

$$Fsupp(X,A) = \frac{\sum_{t \in T} \prod_{j=1..p} (\mu(a_j)(t_i[x_j]))}{nbT} \tag{4}$$

Enfin, dans le cas de *somme comptage seuil,* seuls les degrés d'appartenances qui dépassent le seuil ω, ($\mu(a_j)(t_i[x_j])>$) seront pris en considération .

IV.3.3 confiance d'une règle (X, A)→ (Y,B)

La confiance d'une règle $(X,A) \to (Y,B)$ est le rapport des supports de l'antécédent union le conséquent $(X \cup Y; A \cup B)$, c'est-à-dire le nombre de transactions supportant à la fois l'antécédent et le conséquent de la règle sur le support de l'antécédent [GYEN 2001]:

$$\textbf{\textit{Fconf}} = \frac{FSupp(X \cup Y, A \cup B)}{FSupp(X.A)}$$

IV.4.Les différentes approches des règles d'association floues

IV.4.1 Approche quantitative

Dans la recherche sur les règles d'association quantitatives (voir section II.10)[SRIK 1996], les attributs sont divisés en un ensemble d'intervalles. Certains éléments situés aux extrémités des intervalles, qui peuvent être ignorés par l'algorithme, à cause du problème de *Minsupp* c'est-à-dire que l'Items dans ce cas n'a pas assez de support. Lors du processus de traitement de ces éléments, à forcément conduit à un problème baptisé « *problème d'extrémité d'intervalle*, en anglais *the sharp boundary problem* ».

Par exemple, considérons que l'âge moyen est entre 30 et 50 ans, alors le degré d'appartenance à l'âge moyen pour une personne ayant l'âge de 29 ans est de 0% et celui pour les gens ayant l'âge de 31 ans est de 100%. Cependant, dans la réalité, il n'y a pas une grande

différence entre ces âges. Dans cet exemple, nous pouvons représenter l'attribut « âge » en logique floue par la fonction de degré d'appartenance $\mu(x)$ illustré dans la figure (FIG.25).

Il est intéressant de proposer des approches basées sur la logique floue afin de résoudre cette problématique. *Kuok et al.* [KUOK 1998], sont les premiers qui ont proposé une approche avec l'intégration des concepts de la logique floue pour traiter les donnés quantitatives, afin d'extraire des règles d'association floues. Cette approche est connue sous le nom d'*Approche Quantitative.*

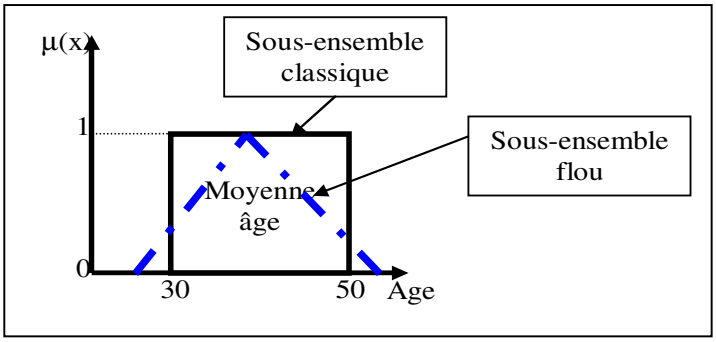


FIG.25 REPRESENTATION DE MOYENNE AGE

Selon *Kuok et al.* [KUOK 1998], l'extraction des règles d'association floues, est la recherche des règles d'association avec l'utilisation des concepts des ensembles flous dans lesquels les attributs quantitatifs peuvent être transformés en des sous-ensembles flous**.**

IV.4.1.1. Le concept de base de l'approche quantitative

Soit $T = \{t_1, t_2, t_3, ...t_n\}$ un ensemble de transaction pour représenter une base de données D, t_i représente le i^{ème} enregistrement dans D, l'ensemble $I = \{i_1, i_2, i_3, ...i_m\}$ pour représenter les attributs apparus dans la base de données, le i_j est le j^{ème} attribut dans la base de données. À chaque attribut i_k on associe un ou plusieurs ensembles des sous-ensemble flous $F_{i_k} = \left\{ f^1_{i_k}, f^2_{i_k} .....f^l_{i_k} \right\}$, $f^l_{i_k}$ représente le l^{ème} sous-ensemble flous dans F_{i_k}. Comme exemple l'attribut salaire est représenté par trois sous-ensembles flous (*haut, moyenne, bas*), nous aurons $F_{salaire}$ = {haut, moyenne, bas}. Les ensembles flous et les fonctions d'appartenance sont donnés par les experts de domaines.

Les règles d'association floues de l'approche quantitative sont de la forme suivante : *Si (X est A) alors (Y est B)* avec : $X = \{x_1, x_2 ....., x_p\}$ et $Y = \{y_1, y_2 ....., y_q\}$ sont des Itemsets du I. Il faut bien noté que X et Y ne contient pas un Items commun, c'est-à-dire, $X \cap Y = \varnothing$, avec A, B des ensembles ou des sous-ensembles flous associés aux Itemsets X et Y respectivement $A = \{F_{x1}, F_{x2}, ...F_{xp}\}$ et $B = \{F_{y1}, F_{y2}, ...F_{yq}\}$. La partie ***Si (X est A)*** est appelée ***antécédent*** ou ***condition***, et la partie, ***alors (Y est B)*** est appelé ***conséquent*** ou ***résultat***. Ainsi, la sémantique de cette règle est donnée comme par : si *(X est A)* est satisfait il peu impliqué que *(Y est B)* est satisfait aussi.

Il est clair que pour valider une règle d'association floue, il faut définir des mesures de qualités. En effet, on aura satisfaction d'une partie *Si* ou la partie *Alors*, ou les deux en même temps, si le support et la confiance de la règle atteignent des seuils minimums *Fminsupp*, *Fminconf* respectivement définie par l'utilisateur.

Cependant, une nouvelle approche a été proposée par *Kuok et al.* [KUOK 1998], pour valider les règles d'association floues où ils utilisent deux facteurs : ***le facteur d'importance*** « *significance factor » et* ***le facteur de la certitude*** *« certainty factor* ».

Les règles intéressantes sont celles ayant des facteurs *d'importance* et de *certitude*s plus grandes que les seuils choisis par un utilisateur. Ces facteurs sont cités ci-après :

IV.4.1.2 Le facteur d'importance

Ce facteur est basé sur la notion du support, c'est la somme des valeurs, en multipliants les degrés d'appartenances d'un Itemset flous ayant satisfait un seuil minimum fixé par un utilisateur sur le nombre de transactions d'une base de données, comme explicité par la formule ci-dessous :

$$S(X, A) = \frac{\sum_{t_i \in T} \prod_{x_j \in X} \{\mu(a_j)(t_i[x_j])\}}{NombreT} \quad (7)$$

où $\mu(a_j)(t_i[x_j]) = \begin{cases} m_{a_j} \in A(t_i[x_j] sim_{a_j} \geq \omega \\ 0 \sin on \end{cases}$

IV.4.1.3 Le facteur de la certitude

La certitude est similaire à la confiance proposée en [AGRA 1993], c'est le rapport, du facteur d'importance de l'itemset constituant la règle sur le facteur d'importance du conséquent.

Formellement : $C((X,A),(Y,B)) = \frac{S((X,A),(Y,B))}{S(X,A)}$ (8)

IV.4.2.Approche structure taxonomique floue

Cette approche est similaire à celle illustrée dans le chapitre deux (section II.10.2). Elle peut être utilisée quand on traite des structures taxonomiques qui ne sont pas classiques, mais floues. Une taxonomique est une catégorisation définie par l'utilisateur avec les articles disponibles [DEGR 2001].

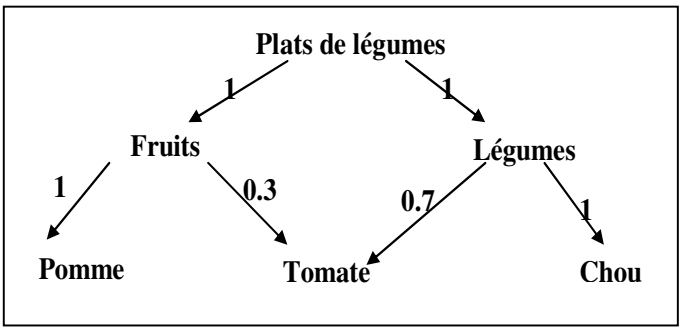


FIG.26 EXEMPLE STRUCTURE TAXONOMIQUE FLOUE

Il s'agit d'une hiérarchie représentée par un arbre où un nœud fils appartient à un seul nœud parent. Dans certains cas, on pourrait avoir besoin, pour permettre un classement des membres à plus d'un noeud parent. Comme exemple, une tomate pourrait être considérée à la fois, un fruit et un légume. Pour démontrer ces dépendances, le concept de structures taxonomiques floues est présenté dans, [WEI 1999]. La figure (FIG.26), illustre un exemple de la structure taxonomique floue.

Dans une structure taxonomique classique, un *nœud fils* appartient à son ancêtre avec un degré d'appartenance, (figure (FIG.27)). Cela signifie que n'importe quel *nœud fils* peut avoir seulement un seul ancêtre. Les structures taxonomiques floues éliminent cette hypothèse en permettant aux *nœuds fils* d'avoir plusieurs degrés d'appartenance à différents noeuds parentaux

en même temps. Chaque *nœud fils* appartient à ses nœuds parentaux par un certain degré μ, où $0 \le \mu \le 1$. N'importe quel nœud (x) est appelé un ancêtre de nœud (y) s'il existe un chemin direct de x à y. Alors que le nœud y est appelé le descendant du noeud x.

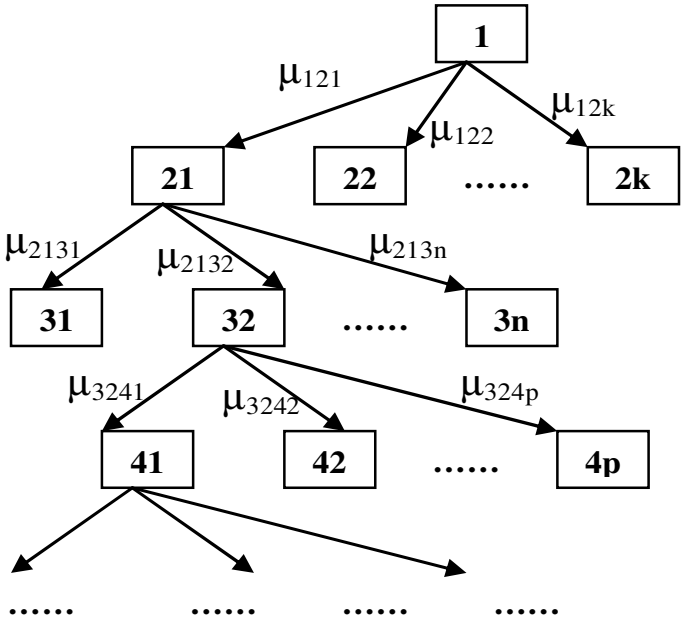


FIG.27 STRUCTURE TAXONOMIQUE DANS LES CAS GENERALE

Les valeurs dans l'arbre ne sont pas des ensembles flous et il n'existe pas de fonction de degré d'appartenance pour eux. Au lieu de cela, ces structures doivent être définies *a priori* par les experts du domaine. Après la définition de ces structures, des procédures normales d'extractions de règles d'association floues, peuvent être utilisées pour traiter des données qui sont organisées sous la forme d'une structure taxinomique floue.

IV.4.3. La découverte d'épisode et Les motifs séquentiels flous

La découverte d'épisodes est une approche, qui a été proposée par *Mannila et al*. en 1995 [MANN 1995] (*Episodes discovery*). L'idée de base de cette approche est qu'un évènement peut être considéré comme un Item [MANN 1995].

La découverte d'épisodes sert *à* identifier des épisodes cycliques à l'intérieur des données séquentielles. En effet, les données peuvent être considérées comme des séquences

d'événements auxquels on impose certaines contraintes de temps. Les épisodes sont définis comme étant une collection d'événements dont les occurrences sont relativement près les unes des autres et celles-ci sont limitées par une fenêtre temporelle. En d'autres termes les épisodes sont des collections partiellement ordonnées d'événements. Par exemple, on observe régulièrement q'un événement Y suit un événement X dans la plupart des cas.

Cependant, d'autres recherches ont été initiées par *Agrawal et Srikant* [AGRA 1995] pour la recherche des règles d'association séquentielles dans les bases de données. Cette approche est un prolongement des règles d'associations classique à deux niveaux :

- ***Premièrement***, en ajoutent un identifiant temporel, ce qui permet d'ordonné les items selon leur date de transaction. Les bases de données utilisées par les règles d'associations séquentielles contiennent deux identifiants un pour la transaction et l'autre pour le client pour des articles achetés. L'identifiant temporel est utilisé pour garder l'historique d'achat.
- ***Deuxièmement***, dans la sémantique des règles, c'est-à-dire dans la forme des règles, comme dans la règle séquentielle : « *Les clients ayant achète un téléviseur achètent un lecteur DVD quelque temps plus tard*. ».

En 2004, *Fiot et al.* [FIOT 2004] ont étudié la question dans un autre contexte. Ils ont proposé une méthode basée sur les notions de la logique floue afin d'extraire des motifs séquentiels flous.

Supposons l'ensemble d'Items I= {A, B, C, D, E} qui représente des articles achetés par des clients. Un motif séquentiel est de la forme : S = < (E) (A D) (C)>. La séquence S peut s'exprimer par : « Si un client achète l'article (E) puis les articles (A D) et enfin l'article (C).

Dans le cadre de la recherche des motifs séquentiels flous la séquence été remplacée par le terme *g-k-séquence*. Cette dernière se *compose de* g *itemsets flous noté* S = (X;A) *regroupant au total* k *items flous de la forme* $[x_p; a_p]$ *où* x_p représente l'Items et le a_p le sous-ensemble flou associe à celui-ci. Enfin une g-k-séquence est décrite comme suit : *S* = < s1 s2 …. sg >. Par exemple *<([chocolat, beaucoup][lait, peu])([lait, peu])>*.

L'extraction des séquences floues repose sur la même procédure que l'extraction des motifs séquentiels classiques. Un algorithme par niveau est utilisé. Les motifs fréquents flous de taille 1 sont d'abord extraits, puis utilisées pour la construction des motifs de taille 2, *etc.*

Dans le cadre des motifs séquentiels flous, deux différences principales apparaissent en comparaison aux motifs classiques :

- les supports sont calculés en considérant le support flou,
- le même item peut se retrouver au sein d'une même séquence mais pas au sein d'un même itemset. Par exemple, il n'est pas possible de considérer la séquence *<([chocolat,peu][chocolat,beaucoup])>*.

IV.5.Approche proposée

Apriori [AGRA 1994] est le premier algorithme d'extraction des règles d'association dans les bases de données gigantesques. Les règles d'association extraites par *Apriori* sont de type binaire. En effet, plusieurs extensions sont émané de l'algorithme *Apriori*, comme : *Partition, FP-tree, et Eclat* [HAN 2000], etc. Ces Algorithmes ont un but commun celui d'extraction des Itemsets fréquents. La variété des types données, *(les données sont : quantitative et/ou qualitative),* cette qualification a donné naissance aux règles d'associations quantitatives [SIRK 1995].

Cependant la théorie des sous ensemble flous, et la logique floue, ont une grande influence sur la technique des règles d'association, ce qui a conduit à l'émergence des règles d'association floues. Plusieurs recherches ont été effectue pour traiter les données quantitatives [SIRK 1995][KUOK 1995] [CHEN 2002].

Le choix des mesures de qualité des règles d'association, en générale constitue un problème important, notamment dans l'extraction des règles d'association floues. Car un Item flou est représenté dans la base de données par son degré d'appartenance. Cette quantité numérique peut être calculée de plusieurs manières (section IV.3). Il est alors intéressant de proposer une approche en se basant sur la façon de manipuler le degré d'appartenance ($\mu(a_j)(t_i[x_j])$) d'un Item (x_j, a_j)dans le calcul du support.

- Notre approche consiste à extraire des règles d'association floues, également une extension de l'algorithme initial *Apriori* [AGRA 1994]. Plus précisément, notre méthode est une modification de la manière par laquelle l'algorithme calcul le support d'un *Itemset* flous en appliquant les mesures discutées en section IV.3.

Nous avons utilisé les mesures basées sur la *t-norme et t-conorme* triangulaire ainsi que le *produit* des degrés d'appartenances. Aussi en se basant sur le *comptage seuil*. Dans ce cas, on fixe un seuil (Ω) de degré d'appartenances, seuls les degrés qui dépassent (Ω*)* sont tenus en

compte. En outre une comparaison entre les différents résultats obtenus par les différentes approches est donnée par la suite.

IV.5.1.Une extension de l'algorithme Apriori aux données floues

Dans cette section on va présenter l'architecture de notre approche, qui procède en trois étapes citées ci-après:

- *la première sert à transformer les données brutes on données floues. Cette transformation consiste à transformer la table de transactions T en une nouvelle table T', en utilisant des partitions floues, concernant les attributs quantitatifs. Généralement ces partitions sont posées par un expert du domaine.*
- *la deuxième étape sert à appliquer l'algorithme Apriori pour générer les Itemsets flous fréquents en basent sur les approches du calcul du support d'un Itemset flous, ainsi en respectant également les seuils minimums fixés par l'utilisateur.*
- *la troisième étape a comme objectif de générer les règles d'association floues Ces règles sont de la forme :* $(X, A) \rightarrow (Y, B)[F\sup f\%, Fconf\%]$, *où* (X, Y) *sont des Itemset et* (A, B) *des sous-ensemble flous et (Fsupf%, Fconf%) sont le support flou et la confiance floue respectivement. pour cela les règles qui ayant un confiance supérieur à un seuil* ***Fminconf*** *sont acceptées.*

Ces étapes sont schématisées par la figure (FIG. 28).

Et dans ce qui suit on utilise les notations ci-dessus pour des raisons de simplification :

- CF_k : Ensemble des Itemsets flous candidats de taille k
- LF_k : Ensemble des Itemsets flous fréquents de taille k
- $F = \{ f_{ij}^1, f_{ij}^2, \ldots, f_{ij}^p \}$: Les partitions floues d'un attribut i_j
- f_{ik}^m.value : pour désigner la valeur d'un degré d'appartenance

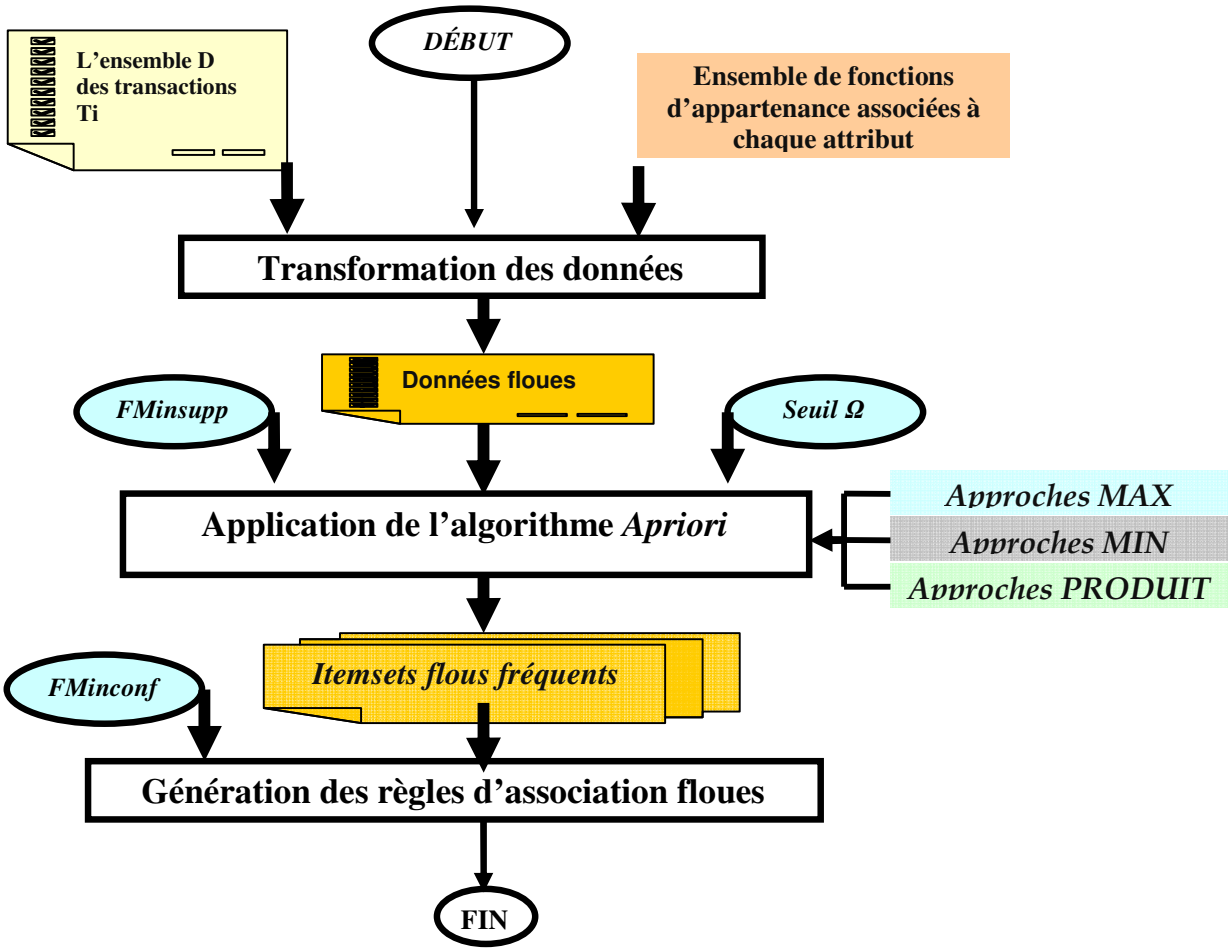


FIG. 28 ARCHITECTURE DE L'APROCHE PROPOSE

IV.5.2. l'algorithme *Apriori* aux données floues

Input : Une table de transactions T contenant M attributs ; F ensemble de fonctions d'appartenance associées à chaque attribut quantitatif, *Fminsup, Fminconf* et un seuil Ω des degrés d'appartenances

Output : LF : *Itemsets flous fréquents*, ER : *ensemble de règles d'association floues*

- ***Etape 1*** **:** Cette étape sert à transformer la table de transaction T en T'. Elle consiste à étendre chaque attribut quantitatif X en plusieurs Attributs (X, F_i) où F_i est un sous-ensemble flou associé à l'attribut X en affectant le degré d'appartenance correspondant.

- **Pseudo-code de l'algorithme de transformation**

  - ***Etape1.1 création de la table T'***

CREATE TABLE *T'* ($f_{i1}^{1}, f_{i1}^{2},..., f_{i1}^{p},....f_{ik}^{1}....f_{ik}^{m}$)

Où $f_{i1}^{1}, f_{i1}^{2},..., f_{i1}^{p},....f_{ik}^{1}....f_{ik}^{m}$ représentent les champs des sous-ensemble flous relativement aux Items $(I_1, I_2,..., I_k)$. Ces partitions sont donnée par un expert à l'aide des fonctions d'appartenances, afin que ces champs prendre leurs valeurs dans l'intervalle [0, 1].

  - ***Etape 1.2 Remplissement de la table T'***

Pour chaque transaction t ∈ T ***faire***

Pour chaque colons (f_{ik}^{m}) de T' ***faire***

f_{ik}^{m} .value = degree^{k} (I_k) { degre^{k} (I_k) est fonction de transfert des données quantitatives ou qualitatives en degrés d'appartenance }

Fin pour

Fin pour

- ***Etape 2*** **:** génération des Itemsets flous fréquents. Cette étape suit la démarche qu'Apriori sauf dans la manière du calcul du support. Ce calcul sera réalisé par les stratégies (*Min*, *Max* ou *Produit* des degrés d'appartenance des Items constituant l'Itemset), vue en section (IV.3), la figure (FIG 29) montre l'organigramme de génération des Itemsets flous fréquent.

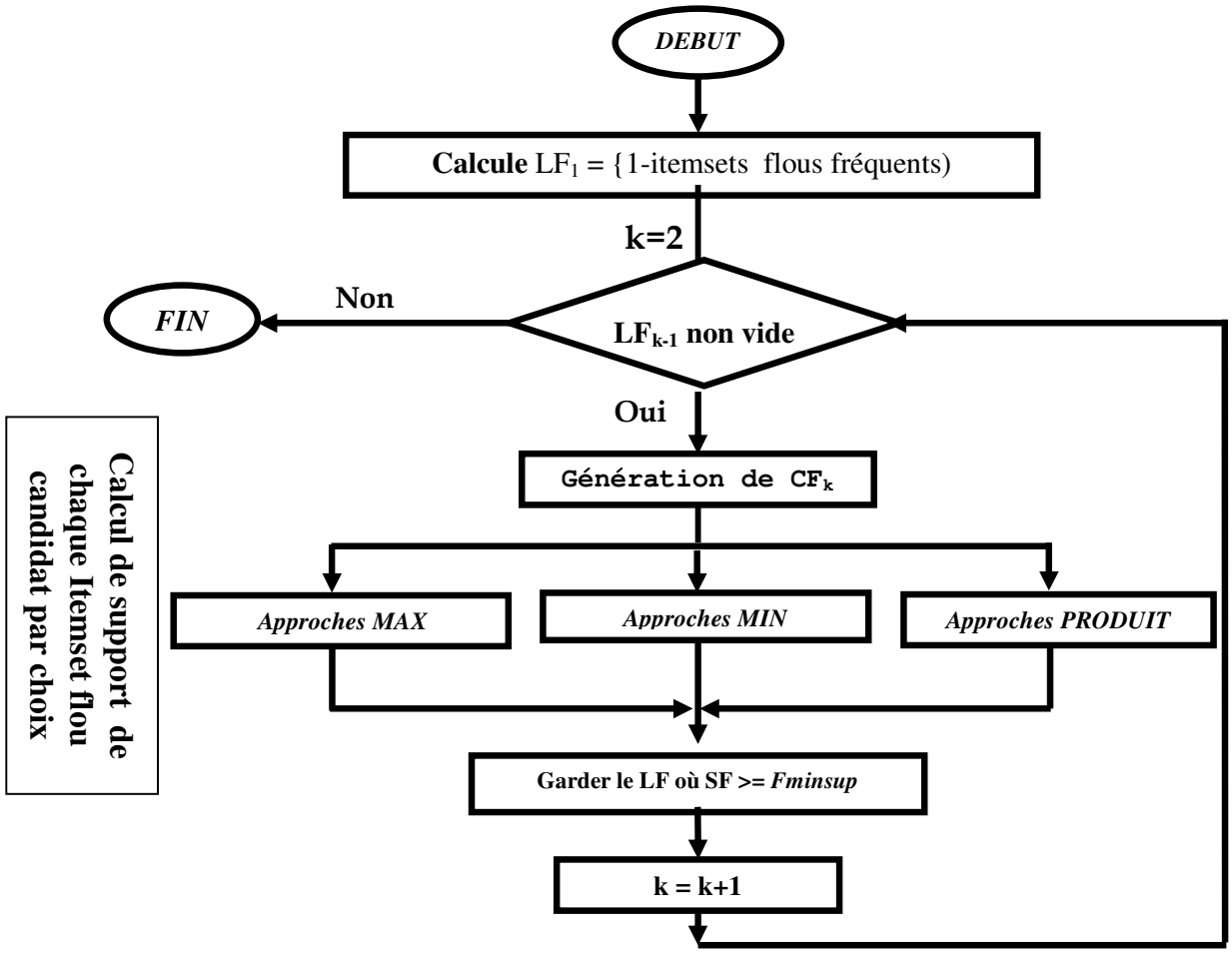


FIG. 29 L'ORGANIGRAMME DE GENERATION DES ITEMSETS FLOUS FREQUENT.

- **Pseudo-code de l'algorithme**

L'algorithme reçoit l'ensemble **T'** des transactions et un seuil minimum ***Fminsup*** de support donné par l'utilisateur et à la fin du processus, on obtient L'ensemble **L** de tous les itemsets fréquents.

Algorithme APRIORIFLOU

Input **: T' :** l'ensemble des transactions

Fminsup : seuil minimum de support

Ω : seuil des degrés d'appartenance

Fs : variable pour le support d'un Itemset flou

Output **: LF :** les Itemsets flous fréquents.

Algorithme

1 LF_1 = {1-itemsets Flous fréquents)

2 k=2 ;

3 ***Tant que*** LF_{k-1} non vide ***faire***

4 CF_k =Aprioriflou-Gen (LF_k) ;

5 ***Pour chaque*** t de T' ***faire***

6 CF_t = **Subset** (CF_k , t) ; {les candidats contenus dans CF_k }

7 Fs =0 ;

8 ***Pour chaque*** cf de CF_t ***faire***

9 $Fs = Fs + \perp(\mu(a_j)(t_i[x_j]))$ ou $j = 1..k$ {calcule de support flou par l'approche min des de grés d'appartenance et pour les autre approche juste changé la formule de calcul}

10 ***Fin pour***

11 ***Fin pour***

12 LF_k = {cf de CF_t / Fs >= F$minsup$};

13 k++ ;

14 ***Fin du tant que***

Return $\cup$ LF_{k-1} ;

ALG. Aprioriflou-Gen

Insert Into CF_k

Select p.item$_1$, p.item$_2$; . . ., p.item$_{k-1}$, q.item $_{k-1}$

From LF_{k-1} p, LF_{k-1} q

Where p.item$_1$ = q.item$_1$; . . . ; p.item$_{k-2}$ = q.item $_{k-2}$q ; p.item$_{k-1}$ < q.item $_{k-1}$;

Pour chaque **itemset flous** cf de CF_k ***faire***

Pour chaque s = **Subset** (k-1) de cf ***faire***

Si s $\notin$ LF_{k-1} ***alors***

Supprime cf de CF_k

Fin pour

Fin pour

Return CF_k

- ***Etape 3*** : sert à générer des règles d'association floues

Début

ER = { } ; // ensemble vide

Pour chaque Itemset flous $I \in LF_k$ où $k > 1$ ***faire***

Pour i= 1 à k ***faire***

$I = LH_{k-i} \cup LH_i$; $FConf = \frac{F\sup(LH_k)}{F\sup(LH_{k-i})}$

Si *Fconf* > *Fminconf* ***alors***

Ajouter ($LH_{k-i} \rightarrow LH_i$) à **ER**

Fin pour

Fin pour

FIN

IV. 5.3.Application

Dans cette section nous présentons de l'application de notre approche sur une base de test. La table (TAB.16) représente un ensemble de Dix-neuf transactions, qui contiennent six Items: Chocolat***,*** Pain***,*** Lait***,*** Fromage***,*** Chips ***et*** Saucisson***.***

Ces transactions décrivent la quantité achetée pour chaque produit. Les cases vides correspondent à une quantité achetée nulle.

Ti	Chocolat	Pain	Lait	Fromage	Chips	Saucisson
T1	2					
T2	1	3	1			
T3	2		1			
T4	3	2	3	4		
T5						2
T6	2			1		
T7			2			
T8		4	1			
T9	3		1		5	
T10			1		2	3
T11	3	1				
T12				4		5
T13			2			
T14		2				
T15					2	
T16	2					4
T17			3			
T18		2				
T19			2			

TAB.16 TABLE DE TRANSACTION DE DONNEES QUANTITATIVES [FIOT 2004]

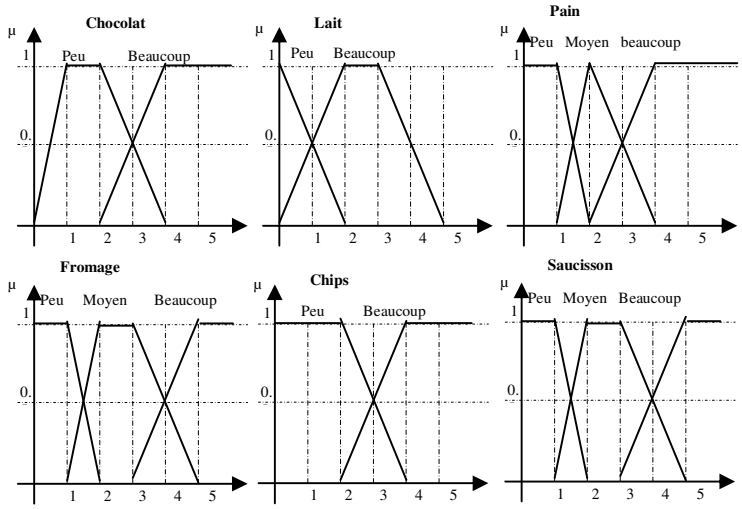


FIG.30. PARTITIONNEMENT FLOU DES ATTRIBUTS QUANTITATIFS [FIOT 2004]

Les partitions floues sont illustrées par la figure (FIG.30). Ces partitions conserne les fonctions des degrés d'appartenance, qui correspondent respectivement aux attributs quantitatifs de la table (TAB.15). On utilisant les abréviations suivantes : ***B*** pour beaucoup, ***P*** pour peu et ***M*** pour moyenne. Dans cette partition l'attribut *«Pain »* par exemple est divisé en trois parties, *« Peu de pain »*, *« Moyen de pain »* et *« Beaucoup de pain »*. Un achat de 1 pain par exemple est considéré comme appartenant à *« Peu de pain »*, de 2 comme *« moyen »* 3 pains est à la fois une quantité *moyenne* et une grande quantité et à partir de 4 pains, l'achat comporte *« beaucoup de pain »*.

1. L'étape 1 sert à définir la table T', à partir T, en utilisant les fonctions d'appartenance ci-dessus. La table (TAB.17) représente les degrés d'appartenance des sous-ensembles flous de chaque attribut quantitatifs.

2. Dans la deuxième étape, par exemple si on fixe les seuils ($\omega = 0{,}49$, *Fsupp* =1,5) on obtient les résultats ci-dessous (voir table TAB.18) : les Itemsets flous fréquents (TAB.18 (a) et (b)). Ces résultas sont relatives aux mesures basées sur les stratégies, *Max, Min et le produit* des degrés d'appartenances des Items constituent L'Itemset

T'_i	chocolat		Pain			Lait		Fromage			Chips		Saucisson		
	P	B	P	M	B	P	B	P	M	B	P	B	P	M	B
1	0,75	0,25				0,50	0,50								
2	1,00			0,50	0,50	0,50	0,50								
3	0,25	0,75				0,50	0,50								
4	0,50	0,50			1,00		1,00			1,00					
5							1,00							1,00	
6	0,25	0,75						1,00							
7							1,00								
8					1,00	0,50	0,50								
9	0,50	0,50				0,50	0,50					1,00			
10											1,00			1,00	
11	0,50	0,50	1,00			1,00									
12										1,00					1,00
13							1,00								
14				1,00											
15											1,00				
16	0,75	0,25						1,00						0,50	0,50
17							1,00								
18				1,00											
19							1,00								

TAB.17 TABLE DE TRANSACTION T'

1-itemset flous	
Items flous	***Fsupp***
(Chocolat, Peu)	4
(Chocolat, Beaucoup)	2,5
(Pain, Moyen)	2,5
(Pain, Beaucoup)	2,5
(Lait, Peu)	3,5
(Lait, Beaucoup)	8,5
Fromage, Beaucoup)	2
(Chips Peu)	2
(Saucisson, Moyen)	2,5
(Saucisson, Beaucoup)	1,5

TAB.18 (A) 1-ITEMSET FLOUS

2-itemsets flous		MAX des degrés d'appartenances	MIN des degrés d'appartenances	Produit des degrés d'appartenances
Itemsets flous		*Fsupp*	*Fsupp*	*Fsupp*
(Chocolat, Peu)	(Pain, Beaucoup)	2		
(Chocolat, Peu)	(Lait, Peu)	3,25	2	1,75
(Chocolat, Peu)	(Lait, Beaucoup)	3,25	2	1,75
(Chocolat, Beaucoup)	(Lait, Peu)	2,25	1,5	
(Chocolat, Beaucoup)	(Lait, Beaucoup)	2,25	1,5	
(Pain, Beaucoup)	(Lait, Peu)	1,5		
(Pain, Beaucoup)	(Lait, Beaucoup)	2,5	2	1,75

TAB.18 (B) 2-ITEMSETS FLOUS SELON LES CHOIX DU METHODE DE CALCULE DE SUPPORT

3. L'étape 3 permet de générer les règles d'association floues. Il est nécessaire de fixer un seuil minimal de confiance, par exemple (*Fconf* =50%), alors l'algorithme donne l'ensemble (ER) des règles d'association floues, qui sont illustrées en table (TAB.19)

règles d'associations floues	***Basé sur MAX***		***Basé sur MIN***		***Basé sur Produit***	
	Fsupp	***Fconf***	***Fsupp***	***Fconf***	***Fsupp***	***Fconf***
(Chocolat, Peu) → (Pain, Beaucoup)	10.53%	50%				
(Pain, Beaucoup) →(Chocolat, Peu)	17.11%	80%				
(Chocolat, Peu) → (Lait, Peu)	17.11%	81,25%	10.53%	50%		
(Lait, Peu)→ (Chocolat, Peu)	17.11%	92,86%	10.53%	57%	9,21%	50%
(Chocolat, Peu) → (Lait, Beaucoup)	11,84%	81,25%	10.53%	50%		
(Chocolat, Beaucoup) → (Lait, Peu)	11,84%	90%	7,89%	60%		
(Lait, Peu)→ (Chocolat, Beaucoup)	11,84%	64%				
(Chocolat, Beaucoup) (Lait, Beaucoup)	11,84%	90%	7,89%	60%		
(Pain, Beaucoup) →(Lait, Peu)	7,89%	60%				
(Pain, Beaucoup) →(Lait, Beaucoup)	13,16%	100%	10.53%	80%	9,21	70%

TAB.19 LES REGLES D'ASSOCIATIONS FLOUES PAR LES TROIS APPROCHE

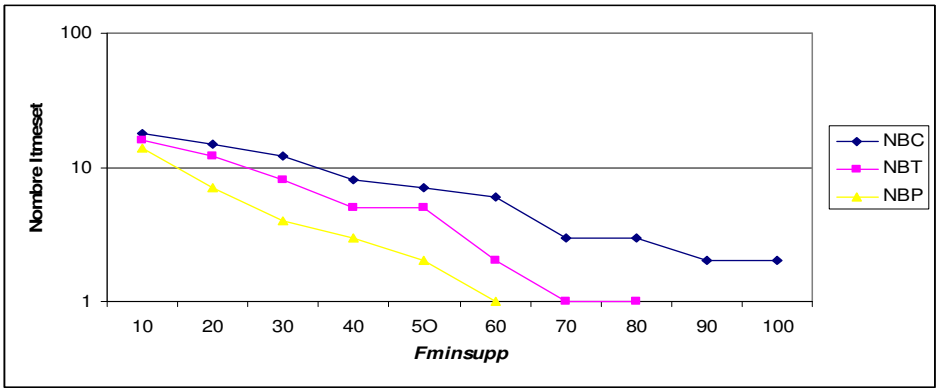
IV.5.4 Discussion des résultats

On utilise ces notations :

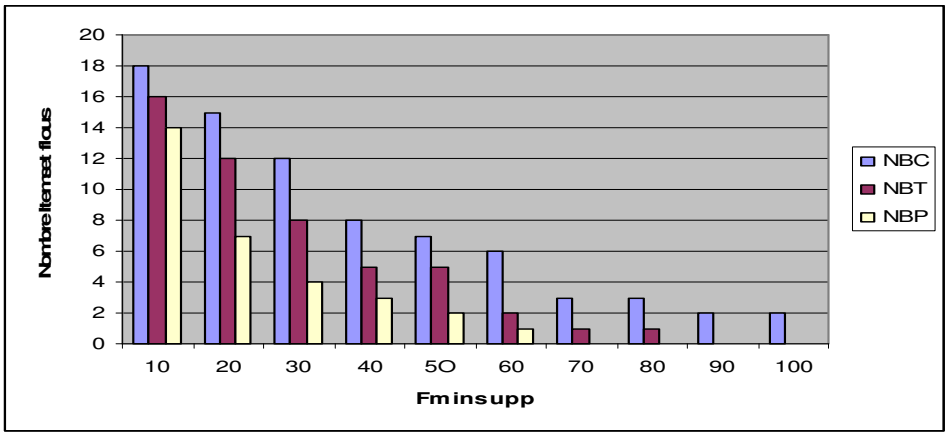
- Le (NBC) pour le nombre d'Itemsets flous ou le nombre des règles d'association floues générées par la mesure *t-conorme*, (le *Max* des degrés d'appartenances).
- Le (NBT) pour le nombre d'Itemsets flous ou le nombre de règles générées par la mesure *t-norme*, (le *Min* des degrés d'appartenances).

- *Le* (NBP) pour le nombre d'Itemsets flous ou le nombre de règles d'association floues générées par la mesure *Produit*, (le *Produit* des degrés d'appartenances).

La figure (FIG.31(A),FIG.31(B)) montre le nombre d'Itemsets flous fréquent en fonction de *Minsup.* D'après cette figure, il est clair que les mesures basées sur *le Min et le produit*, diminuent le nombre d'Itemsets flous fréquent, ce qui peut causer la perte de génération de règles parfois intéressantes.



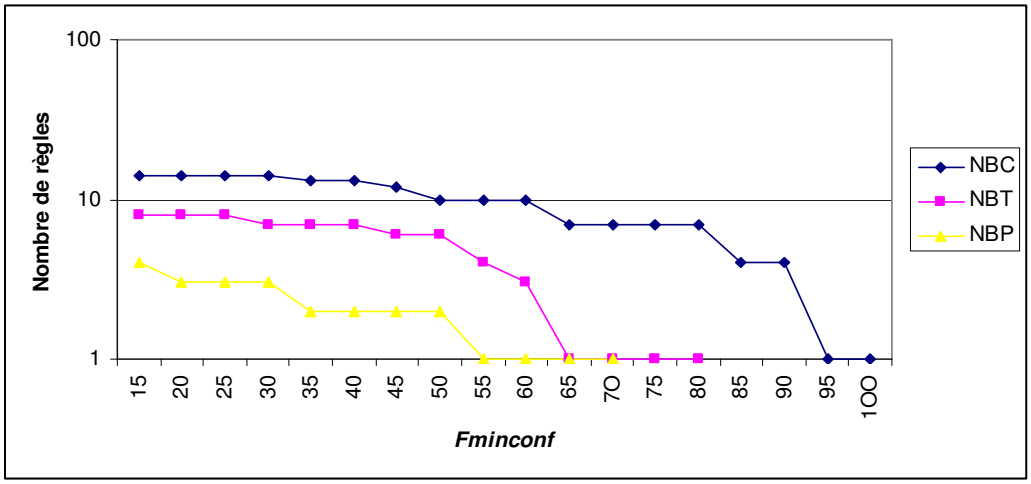
(A) COURBES DEFINIES LES ITEMSETS FLOUS FREQUENTS EN FONCTIONS DE SEUIL DU SUPPORT



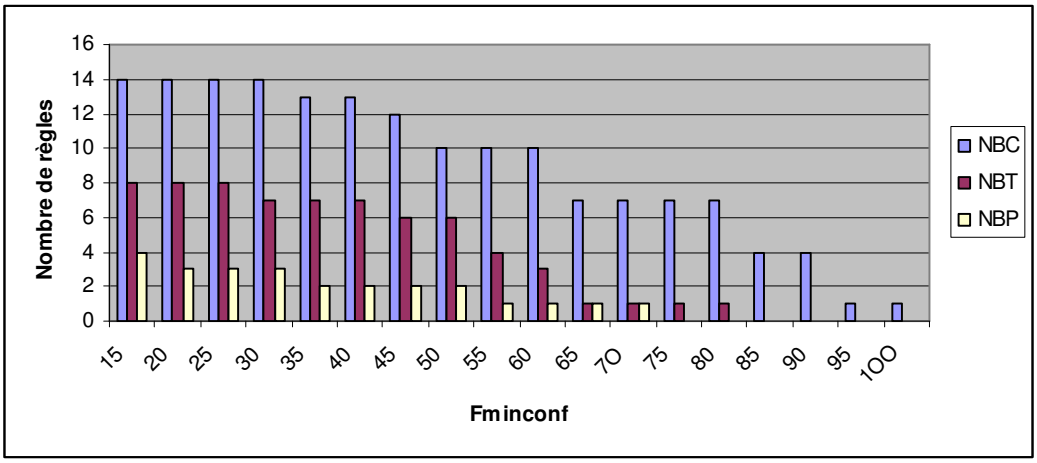
(B) HISTOGRAMME DEFINIS LE NOMBRE D'ITEMSETS FLOUS FREQUENTS

FIG. 31

Les figures (FIG.32 (A),FIG.32(B)), montre le nombre de règles par rapport aux seuils, *Fminconf* correspond aux différentes mesures du support.



(A) COURBES DEFINIES LES REGLES D'ASSOCIATION FLOUES EN FONCTIONS DE SEUIL DU SUPPORT



(B) HISTOGRAMME DEFINIS LE NOMBRE D'ITEMSETS FLOUS FREQUENTS

FIG.32

D'après les résultats obtenus à partir de notre approche, la mesure basée sur le max du degré d'appartenance (*t-conorme*) nous semble efficace par rapport aux deux autres mesures *t-norme* et Produit. De ce fait, dans cet exemple, la mesure (*t-conorme*) génère des règles d'association floues sur le ***lait*** et le ***chocolat***, avec les seuils choisis d'où la règle :

(***Lait***, ***Peu***)→ (***Chocolat***, ***Beaucoup***) [11,84%,64%]

Par contre les mesures basées sur le *Min* et le *Produit* perdent des règles qui peuvent être utiles.

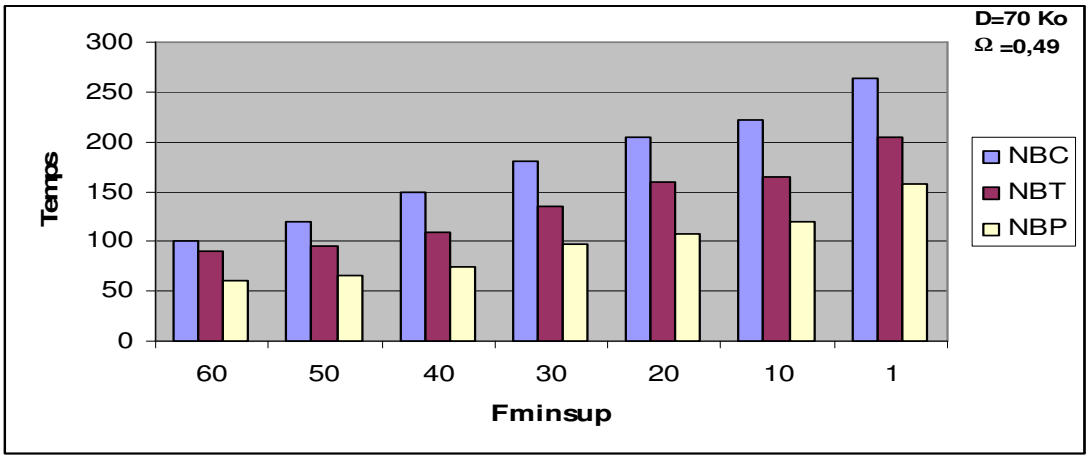
IV.5.5 Performance de l'algorithme

Dans cette section, nous élaborons une série de tests pour valider l'efficacité de notre approche (en terme de temps d'exécution), et de mesurer ses performances par rapport à d'autres approches (calcul du support d'un Item floue que nous avons déjà expliqué dans les sections précédentes). Ces test sont fait par rapport aux valeurs des données en entrée : la taille de la base de données D, ainsi que les seuils minimums de support *Fminsup* et le seuil Ω des degrés d'appartenances.

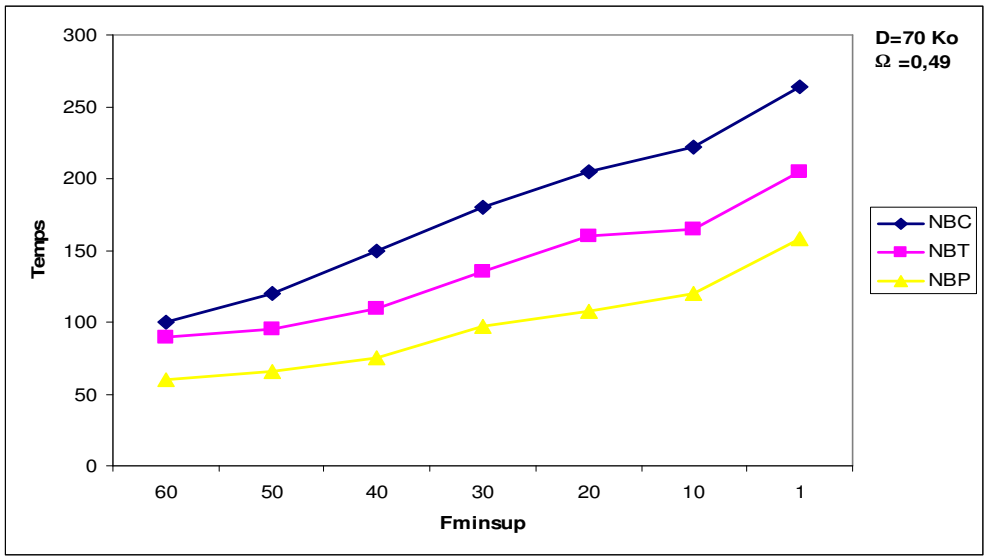
Ces expériences sont effectuées sous la plateforme Windows, sur un PC équipé d'un processeur Pentium IV à 3.1 GHz avec 1Go de mémoire DDR2. L'implémentation de notre algorithme APRIORI_F a été faite avec les trois approches de calcul du support d'un Item floue, il faut bien noter que nous avons utilisé la même structure (données et traitement, tant mieux pour l'objectivité de la comparaison).

On outre la base de données que nous avons utilisée est la même base qui a été utilisée par *Fiot et al.* [FIOT 2004]. Aussi nous avons fait des modifications (Ajout et suppression) dans la taille de la base utilisée pour des raisons du test de comparaison, également nous avons utilisé les tailles suivantes (70Ko et 110Ko).

Cependant, le seuil Ω des degrés d'appartenances est fixé à 0.49 car un degré d'appartenance d'un item flou x dépasse la moitie par conséquent c'est raisonnable de dire que l'item x à la propriété a c'est-à-dire (x est a). À chaque test on fixe le seuil de support flou plus précisément nous avons utilisé les seuils (60%, 50%, 40%, 30%, 20%, 10%) pour les trois approches cité en haut ; les résultats obtenus sont illustrés dans les figure (FIG.33, FIG.34)



(A)



(B)

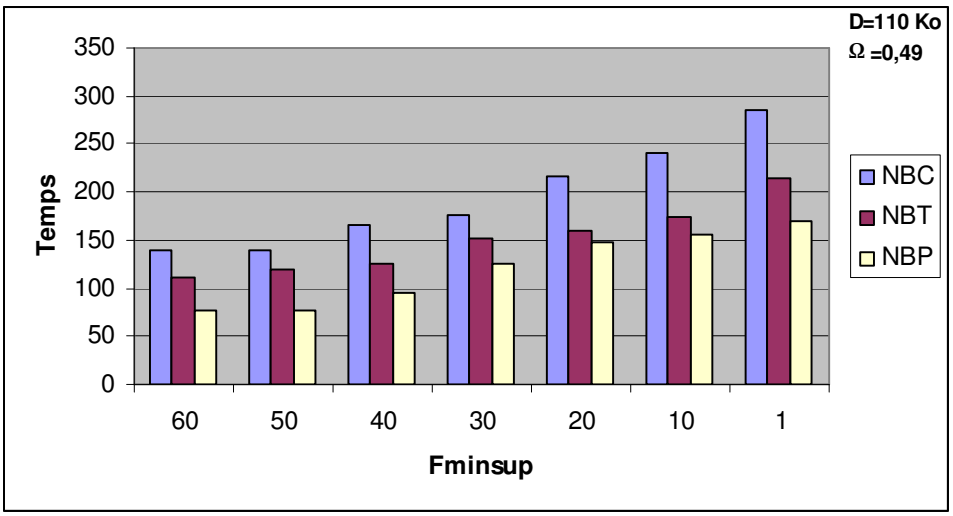
FIG.33 TEMPS DE REPONSE PAR APPORT AU SEUIL MINIMUM DE SUPPORT (TAILLE DB 70KO)

Dans la première expérience nous avons appliqué notre algorithme sur une base de données dont la taille est de 70K, en suivant les démarches de l'expérience cité au-dessous, puis nous avons procédé aux différents choix du seuil *Fminsup*

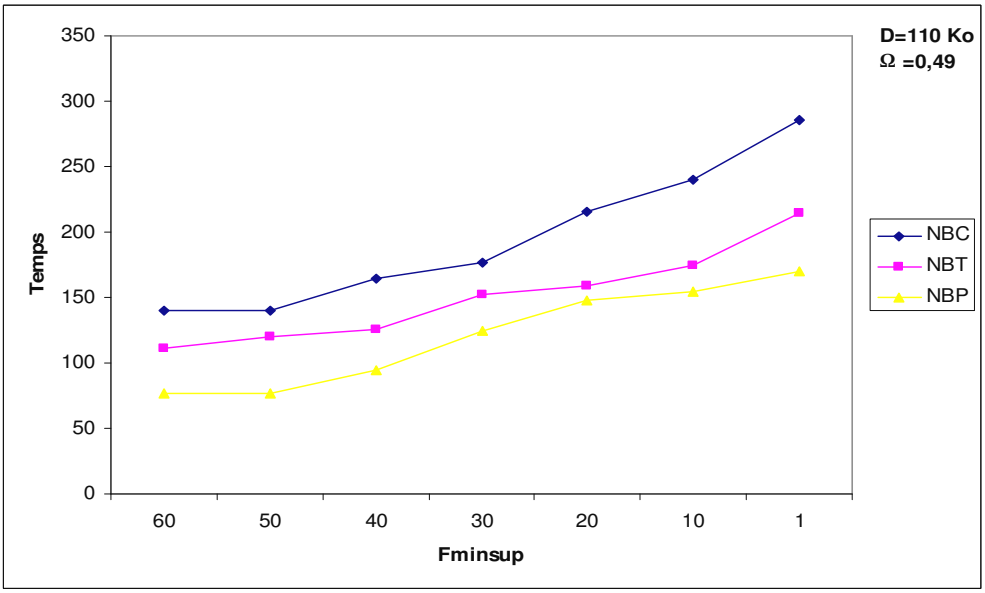
La figure (FIG.31(A) ET (B)) illustre le temps d'exécution en fonction de *Fminsup*. D'après cette figure, il est clair que les mesures basées sur *le Min et le produit*, donnent des résultats beaucoup plus performants et cela est clairement remarqué à travers le temps de calcul qui est largement inférieur au temps de l'approche basée sur le *Max* des degrés d'appartenance.

Nous remarquons que lorsque que nous diminuant le seuil *Fminsup* le temps d'exécution augmente, ce qui explique le fait que quand le seuil est plus bas conduit forcément à l'augmentation dans le nombre *d'itemset* flous fréquents généré. En effet, lorsque le *Fminsup* est faible le nombre *d'itmesets* flous fréquents est grand et l'algorithme devient encore plus lord et vice-versa.

Nous pouvons remarquer également, que les trois courbes possèdent la même allure et ceci s'explique par le fait que nous avons appliqué le même algorithme où la seule modification est dans l'instruction de calcul de support d'un Itemset flou.



(A)



(B)

FIG.34 TEMPS DE REPONSE PAR APPORT AU SEUIL MINIMUM DE SUPPORT (TAILLE DB 110KO)

Nous avons aussi testé les performances de notre algorithme par rapport à la taille de la base de données et nous avons exécuté l'algorithme pour une base de taille 110Ko et nous avons gardé le seuil Ω égal à 0.49 et où seul le seuil *Fminsup* est autorisé à changer. Les figures (FIG.34(A)) et (FIG.34(B) illustrent les résultats de cette expérience. Nous remarquons que trois courbes ont la même allure et lorsque *Fminsup* est grand (tend vers d>60%), le temps

d'exécution est plus bas. Ceci dit, plus *Fminsup* est petit, plus le temps d'exécution est rapide car dans ce cas, le cardinal d'un *Itemset* flous et petit.

Nous remarquons également que le temps d'exécution de l'approche basé sur le *produit* et inférieur à celle de l'approche basé sur le *Min* et cette dernière est inférieur à celle de *Max.* Ce qui montre que l'approche basée sur le *Max* fait plus de parcourt de la base de donnée que les approches basées sur le *Min* et le *Produit*

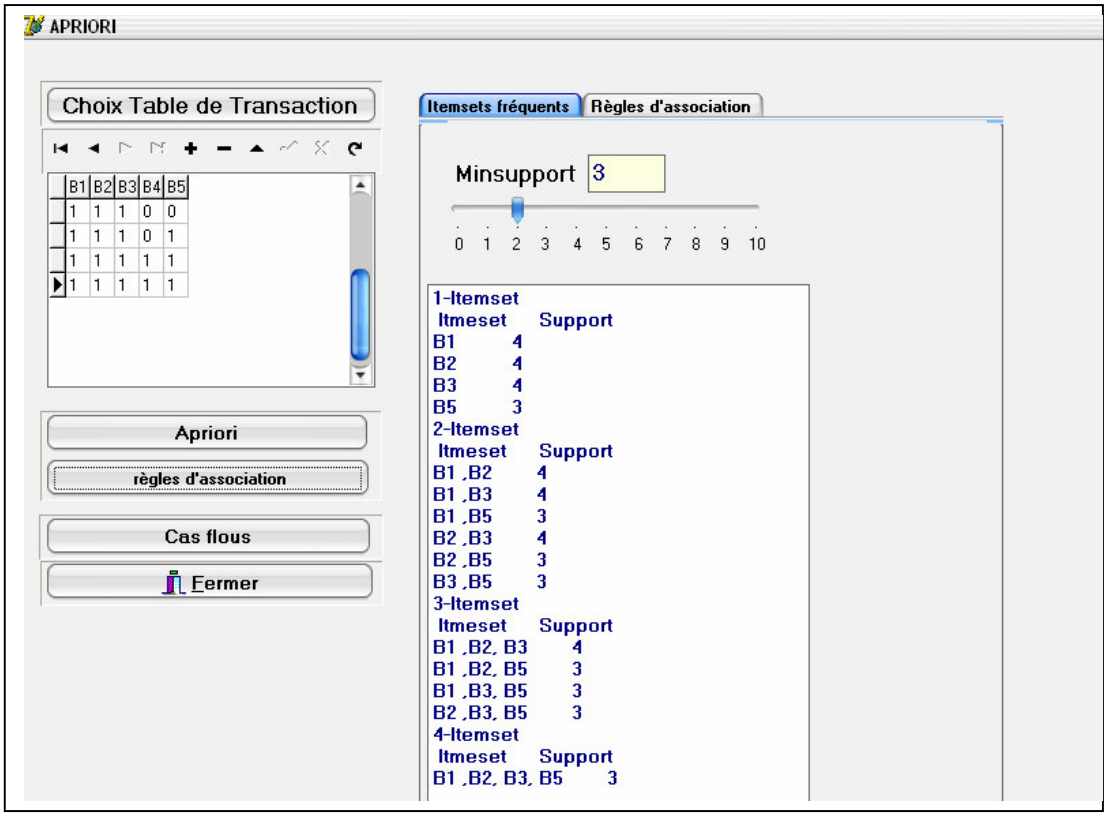
Finalement, nous avons implémenté notre algorithme pour l'extraction des règles d'association floues et nous l'avons testé. Les résultats obtenus étaient fort acceptables.

IV.5.6 Les différentes Interface de logéciel réalisé

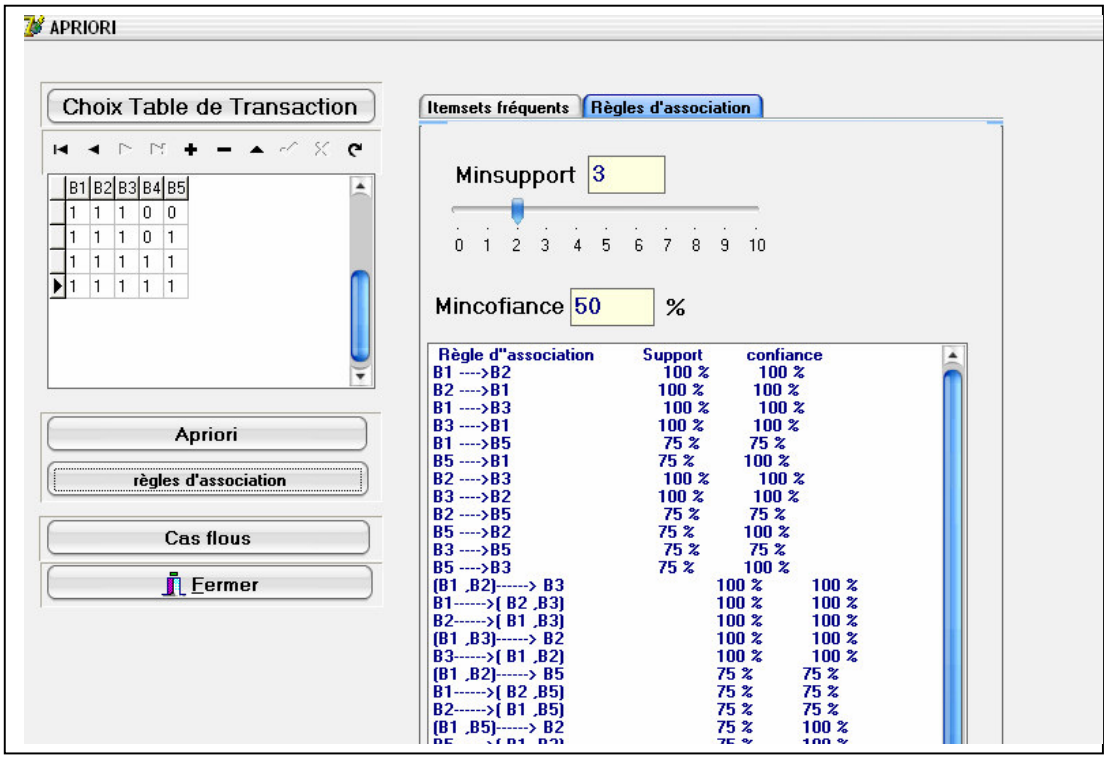
Notre première interface est illustrée par la figure (FIG.35 (A) ET (B)) ce dernier concerne les règles d'association binaire, il contient cinq options :

- ***Choix table de transaction*** *:* pour ouvrir une table de transaction ou nous voulons extraire des règles d'association
- ***Apriori*** : pour générer les Itemset fréquents par *Apriori*
- ***Règles d'associations*** : pour générer les règles d'association par *Apriori*
- ***Cas flous*** : ce buttons vous à conduit à une autre fenêtre (FIG.36 (A) ET (B)) où il y a l'implémentation de notre approche.
- ***Fermer*** : buttons pour quitter l'application

Aussi à gauche il y deux ongles l'un pour afficher les Itemset fréquents extraite par *Apriori* et l'autre pour visualiser les règles d'association généré aussi par Apriori, ainsi il y a des zones pour saisie les seuils minimum de support et la confiance



(A) LES ITEMSET FREQUENT PAR *APRIORI*

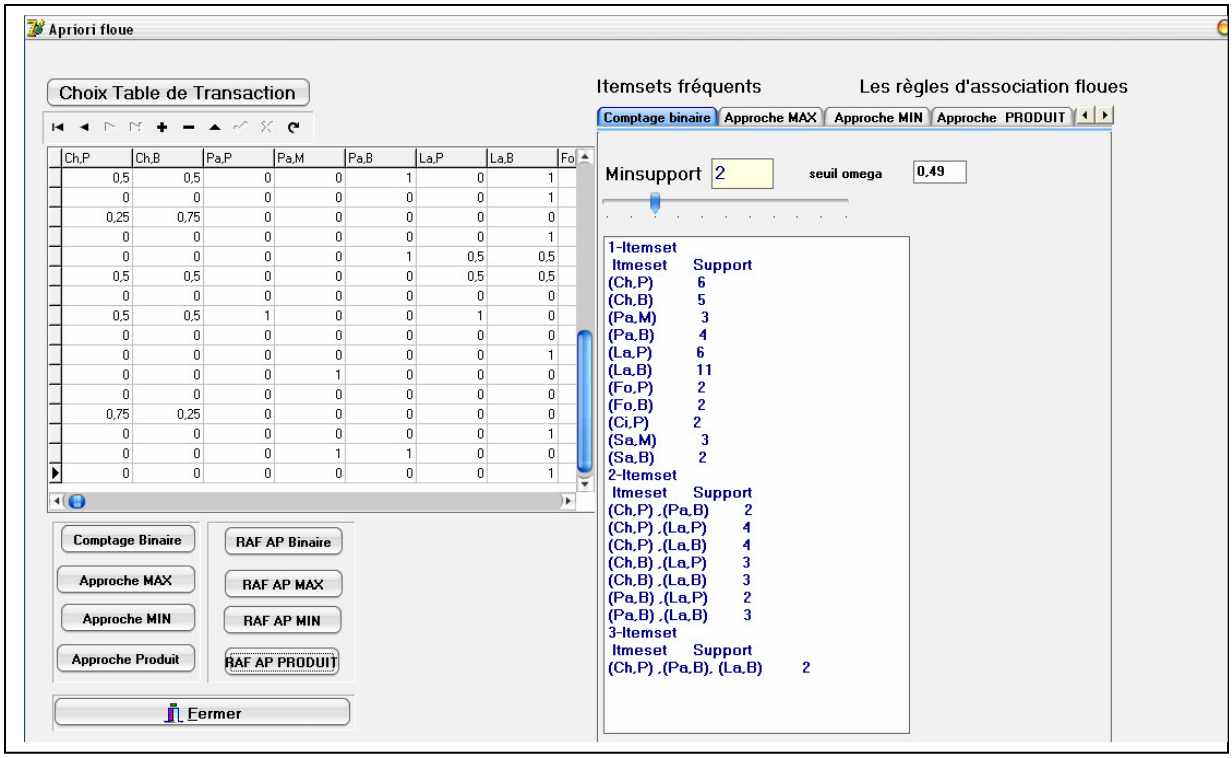


(B) LES REGLES D'ASSOCIATION PAR *APRIORI*

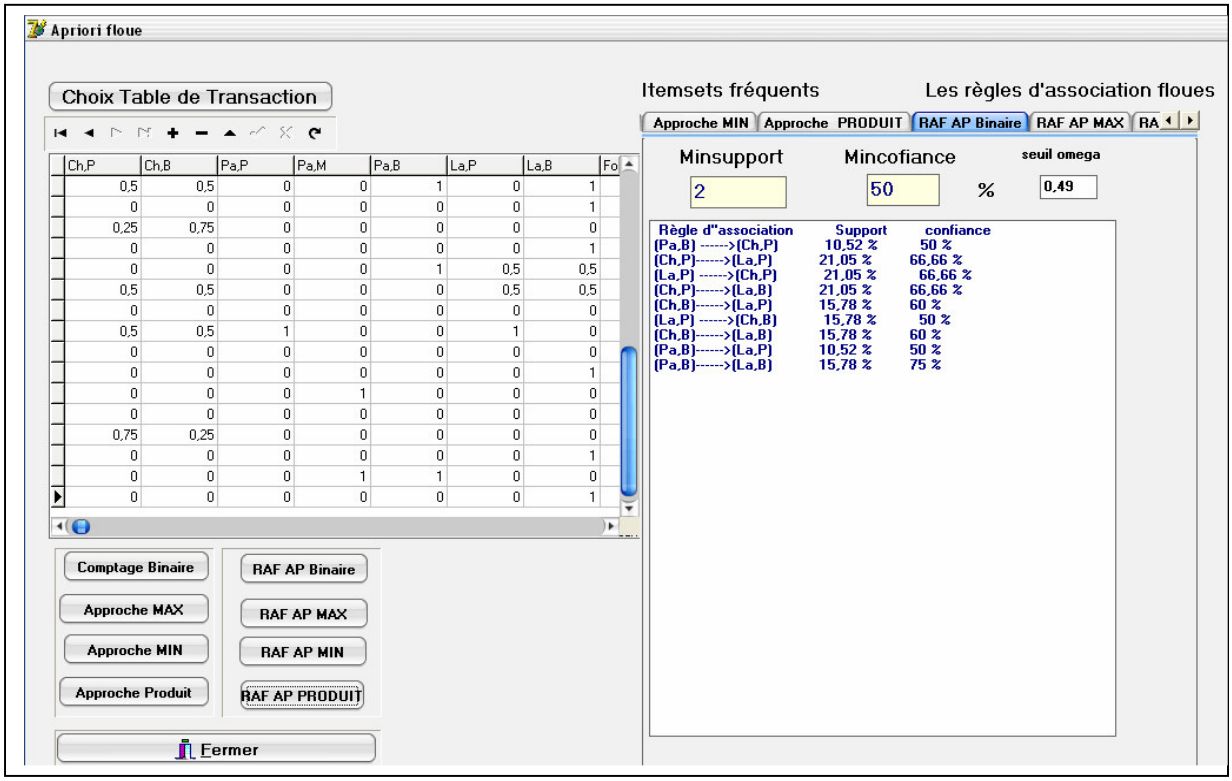
FIG.35

Cependant, le cas flous est illustré par la figure (FIG.36 (A),(B)), où nous avons gardé la même forme de visualisation. Il y a des zones où on peu saisie les seuils *Fminsup, Fminconf,* le seuil Ω des degrés d'appartenances, dans cette fenêtre nous avons utiliser l'abréviation (**RAF**) qui signifie (**R**ègles d'**A**ssociation **F**loues) ,et (AP) c'est (Approche) ainsi dans cette fenêtre on trouve tout les approches étudie auparavant sont les suivant :

- ***Approche Max***
- ***Approche Min***
- ***Approche Produit***



(A) LES ITEMSET FREQUENT PAR *APRIORI FLOUE CAS BINAIRE*



(B)LES REGLES D'ASSOCIATION PAR *APRIORI FLOUE CAS BINAIRE*

FIG.36

IV. 6 Conclusion

L'idée de la recherche des règles d'association floues est très intéressante. Bien que, plusieurs travaux différents aient été effectués sur des règles d'association floues, en utilisant des différentes méthodes et approches, les techniques traitant ce type de connaissances n'ont pas encore suffisamment évoluées.

La détermination de mesures d'intérêts adéquats pour des règles d'association floues est un sujet important car ces mesures permettent de découvrir de nouvelles règles d'association floues qui soient utiles et valides. Dans ce chapitre, nous avons présenté un état de l'art sur les règles d'association floues, notamment les différentes approches des règles d'association floues : *l'approche quantitative*, *l'approche structures taxonomiques floues,* ainsi que *la recherche des motifs séquentiels flous.* Aussi, nous avons exposé notre approche qui consiste en une extension de l'algorithme *Apriori* visant à extraire des règles d'associations floues. Nous avons comparé, à cet effet, notre approche à trois autres approches pour calculer le support d'un itemset flou. Les résultats obtenus montrent que la mesure basée sur le *Max* des degrés d'appartenance d'un itemset flou est efficace et meilleure que les mesures basées sur le *Min et le Produit*, qui peuvent parfois ignorer des règles parfois très intéressantes.

Conclusion générale

Conclusion

L'objectif de ce travail est l'extraction des connaissances floues par le biais des règles d'associations floues par extension de l'algorithme *Apriori*. Pour cela, nous avons consacré le premier chapitre pour l'extraction des connaissances à partir de données (ECD) et nous avons donné une brève description de ses méthodes qui sont relatives à l'analyse des données. Nous avons évoqué dans le même chapitre les notions de base de la fouille de données ainsi que le processus d'ECD. Dans le deuxième et le quatrième chapitre, nous nous sommes rapprochés encore plus de notre sujet de recherche en présentant l'état de l'art d'extraction des règles d'associations dans les deux cadres, (*binaire*, *floue*).

En conclusion, les règles d'association préoccupent les chercheurs depuis leurs émergences. Beaucoup de travaux ont été effectués sur les règles d'association. En effet, plusieurs méthodes et différentes approches ont été utilisées. Dans ce mémoire, nous présenté les approches existant dans les littératures, ainsi que la signification de terme *règles d'association*. En générale on peu dire qu'une règle d'association est un type de connaissance extrait d'une base de données. Nous pouvons les classer en deux grandes catégories :

1. Les règles d'association classiques qui comprend :
    - *Approche binaire*
    - *Approche quantitative*
    - *Approche des règles d'associations généralisées*
    - *Approche séquentiels*
2. Les règles d'association floues qui comprend :
    - *Approche quantitative*
    - *Approche structure taxonomique floue*
    - *Approche des motifs séquentiels flous*

En outre, nous avons défini dans ce mémoire une nouvelle approche qui consiste à une extension de l'algorithme *Apriori.* Cette approche a pour but d'extraire des règles d'associations floues. En effet, nous nous sommes concentré sur les règles d'association floues de la forme: $(X, A) \rightarrow (\mathrm{Y}, \mathrm{B})$, où X et Y sont des *Itemsets* non vides et $X \cap Y = \emptyset$,

A et B sont des sous-ensemble flous associes à X, Y respectivement. Ainsi, nous avons utilisé trois différentes approches de calcul de support d'un *Itemset* flous. Chacune de ces approches donne des règles différentes des autres. Dans la réalité, nous ne pouvons pas prédire la meilleure approche, nous aurons toujours recours à un expert. Cependant, nous pouvons affirmer que l'approche basée sur le *Max* des degrés d'appartenance, est plus efficace par rapport aux approches basées sur le *Min* et le *Produit*.

L'algorithme que nous avons présenté se base sur *Apriori*, il est alors efficient pour l'extraction des règles d'association floue, mais un peu lourde en terme de temps d'exécution.

Bien que, plusieurs travaux différents aient été effectués sur les règles d'association floues en utilisant différentes méthodes et approches, il reste néanmoins un domaine de recherche en pleine émergence car toutes les techniques développées jusqu'à lors n'ont pas suffisamment évoluées. A cet effet, nous proposons comme perspective de faire des extensions pour les algorithmes *Partition,* [SAVA 1995], *Eclat* [ZAKI 1997], et *FP-Growth* [HAN 2000], aux cas flous. Nous pensons, également, qu'il serait très judicieux de développer de nouvelles approches ayant pour but de construire des algorithmes de maintenance de règles d'association.

Références

[AGRA 1993] Agrawal R., Imielinski, T., Swami, A. « Mining Association Rules Between Sets of Items in Large Databases ». In Proceedings ACM SIGMOD International Conference on Management of Data, pp207, Washington DC, May1993.

[AGRA 1994] Agrawal, R., Srikant, R. « Fast Algorithms for Mining Association Rules ». In Proceedings of the 20th International Conference on Very Large Data Bases (VLDB'94), pp 487-499, Santiago, Chile, September 1994.

[AGRA 1995] Agrawal, R. et Srikant, R. 1995. « Mining sequential patterns ». In Proc. 1995 International Conference Data Engineering (ICDE'95),Taipei,Taiwan, Mars 1995.pp3-14.

AGRA 1996] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A.I. Verkamo, « Fast Discovery of Association Rules». In U.M. Fayyad, G. Piatetski-Shapiro, P. Smyth and R.Uthurusamy, editors, Advances in Knowledge Discovery and Data Mining, 307--328. AAAI Press, 1996

[BOUC 1995] B. Bouchon-Meunier. « *La logique floue et ses applications*. » Addison-Wesley, 1995.

[BOUC 2003] B. Bouchon-Meunier , C. Marsala, editors. « *Logique Floue, Principes, Aide à la Décision* ». Hermès-Lavoisier, 2003.

[BREI 1984] Breiman, J. Friedman, H., Olshen, R. A. and Stone, C. J. «Classification and Regression Trees». Wadsworth, Belmont, Ca., 1984.

[CLAU 1998]Claude Rosental, «Histoire de la logique floue. Une approche sociologique des pratiques de démonstration», Revue de Synthèse, vol. 4, 4, octobre-décembre 1998, pp. 575-602.

[CHEN 2002]Chen, G., Wei, Q.: «Fuzzy Association Rules and the Extended Mining Algorithms», Information Sciences, Vol. 147, (1-4) pp. 201–28 (2002)

[DEGR 2001] DeGraaf, Jeannette M.; Kosters, Walter A.; Witteman, Jeroen J.W.: « Interesting fuzzy association rules in quantitative databases ». Lecture Notes in Computer Science, Volume 2168, 2001.

[DUVA 2000] Duval Béatrice « ECD : Extraction de connaissances à partir des données (KDD : Knowledge Discovery In Databases) », Laboratoire d'Informatique Université d'Angers, 2000.

[FAYY 1996] Fayyad, U., Piatetsky-Shapiro, G., et Smyth, P. « From Data Mining to Knowledge Discovery: An Overview ». In Fayyad, U., Piatetsky-Shapiro, G., Amith, Smyth, P., and Uthurnsamy, R. (eds.), Advances in Knowledge Discovery and Data Mining, MIT Press, 1-36, Cambridge, 1996.

[FIOT 2004] C. Fiot, G. Dray, A. Laurent, and M. Teisseire. «A la recherche des motifs sequentiels flous». In Actes des Rencontres Francophones sur la Logique Floue et ses Applications (LFA 04), pages 131–138, Nantes, France, 2004.

[FU 1998] A. Fu, M. Wong, S. Sze, W. Wong, W. Yu. Finding fuzzy sets for the mining of fuzzy association rules for numerical attributes. the First International Symposium on Intelligent Data Engineering and Learning (IDEAL), 1998, pp. 263-268

[FRAW 1992] Frawley, William J.; Piatetsky-Shapiro, Gregory; Matheus, Christopher J.: Knowledge Discovery in Databases: an Overview. AAAI/MIT Press, 1992.

[GARD 2001] Gardarin Georges « Bases de données », Eyrolles, Paris, 2001

[GOET 03] Goethals, Bart « Survey on Frequent Pattern Mining ». Manuscrit (partie de these), 2003. Department of Computer Science University of Helsinki
http://www.adrem.ua.ac.be/bibrem/pubs/fpm_survey.pdf

[GOTT 2006] Gottwald, Siegfried: «Universes of Fuzzy Sets and Axiomatizations of Fuzzy Set Theory». Studia Logica v. 82, 2006.

[GYEN 2001]Gyenesei, A.: « A Fuzzy Approach for Mining Quantitative Association Rules», Acta Cybernetical, Vol. 15, (2) (2001) 305 – 320

[HART 1975] Hartigan, J. A. « Clustering algorithms ». New York: John Wiley & Sons,1975.

[HAJEK 1966]Hajek P., Havel et Chytil (1966), The GUHA method of automatic hypotheses determination, Computing, 1, pp. 293-308, 1966.

[HAJEK 1999]Hajek P. et Rauch J. (1999), Logics and statistics for association rules and beyond, Tutorial PKDD'99, Prague, 1999.

[HIPP 2000] Hipp, J., Güntzer U., Nakhaeizadeh, G. « Algorithms for Association Rule Mining. A General Survey and Comparaison ». In Proceedings, ACM SIGKDD 2000, Volume2, Issue 1, pp58-64, 2000.

[Han 2000] Han, J., Pei, J., Yin, Y. « Mining Frequent Patterns without Candidate Generation ». In Proceedings of the 2000 ACM-SIGMOD Int'l Conf. On Management of Data, Dallas, Texas, USA, May 2000.

[HAND 2001] Hand, D., Mannila, H., Smyth,P. " Principles of Data Mining", MIT Press, Cambridge, CA, 2001.

[IDRI 2003] A. Idri , « un modèle intelligent d'estimation des coût de développement de logiciels », thèse doctorat, université du Québec à Montréal, septembre 2003 (www.lrgl.uqam.ca_publications_pdf_805)

[Ion 2006] Ion IANCU, Mihai GABROVEANU, Adrian GIURCA Some Quality Measures for Fuzzy Association Rules, in Proc. of the 6th Int. Conf. on Artificial Intelligence and Digital Communications - AIDC 2006, Thessaloniki, Greece, pp. 26-36 (in collaboration with M. Gabroveanu, A. Giurca)

[JELE 1999] Jelena. Godjevac, « Idées nettes sur la logique floue » , presses polytechniques et universitaires romandes , lausanne,1999.

[JESU 2009]Jesús Alcalá-Fdez, Rafael Alcalá, María José Gacto, Francisco Herrera: Learning the membership function contexts for mining fuzzy association rules by using genetic algorithms. Fuzzy Sets and Systems 160(7): 905-921 (2009)

[JIAW 2000] Jiawei Han et Micheline Kamber «Data Mining concepts and techniques », Morgan Kaufmann Publishers, 2000.

[LUDO 2006]Ludovic Lebart, Alain Morineau, Marie Piron:« Statistiques exploratoire multidimensionnelle : visualisations et inférences en fouille de données »4eme Edition Dunod, Paris 2006

[KUOK 1998] Kuok, Chan Man; Fu, Ada; Wong, Man Hon: Mining Fuzzy Association Rules in Databases. SIGMOD Record Volume 27, 1998.

[LU 2000] Lu, H., Feng, L. et Han, J. 2000 Beyond Intra-Transaction Association Analysis : Mining Multi-Dimensional Inter-Transaction Association Rules. ACM Transaction on Information System. vol. 18 :4, pp. 423-454.

[LAVR 1999] N. Lavrac and P. Flach and B. Zupan (1999). « Rule Evaluation Measures : A Unifying View ». Dans S. Džeroski et P. Flach, editors, Ninth International Workshop on Inductive Logic Programming (ILP'99), volume 1634 of Lecture Notes in Artificial Intelligence, pages 174-185. Springer-Verlag.

[LALL 2004] S. Lallich and O. Teytaud (2004). «Évaluation et validation de l'intérêt des règles d'association ». RNTI-E-1, pp 193–217.

[LALL 2004] Stéphane Lallich & Olivier Teytaud. « Evaluation et validation de l'intérêt des règles D'association ». Dans : Revue des Nouvelles Technologies de l'Information, tome E-1, 2004, Pages 193..218.

[MEHM 2001]Mehmed Kantardzic, "Data Mining: Concepts, Models, Methods, and Algorithms" Paperback, IEEE Press/Wiley, 2001, xii + 345 pages

[MART 2006] Martine CADOT « Extraire et valider les relations complexes en sciences humaines : Statistiques, motifs et règles d'association » Thèse de doctorat de l'université de Franche-Comté – Besançon Présentée et soutenue publiquement le 12 décembre 2006

[Plas 2005] M. Plasse, N. Niang-Keita et G. Saporta. Utilisation conjointe des règles d'association et de la classification de variables . In 37 èmes Journées de Statistique, 2005. Pau, 6 au 10 juin . (ref. CEDRIC 831)

[MM2008] Muyeba, M., M. Sulaiman Khan and Coenen, F.P. "A Framework for Mining Fuzzy Association Rules from Composite Items.", In Chawla et al (Eds.), New Frontiers in Applied Data Mining, Springer LNAI5433, pp62-74. (Revised and updated version of Information Processing in Knowledge Discovery (ALSIP 2008), colocated with PAKDD 2008

[MANN 1995] Mannila, H., Toivonen, et Inkeri Verkamo, A. 1995. Discovering Frequent Episodes in Sequences. Proc. 1995 International Conference on Knowledge Discovery and Data Mining(KDD'95), Montreal, Canada, August 1995. pp. 210-215.

[MANN 1997] Mannila, Heiki; Toivonen, Hannu; Verkamo, A. Inkeri: Discovery of Frequent Episodes in Event Sequences. Data Mining and Knowledge Discovery, Volume 1, Issue 3, 1997.

[MART 2003] C, Martin; C, Chris; E. Kerre, Etienne.: "Fuzzy Association Rules: a Two-Sided Approach". Proceedings Int. Conf. on Fuzzy Information Processing - Theories and Applications, 2003.

[MESI 1999] Mesiar, Radko; Navara, Mirko: Diagonals of continuous triangular norms. Fuzzy Sets and Systems v.104, 1999.

[Pasq1999] Pasquier, N., Bastide, Y., Taouil, T., Lakhal, L. « Efficient Mining of Association Rules Using Closed Itemset Lattices ». Information Systems, 24(1):25–46, 1999.

[PAUL 2008] McNicholas, P. D., Murphy, T. B. and O'Regan, M. (2008), 'Standardising the lift of an association rule', *Computational Statistics and Data Analysis* 52(10), 4712-4721

[Pei 2000] Pei, J., Han, J., Mao, R. « CLOSET : An Efficient Algorithm for Mining Frequent Closed Itemsets ». In Proceedings, ACM SIGMOD Workshop DMKD'00, pp 21–30, Dallas (TX), USA, 2000.

[PIAT 1991] G. Piatetsky-Shapiro (1991). " Discovery, Analysis, and Presentation of Strong Rules". Dans Knowledge Discovery in Databases, pages 229 _ 248. AAAI Press / The MIT Press.

[QUIN 1986] Quinlan, J.R. « Induction on decision trees ». Machine Learning , vol. 1, pp. 81-106, 1986.

[RIAD 2007]Riadh Ben Messaoud « cours Data-Mining » Institut Universitaire de Technologie LumiµereLicence C.E.STAT Avril 2007
http://eric.univ-lyon2.fr/~rbenmessaoud/supports/datamining/data_mining.pdf.

[STEP 2005] Stéphane TUFFÉRY « Data Mining et Statistique Décisionnelle », Edition TTechnip, Paris 2005

[SAVA 1995] Savasere, A., Omiecinski, E., Navathe, S. « An Efficient Algorithm for MiningAssociation Rules in Large Databases ». In Proceedings of the 21th conference on VLDB (VLDB'95), Zurich, Switzerland, September 1995.

[SITE01] Université René Descartes Paris.5 « Cours de Bioinformatique » : http://www.dsi.univ-paris5.fr/bio2/biocours/intro.html

[SRIK 1996] Srikant, R. and Agrawal, R., Mining quantitative association rules in large relational tables. In Proceedings of the the 1996 ACM SIGMOD international conference on Management of data, 1996, 1-12

[SRIK1995] Srikant, R. Agrawal, R. (1995) «Mining Generalized Associations Rules", Proceedings of 21St INt'l conf. on Very Large Databases (VLDB'95), Zurich, Suisse

[THOM 2003]Thomas Daurel, thèse du doctorat en informatique «Représentations condensées d'Ensembles de Règles d'Association », Cette thèse a été préparée au Laboratoire d'InfoRmatique en Image et Systèmes d'information (FRE 2672 CNRS) de l'INSA de Lyon

[TWO 1999] Two Crows Corporation. «Intorduction to Data Mining and Knowledge Discovery». 3d ed. Potomac: Two Crows Corporation, 1999.

[VALE 2006]Valerie FIOLET Thèse du Doctorat ; « Algorithme distribues d'extraction de connaissances» UNIVERSITE DES SCIENCES ET TECHNOLOGIES DE LILE année 2006

[Vail 2005]Vaillant B., Meyer P., Prudhomme E., Lallich S., Lenca P., Bigaret S., Mesurer l'intérêt des règles d'association , pp. 69-78, Atelier Qualité des Données et des Connaissances (associé à la conférence Extraction et Gestion des Connaissances (2005)), Paris, 18 janvier 2005

[WEI 1999] Wei, Qiang; Chen, Guoqing: Mining Generalized Association Rules with Fuzzy Taxonomic Structures. Fuzzy Information Processing Society, 1999. NAFIPS, 1999.

[ZADE 1965] Zadeh, Lofti A.: Fuzzy Sets. Information and Control 8 (3) 338-353, 1965.

[ZADE 1978] Zadeh Lofti., «Fuzzy sets as a basis for a theory of possibility», Fuzzy Set and Systems, vol. 1, 1978, p. 3-28.

[Zaki 1997] Zaki, M.J., Parthasarathy, S., Ogihara, M., Li, W. « New Algorithms for fas discovery of Association Rules ». In Proceedings of the 3rd Int'l Conference on KDD and data mining (KDD'97), Newport Beach, California, August 1997.

[ZIGH 2003] Zighed D.A., Rakotomalala R., « Extraction de connaissances à partir des données (ECD) », in Techniques de l'Ingénieur, H 3 744, 2003.

Résume

Résumé. La logique floue a un fort impact sur les méthodes de la fouille de données, particulièrement dans les règles d'association. Plusieurs recherches ont été effectuées pour aborder le sujet et plusieurs problèmes ont été rencontrés. Parmi ces problèmes, on trouve celui des mesures d'intérêt d'une règle d'association floue, par exemple la façon de calcul du support, sachant qu'un Item flou est représenté dans les enregistrements d'une base de données par des degrés d'appartenance qui sont inclus dans l'intervalle [0,1]. Ces quantités numériques offrent plusieurs façons de calcul du support et de la confiance. Notre approche consiste en une modification de l'algorithme initial *Apriori* et concerne plus précisément la manière de calcul du support en utilisant trois approches différentes : la *t-norme* (en choisissant le minimum des degrés d'appartenance), la t-conorme (en choisissant le maximum des degrés d'appartenance), et le produit des degrés d'appartenances. Enfin, les résultats de ces différentes approches seront présentés et discutés.

***Mots clés** : Fouille de données , Règles d'association floues, Support et confiance d'un Itemset flous.*

Abstract— Fuzzy logic has a strong impact on methods of Data Mining, especially in association rules. Several studies have been conducted to address the topic and many problems were encountered. These problems include the measures of interest a fuzzy association rule, such as computing support, knowing fuzzy Item is represented in the records in a database by the membership degrees. It's included in an interval [0,1]. These numerical quantities have many ways to calculating support and confidence. Our approach is to use respectful support measures, specifically how to calculate it by using three different approaches: the t-norm (choosing the minimum of the membership degrees), the t-co-norm (choosing the maximum of the membership degrees), and the multiplication of the membership degrees. At the end, the results of these different approaches will be presented and discussed

***Key words** : Data Mining , Fuzzy association rules, supports and confidence of fuzzy Itemset .*

خلاصة :المنطق الغامض له الفضل كبير على طرق و خوارزميات البحث و التفتيش في عمق المعطيات و بالخصوص على قواعد البيانات عدة بحوث تم إعدادها لطرح هذا الموضوع و لكن هناك عدة مسائل . و نذكر من بينها مسألة ما هي أحسن المقاييس لاجاد العلاقات بين العناصر الغامضة؟ ، مثال كيفية حساب النسب (الحضور و الثقة) علما أن العنصر الغامض يتم تمثيله في سجلات الخاصة بقاعدة البيانات بواسطة درجة التباين و هذه الآخيرة تنتمي الى المجال [0،1] .هذه القيم (الكميات) العددية تتيح لنا عدة طرق لحساب النسب (الحضور و الثقة). العمل منجز يتمثل في تعديلات على الخوارزمية الأساسية ***APRIORI*** بالخصوص على طرق حساب نسبة الحضور و ذلك بالاعتماد على 3 طرق مختلفة , t−norme (باختيار أصغر درجة التباين) t-conorme (باختيار أكبر درجة التباين) ; و الجداء درجات التباين. أخرا قمنا بدراسة و مناقشة النتائج المتحصل عليها زيادتا على هذا قمنا بعملية مقارنة بين هذه الطرق المطروحة.

***المفاتيح :** البحث في عمق المعطيات ، العلاقات بين العناصر الغامضة ، نسب الحضور و الثقة للعناصر الغامضة .*