République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université M'hamed BOUGARA de BOUMERDES

Faculté des Sciences

Département d'Informatique

MEMOIRE DE MAGISTER

Spécialité : Système informatique et génie des logiciels

Option : Spécification de Logiciel et Traitement de l'Information

Ecole Doctorale

Présenté par :

Alouane Basma

Thème

Recherche de partitions floues optimales par segmentation floue pour la fouille de données quantitatives.

Devant le jury de soutenance composé de:

Mr. MEZGHICHE Mohamed	Professeur	UMBB	Président.
Mr. DJOUADI Yassine	Maître de conférence	UMMTO	Rapporteur.
Mr. AHMED NACER Mohamed	Professeur	USTHB	Examinateur.
Mr. BABA-ALI Ahmed Riadh	Maître de conférence	USTHB	Examinateur.

Année Universitaire : 2007/2008

Remerciements

Je tiens à remercier, Monsieur DJOUADI, pour ses conseils judicieux, sa grande disponibilité et les précieuses discussions que nous avons eues ensemble. Je lui exprime ma profonde gratitude pour m'avoir fait profiter de ses connaissances, mais aussi de ses méthodes de travail, et surtout de sa rigueur scientifique. Grâce à lui, j'ai découvert un domaine de recherche qui aujourd'hui me passionne.

J'adresse mes plus vifs remerciements à Monsieur M. Mezghiche Professeur à l'UMBB, pour avoir accepter de présider ce jury et pour m'avoir accueilli dans son laboratoire LIFAB.

Je remercie également et pleinement Monsieur AHMED NACER, Professeur à l'USTHB et Monsieur BABA-ALI, Maître de conférence à l'USTHB qui m'ont fait l'honneur d'accepter de juger ce modeste travail.

Je remercie, Monsieur le professeur BELKAID, M.S, Doyen de la faculté de Génie Electrique et Informatique pour m'avoir accueilli.

Je remercie également l'équipe DATA MINING de l'université de Tizi-Ouzou. Je souhaite aussi témoigner toute mon amitié à l'ensemble de mes collègues de poste de graduation.

Enfin un grand merci à mes parents et à mes frères et sœurs, qui mon toujours soutenu, et je leur dédie ce travail.

A mes parents

Table de matière

Chapitre II : Extraction de connaissance et fouille de données

Abstract

The original problem of research of association rules was to extract some correlations from binary data.

Noting that the data are often quantitative, the problem has been extended for quantitative attributes by partitioning attribute domain and consequently mapping the quantitative problem to a binary one. However, such mapping generates sharp boundary problem.

In order to avoid such problem, fuzzy sets have been considered. All existing approaches, assume that fuzzy sets are empirically given. For this purpose, an original approach is proposed in this work in order to generate automatically fuzzy partitions. Our approach is based on fuzzy clustering method. We also suggest how to find automatically the optimal number of the fuzzy sets by using validity indices.

Key words: Data mining, association rules, fuzzy sets, fuzzy association rules, fuzzy clustering, validity indices, fuzzy resemblance measure.

ملخص:

عرفت إشكالية البحث عن قواعد التجميع اهتمام متزايد. طرحت هذه الإشكالية لأول مرة للبحث عن قواعد التجميع للمعطيات الثنائية.

و قد تم تمديد الإشكالية لمعالجة المعطيات الكمية و ذلك باقتراح تقسيم مجال المعلومة، و بذلك يتم تحويل إشكالية المعطيات الكمية إلى إشكالية المعطيات الثنائية.

و لكن مثل هذا التحويل سبب ضياع المعلومات. و لاحتواء هذا المشكل اعتمد إدخال نظرية المجموعات الضبابية. و كل الأعمال الموجودة بخصوص هذا الاقتراح، تعتبر أن المجموعات الضبابية معطاة من قبل خبير في الميدان.

في هذا المجال نقترح أول مقاربة من هذا النوع لإيجاد المجموعات الضبابية و ذلك بطريقة آلية. و هذه المقاربة أساسا تعتمد منهج العنقدة الضبابية كما نقترح منهج لايجاد عدد المجموعات الضبابية آليا و هذا باستعمال قرائن الصلاحية.

الكلمات المفتاح: التنقيب عن البيانات، قواعد التجميع، المجموعات الضبابية، قواعد التجميع الضبابي، العنقدة الضبابية، قرائن الصلاحية، مقياس التشابه الضبابي.

Résumé

Le problème original de recherche de règles d'association consistait à extraire certaines corrélations à partir de données binaires.

Constatant que souvent les données sont quantitatives, le problème a été étendu. L'idée consiste à ramener le problème à un cas binaire. Cependant une telle transformation cause le problème de valeurs aux limites.

Afin de pallier à ce problème, les ensembles flous ont été proposés. Toutes les approches existantes dans la littérature considèrent que les ensembles flous sont donnés d'une manière empirique. A cette fin, nous proposons dans notre mémoire une approche originale qui permet de générer automatiquement les partitions floues. Nous proposons aussi deux méthodes pour trouver le nombre de partitions floues.

Mots clés : Fouille de données, Règles d'association, ensembles flous, Règles d'association floues, Segmentation floue, Indice de validité, Mesure de ressemblance floue.

L'Extraction automatique de Connaissances à partir de Données (ECD) ou Knowledge Discovery in Databases (KDD) pour les anglos-saxons, consiste en la recherche d'informations non triviales, cachées dans un ensemble volumineux de données. La fouille de données est une étape dans un processus d'Extraction de Connaissances. Ces deux notions (l'Extraction de Connaissance à partir de Données et la fouille de données) seront définies plus formellement dans les paragraphes II.1 et II.2.

Les techniques d'extractions de connaissances à partir de données sont utilisées dans des domaines très variés, mais trouvent une application évidente comme outil d'aide à la décision. L'exploitation, par des techniques d'extractions, des données sauvegardées sur de longues périodes de temps est représentant la mémoire de l'entreprise permet de faire émerger des phénomènes, modèles ou règles insoupçonnés qui constituent un réservoir d'informations et connaissances essentielles pour la prise de décision.

La fouille de données est actuellement un domaine de recherche en plein essor visant à exploiter les grandes quantités de données collectées chaque jour. Ce domaine pluridisciplinaire se situe au confluent de l'intelligence artificielle (notamment de l'apprentissage automatique), des statistiques et des bases de données.

L'idée de la fouille de données est d'extraire des connaissances cachées à partir d'un gisement de données disponible. Diverses formes de connaissances peuvent êtres apprises à partir de données : elles peuvent être sous forme de règles d'association, de modèles, de régularité, de concepts etc...

Dans ce travail nous nous intéresserons aux règles d'association. Les règles d'association constituent un des modèles les plus puissants en fouille de données [6]. Elles permettent de traiter des gros volumes de données et d'en extraire des règles significatives de la forme : pâte $\rightarrow$ fromage, signifiant que les clients qui achètent des pâtes achètent aussi du fromage.

La plupart des algorithmes proposés à ce jour pour induire les règles d'association traitent des données binaires. De telles données indiquent l'absence ou la présence d'éléments. Constatant que souvent les données sont numériques une approche a été proposée pour généraliser les données binaires aux données numériques.

Ils s'avère que l'idée de cette approche consiste à partitionner le domaine où prend ses valeurs une donnée numérique en différent intervalles. Ces intervalles sont disjoints et l'appartenance d'une valeur à un intervalle est binaire. Cette approche soulève toutefois la difficulté de :

1. trouver le nombre judicieux d'intervalles.
2. trouver les bornes *inférieure* et *supérieure* de chaque intervalle.

Ce problème se pose de manière évidente en remarquant qu'un nombre élevé d'intervalles fait forcément diminuer le support d'un item tandis qu'un nombre faible d'intervalles fait diminuer la confiance.

Constatant ce dilemme, les ensembles flous ont été proposés pour la fouille de données quantitatives [42]. Ils permettent un découpage graduel d'un domaine (fini) en différents sous ensembles flous. Ils permettent aussi un traitement symbolique (qualitatif) de l'intervalle considéré. Il s'entend qu'un tel traitement étant plus proche du modèle humain de raisonnement [14]. Si l'utilité des ensembles flous dans la fouille de données est reconnue au vu des performances et résultats obtenus [42] [53] [54] [55], il s'avère néanmoins que tous ces travaux considèrent des **partitions empiriques**. **Aucune des approches** existantes dans la littérature ne propose **une partition automatique (intelligente)** du domaine de l'attribut considéré.

Sur la base de cette constatation, nous proposons dans ce mémoire une méthode qui permet de découvrir automatiquement les partitions floues correspondant à un attribut quantitatif donnée. A cette fin, nous proposons d'utiliser la segmentation floue pour générer les partitions des attributs numériques.

Notre mémoire est organisé comme suit :

Le chapitre II présente la découverte de connaissance dans les bases de données et la fouille de données, il décrit le processus d'extraction de connaissances et situe la fouille de données dans ce processus, puis en présente les différentes techniques de fouille de données. Une section est consacrée à la recherche des règles d'association. On y trouve la présentation

formelle et la présentation informelle, les algorithmes d'extractions des règles d'association, enfin on introduit le problème des règles d'association quantitatives.

Le chapitre III présente la théorie des sous ensembles flous, puis détaille les différentes approches existantes concernant les règles d'association floues. Enfin on y présente le problème de partitionnement flou du domaine des attributs numériques (quantitatives).

Le chapitre IV intitulé la segmentation, introduit la segmentation classique, et présente les différents algorithmes de segmentation classique, une section est consacrée pour présenter les algorithmes de segmentation floue. On y présente les avantages de ces algorithmes dans le cadre de notre travail. Enfin on présente l'algorithme proposé pour notre étude.

Le chapitre V présente l'approche proposée, nous présentons les différents algorithmes utilisés dans notre approche. Ainsi nous présentons la méthode adoptée pour la comparaison des différents modèles de règles d'association issus de nos différentes propositions.

Dans le chapitre VI intitulé validation expérimentale, nous présentons les différents résultats de l'approche proposée, et nous présentons aussi les différentes bases de données sur les quelles nous avons testé la faisabilité de la solution proposée.

Enfin, nous terminons ce mémoire par une conclusion et nous présentons nos perspectives de recherche futures.

Ce chapitre présente l'extraction de connaissance à partir des données. Les différentes étapes d'un processus d'extraction de connaissance à partir des données seront aussi décrites. Parmi ces étapes, nous détaillerons la fouille de données ainsi que les différentes approches de mise en œuvre d'un modèle de fouille de données

II.1 Extraction de connaissances à partir des données

Selon certains experts, les quantités de données doublent en général tous les neuf mois [93]. Si l'on se réfère à des applications scientifiques, ce sont de gigaoctets de données qui sont collectées et stockées. Dans des entreprises commerciales comme les assurances, la grande distribution ou encore dans le domaine bancaire, de nombreuses données sont collectées sur les clients et ne sont pas nécessairement exploitées par la suite.

Dans ce cas, comment répondre aux besoins des entreprises qui souhaitent pouvoir exploiter ces données dans des temps acceptables, tout en sachant que les requêtes traditionnelles, type **SQL**, sont limitées au niveau du type d'informations qu'elles peuvent obtenir d'une base de données ? Tous ces facteurs sont les éléments d'un domaine de recherche et de développement très actif actuellement appelé : l'**E**xtraction de **C**onnaissances dans les **D**onnées (**ECD**) ou en anglais ***K**nowledge **D**iscovery in **D**atabase (**KDD**)* [23].

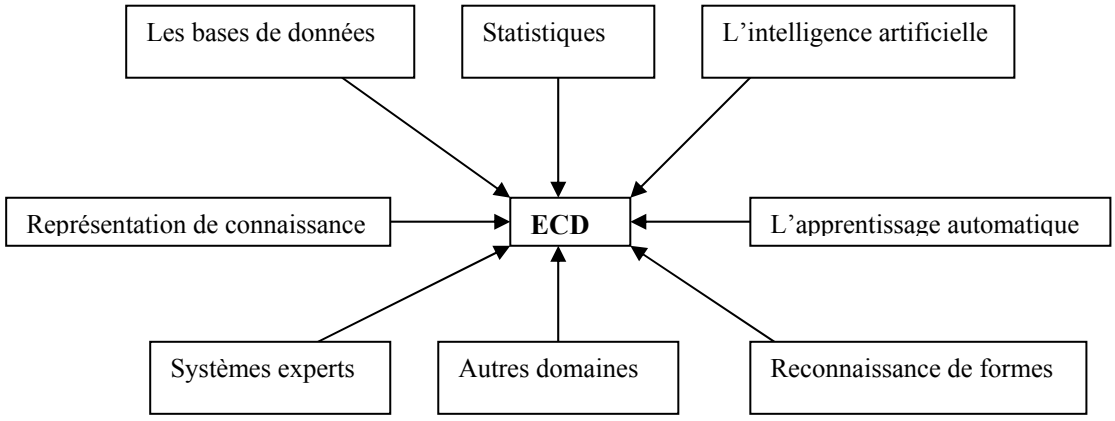


Figure II.1 : *La fouille de données à la confluence de nombreux domaines* [43].

Grâce aux techniques d'extraction de connaissance, les bases de données volumineuses sont devenues des sources riches et fiables pour la génération et la validation de connaissances. La fouille de données (en anglais Data Mining) n'est qu'une phase du

processus d'extraction de connaissances à partir des données, et consiste à appliquer des algorithmes d'apprentissage sur les données afin d'en extraire des modèles (ou motifs). L'extraction de connaissances à partir des données se situe à l'intersection de nombreuses disciplines [43], [6] comme l'apprentissage automatique, la reconnaissance de formes, les bases de données, les statistiques, la représentation de connaissance, l'intelligence artificielle, les systèmes experts, etc… (cf. Figure II.1).

II.1.1 Définition générale

Afin de définir la notion d'extraction de connaissance à partir des données (en anglais knowledge discovery in database), nous présentons les deux définitions suivantes :

Definition 1 (Knowledge Discovery in Databases)

Knowledge Discovery in Databases is the non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data [1].

Définition 2 (Extraction de Connaissances à partir des Données)

L'Extraction de Connaissance à partir des Données (ECD) est un processus itératif et interactif d'analyse d'un grand ensemble de données brutes afin d'en extraire des connaissances exploitables par un utilisateur- analyste qui y joue un rôle central [2].

D'après ces deux définitions, l'utilisateur fait partie intégrante du processus. L'interactivité est liée aux différents choix que l'utilisateur est amené à effectuer et est liée au fait que l'extraction de connaissances à partir des données soit composée de plusieurs phases et que l'utilisateur peut décider de revenir en arrière à tout moment si les résultats ne lui conviennent pas.

Dans la suite de document nous utiliserons l'abréviation ECD pour signifier Extraction de Connaissances à partir des Données.

II.1.2 Les étapes d'un processus d'Extraction de Connaissances à partir des données

D'après [1], un processus d'ECD est constitué de quatre phases qui sont : *le nettoyage et intégration des données, le pré-traitement des données, la fouille de données* et enfin *l'évaluation et la présentation des connaissances.*

La figure II.2 récapitule ces différentes phases ainsi que les enchaînements possibles entre ces phases. Cette séparation est théorique, en pratique, ce n'est pas toujours le cas. En effet, dans de nombreux systèmes, certaines de ces étapes sont fusionnées [43], [6].

1. Nettoyage et intégration des données

Le nettoyage des données consiste à retravailler des données bruitées, soit en les supprimant, soit en les modifiant de manière à tirer le meilleur profit.

L'intégration est la combinaison des données provenant de plusieurs sources (base de données, sources externes, etc.…).

Le but de ces deux opérations est de générer des entrepôts de données et/ou des magasins de données spécialisés contenant les données retravaillées pour faciliter leurs exploitations futures.

Exemple :

Soit l'exemple suivant qui présente une base de donnée d'un éditeur qui vend 5 sortes de magazines : sport, voiture, maison, musique et BD. Il souhaite mieux étudier ses clients pour découvrir de nouveaux marchés ou vendre plus de magazines à ses clients habituels.

client	Nom	Adresse	Date d'abonnement	Magazine
23134	Bemol	Rue du moulin, Paris	7/10/96	Voiture
23134	Bemol	Rue du moulin, Paris	12/5/96	Musique
23134	Bemol	Rue du moulin, Paris	25/7/95	BD
31435	Bodinoz	Rue verte, Nancy	11/11/11	BD
43342	Airinaire	Rue de la source, Brest	30/5/95	Sport
25312	Talonion	Rue du marché, Paris	25/02/98	NULL
43241	Manvussa	NULL	14/04/96	Sport
23130	Bemolle	Rue du moulin, Paris	11/11/11	Maison

Table II.1 : *la base de données avant le nettoyage*

Intégrité de domaine : Dans notre exemple, la date d'abonnement des client 23130, 31435 (11/11/11) semble plutôt correspondre à une erreur de saisie ou encore à une valeur par défaut en remplacement d'une valeur manquante.

Informations manquantes : Dans notre exemple, nous n'avons pas le type de magazine pour le client 25312. Il sera écarté de notre ensemble. L'enregistrement du client 43241 sera conservé bien que l'adresse ne soit pas connue.

Après le nettoyage on aura la base de donnée suivante :

client	Nom	Adresse	Date d'abonnement	Magazine
23134	Bemol	Rue du moulin, Paris	7/10/96	Voiture
23134	Bemol	Rue du moulin, Paris	12/5/96	Musique
23134	Bemol	Rue du moulin, Paris	25/7/95	BD
31435	Bodinoz	Rue verte, Nancy	NULL	BD
43342	Airinaire	Rue de la source, Brest	30/5/95	Sport
43241	Manvussa	NULL	14/04/96	Sport
23130	Bemolle	Rue du moulin, Paris	NULL	Maison

Table II.2 : *la base de données après le nettoyage*

2. Pré-traitement des données

Il peut arriver parfois que les bases de données contiennent à ce niveau un certain nombre de données incomplètes et/ou bruitées. Ces données erronées, manquantes ou inconsistantes doivent être retravaillées si cela n'a pas été fait précédemment. Dans le cas contraire, durant l'étape précédente, les données sont stockées dans un entrepôt. Cette étape

permet de sélectionner et transformer des données de manière à les rendre exploitables par un outil de fouille de données.

Cette seconde étape du processus d'ECD permet d'affiner les données. Si l'entrepôt de données est bien construit, le pré-traitement de données peut permettre d'améliorer les résultats lors de l'interrogation dans la phase de fouille de données.

Exemple : soit la base de donnée nettoyée précédemment, la table II.3 présente le résultat de pré-traitement. Les clients qui ont des informations manquantes seront supprimés de la base.

client	Nom	Adresse	Date d'abonnement	Magazine
23134	Bemol	Rue du moulin, Paris	7/10/96	Voiture
23134	Bemol	Rue du moulin, Paris	12/5/96	Musique
23134	Bemol	Rue du moulin, Paris	25/7/95	BD
43342	Airinaire	Rue de la source, Brest	30/5/95	Sport

Table II.3: *la base de données après le pré-traitement*

3. Fouille de données (Data Mining)

La fouille de données (*data mining* en anglais), est le cœur du processus d'ECD. Il s'agit à ce niveau de trouver des pépites de connaissances à partir des données. Tout le travail consiste à appliquer des méthodes intelligentes dans le but d'extraire cette connaissance. Il est possible de définir la qualité d'un modèle en fonction de critères comme les performances obtenus, la fiabilité, la compréhensibilité, la rapidité de construction et d'utilisation et enfin l'évolutivité. Tout le problème de la fouille de données réside dans le choix de la méthode adéquate à un problème donné. Il est possible de combiner plusieurs méthodes pour essayer d'obtenir une solution optimale globale.

Nous ne détaillerons pas d'avantage la fouille de données dans ce paragraphe car elle fera l'objet d'un paragraphe complet (cf.II.2)

4. Evaluation et présentation

Cette phase est constituée de l'évaluation, qui mesure l'intérêt des motifs extraits, et de la présentation des résultats à l'utilisateur grâce à différentes techniques de visualisation. Cette étape est dépendante de la tâche de fouille de données employée. En effet, bien que

l'interaction avec l'expert soit importante quelle que soit cette tâche, les techniques ne sont pas les mêmes. Ce n'est qu'à partir de la phase de présentation que l'on peut employer le terme de *connaissance* à condition que ces motifs soient validés par les experts du domaine. Il y a principalement deux techniques de validation qui sont la technique de validation statistique et la technique de validation par expertise.

La validation statistique consiste à utiliser des méthodes de base de statistique descriptive. L'objectif est d'obtenir des informations qui permettront de juger le résultat obtenu, ou d'estimer la qualité ou les biais des données d'apprentissage. Cette validation peut être obtenue par :

- le calcul des moyennes et variances des attributs,
- si possible, le calcul de la corrélation entre certains champs,
- ou la détermination de la classe majoritaire dans le cas de la classification.

La validation par expertise, est réalisée par un expert du domaine qui jugera la pertinence des résultats produits. Par exemple pour la recherche des règles d'association, c'est l'expert du domaine qui jugera la pertinence des règles.

Pour certains domaines d'application (le diagnostic médical, par exemple), le modèle présenté doit être compréhensible, une première validation doit être effectuée par un expert qui juge la compréhensibilité du modèle. Cette validation peut être, éventuellement, accompagnée par une technique statistique.

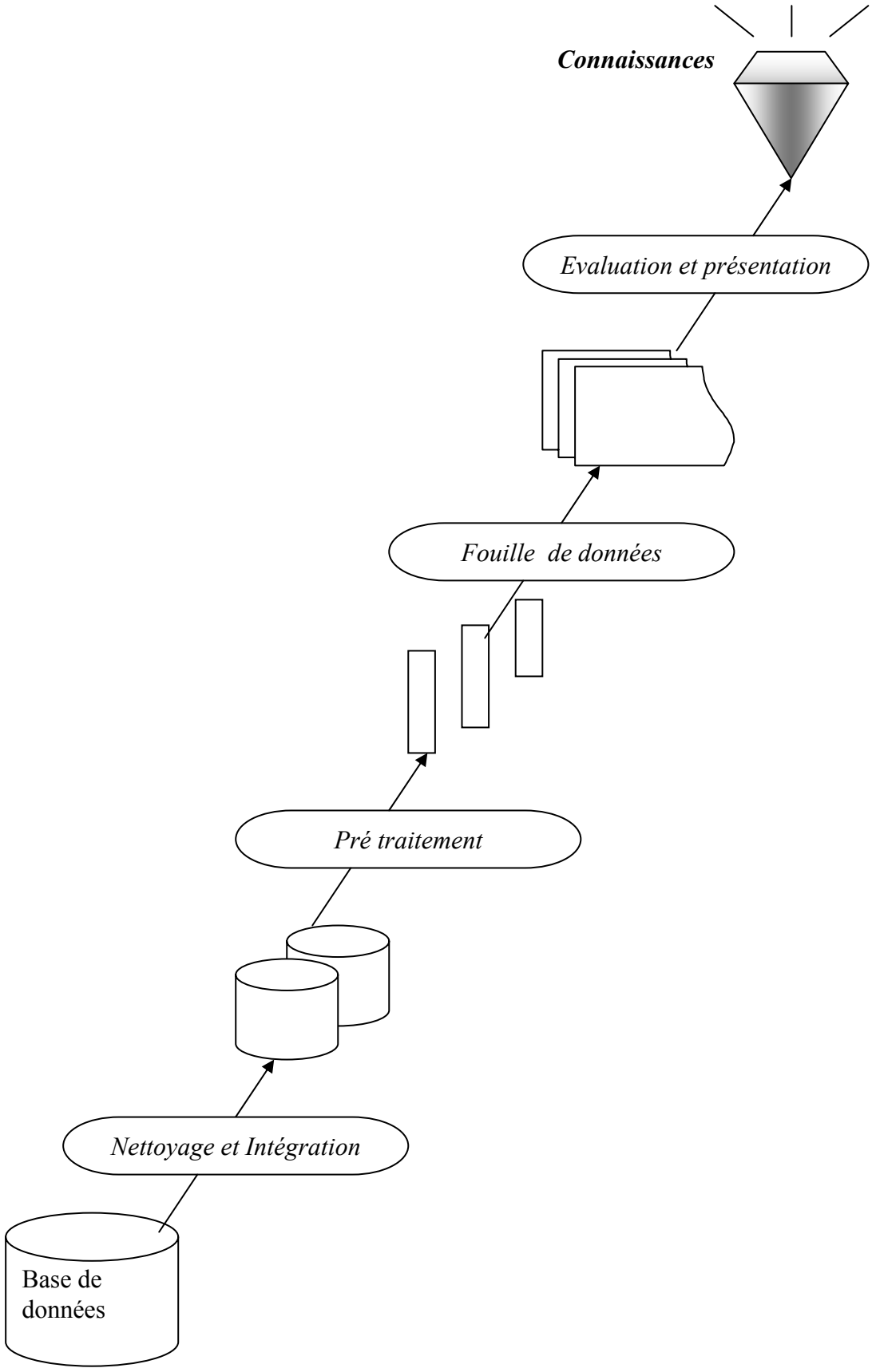


Figure II.2 *: processus d'extraction de connaissances à partir des données*

II.2 Fouille de données (Data Mining)

Les concepts de fouille de données et d'extraction de connaissances à partir de données sont parfois confondus et considérés comme synonymes. Mais, formellement on considère la fouille de données comme une étape essentielle intervenant dans le processus d'ECD.

La fouille de données fait appel à un lot de méthodes issues de la statistique, de l'analyse des données, de la reconnaissance des formes, de l'intelligence artificielle ou de l'apprentissage automatique. L'objectif de la mise en œuvre des techniques de fouille de données est d'aboutir à des connaissances opérationnelles. Ces connaissances sont exprimées sous forme de modèles : une série de coefficients pour un modèle de prévision numérique, des règles logiques du type "Si Condition alors Conclusion" ou des instances.

II.2.1 Historique

L'expression "*data mining*" est apparue vers le début des années 1960 et avait, à cette époque, un sens péjoratif. En effet, les ordinateurs étaient de plus en plus utilisés pour toutes sortes de calculs qu'il n'était pas envisageable d'effectuer manuellement jusque là. Certains chercheurs ont commencé à traiter sans à priori statistique les tableaux de données relatifs à des enquêtes ou des expériences dont ils disposaient. Comme ils constataient que les résultats obtenus, loin d'être aberrants, étaient tout au contraire prometteurs, ils furent incités à systématiser cette approche opportuniste. Les statisticiens officiels considéraient toutefois cette démarche comme peu scientifique et utilisèrent alors les termes "*data mining*" ou "data fishing" pour les critiquer.

Cette attitude opportuniste face aux données coïncida en France avec la diffusion dans le grand public de l'analyse de données dont les promoteurs, comme Jean-Paul Benzecri [4], ont également du subir dans les premiers temps les critiques venant des membres de la communauté des statisticiens.

Le succès de cette démarche empirique ne s'est pas démenti malgré tout. L'analyse des données s'est développée et son intérêt grandissait en même temps que la taille des bases de données.

Vers la fin des années 1980, des chercheurs en base de données, tel que Rakesh Agrawal, ont commencé à travailler sur l'exploitation du contenu des bases de données volumineuses comme par exemple celles des tickets de caisses de grandes surfaces, convaincus de pouvoir valoriser ces masses de données dormantes. Ils utilisèrent l'expression "*database mining* " mais, celle-ci étant déjà déposée par une entreprise (*Database mining workstation*), ce fut "*data mining*" qui s'imposa. En mars 1989, Shapiro Piatetski proposa le terme "*knowledge discovery* " à l'occasion d'un atelier sur la découverte des connaissances dans les bases de données.

La communauté de "*data mining*" a initié sa première conférence en 1995 à la suite de nombreux atelier (*workshops*) sur le KDD entre 1989 et 1994. En 1998 s'est créé, sous les auspices de l'ACM, un chapitre spécial baptisé ACM-SIGKDD, qui réunit la communauté internationale du KDD. La première revue du domaine "*Data mining and knowledge discovery journal* " publiée par "Kluwers" a été lancée en 1997.

La fouille de données, dans sa forme et compréhension actuelle, à la fois comme champ scientifique et industriel, est apparue au début des années 90. Cette émergence n'est pas le fruit du hasard mais le résultat de la combinaison de nombreux facteurs à la fois technologiques, économiques et même socio-politiques.

On peut voir la fouille de données (*data mining*) comme une nécessité imposée par le besoin des entreprises de valoriser les données qu'elles accumulent dans leurs bases.

II.2.2 Définition de la fouille de données

La définition la plus admise pour la fouille de données est celle de Fayyad [3] :

« Extraction of interesting (*non-trivial, implicit, previously unknown and potentially useful*) *information or patterns from data in large databases"*

« *Le Data Mining est un processus non trivial qui consiste à identifier, dans des données, des schémas nouveaux, valides, potentiellement utiles et surtout compréhensibles et utilisables ».*

La métaphore qui consiste à considérer les grandes bases de données comme des gisements d'où l'on peut extraire des pépites à l'aide d'outils spécifiques n'est certes pas nouvelle. Dès les années 1970 Jean-Paul Benzécri assignait le même objectif à l'analyse des données [4] :

« L'analyse des données est un outil pour dégager de la gangue des données le pur diamant de la véridique nature ».

On a pu donc considérer que bien des praticiens faisaient de fouille de données sans le savoir.

Les premières applications se sont faites dans le domaine de la gestion de la relation client qui consiste à analyser le comportement de la clientèle pour mieux la fidéliser et lui proposer des produits adaptés.

Quand elle est bien menée, la fouille de données a apporté des succès certains, à tel point que l'engouement qu'elle suscite a pu entraîner la transformation (au moins nominale) de services statistiques de grandes entreprises en services de fouille de données.

La fouille de données est l'art d'extraire des connaissances à partir des données [47]. Les données peuvent être stockées dans des entrepôts (*data warehouse*) [83], dans des bases de données distribuées ou sur Internet : *web mining*. La fouille de données ne se limite pas au traitement des données structurées sous forme de tables numériques ; il offre des moyens pour aborder les corpus en langage naturel (*text mining*), les images (*image mining*), le son (*sound mining*) ou la vidéo et dans ce cas, on parle alors plus généralement de *multimedia mining*.

La fouille de données consiste en l'ensemble des techniques [51] [49] qui permettent de transformer les données en connaissances. Parmi ces techniques, nous pouvons citer :

- Classification
- Estimation
- Prédiction
- Regroupement par similitudes

- Segmentation
- Description
- Sequencing, séries temporelles
- Association

II.2.3 Les méthodes de fouille de données

Les méthodes de fouille de données permettent de découvrir ce que contiennent les données comme informations ou modèles utiles. Si nous essayons de distinguer les méthodes de fouille de donnés utilisées, trois catégories se dégagent [6],[43] [52] :

- Les méthodes de visualisation et de description.
- Les méthodes de classification et de structuration.
- Les méthodes d'explication et de prédiction.

Chacune de ces familles de méthodes comporte plusieurs techniques appropriées aux différents types de tableaux de données. Certaines sont mieux adaptées à des données numériques continues alors que d'autres sont plus généralement dédiées aux traitements de tableaux de données qualitatives. Nous allons donner à présent un aperçu général sur les principales méthodes.

1. Les méthodes de visualisation et de description

L'objectif de ces méthodes est de permettre à l'analyste d'avoir une compréhension synthétique de l'ensemble de ses données. Il s'agit donc principalement d'outils de synthèse d'information. Cette synthèse peut s'exprimer par des indicateurs statistiques. Par exemple, pour des attributs quantitatifs, les indicateurs les plus utilisés sont la moyenne, l'écart-type, le mode et la médiane. Pour des attributs qualitatifs, on associe généralement la distribution selon les modalités de l'attribut.

2. Les méthodes de structuration et de classification

En ECD, nous avons affaire à une profusion de données. Décrire ces données s'avère parfois difficile à cause de cette volumétrie. L'utilisateur cherche souvent à identifier des

groupes d'objets semblables au sens d'une métrique donnée. Ces groupes peuvent par exemple correspondre à une réalité ou à des concepts particuliers.

Par exemple, dans la réalité, l'homme à souvent beaucoup de mal à mémoriser de façon individualisée un ensemble d'objets, surtout quand ils sont en très grand nombre. Par exemple, mémoriser toutes les espèces végétales ou animales est une tâche extrêmement laborieuse, voire impossible, pour un humain. L'homme préfère généralement catégoriser ces objets en classes en fonction de certaines propriétés communes ou en fonction d'un critère donné. Ces classes ou ces catégories d'objets sont ensuite nommées. Par exemple, le monde animal est structuré en groupes : vertébrés ou invertébrés, mammifères ou non, etc. Ainsi, toutes les espèces sont ventilées en fonction de la présence ou non de certains attributs communs.
Les principales techniques de cette méthode se répartissent en trois groupes [52] :

- *Les méthodes monothétiques* : Dont l'objet est la recherche de partitions sur l'ensemble des objets à classer, telles que sur chaque classe, l'un des attributs soit constant ou de très faible variance. Par exemple, dans la classe des vertébrés, toutes les espèces ont en commun la présence de vertèbres.

- *Les méthodes polythétiques* : Ces méthodes recherchent des partitions dans lesquelles les éléments d'une même classe ont, entre eux, une certaine ressemblance, et des éléments appartenant à des classes différentes d'une même partition qui doivent être les plus dissemblables possibles au sens d'un certain critère préétabli. La ressemblance doit prendre en compte la totalité des attributs descriptifs.

Parmi les techniques fréquemment employées, on trouve les méthodes de classification hiérarchique [63], les nuées dynamiques proposées par Diday [94].

- *Les méthodes basées sur les réseaux de neurones* : Le processus d'apprentissage est incrémental, c'est-à-dire que les objets sont affectés séquentiellement à des groupes en fonction de leur proximité. Nous retrouvons, dans cette catégorie de méthodes, les techniques dites des cartes topologiques de Kohonen , et les techniques basées sur la résonance adaptative de Grossberg et Carpenter.

3. Les méthodes d'explication et de prédiction :

Ces méthodes ont pour objectif de rechercher à partir des données disponibles un modèle explicatif ou prédictif entre, d'une part, un attribut particulier à prédire et, d'autre part, des attributs prédictifs. Dans le cas où un tel modèle serait produit et qu'il s'avérait valide, il pourrait alors être utilisé à des fins de prédiction.

Considérons le cas d'un médecin qui s'intéresse à la nature d'une affection dont il veut connaître la nature cancéreuse ou non. Imaginons qu'il souhaite construire une règle lui permettant de prévoir, à l'avance sur la base d'examens cliniques simples, la nature cancéreuse ou bénigne de l'affection. Pour cela, il peut procéder par apprentissage à partir de données. Cela consiste, pour lui, à recueillir des informations sur des patients déjà traités pour cette affection et dont il sait si elle a été cancéreuse ou bénigne. Sur la base de ce corpus, il mettra en œuvre une méthode d'apprentissage qui l'aiderait à bâtir son modèle d'identification. Dans ce contexte on parle d'apprentissage supervisé car l'attribut à prédire est déjà préétabli. Il s'agit alors de mettre au point un processus permettant de le reconstituer de façon automatique à partir des autres attributs.

Il existe une multitude de méthodes d'explication et ou de prédiction développées dans différents contextes. Nous allons présenter synthétiquement les principales familles de méthodes d'explication et de prédiction.

- *Les graphes d'induction :* Les graphes d'induction, dont les modèles les plus utilisés sont les arbres de décision [29].

- *Les réseaux de neurones :* Les réseaux de neurones sont parmi les outils de prédiction les plus utilisés pour les problèmes difficiles où le prédicateur que l'on cherche à construire repose sur de nombreuses interactions complexes entre les attributs exogènes.

- *Les méthodes de régression :* En régression, il s'agit d'expliciter une relation de type linéaire ou non entre un ensemble de variables exogènes et une variable endogène. Généralement, dans le cadre de la régression, toutes les variables sont considérées comme continues.

- *L'analyse discriminante :* L'analyse discriminante est l'une des plus anciennes techniques de discrimination. Elle a été proposée par Fischer en 1936.

- *Les réseaux bayésiens :* Les réseaux bayésiens sont apparus au début des années 1980. Rendus populaires par le groupe de recherche de la firme Microsoft qui les introduits dans les systèmes d'aide contextuelle d'Office, ils sont maintenant très utilisés dans la modélisation des processus complexes de décision.

- *Les règles d'association :* La recherche de règles d'association dans une base de données est certainement le problème qui a le plus fortement contribué à l'émergence de fouille de données en tant que domaine scientifique à part entière [7]. La grande distribution, les télécommunications et plein d'autres secteurs de la grande consommation enregistrent, dans un but de facturation, l'ensemble des transactions commerciales avec leurs clients. Pour une grande surface de distribution, cela peut atteindre plusieurs centaines de millions de transactions effectuées par jour. Les données enregistrées au passage en caisse servent d'une part à la facturation au client et d'autre part à des actions de gestion comme le suivi des stocks ou encore à l'étude de la composition des paniers dans un but de marketing. L'étude des transactions en vue d'identifier des associations entre produits permet de mieux caractériser un client et ainsi de définir des actions commerciales ciblées envers lui.

La recherche des règles d'association est une méthode d'apprentissage non supervisée, car on ne dispose en entrée que de la description des données. Les règles d'association, objet de notre travail, seront particulièrement décrites dans ce qui suit.

II.3 Les règles d'association

Cette section présente en détail les règles d'association qui constituent la problématique centrale de ce mémoire. Nous présentons dans cette section le cadre formel et le cadre informel pour la recherche des règles d'association.

II.3.1 Cadre informel

Le concept d'extraction des règles d'association introduit en 1993 par Agrawal et al. [7] est une méthode qui a vu le jour avec la recherche en bases de données. Agrawal et al. ont été amenés à travailler sur une application de vente de produits dans les supermarchés [7]. Cette application est également appelée "Analyse du panier de la ménagère" et elle est à l'origine des règles d'association. Il s'agit d'obtenir des relations ou des corrélations du type "Si Condition alors Résultat". Pour ce problème, chaque panier n'est significatif que pour un client en fonction de ses besoins et de ses envies, mais si le supermarché s'intéresse à tous les paniers simultanément, des informations utiles peuvent être induites et exploitées. Tous les clients sont différents et achètent des produits différents, en quantités différentes, l'analyse du panier de la ménagère étudie qui sont les clients et pourquoi ils effectuent tel ou tel type d'achat. Cette analyse permet d'étudier quels produits tendent à être achetés en même temps.

Par exemple, on sera capable de dire que 75% des clients qui achètent du lait achètent en même temps des oeufs (lait → oeuf : 0.75), une telle constatation est très intéressante puisqu'elle aide le gestionnaire d'un supermarché à ranger ses rayons de telle sorte que le lait et les œufs soient à proximité.

Dans les bases de données de vente, un tuple consiste en une transaction regroupant l'ensemble des articles achetés appelés items. Ainsi une base de données est un ensemble de transactions, qu'on appelle aussi base transactionnelle ou base de transactions.

Une règle d'association décrit une corrélation entre des ensembles d'items dans une base de transactions. Autrement dit, étant donné un ensemble d'items, le but est de découvrir si l'occurrence de cet ensemble dans une transaction est associée à l'occurrence d'un autre ensemble d'items. Par exemple, "80% des clients qui achètent un ordinateur achètent aussi une imprimante et un abonnement à Internet" est une règle d'association associant l'item ordinateur aux items imprimante et abonnement à Internet.

II.3.2 Cadre formel

Formellement, le problème de règle d'association est présenté dans [7] de la façon suivante :

Soit $I = \{ i_0, i_1, ..., i_n \}$ un ensemble de *n* items,

Soit $T = \{t_1,...,t_m\}$ un ensemble de *m* transactions telle que chaque transaction t_i soit dotée d'un identifiant unique (noté TID) appelé transaction identificateur, et chaque transaction t_i est un sous ensemble d'items $t_i \subseteq I$.

Définition (Itemset)

Un itemset (noté I_0) est un sous ensemble d'items autrement dit c'est un élément de l'ensemble des parties de I $\left(I_0 \in 2^I\right)$.

Exemple d' itemset

L'itemset $I_0 = \{banane, tomate, fromage\}$ est composée de trois items : banane, tomate, fromage.

Définition (*k*-itemset)

Un *k*-itemset est un sous-ensemble d'items I_0 $(I_0 \subseteq I)$ tel que $|I_0| = k$.

Exemple d'un k-itemset

L'itemset $I_0 = \{banane, tomate, fromage\}$ est un 3-itemset.

Définition (Support d'un itemset)

Le support d'un itemset I_k (noté $\text{supp}(I_k, T)$) est la probabilité qu'une transaction t_i contienne cet itemset I_k .

$$\text{supp}(I_k, T) = \frac{\left|\{t_i \in T / I_k \subseteq t_i\}\right|}{|T|} \qquad \text{(II.1)}$$

Où $|\ |$: désigne la cardinalité.

Le support prend sa valeur dans l'intervalle [0,1]. Il est souvent exprimé en pourcentage.

Définition (Itemset fréquent)

Etant donné un seuil γ, appelé support minimum (noté *minsupp*), un itemset I_k est dit fréquent (relativement à γ) dans une base de transactions T, si son support dépasse le seuil γ.

I_k est fréquent si seulement si $\text{supp}(I_k, T) \geq \gamma$

Définition (Règle d'association)

On appelle règle d'association, une relation de la forme $I_1 \Rightarrow I_2$ entre deux itemsets I_1, I_2 tels que $I_1, I_2 \subseteq I, I_1 \cap I_2 = \phi, I_1, I_2 \neq \phi.$

La partie gauche de la règle I_1 est appelée la *prémisse* de la règle et la partie droite I_2 est appelée la *conclusion* de la règle.

Définition (Support d'une règle d'association)

Le support d'une règle d'association $I_1 \Rightarrow I_2$ (note $Supp(I_1 \Rightarrow I_2, T)$) est le pourcentage des transactions de T qui contiennent $I_1 \cap I_2$

$$Supp(I_1 \Rightarrow I_2, T) = \frac{|t_i \in T / I_1 \cap I_2 \subseteq t_i|}{|T|} \quad \text{(II.2)}$$

$$Supp(I_1 \Rightarrow I_2, T) = Supp(I_1 \cap I_2)$$

Définition (Confiance) :

La confiance d'une règle d'association $I_1 \Rightarrow I_2$, notée ($conf(I_1 \Rightarrow I_2, T)$) représente la proportion de transactions couvrant I_1 et qui couvrent aussi I_2.

$$conf(I_1 \Rightarrow I_2, T) = \frac{|t_i \in T / I_1 \cap I_2 \subseteq t_i|}{|t_i \in T / I_1 \subseteq t_i|} \quad \text{(II.3)}$$

$$conf(I_1 \Rightarrow I_2, T) = \frac{Supp(I_1 \cap I_2)}{\text{supp}(I_1)}$$

Définition (Ensemble de règles d'association valides)

Soit un ensemble *F* d'itemsets fréquents pour un seuil minimal de support *minsupp*. Etant donné un seuil minimal de confiance *minconf*, l'ensemble *AR* des règles d'association valides est :

$$AR = \{ R : l_2 \rightarrow (l_1 - l_2) / l_1, l_2 \in F \wedge l_2 \subset l_1 \wedge \text{confiance}(R) > minconf \}$$

Pour chaque itemset fréquent l_1 dans *F*, touts les sous-ensembles l_2 de l_1 sont déterminés et la valeur de la confiance de la règle (*R*) est calculée. Si cette valeur est supérieure ou égale au seuil minimal de confiance alors la règle d'association $l_2 \rightarrow (l_1 - l_2)$ est générée. Cette partie sera développée plus en détail dans le paragraphe II.3.3.2.

II.3.3 La découverte des règles d'association

Le problème de découverte des règles d'association peut être scindée en deux sous problèmes

1. Le premier sous problème consiste à déterminer l'ensemble des itemsets fréquents dans la base de transaction *T*.
2. Le deuxième sous problème consiste à générer les règles d'association à partir de l'ensemble des itemesets fréquents.

II.3.3.1 Extraction des itemsets fréquents

Cette phase constitue la première partie de problème de recherche de règles d'association, il existe trois grandes approches algorithmiques pour la recherche d'itemsets fréquents pour la génération des règles d'association.

- Approche d'extraction d'itemsets fréquents [7],
- Approche d'extraction d'itemsets fréquents maximaux [77],
- Approche d'extraction d'itemsets fermés fréquents [25].

Dans ce travail nous nous intéressons à l'approche orientée "fréquent". Dans le cadre de cette approche Agrawal et al. ont proposé dans [7], le premier algorithme d'extraction des

règles d'association dans les bases de données transactionnelles, il s'agit de l'algorithme **Apriori** que nous décrivons ci-après.

Principe de base de l'algorithme Apriori

L'idée générale de cet algorithme est de générer à chaque itération k, un ensemble d'itemsets potentiels et de le tester (calculer le support). Un balayage est réalisé pour éliminer les itemsets non fréquents. Les k-itemsets fréquents obtenus sont réutilisés lors de l'itération k+1 pour générer les itemsets candidats de taille k+1. Afin de limiter, le nombre d'itemsets fréquents lors de chaque itération, cet algorithme se base sur le principe d'anti-monotonicité de la fréquence.

Propriété II.1 (*propriété sur les sous-ensembles)*

Tous les sous-ensembles d'un itemset fréquent sont fréquents.

Cette propriété permet de limiter le nombre des candidats de taille k générés lors de la k^{ime} itération en réalisant une jointure conditionnelle des itemsets fréquents de taille k-1 découverts lors de l'itération précédents [7].

Propriété II.2 (*propriété sur les sur-ensembles)*

Tous les sur-ensembles d'un itemset non fréquent sont non fréquents.

Cette propriété permet de supprimer un candidat de taille k lorsque au moins un de ses sous-ensembles de taille k-1 ne fait par partie des itemsets fréquents découverts lors de l'itération précédente [7].

Présentation de l'algorithme Apriori

Durant la première itération de l'algorithme (ligne1), tous les itemsets de taille 1 sont considérés et un balayage de la base de transaction T est réalisé afin de déterminer l'ensemble des 1-itemsets fréquents F_1. Les k itérations suivantes se subdivisent en deux étapes :

1- Durant la première étape, l'ensemble C_k des k-itemsets candidats est construit en joignant les (k-1)-itemsets fréquents, c'est-à-dire $C_k=F_{k-1}\square F_{k-1}$. Cette étape est réalisée dans la procédure *Apriori-Gen* (F_{k-1}).

2- Durant la seconde étape, un balayage de la base de données est réalisé afin de déterminer le support de chacun des k-itemsets candidats dans C_k et les k-itemsets fréquents sont insérés dans l'ensemble F_k. Lors de ce balayage, pour chaque transaction t de T, l'ensemble C_t des k-itemsets candidats qui sont contenus dans t est déterminé et le support de chacun de ces itemsets est recalculée.

La détermination de la fréquence des k-itemsets candidats est réalisée par la fonction *subset*(C_k,t). Cette fonction reçoit un ensemble C_k de k-itemsets candidats et une transaction t comme paramètres. Elle retourne un ensemble C_k de k-itemsets candidats de C_k qui sont contenu dans t. Ces itérations s'arrêtent lorsque aucun nouvel itemsets candidat ne peut être généré, c'est-à-dire $F_{k-1}=\phi$.

La procédure *exist_subset_infrequent*(c, F_{k-1}) renvoie une booléen afin d'indiquer s'il existe un sous-ensemble de c qui n'est pas fréquent.

La procédure *Apriori-Gen*(F_{k-1}), reçoit un ensemble de F_{k-1} *de* (k-1)-itemsets fréquents comme paramètre. Elle retourne un ensemble C_k de k-itemsets candidats qui est un sur-ensemble de l'ensemble des k-itemsets fréquents.

Deux (k-1)-itemsets fréquents f_1 et f_2 *de* F_{k-1} sont joints si et seulement si les k-2 premiers items qui les composent sont identiques. La jointure de deux (k-1)-itemsets résulte en un k-itemset candidat.

En utilisant la procédure *exist_subset_infrequent*(c, F_{k-1}), si le candidat c (k-itemsets) dont l'un des sous-ensembles s de taille k-1 ne se trouve pas dans F_{k-1}, alors c est supprimé de C_k , sinon il est ajouté à C_k.

Le pseudo-code de l'algorithme Apriori est présenté dans l'algorithme II.1.

Entrée : Base de transactions T ; Support minimum

Sortie : Ensemble des itemsets fréquents F

1. $F_1 \leftarrow \{1-itemsets\ \ frequents\}$
2. **Pour** $(k=2; F_{k-1} \neq 0; k++)$ **faire**
3. $C_k \leftarrow$ Apriori-gen(F_{k-1}) ;
4. **pour chaque** $(transaction\ \ t \in T)$ **faire**
5. $C_t \leftarrow subset(C_k, t)$ **;**
6. **pour chaque** (candidat $c \in C_t$) **faire**
7. $c.support++$;
8. **fin pour**
9. **fin pour**
10. $F_k \leftarrow \{c \in C_k \backslash c.support \geq misupp\}$
11. **fin pour**
12. retourner F_k

Fonction *Apriori-Gen*(F_{k-1})

1. **pour chaque** itemset $f_1 \in F_{k-1}$ **faire**
2. **pour chaque** itemset $f_2 \in F_{k-1}$ **faire**
3. **si** $(f_1[1] = f_2[1]) \vee (f_1[2] = f_2[2]) \vee ... \vee (f_1[k-2] = f_2[k-2]) \vee (f_1[k-1] < f_2[k-1])$ **alors**
4. $c = f_1[1], ..., f_1[k-1], f_2[k-1]$ //étape de jointure : générer des candidats
5. **si** exist_subset_infréquent(c, f_{k-1}) **alors**
6. supprimer c de C_k
7. **sinon**
8. ajouter c à C_k
9. **fin si**
10. **fin si**
11. **fin pour**
12. **fin pour**
13. retourner C_k

Fonction *exist_subset_infrequent*(c, f_{k-1})

1.**pour chaque** $(k-1)$-subset s de c **faire**

2. **si** $s \notin F_{k-1}$ **alors**

3. retourner TRUE ;

4. **fin si**

5. retourner FALSE ;

6.**fin pour**

Algorithme II.1 : *Algorithme Apriori pour la génération des itemsets fréquents*

Exemple : application de l'algorithme Apriori

Soit une base de transaction T contenant huit items et soit un seuil minimal de support *minsupp* égal à 3/8. Le tableau (II.4) présente la base de transactions.

TID	Tomate	Banane	Lait	Fromage	Pâte	Riz	Programme TV	Pile
T_1	1	0	0	1	1	0	0	0
T_2	0	0	1	0	0	1	0	0
T_3	0	0	0	0	0	1	1	1
T_4	1	0	0	0	1	0	0	0
T_5	1	0	0	0	1	0	0	0
T_6	0	1	1	0	0	1	0	1
T_7	1	0	0	1	1	0	0	0
T_8	1	0	0	1	1	0	0	0

Table II.4 : *La base de transaction T.*

Itemset	Support
Tomate	5/8
Banane	1/8
Lait	2/8
Fromage	3/8
Pâte	5/8
Riz	3/8
Programme TV	1/8
Pile	2/8

Génération de 1-Itemsets fréquents →

Itemset	Support
Tomate	5/8
Fromage	3/8
Pâte	5/8
Riz	3/8

Génération de 2-Itemsets candidats →

Itemset
Tomate & Fromage
Tomate & Pâte
Tomate & Riz
Fromage & Pâte
Fromage & Riz
Pâte & Riz

Calcul des Supports →

Itemset	Support
Tomate & Fromage	3/8
Tomate & Pâte	5/8
Tomate & Riz	0/8
Fromage & Pâte	3/8
Fromage & Riz	0/8
Pâte & Riz	0/8

Génération de 2-Itemsets fréquents →

Itemset	Support
Tomate & Fromage	3/8
Tomate & Pâte	5/8
Fromage & Pâte	3/8

Génération de 3-Itemsets candidats →

Itemset
Tomate & Fromage & Pâte

Génération de 3-Itemsets fréquents →

Itemset	Support
Tomate & Fromage & Pâte	3/8

Table II.5 : *recherche des itemsets fréquents*

II.3.3.2 Génération des règles d'association

La génération des règles d'association constitue la seconde étape de processus de recherche des règles d'association, elle s'effectue à partir des itemsets fréquents générés précédemment. Agrawal et al. [7] ont proposé un algorithme de génération de règles efficaces.

Le principe général de la génération de l'ensemble des règles d'association est le suivant :

Pour chaque itemset fréquent l_1 dans F de taille supérieure ou égale à deux, tous les sous-ensembles l_2 sont déterminés et la valeur du rapport *support* (l_1)*/support* (l_2) est calculée. Si ce rapport est supérieur ou égal au seuil de confiance *minconf* fixé par l'utilisateur, la règle d'association $l_2 \rightarrow (l_1 - l_2)$ est générée. L'algorithme proposé dans [7] se base sur la propriété (II.1) concernant les supports des itemsets pour réduire le nombre d'opérations réalisées par la génération.

Etant donné trois itemsets fréquents l_1, l_2 *et* l_3 tels que $l_1 \supset l_2 \supset l_3$, il est possible de déduire de la propriété (II.1) que *support* (l_3) $\geq$ *support* (l_2) $\geq$ *support* (l_1). En conséquence, la confiance de la règle R_2 : $l_3 \rightarrow (l_1 - l_3)$ est inférieure ou égale à la confiance de la règle R_1: $l_2 \rightarrow (l_1 - l_2)$. Si la règle R_1 n'est pas valide alors la règle R_2 ne le sera pas non plus.

Cela signifie par exemple, que si la règle d'association $AC \rightarrow DE$ n'est pas valide, par conséquent les règles $A \rightarrow CDE$ et $C \rightarrow ADE$ ne seront pas valides non plus et il n'est pas nécessaire de calculer leur confiance. Cette constatation permet de diminuer le nombre de règles d'association testées par l'algorithme. Réciproquement, la confiance de la règle $R_3 : (l_1 - l_2) \rightarrow l_2$ est supérieure ou égale à la confiance de la règle $R_4 : (l_1 - l_3) \rightarrow l_3$. Si la règle R_4 est valide alors la règle R_3 le sera également.

Cela signifie que si la règle d'association $A \rightarrow BC$ est valide, alors les règles $AB \rightarrow C$ et $AC \rightarrow B$ le seront également.

Algorithme de génération des règles d'association

F est un ensemble d'itemsets fréquents dans lequel chaque élément de cet ensemble possède deux champs qui sont l'itemset lui-même et son support. H_m représente l'ensemble des *m*-itemsets qui sont les conséquences de règles valides générées à partir de l'itemset l_k. L'algorithme considère successivement chaque itemset fréquent de *F* de taille supérieure à 1. Pour chacun de ces itemsets l_k, l'ensemble H_1 des itemsets de taille 1 qui sont des sous-ensembles de l_k est généré et pour chaque élément h_1 de H_1, la règle $(l_k - h_1) \rightarrow h_1$ est générée si sa confiance est supérieure ou égale à *minconf*. Sinon, si cette règle n'est pas valide, alors le 1-itemset h_1 est supprimé de H_1. Lorsque tous les 1-itemsets de H_1 ont été testés, H_1 contient la liste des 1-itemsets qui sont conséquences des règles valides générées à partir de l_k. Les règles valides générées à partir de l_k sont les règles dont l'union de l'antécédent et de la conséquence donne l'itemset l_k. La procédure Gen-Rules est alors appelée afin d'insérer dans *AR* les règles valides générées à partir de l_k dont la conséquence contient plus d'un item. L'algorithme se termine lorsque tous les k-itemsets fréquents pour $k \geq 2$ ont été considérés. L'ensemble *AR* renvoyé par l'algorithme contient alors toutes les règles d'association valides pour le seuil minimal de confiance *minconf* générées à partir de l'ensemble *F*.

Le pseudo-code de l'algorithme de génération de règles d'association est présenté dans l'algorithme (II.2).

Entrée : - un ensemble *F* des itemsets fréquents.

- un seuil minimal de confiance *minconf*.

Sortie : Un ensemble *AR* des règles d'association valides

1. **pour chaque** (*k*-itemsets fréquents $l_k \in F$ tel que $k \geq 2$) faire
2. $H_1 \leftarrow$ 1-itemsets étant des sous ensemble de l_k ;
3. **pour chaque** $(h_1 \in H_1)$ **faire**
4. *confiance* $(r) \leftarrow$ *support* (l_k)/*support* $(l_k - h_1)$;
5. **si** *confiance* $(r) \geq$ *minconf* **alors**
6. $AR \leftarrow AR \cup \{r : (l_k - h_1) \rightarrow h_1\}$ **;**
7. **sinon**
8. $H_1 \leftarrow H_1 \setminus h_1$
9. **fin si**
10. **fin pour**
11. Gen-Rules (l_k, H_1)
12. **fin pour**
13. retourner *AR*

Algorithme II.2 *: Algorithme de génération des règles d'association* (*GenRegleAssociation*)

Algorithme d'insertion des règles valides

La procédure *Gen-Rules* reçoit un *k*-itemset fréquent l_k, un ensemble H_m qui contient les *m*-itemsets qui sont les conséquences de règles valides générées à partir de l_k et du seuil minimal de confiance *minconf* . Elle met à jour l'ensemble *AR* de règles d'association en y insérant les règles valides générées à partir de l_k dont la conséquence est un (*m*+1)-itemset. Cette procédure est récursive et réalise en fin d'exécution un appel afin de générer à partir de l_k les règles valides dont la conséquence est un (m+2)-itemsets. Ces appels ce répète récursivement jusqu'à ce que les règles dont la conséquence est un ($|l_k|$-1)-itemset aient été insérées dans *AR*.

Le premier test correspond au test d'arrêt des appels récursifs de la procédure. Ces appels cessent lorsque l'ensemble H_m reçu comme paramètre contient des itemsets de taille $m=|l_k|-1$. Dans ce cas, toute les règles valides générées à partir de l_k, ont été insérées dans *AR*.

Ensuite, l'ensemble H_{m+1} des (m+1)-itemsets qui peuvent être des conséquences des règles valides générées à partir de l_k est crée. Cette création est réalisée en appliquant la procédure *Apriori-Gen,* présentée précédemment, à l'ensemble H_m des m-itemsets qui sont les conséquences de règles valides générées à partir de l_k. Chaque règle dont la conséquence est un (m+1)-itemset de H_{m+1} est alors testée. Si la règle testée est valide, elle est insérée dans *AR*. Sinon, le ($m+1$)-itemset qui est la conséquence est supprimé de H_{m+1}. Cette suppression correspond à la diminution du nombre de règles testées basée sur la propriété II.1.

En effet, si la règle d'association $AC \rightarrow DE$ n'est pas valide, l'itemset *DE* est supprimé de H_2. Lors de l'appel récursif suivant (avec h_2 en paramètre), les itemsets *CDE* et *ADE* ne seront pas crées par *Apriori-Gen* dans h_3 car *DE* est un sous-ensemble de *CDE* et *ADE*. Les règles $A \rightarrow CDE$ et $C \rightarrow ADE$ ne seront donc pas testées.

L'appel récursif de *Gen-Rules* est réalisé en fin de procédure avec comme paramètre l'itemset l_k et l'ensemble H_{m+1}.

Le pseudo-code de la procédure *GenRules* est présenté dans l'algorithme (II.3).

Entrée : - k-itemsets fréquent l_k.

- ensemble H_m de m-itemsets conséquences de règles valides générées à partir de l- k.
- un seuil minimal de confiance *minconf*.

Sortie : un ensemble AR de règles d'association valides augmenté des règles valides générées à partir de l_k dont la conséquence est un (m+1)-itemset

1. **si** $k > m+1$ **alors**
2. $H_{m+1} \leftarrow$ Apriori-Gen (H_m) ;
3. **pour chaque** $h_{m+1} \in H_{m+1}$ **faire**
4. *confiance* (r) $\leftarrow$ *support* (l_k) / *support* (l_k- h_{m+1}) ;
5. **si** *confiance* (r) $\geq$ *minconf* **alors**
6. $AR \leftarrow AR \cup \{r : (l_k - h_{m+1}) \rightarrow h_{m+1}\}$;
7. **sinon**
8. supprimer h_{m+1} de H_{m+1} ;
9. **fin si**
10. **fin pour**
11. Gen-Rules (l_k, H_{m+1}) ;
12. **fin si**

Algorithme II.3 *: Génération des règles d'association dans AR (GenRules)*

Exemple : génération des règles d'association

L'exemple précèdent induit les règles d'association suivants lorsque le seuil de minimum de confiance égal 1/2.

Règle	Confiance
Tomate $\rightarrow$ Fromage	3/5
Tomate $\rightarrow$ Pâte	5/5
Fromage $\rightarrow$ Pâte	3/3
Tomate $\rightarrow$ Fromage & Pâte	3/5
Fromage $\rightarrow$ Tomate & Pâte	3/3
Pâte $\rightarrow$ Tomate & Fromage	3/5

Table II.6 : *Exemple des règles d'association avec minconf=1/2*

II.3.4 Les amélioration de l'algorithme Apriori

L'algorithme *Apriori* [7] est un algorithme très utilisé afin d'extraire des itemsets fréquents. Ses performances diminuent en présence de données denses ou fortement corrélées. L'idée de base d'*Apriori* est de parcourir l'espace de recherche en "largeur d'abord", pour ne retenir que les itemsets fréquents et de générer d'autres éléments dans le niveau suivant par auto jointure. Les supports des itemsets candidats sont calculés et les candidats non fréquents sont supprimés. Cette suppression est basée essentiellement sur la propriété d'anti-monotonicité.

Cependant, cette approche souffre de la gestion du nombre de candidats qu'elle pourrait générer, surtout pour des contextes fortement corrélés et/ou des valeurs de support relativement faibles. Quelques algorithmes d'extraction d'itemsets fréquents basés sur l'algorithme *Apriori* ont été proposés dans le but d'améliorer son efficacité. Nous citons notamment :

L'algorithme AprioriTid [7]

Apriori Tid cherche à garder le contexte en mémoire afin de limiter les accès à la base. A supposer (ce qui est en général le cas sur des données réelles) que les ensembles de candidats décroissent en même temps que la taille des candidats, ce qui permet de diminuer la taille de la base de données progressivement. Dans ce cas, cet algorithme s'avère bien plus efficace qu'Apriori. Une approche mixte [7] de l'algorithme *Apriori* et *Apriori Tid* a été proposée, s'agit d'une hybridation des deux algorithmes.

L'algorithme Sampling [89]

Cet algorithme extrait un échantillon de la base qui tient en mémoire. A partir de cet échantillon, l'ensemble des itemsets fréquents dans l'échantillon, est construit ainsi que sa bordure négative constituée des itemsets non fréquents minimaux dont toutes les parties sont fréquentes, ce qui limite le risque de non exhaustivité.

L'algorithme Partition [90]

Cet algorithme partitionne la base de données en plusieurs sous ensembles pour l'extraction d'itemsets fréquents, lorsque les données ne tiennent pas en mémoire par exemple.

L'algorithme Count Distribution [8]

Cet algorithme est une version parallélisée de l'algorithme *Apriori*. Chaque processus traite sa portion locale de la base de transactions. Il minimise le traitement d'une base de données de grande taille impose de coûteuse opérations d'entrée/sortie.

II.3.5 Réduction de l'ensemble de règles d'association

Le nombre d'itemset fréquents extraits et leur taille moyenne étant élevés dans la plupart des cas [93], le nombre de règles d'association générées est encore beaucoup trop élevé. Ce nombre important de règles d'association extraites constitue un problème majeur pour la pertinence et l'utilité du résultat, problème qui est accentué par la présence de nombreuses règles redondantes et triviales. La réduction de l'ensemble de règles d'association générées afin d'améliorer la pertinence du résultat a fait l'objet de plusieurs travaux de recherche [88] [26] [44] [48]. Les approches proposées pour résoudre ce problème peuvent se classer en deux catégories : les approches orientées données, et les approches orientées utilisateur, Ces approches seront explicitées dans les sous sections suivantes.

II.3.5.1 Approche orientée données

Les approches orientées données se basent sur les propriétés structurelles des règles d'association afin de réduire l'ensemble des règles d'association extraites. Nous distinguons quatre catégories parmi ces approches :

- Approche qui utilise une taxonomie, ou hiérarchie de classe des items(règles d'association généralisées)
- Approche qui utilise des mesures statistiques autres que la confiance pour déterminer la précision des relations entre itemsets.
- Approche qui utilise des mesure de *déviation* des règles d'association autres que le support et la confiance.
- Approche qui consiste à supprimer les règles d'association redondantes de l'ensemble des règles d'association valides extraites.

A) Règles d'association généralisées

Les règles d'association généralisées [10] [91], également appelées règles d'association multi niveaux [11] [24], sont définies en utilisant une *taxonomie* des items du contexte d'extraction. Cette taxonomie est un DAG (Direct Acyclic Graph) dont les sommets sont les items et les arcs sont des relations *is-a* entre deux items. S'il existe un arc d'un item *i* vers un item *i'* dans la taxonomie alors *i* est appelé père de *i'* et *i'* est appelé fils de *i*. S'il existe un arc d'un item *i* vers un item *i'* dans la fermeture transitive de la taxonomie, *i* est appelé ancêtre de *i'* et *i'*est appelé descendant de *i*. Ce qui signifie que l'item *i* est une généralisation de l'item *i'* et que *i'* est une spécialisation de *i*. un item n'est pas un ancêtre de lui-même car le graphe est acyclique.

Soit:

T: Base de transaction,

I: Ensemble des items contenus dans les transactions,

R: Relation liant *O* et *I,*

τ : Une taxonomie

Les items *i* ($i \in I$) sont tels que descendant$(i) = \phi$ dans la taxonomie τ . Les items de la taxonomie qui possèdent des descendants sont des concepts de généralisation des items du contexte

Exemple :

N°	Transaction
1	Baskets, Caleçon
2	T-shirt, Pantalon
3	Baskets, Pantalon, Caleçon
4	T-shirt, Pantalon
5	Pantalon, Caleçon
6	Baskets, T-shirt

Table II.7 : *Base de transactions*

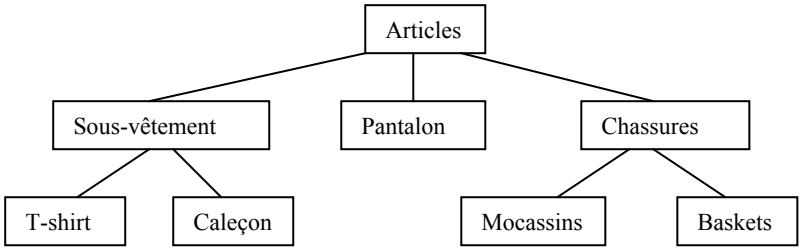


Figure II.3 : *Taxonomie τ des items de T*

1-Itemset généralisé	Support
{t-shirt}	3/6
{Caleçon}	3/6
{Pantalon}	4/6
{Baskets}	3/6
{sous-vetement}	6/6
{chaussures}	3/6

2-Itemset généralisée	Support
{caleçon, Pantalon}	2/6
{sous-vêtement, Pantalon}	4/6
{Caleçon,Basketes}	2/6
{Caleçon, Chaussures}	2/6
{sous-vêtement, Baskets}	3/6
{Sous-vêtement,Chaussures}	3/6
{t-shirt,Pantalon}	2/6

Table II.8 *: Itemsets généralisés fréquents pour minsupp =2/6*

Règles d'association généralisées	support	confiance
T-shirt→ Pantalon	2/6	2/3
Caleçon→Pantalon	2/6	2/3
Sous-vêtement→Pantalon	4/6	4/6
Pantalon→ Sous-vêtement	4/6	4/4
Caleçon→Baskets	2/6	2/3
Baskets→ Caleçon	2/6	2/3
Caleçon→chaussures	2/6	2/3
Chaussures→Caleçon	2/6	2/3
Baskets→ Sous-vêtement	3/6	3/3
Chaussures→ Sous-vêtement	3/6	3/3

Table II.9 : *Règles d'association généralisées valides pour Minsupp=2/6 et Minconf=2/3*

Les règles d'association généralisées sont des règles d'association entre ensembles d'items pouvant appartenir à différents niveaux de la taxonomie. Elles sont de la forme :

$R : l_1 \rightarrow l_2$ Avec $l_1 \cap l_2 = \phi$

Telles que aucun item de l_2 n'est ancêtre d'un item de l_1 (pour éviter les règles triviales du genre $i \rightarrow ancêtre\ (i)$).

Les règles d'association généralisées valides sont celle dont le support et la confiance sont supérieurs ou égaux aux seuils minimaux *minsupp* et *minconf* respectivement. Les itemsets fréquents, à partir desquels les règles d'association généralisées sont générées, sont constituées des items de la taxonomie τ

L'ensemble des itemsets généralisés fréquents est un sur ensemble de l'ensemble des itemsets fréquents. L'ensemble des règles d'association généralisées est donc un sur ensemble de l'ensemble des règles d'association valides.

Toutefois, il est possible de supprimer certaines règles d'association généralisées valides lorsque celles-ci sont résumées par une règle plus générale. Une règle d'association généralisée $R' : A' \rightarrow C$ est plus générale qu'une règle $R : A \rightarrow C$ si les items de *A'* sont des ancêtres des items de *A*. La règle *R* est résumée par la règle *R'* s'il est possible de réduire *R*, son support et sa confiance à partir de la règle *R'*et des supports des itemsets *A* et *A'*. Dans ce cas la règle *R* est redondante par rapport à la règle *R'*.

Trois algorithmes d'extraction des itemsets généralisés fréquents nommés *Basic, Cumulate* et *EstMerge* ont été proposés dans [10]. Pour l'algorithme *Basic*, les objets du contexte d'extraction sont remplacés par des objets "étendus" contenant tous les items de l'objet original ainsi que les items ancêtres de ces items. A partir de ces objets étendus, une version modifiée de l'algorithme Apriori extrait les itemsets généralisés fréquents. Dans l'algorithme *Cumulate,* une liste des items ancêtre de chaque item de la taxonomie est construite et les items les moins généraux sont remplacés au cours de processus par leur item *père* dans les itemsets déterminés infréquents après un balayage du contexte. Dans l'algorithme *EstMerge*, les itemsets généralisés fréquents constitués des items les plus généraux et substitue successivement chaque item dans ces derniers par ses items *fils* jusqu'à ce que l'itemset produit soit infréquent.

B) *Mesures de déviation*

Les mesures de déviation sont des mesures de distance entre règles d'association définies en fonction de leur support et confiance. Elles sont utilisées de deux manières distinctes afin de supprimer certaines règles de l'ensemble des règles d'association découvertes.

- La première consiste à identifier les règles d'association fortement semblables, caractérisées par une faible distance, et classer ou supprimer certaines de ces règles en fonction des mesures de déviation qui leurs sont associées [12], [13].

- La seconde permet d'identifier les règles d'association qui sont inattendues pour l'utilisateur et apportent donc une connaissance importante car nouvelle [15], [16]. Les connaissances de l'utilisateur sont représentées en utilisant des modèles probalistiques auxquels sont confrontés les règles d'association extraites. La déviation d'une règle correspond à la différence entre la valeur attendue pour la règle dans le modèle probabiliste et la valeur réelles pour la règle dans le contexte. Nous pouvons citer :

- Une mesure de déviation basée sur la confiance des règles d'association a été proposée dans [17]. Cette mesure est calculée pour chaque règle d'association R parmi un ensemble $\boldsymbol{R}$ de règles extraites. On note $moyenne_{conf}(\boldsymbol{R})$ la valeur moyenne et $déviation_{conf}(\boldsymbol{R})$ la déviation standard des confiances des règles de l'ensemble $\boldsymbol{R}$. La mesure de distance associée à une règle d'association $R : l_1 \rightarrow l_2$ dans l'ensemble $\boldsymbol{R}$.

$$distance_{std}(R,\boldsymbol{R}) = (confiance\ (R) - moyenne_{conf}(\boldsymbol{R})) - déviation_{conf}(\boldsymbol{R})$$

L'approche proposée dans [17], consiste à fractionner l'ensemble AR des règles d'association valides extraites en plusieurs sous-ensembles et supprimer dans chacun de ces sous-ensembles $\boldsymbol{R}$ les règles R dont la $distance_{std}(R, \boldsymbol{R})$ est inférieure à un seuil minimal défini par l'utilisateur.

- Une mesure de déviation spécifiant une distance entre deux règles d'association possèdant la même conséquence a été proposée dans [14]. La distance entre deux règles est définie en fonction des ensembles d'objets qui vérifient chacune des deux règles. Soit $O(I)$ le nombre d'objet du contexte d'extraction contenant l'itemset l.

Pour deux régles d'association $R : A \to C$ et $R' : A' \to C$ la distance entre *R et R'* est :

$$distance_{sem}(R,R') = O(A \cup C) + O(A' \cup C) - 2 \times O(A \cup A' \cup C)$$

La valeur obtenue correspond au nombre d'objets du contexte qui vérifient l'une des deux règles mais pas l'autre. L'auteur a utilisé cette mesure afin de faciliter la visualisation de l'ensemble des règles d'association extraites en formant des groupes de règles qui concernent les mêmes contextes. Les groupes sont construits en minimisant la distance entre les règles de chaque groupe.

C) *Couverture structurelle*

Dans [91], l'ensemble des règles d'association valides *AR* est extrait et une *couverture structurelle* pour les règles d'association est construite en supprimant de cet ensemble les règles redondantes d'un point de vue syntaxique. Cette approche ne tient pas compte de la confiance des règles, ce qui entraîne une perte d'information importante dans de nombreux cas. De plus, la construction de la couverture nécessite pour chaque règle la liste des objets contenant l'itemset union de l'antécédent et de la conséquence de la règle. La couverture structurelle Δ d'un ensemble *AR* de règles d'association est définie par :

$$\Delta = \left\{ A \to C \in AR \middle| \exists A' \to C \in AR \ telle \ que \ A' \subset A \right\}$$

Un algorithme de génération de couvertures structurelles appelé *RuleCover* est présenté dans [14]. Afin de générer la couverture structurelle, cet algorithme requiert qu'a chaque règle d'association de *AR* soit associée la liste des objets vérifiant la règle, c'est-à-dire la liste des objets contenant l'itemset union de l'antécédent et de la conséquence de la règle.

***Définition d'une règle d'association redondante* [27]**

Soit *AR*, l'ensemble de règles d'association générées, Une règle $\left(R_1 : X \to Y \in AR \ / conf = 0\right)$ est considérée comme redondante par rapport à une autre règles $R_2 : X_1 \to Y_1 \ / conf = c$ si R1 vérifie les deux conditions suivantes :

1. Supp(R_1)=Supp(R_2) et conf(R_1)=conf(R_2)=c
2. $(X_1 \subset X \ et \ Y \subset Y_1)$ où $\left(X_1 = X \ et \ Y \subset Y_1\right)$.

II.3.5.2 Approche orientée utilisateur

Les approches orientées utilisateur pour la réduction de l'ensemble des règles d'association requièrent l'intervention de l'utilisateur afin de définir des critères de sélection des règles qui figureront dans l'ensemble résultat. Nous distinguons trois catégories différentes parmi ces approches :

- L'utilisation d'expressions régulières appelées *templates utilisées afin de filtrer l'ensemble de règles valides extraites précédemment*
- L'utilisation d'un opérateur appelé *MINE RULE* qui est une extension du langage SQL
- L'utilisation de critères de sélection appelés *contrainte sur les items* pour extraire seulement les itemsets fréquents permettant de générer les règles vérifiant ces contraintes.

A. *Templates*

Dans [18], les templates sont définis comme des expressions régulières spécifiant des critères généraux de sélection d'un sous-ensemble est construit à partir de l'ensemble des règles d'association valides, en conservant les règles qui vérifient les critères spécifiés par les templates parmi cet ensemble. Ces critères stipulent quels items doivent apparaître dans l'antécédent et la conséquence des règles du sous-ensemble.

Les templates sont des expressions de la forme : $X_1 \cap ... \cap X_j \rightarrow X_{j+1}$. Dans laquelle chaque X_h est un item $i \in I$.

B. *Opérateur Min Rule*

Dans [19], les critères de sélection des règles sont définis par un opérateur nommé *MIN RULE*. Cet opérateur est une extension du langage de requête SQL implémenté dans les SGBD relationnels. L'extraction des règles d'association est réalisée à partir des tuples des relations de la base de données par une requête qui est une instance de l'opérateur *MIN RULE*.

Les itemsets fréquents sont des ensembles de valeurs d'attributs de tuples extraits des tables (ou requêtes) qui sont les sources de données de la requête. Les règles d'association sont générées à partir de ces itemsets fréquents selon les critères définis par la requête.

C. *Contraintes sur les items* [92]

Dans l'approche par contraintes sur les items, l'utilisateur définit des contraintes, spécifiant les règles d'association à extraire, qui sont des expressions portant sur l'antécédent, la conséquence ou l'antécédent et la conséquence simultanément des règles.

Ces contraintes sont utilisées lors de la phase d'extraction des itemsets fréquents afin de limiter l'espace de recherche de cette phase. Elles sont prises en compte lors de la génération des itemsets candidats afin de considérer seulement les candidats permettant de générer des règles satisfaisant les contraintes.

II.3.6 Domaine d'applications

Le fait d'identifier des relations significatives entre les données contenues dans une base peut s'avérer utile à différentes domaines commerciaux, scientifiques est industriels dans leurs but de rentabiliser leur profit.

Plusieurs travaux basés sur la recherche des règles d'association ont été appliqués dans des applications réelles comme :

- la planification commerciale [1] [5]
- les réseaux de télécommunication [84] [61]
- la recherche médicale [65]
- les données de web [85], [19]
- le marketing bancaire [86]
- le multimédia [87]

Selon ces domaines d'application on constate que les règles d'association traitent des données de différents types.

II.3.7 Types des données considérées

D'après [20] [82], on peut constater que les données peuvent être de différents types, quantitatives, qualitatives.

Cas quantitatif

Une variable quantitative prend des valeurs entières ou réelles, elle est dite alors discrète ou continue.

Cas qualitatif

Par définition, les observations d'une variable qualitative ne sont pas des valeurs numériques, mais des caractéristiques, appelées *modalités* [20]. Lorsque ces modalités sont naturellement ordonnées (par exemple, la mention au bac ou une classe d'âge), la variable est dite *ordinale*. Dans le cas contraire (par exemple, la profession dans une population de personnes actives ou la situation familiale) la variable est dite *nominale*.

II.4 Règles d'association quantitatives

Jusque là, nous avons traité le cas des règles d'association portant sur des attributs booléens [7] [8] [11] [12] [50]. Par exemple, la transaction 1 de la table (II.4) exprime la présence de tomates, fromage, pâte. Cette transaction n'indique pas la quantité de tomates achetées.

Nous traitons dans ce paragraphe le cas des règles d'association sur des attributs quantitatifs. La recherche des règles d'association quantitatives a été introduite par Agrawal et al. [21]. Agrawal propose de découper le domaine où prend ses valeurs une donnée numérique en différents intervalles. Ces intervalles sont disjoints et l'appartenance d'une valeur à un intervalle est binaire.

L'algorithme présenté dans [21] est décrit par les étapes suivantes :

1- Déterminer le nombre de partition pour chaque attribut quantitatif puis partitionner le domaine de chaque attribut à des petits intervalles appeler intervalles de base

2- Fusionner les intervalles adjacents à un intervalle large, telle que l'intervalle large aura un support suffisant.

3- Faire correspondre à chaque intervalle un entier positif

4- Remplacer la valeur originale d'un attribut par l'entier positif correspond à l'intervalle qui présente.

Nous pouvons remarquer qu'à travers ces étapes, l'auteur fait un mapping du cas quantitatif au cas booléen.

Exemple

La table II.10 présente l'exemple introduit dans [21], cet exemple illustre l'application de l'algorithme introduit par Agrawal et al. On considère *Minsupp*=2/5 et *minconf* =1/2

TID	Age	Marier	Numvoiture
100	23	Non	1
200	25	Oui	1
300	29	Non	0
400	34	Oui	2
500	38	Oui	2

L'ensemble de données

Intervalle	Nom
20…24	I_1
25…29	I_2
30…34	I_3
35....39	I_4

Partitionnement et mapping de l'attribut *Age*

TID	Age	Marier	Numvoiture
100	20..24	Non	0
200	25..29	Oui	1
300	25..29	Non	1
400	30..34	Oui	2
500	35..35	oui	2

Après partitionnement

valeur	nom
0	P_1
1	P_2
2	P_3

Partitionnement et mapping *numvoiture*

L'attribut *Marier* est un attribut booléen , l'auteur associe la valeur 1 si la valeur de l'attribut *Marier* est Oui, est la valeur 0 si la valeur de l'attribut *Marier* est Non.

TID	Age. I_1	Age. I_2	Age. I_3	Age. I_4	Marier	Numvoiture. P_1	Numvoiture. P_2	Numvoiture. P_3

100	1	0	0	0	0	1	0	0
200	0	1	0	0	1	0	1	0
300	0	1	0	0	0	0	1	0
400	0	0	1	0	1	0	0	1
500	0	0	0	1	1	0	0	1

Après le mapping de tous les attributs

Itemset	Support
<Age : 20…29>	3/5
<Age : 30…39>	2/5
<Marier : Oui>	3/5
<Marier : Non>	2/5
<Num voiture : 0…1>	3/5
<Age : 30…39><Marier : Oui>	2/5

itemsets fréquents

Règles d'association	Support	Confiance
<Age : 30…39> and <Marier : oui> $\rightarrow$ <Num voiture: 2>	40%	100%
<Age : 20…29> $\rightarrow$ <Num voiture : 0…1>	60%	66.6%

Règle d'association

Table II.10 : *exemple de problème des règles d'association quantitatives*

Après le mapping du cas quantitatif au cas booléen, la génération des règles d'association peut être réalisée par l'application de l'algorithme *Apriori* introduit dans le paragraphe II.3.3.

Cette technique de partitionnement suppose d'abord un nombre empirique de partitions pour chaque attribut quantitatif puis projette toutes les valeurs possibles sur un ensemble d'items consécutifs. Cependant cette méthode engendre un problème de valeur aux limites entre les intervalles.

Pour remédier à ce problème Agrawal et al. [21] proposent une mesure (appelée partial completeness) qui quantifie l'information perdue à cause de ce partitionnement. Cette mesure est utilisée pour déterminer le nombre de partitions.

II.4.1 Problème des règles d'association quantitatives

Dans ce paragraphe en met en évidence le problème des règles d'association quantitatives, on utilise l'exemple présenté précédemment. Deux cas sont envisagés. Un nombre élevé d'intervalles (des intervalles petits) et un nombre réduit d'intervalles (des intervalles larges).

1.***Petits intervalles***

On considère dans ce cas notre partition comporte un nombre élevé d'intervalles. Conséquemment, les intervalles seront de taille réduite. Donc le partitionnement de l'attribut *Age* peut être présenté par la figure (II.4).

Nous considérons que le *Minsupp*=2/5.

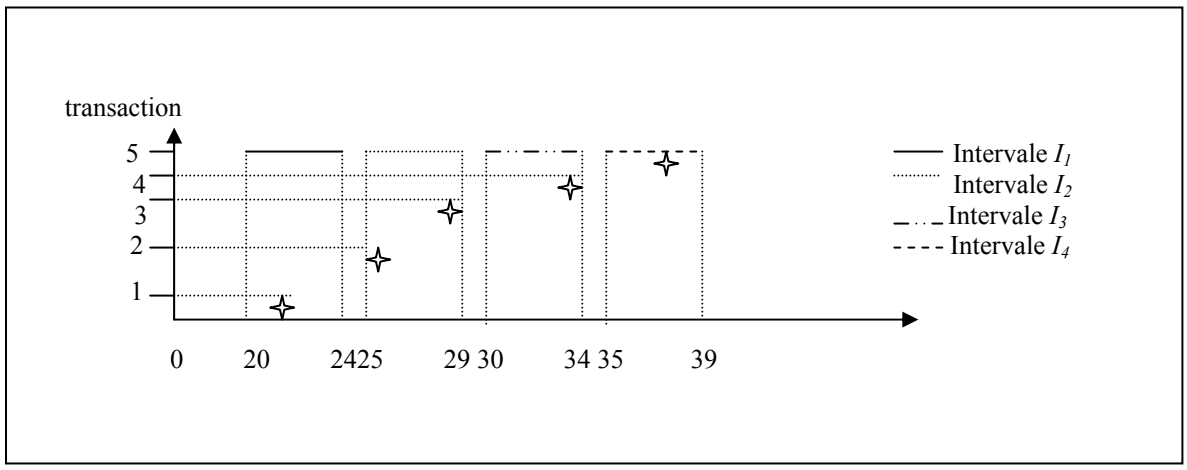


Figure II.4 : *partitionnement de l'attribut Age*

Pour ce cas, *support*(I_1)=1/5, *support*(I_2)=2/5, *support*(I_3)=1/5, *support*(I_4)=1/5

L'intervalle I_1, l'intervalle I_2 et l'intervalle I_4, ne vérifient pas le *minsupp,* alors pour l'extraction des itemsets fréquents ces intervalles seront rejetés, et par ce que l'extraction des règles d'association ce fait après l'extraction des itemsets fréquents, ce cas peut engendrer une

perte d'information. Pour ce cas on aura le problème de support insuffisant, ce qui engendre une perte d'information.

2.***Intervalles larges***

Nous supposons maintenant que le nombre d'intervalles est réduit, aussi les intervalles deviennent plus larges (cf. Figure II.5)

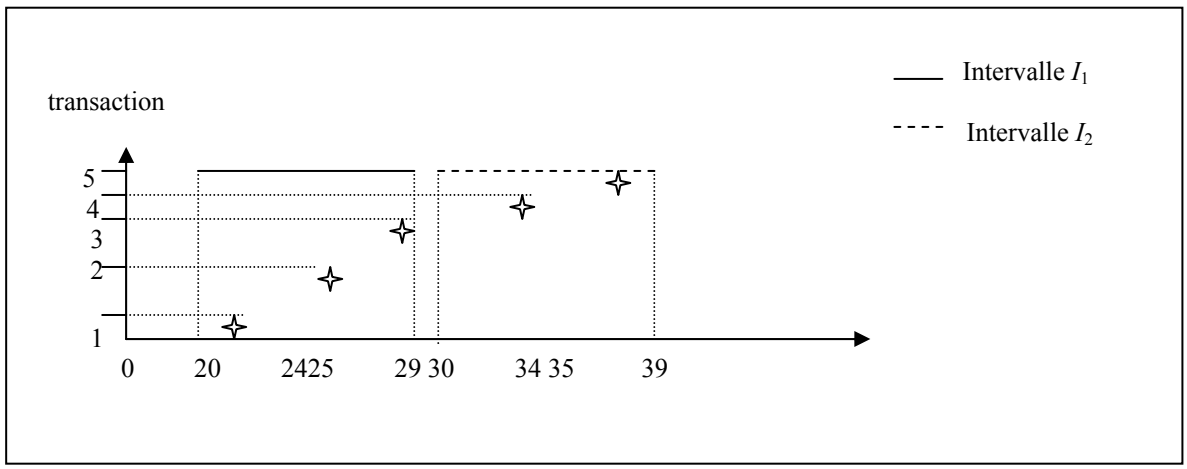


Figure II.5 : *partitionnement de l'attribut Age*

Dans ce cas, *support* (I_1)=3/5, *support*(I_2)=2/5, les deux intervalles vérifient le *minsupp*. Mais cette solution introduit le problème de confiance, car $conf(A \Rightarrow B) = \frac{Supp(A \cup B)}{\text{supp}(A)}$, alors si supp($A$) est grand, $conf(A \Rightarrow B)$ diminue, donc il existe un risque de rater les règles vraiment intéressantes pour l'utilisateur, vu qu'elle ne rempliront pas la condition de confiance minimale

II.4.2 Constat

Afin d'apporter une solution au dilemme ainsi mis en évidence, un partitionnement graduelle des attributs quantitatifs a été proposé [42]. Un telle partitionnement consiste à attribuer à chaque élément de l'intervalle un degré de vérité qui mesure le degré d'appartenance de cet élément à l'intervalle. La figure (II.6) illustre cette représentation.
Nous considérons que le *minsupp*= 2/5

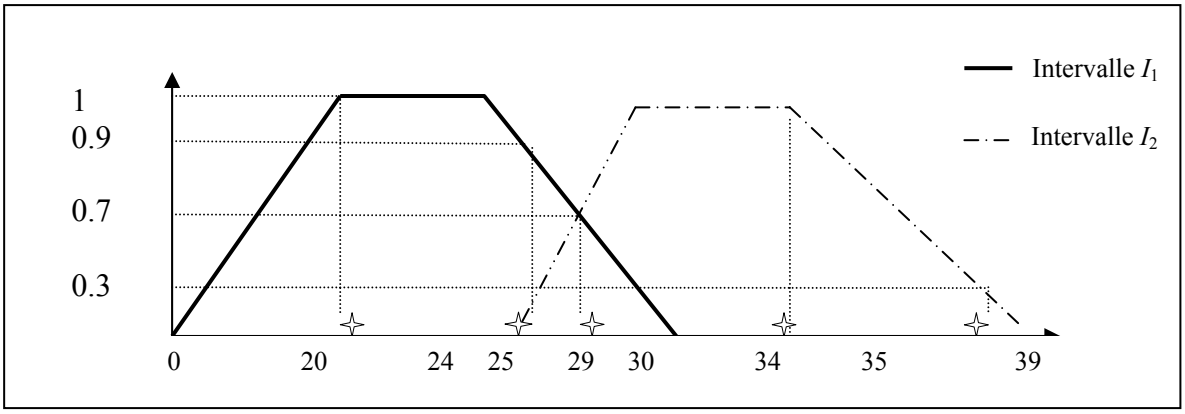


Figure II.6 : *partitionnement de l'attribut Age*

Par cette présentation on aura deux intervalles. Le support de chaque intervalle sera calculer comme suit :

support (I_1)=support(23)+support(25)+ support(29)= 1+0.9+0.7=2.6

support (I_2)= support(29)+support(34)+support(38)=0.7+1+0.3=2

Par cette présentation il y'aura moins de perte d'information, car chaque élément appartient à tous les intervalles avec un degré particulier.

La théorie qui nous permet de modéliser une telle gradualité est la théorie des sous ensembles flous (cf. chapitre III).

II.5 Conclusion

Nous venons de présenter dans ce chapitre le fondement de l'ECD ainsi que la recherche de règle d'association. Nous avons situé cette problématique au sein d'un processus d'ECD.

Nous avons cité également le problème des règles d'association quantitatives, et nous avons montré les conséquences du choix empirique d'une partition sur les mesures de support et confiance et conséquemment les conséquences sur la découverte de règles.

Nous avons vu que la théorie des ensembles flous permet un partitionnement graduel et par conséquent permet de diminuer la perte d'information. Pour cela, nous présentons cette théorie dans le chapitre suivant. Nous y présentons aussi les règles d'association floues.

Nous avons vu dans le chapitre précédent que les ensembles flous ont été proposés par certains auteurs [42] afin de définir une partition floue du domaine des valeurs d'un attribut quantitatif. L'utilisation de tels ensembles flous entraîne la découverte de règles d'association dites floues.

Aussi, nous présentons d'abord dans ce chapitre la théorie des sous-ensembles flous et nous détaillons ensuite les règles d'association floues.

III.1 La théorie des sous ensembles flous

La théorie des sous ensembles flous a été introduite en 1965 par le professeur Lotfi Zadeh [95]. Zadeh a constaté que les ensembles classiques sont incapables de représenter des données subjectives. Le concept de base de la théorie des sous ensembles flous consiste en l'appartenance graduelle à un ensemble, ce qui constitue un bon moyen de formaliser les processus de raisonnement humain. Ce qui permet aussi de représenter des données subjectives et a donné, par la même, naissance à la logique floue qu'il ne faut d'ailleurs pas confondre avec la théorie des ensembles flous.

Considérons aussi qu'un sous-ensemble quelconque est lui-même un ensemble, nous confondrons dorénavant ensemble flou et sous-ensemble flou et utiliserons indifféremment dans ce qui suit ces deux terminologies. Nous donnons ci-après les concepts et notions utilisées dans le cadre de notre étude.

III.1.1 Ensemble classique et Ensemble flou

Ensemble classique

Dans la théorie ensembliste classique, un sous-ensemble d'un ensemble de référence est constitué d'éléments qui possèdent une ou plusieurs propriétés communes caractéristiques de ce sous-ensemble. Chaque élément "*e*" de l'univers de référence Ω est ainsi caractérisé par son appartenance ou sa non appartenance aux différents sous-ensembles E_i définis.

La notion d'appartenance s'exprime par la fonction caractéristique appelée aussi fonction indicatrice. Cette fonction est définie pour chaque sous-ensemble E_i comme suit :

$$\mu_{E_i} : \Omega \to \{0,1\} \qquad e \mapsto \mu_{E_i}(e) = \begin{cases} 1 & \text{si} \quad e \in E_i \\ 0 & \text{sinon} \end{cases} \qquad \text{(III.1)}$$

Exemple

Par exemple, si E est l'ensemble des tailles des personnes vivant en Algérie, le sous ensemble E_{grand} correspondant aux grandes tailles est défini par :

$$\mu_{E_{grand}} = \begin{cases} 1 & \text{Si} \quad e \geq 180\ cm \\ 0 & \text{Sinon} \end{cases}$$

Ce concept ce traduit par la fonction illustrée dans la figure (III.1)

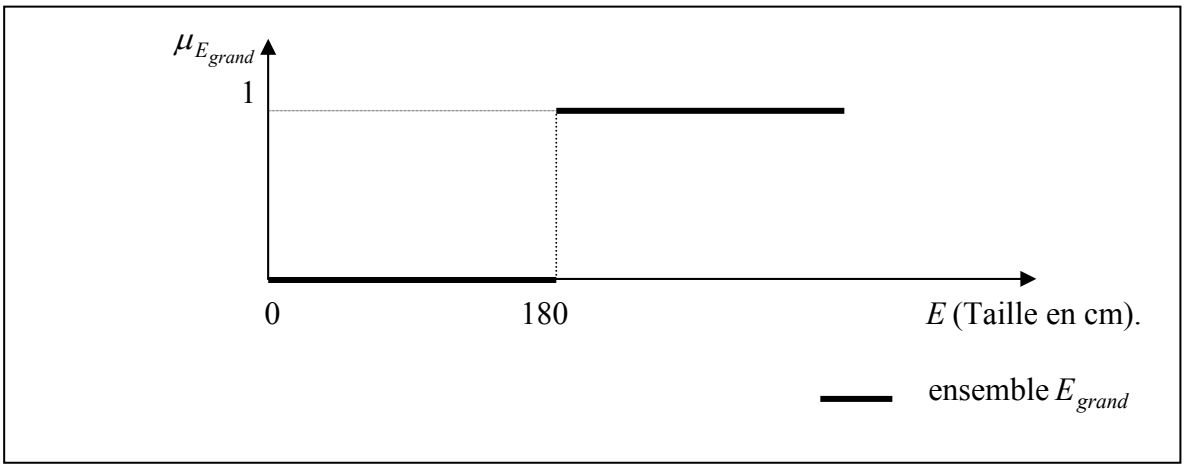


Figure III.1 *: Fonction caractéristique de* E_{grand} (ensemble classique)

III.1.1.2 Ensemble flou

Les limites de la représentation classique apparaissent très clairement. La grande taille d'une personne est un concept que l'être humain est capable de formuler et d'utiliser pour raisonner. Sa définition se fonde essentiellement sur des observations et l'expérience acquise dans son environnement.

Le concept d'ensemble flou généralise le concept d'ensemble classique en associant à chaque élément "un degrés d'appartenance" défini dans l'intervalle réel [0,1] [73] [95]. Le degré "0" indique qu'un élément n'appartient pas du tout à l'ensemble et le degré "1" indique qu'un élément appartient "totalement" à l'ensemble. Les ensemble flous autorisent donc des états intermédiaires entre la non appartenance et l'appartenance totale.

Définition (Ensemble flou)

On reprend la définition de Zadeh citée en [95] des sous ensembles flous:

Un ensemble flou E *d'un ensemble de référence* Ω *est entièrement défini par l'application* μ_E *de* Ω *dans* [0,1]. *Cette application étant interprétée comme le degré d'appartenance des éléments de* Ω *à* E.

Donc, un ensemble flou E_i étend la notion d'ensemble classique en généralisant la fonction caractéristique μ_{E_i} par la fonction d'appartenance $\mu_{E_i} : \Omega \to [0,1]$. Ceci permet de rendre compte de l'appartenance partielle (ou encore graduelle) d'un élément au sous-ensemble flou. Ainsi l'appartenance ensembliste est représentée par une variable qui appartiennent à l'intervalle [0,1] et non plus à $\{0,1\}$.

Exemple

En reprenant l'exemple sur les personnes de grande taille, la figure (III.2) illustre la fonction d'appartenance μ_E de l'ensemble flou E_{grand}.

On peut remarquer que la forme de la fonction d'appartenance pour les ensembles classiques est linéaire (cf. figure III.1). Toutefois, la théorie des ensembles flous ne limite pas la forme d'une fonction d'appartenance et différentes formes sont possibles (cf. III.1.2).

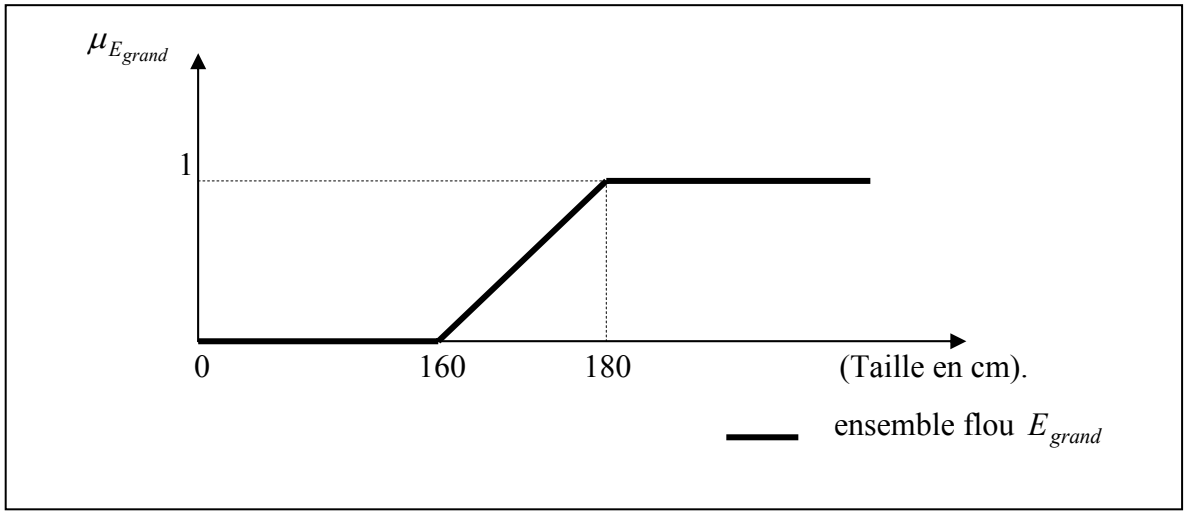


Figure III.2 : *Fonction d'appartenance à* E_{grand}

Par cette présentation, on modélise une transition graduelle entre l'état "grand" et l'état "moins grand", ce qui traduit que les personnes de plus de 180cm sont considérées comme étant grandes et que celles dont la taille se trouve entre 160 et 180cm le sont plus au moins. Il n'y a donc pas la transition stricte illustrée dans la figure III.1 qui indique que l'on devient brusquement grand par une taille de 180cm, et que l'on n'est pas considéré comme grand quand on mesure 179cm.

On remarquera à ce stade de la présentation qu'en dehors de la prise en compte des imprécisions, les sous-ensembles flous permettent de représenter des concepts de façon intuitive, ce qui facilite leur compréhension. De plus, de part leur granularité, les ensembles flous sont parfaitement adaptés pour représenter des données de façon synthétique et compacte, notamment lorsque l'univers de référence est numérique. Par exemple, si un échantillon de 10000 personnes est utilisé pour étudier la taille d'une population, celui-ci peut être synthétisé par trois (voir plus) ensembles flous (petit, moyen et grand), évitant ainsi d'avoir à manipuler et/ou stoker l'ensemble des données.

III.1.2 Fonction d'appartenance

Théoriquement, un ensemble flou est complètement défini par la donnée de sa fonction d'appartenance. Aussi, toutes les opérations possibles sur les ensembles flous ont été

définies à travers les fonctions d'appartenances. Les fonctions d'appartenances possèdent les caractéristiques suivantes :

Le type

Le type correspond à la forme de la fonction, par exemple triangulaire, trapézoïdale, gaussienne (Cf. figureIII.3).

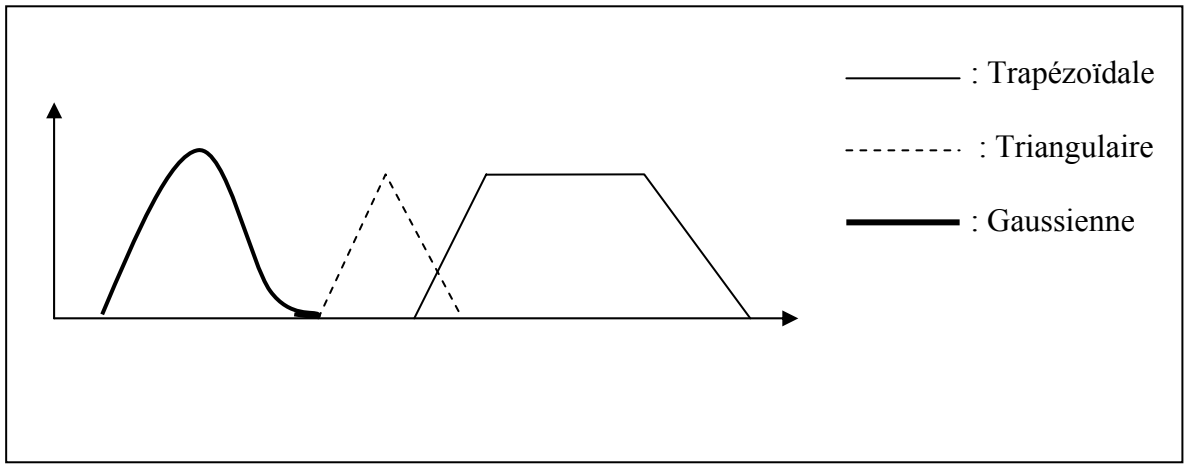


Figure III.3 : *Exemple de fonction d'appartenance*

Le noyau

Le noyau *Ker*(*A*) d'un ensemble flou *A* est l'ensemble des éléments de Ω dont le degré d'appartenance à *A* vaut 1 :

$$Ker(A) = \{ e \in \Omega \ / \ \mu_A(e) = 1 \ \} \qquad \text{(III.2)}$$

Pour les triangles, le noyau est plus communément appelé valeur modale car il se réduit à la valeur du sommet.

La hauteur

La hauteur *h (A)* d'un ensemble flou *A* est la valeur maximale de la fonction d'appartenance.

$$h\,(\mathrm{A}) = \max \mu_A(e) \qquad e \in \Omega \qquad \text{(III.3)}$$

Si $h(A)=1$, on dit que le sous-ensemble flou A est normalisé.

Les coupes de niveau α

On appelle coupe de niveau α ou α-coupe (en anglais α-cut) d'un ensemble flou A, l'ensemble A_α de tous les éléments de Ω qui appartiennent à A avec un degré au moins égal à α :

$$A_\alpha = \{e \in \Omega / \mu_A(e) \geq \alpha \} \quad \text{(III.4)}$$

La notion de α-coupe est très largement utilisée car elle permet à partir de sous-ensembles flous de revenir à la notion d'ensemble classique.

Le support

Le support noté *Support* (A) d'un ensemble flou A est l'ensemble des éléments de Ω dont le degré d'appartenance à A est non nul :

$$Support\ (A) = \{e \in \Omega / \mu_A(e) > 0\} \quad \text{(III.5)}$$

La figure III.4 illustre les principales notions définies ci-dessus.

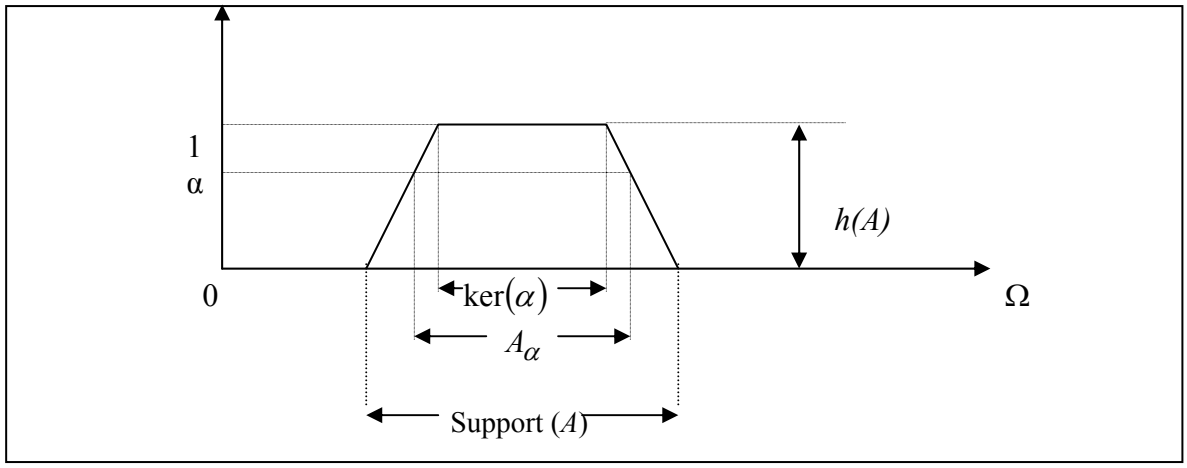


Figure III.4 : *Principaux éléments caractéristiques d'un sous ensemble flou A*

III.1.3 La cardinalité

La cardinalité d'un ensemble classique correspond au nombre d'éléments qui appartiennent à cet ensemble. La cardinalité $|A|$ d'un ensemble flou A est la quantité d'éléments de Ω qui appartiennent à A.

Nous détaillons plus particulièrement la notion de cardinalité d'un ensemble flou car elle sera largement utilisée dans notre travail pour le calcul du support et de la confiance d'une règle d'association floue.

III.1.3.1 La cardinalité scalaire $\sum$-count

Soit un ensemble flou A défini sur un univers de référence Ω (Ω fini). La cardinalité scalaire [108] de l'ensemble flou A (noté $|A|$) est définie comme suit :

$$|A| = \sum_{e \in \Omega} \mu_A(e) \qquad \text{(III.6)}$$

III.1.3.2 La cardinalité floue de Zadeh

La première définition d'une cardinalité floue d'un ensemble flou est due à Zadeh [109]. Elle est basée sur une coupe de niveau α .

$$Z(A,k) = \begin{cases} 0 & \text{si il existe pas un } \alpha \,/\, |A_\alpha| = k \\ \max \{\alpha \,/\, |A_\alpha| = k\} & \text{sinon} \end{cases} \qquad \text{(III.7)}$$

A_α : représente une coupe de niveau α , $k \in \mathrm{N}$.

III.1.3.3 Cardinalité relative d'un ensemble flou

La cardinalité relative [72] d'un ensemble A sachant D est la mesure du pourcentage d'éléments de D qui sont aussi élément de A.

$$\text{Rel card } (A/D) = \frac{card(A \cap D)}{card(D)} \qquad \text{(III.8)}$$

Cette cardinalité est souvent utilisé dans le cas de règle d'association basée sur l'approche sémantique.

III.1.4 Opérations sur les sous-ensembles flous

La théorie des sous-ensembles flous étant une généralisation de la théorie ensembliste classique, la plupart des opérations ensemblistes classiques ont été généralisées aux ensembles flous. Nous présentons ici les principales opérations ainsi définies sur les ensembles flous.

III.1.4.1 L'égalité

Deux sous-ensembles flous A et B sont égaux si et seulement si leur fonctions d'appartenances sont égales en tout point de Ω :

$$A=B \Leftrightarrow \forall e \in \Omega,\ \ \mu_A(e) = \mu_B(e) \qquad \text{(III.9)}$$

III.1.4.2 L'inclusion

L'ensemble flou A est inclus dans l'ensemble flou B, si et seulement si leur fonctions d'appartenances vérifient la condition suivante :

$$A \subseteq B \Leftrightarrow \forall e \in \Omega, \mu_A(e) \leq \mu_B(e) \qquad \text{(III.10)}$$

III.1.4.3 L'union :

L'union de deux ensembles flous est une opération ensembliste définie à partir d'un opérateur appelé *t-conorme* (conorme triangulaire notée $\perp$). L'union de deux ensembles flous A et B est un ensemble flou C. La fonction indicatrice de C $\left(\mu_C = \mu_{A \cup B}\right)$ est définie à travers une *t-conorme* comme suit :

$$\forall e \in \Omega, \mu_c(e) = \perp (\mu_A(e), \mu_B(e)) \qquad \text{(III.11)}$$

Un exemple illustrant l'union de deux sous-ensembles flous est présenté sur la figure (III.6) dans le cas où l'operateur max est choisie comme *t-conorme*.

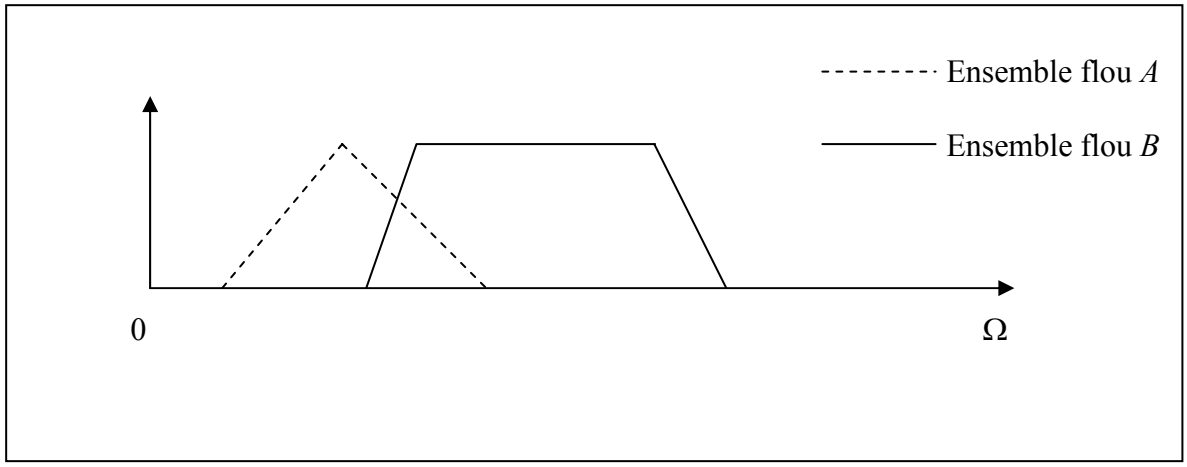


Figure III.5 *: fonction d'appartenance des ensemble flou A et B*

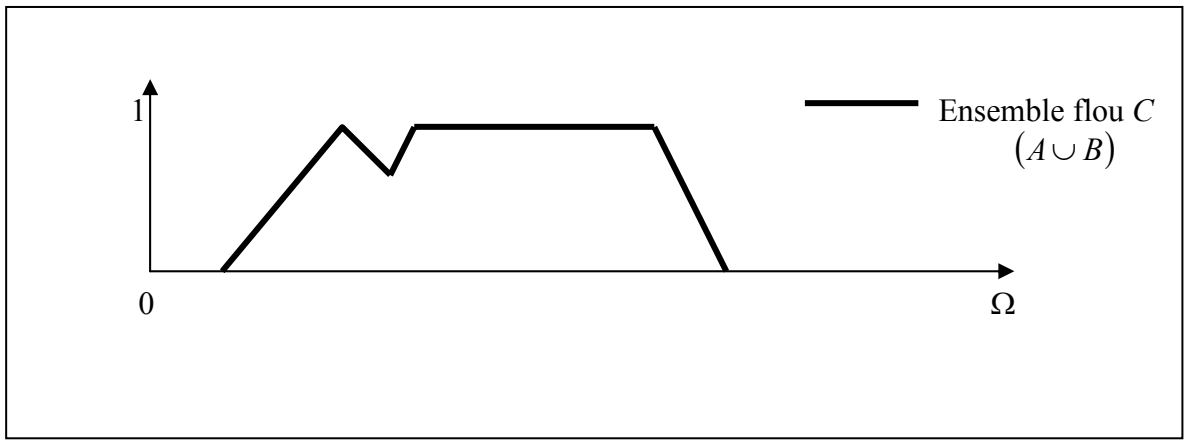


Figure III.6 *: Résultat de l'union des ensemble flou A et B*

III.1.4.4 L'intersection

L'intersection de deux ensembles flous est une opération ensembliste définie à partir d'un opérateur appelé *t-norme* (norme triangulaire notée $\top$). L'intersection de deux ensembles flous *A* et *B* est un ensemble flou *C*. La fonction indicatrice de *C* $(\mu_C = \mu_{A\cap B})$ est définie à travers une *t-norme* comme suit :

$$\forall e \in \Omega, \quad \mu_c(e) = \mu_{A\cap B}(e) = \top\left(\mu_A(e), \mu_B(e)\right) \qquad \text{(III.12)}$$

Un exemple illustrant l'intersection entre deux ensembles flous est présenté sur la figure (III.7) dans le cas où l'operateur min est choisie comme *t-norme*.

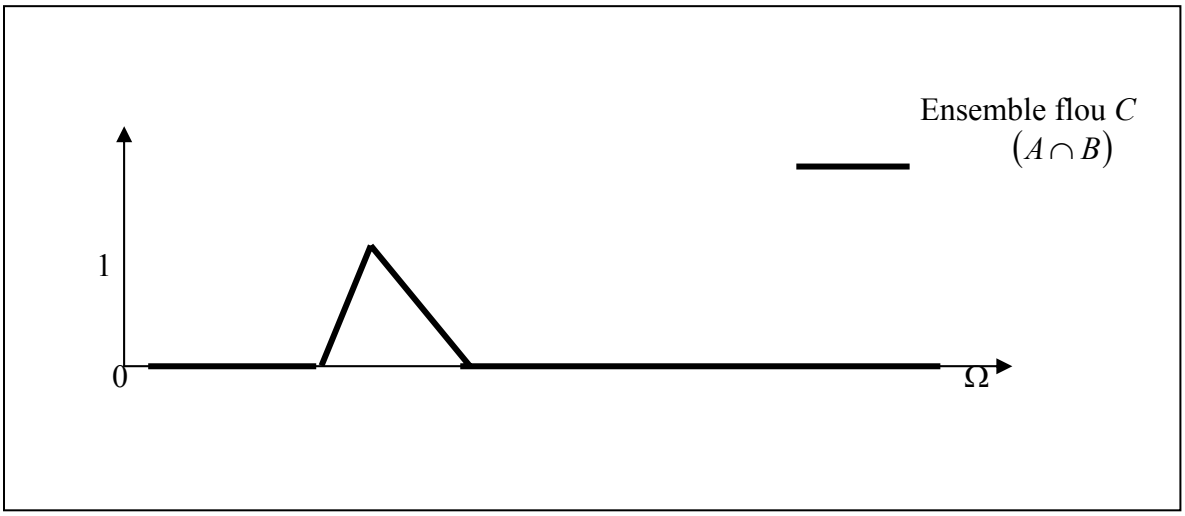


Figure III.7 : *Résultat de l'intersection des ensemble floue A et B*

III.1.4.5 Le complément

Le complément $\overline{A}$ d'un ensemble flou A est un ensemble flou constitué d'éléments de Ω associés à un degré d'appartenance à $\overline{A}$ qui est d'autant plus grand que le degré d'appartenance à A est faible. L'ensemble flou $\overline{A}$ est définie à travers sa fonction d'appartenance comme suit:

$$\forall e \in \Omega,\ \mu_{\overline{A}}(e) = 1 - \mu_A(e) \qquad \text{(III.13)}$$

III.1.4.6 Le produit cartésien

Les problèmes considérés sont souvent décrits dans plusieurs univers de référence $\Omega_1,...,\Omega_n$. Il peut être intéressant de pouvoir raisonner dans un univers de référence Ω globale composé de chacun des univers initiaux. Ω correspond donc au produit cartésien de $\Omega_1,...,\Omega_n$, $\Omega = \Omega_1 \times ... \times \Omega_n$

Le produit cartésien de n ensembles flous $A_1.....A_n$ définis respectivement sur les univers de référence $\Omega_1,...,\Omega_n$ est un ensemble flou A défini sur Ω par sa fonction d'appartenance μ_A telle que :

$$\forall e = (e_1,...,e_n) \in \Omega,\ \mu_A(e) = \top\left(\mu_{A_1}(e_1),...\mu_{A_n}(e_n)\right) \qquad \text{(III.14)}$$

III.1.4.7 Normes et conormes triangulaires

Les *t-normes* et *t-conormes* permettent de généraliser les opérations ensemblistes d'union et d'intersection (et par extension la disjonction et conjonctions de propositions) sur les sous ensembles flous. Pour que cette généralisation soit correcte, un certain nombre de propriétés doivent être vérifiées.

t-norme

Une norme triangulaire est une fonction $\top : [0,1] \rightarrow [0,1]$ qui possède les propriétés suivantes :

Commutativité : $\forall x, y \quad \top(x,y) = \top(y,x)$
Associativité : $\forall x, y, z \quad \top(x, \top(y,z)) = \top(\top(x,y),z)$
Monotonie : $\forall x, y, z, t \quad \top(x,y) \leq \top(z,t)$ si $x \leq z$ et $y \leq t$
1 élément neutre $\forall \; x \; \top(x,1) = x$

De plus, elle assure que $\forall x, y \in [0,1]$, $\top(x,y) \leq x$ et $\top(x,y) \leq y$. Le tableau (III.1) regroupe les *t-normes* les plus courantes.

Nom	fonction
min	$\top(x,y) = \min(x,y)$
produit algébrique	$\top(x,y) = x.y$
le produit bornné	$\top(x,y) = \max(0, x+y-1)$
produit drastique	$\top(x,y) = \begin{cases} x & \text{si } y=0 \\ y & \text{si } x=0 \\ 1 & \text{si } x.y > 0 \end{cases}$

Table III.1 : *Exemple de normes triangulaires*

t-conorme

La conorme triangulaire est une fonction $\perp: [0,1]\times[0,1] \to [0,1]$, possède les propriétés suivantes :

Commutativité : $\forall x, y \quad \perp (x, y) = \perp (y, x)$
Associativité : $\forall x, y, z \quad \perp (x, \perp (y, z)) = \perp (\perp (x, y), z)$
Monotonie : $\forall x, y, z \quad \perp (x, y) \leq \perp (z, t)$ si $x \leq z$ et $y \leq z$
0 élément neutre : $\forall x \quad \perp (x, 0) = x$

Elle assure aussi que $\forall x, y \in [0,1], \perp (x, y) \geq x$, et $\perp (x, y) \geq y$. Le tableau (III.2) regroupe les *t-conormes* les plus courantes.

Nom	Fonction
max	$\top(x,y)=\max(x,y)$
somme algébrique	$\top(x,y)=x+y - x\ .y$
somme borné	$\top(x,y)=\min(x+y,1)$

***Table III.2** : Exemple de conormes triangulaires*

III.1.5 Raisonnement à partir des ensembles flous

Lors de leur définition, les ensembles flous peuvent être associés à un concept à décrire. Il existe alors un lien sémantique fort entre l'ensemble flou décrit par sa fonction d'appartenance et l'univers de référence sur lequel il est défini. Dans ce contexte, des modèles de raisonnement peuvent être mis en œuvre pour étendre les principes de la logique classique à ceux de la logique floue.

III.1.5.1 Variable linguistique

En logique, les concepts manipulés sont décrits par les attributs qui prennent leurs valeurs sur un univers de référence. En logique floue [73], ces concepts sont généralement représentés par des variables linguistiques. Une variable linguistique et définie par un triplet

(A, Ω_A, F_A) ou A est un attribut (nom de variable), Ω_A son univers de référence, et $F_A = \{A_1, ..., A_n\}$ est un ensemble d'ensembles flous décrivant A.

Les ensembles flous $A_1, ..., A_n$ sont souvent désignés par un terme linguistique (ou valeur) précisant leur rôle. La figure (III.8) montre un exemple de variable linguistique associée au concept de taille d'une personne :

(taille, [0,300], {petit, moyen, grand}).

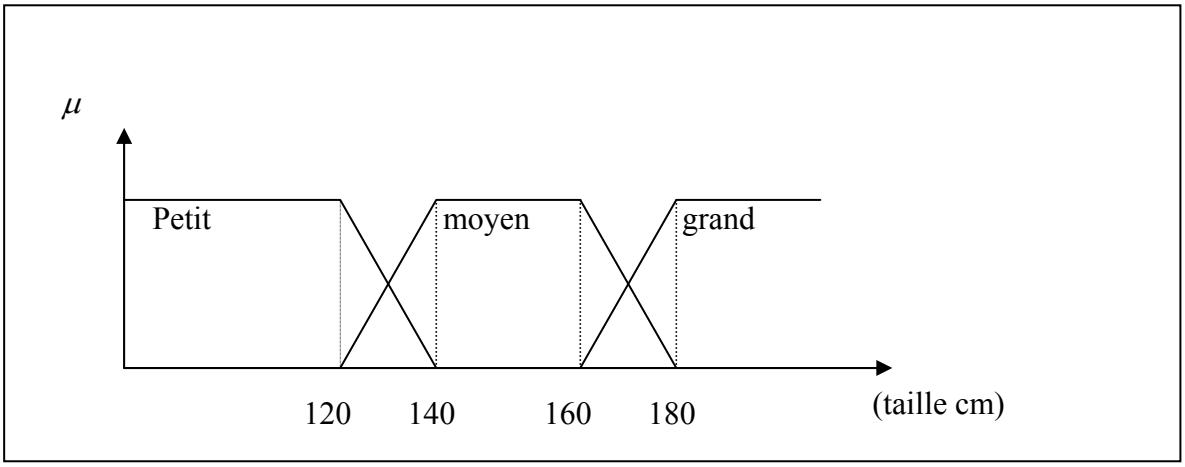


Figure III.8 *: variable linguistique associée à la taille d'une personne.*

Une variable linguistique permet donc d'une part de synthétiser l'information manipulée grâce aux ensembles flous (modélisation qualitative) et d'autre part de représenter des concepts imprécis tel que l'homme en manipule quotidiennement.

La détermination de la forme et de la position de ces ensembles floues est un élément essentiel pour pouvoir effectuer des raisonnements valides, robustes et compréhensibles. C'est pour cette raison que dans beaucoup de systèmes comme ceux destinés à l'aide à la décision, les ensembles flous sont définis par des experts du domaine afin qu'ils représentent exactement leurs connaissances.

Cependant, il n'est pas toujours possible d'obtenir une telle expertise, que ce soit à cause de la complexité du problème ou bien parce que les experts sont trop rares voir inexistants. Dans ces conditions, des algorithmes peuvent être mis en œuvre pour extraire automatiquement les sous ensembles flous. Une expertise du résultat peut éventuellement être

faite par la suite afin de déterminer la signification (le terme) linguistique des sous- ensembles floue obtenus.

III.1.5.2 Proposition floues

La logique classique manipule des propositions qui sont soit vraies, soit fausses. En logique floue, les propositions sont généralisées sous la forme de propositions floues. Si (x, Ω_A, F_A) est une variable linguistique, "x est A_1" est un exemple de proposition floue. Celle-ci n'est plus caractérisée par une valeur de vérité dans {Vrai, Faux} mais par un degré de vérité dans [0,1] qui est déduit de la fonction d'appartenance au ensemble flou mis en jeu. Ce degré de vérité est aussi appelé "fuzzy truth-value".

En reprenant l'exemple précédent, le degré de vérité de "x est A_1" correspondra à la valeur $\mu_{A_1}(x)$ pour chaque $x \in \Omega_A$.

Ces propositions floues élémentaires peuvent ensuite être utilisées pour former des propositions floues plus complexes, en les combinant entre elle par différents opérateurs.

Ainsi, la négation d'une proposition floue "*x est A_1*" sera désignée par " *x n'est pas A_1*" son degré de vérité peut être obtenu à partir de la fonction d'appartenance ou complément de l'ensemble flou A_1 soit $\mu_{\overline{A1}}$.

On peut construire des formules bien formées (wff) à partir de propositions P_A *et* P_B dont le degré de vérité dépendra de celui de P_A *et* P_B. Les opérateurs considérés sont :

- La conjonction ($\wedge$),
- La disjonction ($\vee$),
- L'implication ($\Rightarrow$),
- La négation ($\neg$)

En logique classique, la conjonction $P_{A \wedge B}$ est vraie si et seulement si P_A et P_B sont toutes les deux vraies.

En logique floue, l'imprécision relative aux propositions se traduira par $P_{A\wedge B}$ est d'autant plus vrai que P_A *et* P_B le sont ensemble. Le degré de vérité associé à $P_{A\wedge B}$ est généralement obtenu en utilisant une *t-norme* : $\mu_{A\wedge B} = \top(\mu_{A,}, \mu_B)$.

Par exemple, si l'on choisit l'opérateur min comme *t-norme*, le degré de vérité sera: $\mu_{A\wedge B} = \min(\mu_A, \mu_B)$

De façon similaire, la disjonction ordinaire veut que $P_{A\vee B}$ soit vraie si P_A est vraie ou bien si P_B est vraie. En théorie des sous ensembles flous, $P_{A\vee B}$ sera d'autant plus vraie que P_A ou P_B le sera. Le degré de vérité s'obtient généralement par une t-conorme : $\mu_{A\vee B} = \perp(\mu_{A,}\mu_B)$.

Par exemple, si l'on choisit l'opérateur min comme *t-conorme*, dans ce cas, le degré de vérité sera: $\mu_{A\vee B} = \max(\mu_{A,}\mu_B)$.

III.1.6 Règles floues

Une règle floue *R* : "*Si A est* A_i *Alors B est* B_j" est une relation entre deux propositions floues ayant chacune un rôle particulier. La première (*A* est A_i) est appelée prémisse de la règle alors que la seconde (*B* est B_j) est la conclusion. Dans le cas de propositions floues élémentaires la prémisse et la conclusion sont définies à partir de deux variables linguistiques (A, Ω_A, F_A) *et* (B, Ω_B, F_B) qui décrivent les connaissances relatives aux univers de référence Ω_A et Ω_B de manière à prendre en compte l'imprécision relative aux modalités de *A et B*. Grâce à ce mode de représentation des relations imprécises comme " Si il fait beau temps alors la visibilité en mer est bonne" peuvent être exprimées alors que ce n'est pas le cas en logique classique.

Une proposition floue élémentaire est souvent insuffisante pour représenter l'ensemble des informations à manipuler. Plusieurs propositions floues peuvent alors être combinées pour enrichir et détailler la représentation. Ainsi la prémisse correspondant à "il fait beau temps" peut correspondre à la conjonction de deux autres propositions : "La mer est calme" et "le taux d'humidité est faible".

Les opérateurs de conjonction disjonction et de négation énoncés dans la partie précédente sont souvent utilisés à cet effet.

La relation *R* entre la prémisse et la conclusion de la règle est déterminée par une implication floue dont le degré de vérité est défini par une fonction d'appartenance μ_R qui dépend du degré de vérité de chacune des deux propositions. Si μ_{Ai} et μ_{Bj} désignent les fonctions d'appartenance aux ensembles flous A_i et B_j caractérisant la prémisse et la conclusion de la règle *R* la fonction d'appartenance décrivant la proposition floue *R* est de la forme.

$$\mu_R(e, y) = I\left(\mu_{Ai}(e),\ \mu_{Bj}(y)\right) \qquad \text{(III.15)}$$

Où (e, y) appartient à $E_A \times E_B$ et $I:[0,1]\rightarrow[0,1]$ est une fonction correspondant à l'implication floue. Il existe un certain nombre de fonctions permettant d'implémenter l'implication floue et de prendre en compte l'aspect graduel des propositions floues qu'elle reliée. Le tableau (III.3) en précise quelques unes.

Nom	Degré de vérité
Reichenbach	$1-\mu_A(e)+\mu_A(e)\mu_B(y)$
Rescher_Gaines	$\begin{cases} 1 & \text{si } \mu_A(e)\le\mu_B(y) \\ 0 & \text{sinon} \end{cases}$
Kleene_Dienes	$\max(1-\mu_A(e),\mu_B(y))$
Lukasiewicz	$\min(1-\mu_A(e)+\mu_B(y),1)$
Brouwer-Gödel	$\begin{cases} 1 & \text{si } \mu_A(e)\le\mu_B(y) \\ \mu_B(y) & \text{sinon} \end{cases}$

TableIII.3: *Exemples d'implications floues* $I(\mu_A(e),\mu_B(y))$

Ces implications généralisent l'interprétation des règles implicatives de la logique classique et ont des comportements plus ou moins analogues dans les applications. Elles possèdent cependant des propriétés différentes qui peuvent être plus ou moins adaptées selon le contexte d'utilisation.

III.2 Règles d'association floues

Une règle d'association floue se présente sous la forme "*Si X est A, alors Y est B*" où la partie "*X est A* " est l'antécédent (ou condition) de la règle et la partie "*Y est B*" est le conséquent (ou conclusion) de la règle. Cette règle se note $(X, A) \Rightarrow (Y, B)$, tel que $X = \{x_1,...,x_p\}$ et $Y = \{y_1,...,y_n\}$ sont deux itemsets disjoints, et $A = \{a_1,...,a_n\}$ et $B = \{b_1,...,b_m\}$ sont les ensemble des sous-ensembles flous associés aux éléments de *X* et *Y*.

Une règle d'association floue est *valide* si un nombre suffisant de transactions de *T* supportent (contiennent) les paires [*attribut x, ensemble flou a*] et [*attribut y, ensemble flou b*].

Les approches fondamentales pour évaluer une règle d'association floue $(X, A) \Rightarrow (Y, B)$ sont :

- Approche ensembliste : remplacé les opérations des ensembles classiques par les opérations des ensembles flous correspondant**.**
- Approche logique : considéré les règles d'association floues comme des règles logiques est interpréter l'implication.

III.2.1 Approche ensembliste

III.2.1.1 Principe générale

Comme son nom l'indique, l'approche ensembliste est basée sur la notion centrale d'ensemble. Conséquemment, toutes les opérations nécessaires à la découverte des règles d'association floues reposent sur les opérations définies sur les ensembles flous notamment la cardinalité et l'intersection. Ces opérations ont été présentées dans la section III.1.3 et III.1.4. Nous avons aussi vu dans le paragraphe II.3.3 que la recherche d'une règle d'association classique nécessitait le calcul du support et de la confiance de cette règle.

L'approche ensembliste consiste donc à remplacer d'une manière Ad Hoc dans les expressions II.2 et II.3 les opérations ensemblistes à savoir l'intersection et la cardinalité, par leur équivalent opérations floues (resp. intersection floue, cardinalité floue).

$Supp(A \Rightarrow B) =$ card ($\top (A(x_i), B(x_i))$) / card (T)

$conf(A \Rightarrow B) =$ card ($\top (A(x_i), B(x_i))$) / card $(A(x_i))$

card (A) représente la cardinalité d'un ensemble flou.

III.2.1.2 Evaluation algébrique du support et de la confiance

Pour le calcul du support et de la confiance d'une règle d'association floue, la méthode algébrique utilise des cardinalités scalaires. Cette méthode agrége de manière algébrique (exemple somme algébrique) les cardinalités. A travers notre état de l'art, nous avons remarquez que les différentes propositions étaient assez similaires. Elles différent généralement sur l'introduction de mesures supplémentaires (facteur de certitude, facteur de signification). Aussi, nous avons décidé de présenter les travaux de TP Hong et donner quelques indications sur les travaux de Kuok.

A) ***Travaux de Kuok***

Dans [42] l'auteur propose deux mesures pour évaluer les règles d'association floues, *le facteur de signification* (*significance factor*) et *le facteur de certitude* (*certainty factor*) avant de présenter les deux mesures nous donnons les notations utilisées.

Soit *T* une base de transactions, où chaque transaction *t* est n-uplet de *T*. Soit l'ensemble *I* des attributs *i* apparaissant dans *T*. L'auteur note *t*[*i*] la valeur de l'attribut *i* pour la transaction *t*. Chaque attribut *i* est partitionné en sous-ensembles flous. Soit l'ensemble $F_i = \{ F_i^1, F_i^2, ... F_i^{li} \}$ de sous-ensembles flous associés à l'attribut *i*.

L'auteur note $\mu_{F_i^{\lambda_i}}(t[i])$ la fonction d'appartenance de l'attribut *i* de la transaction *t* au sous-ensemble flou $F_i^{\lambda_i}$, **l'auteur considère que ce découpage ainsi que les fonctions d'appartenance aux ensembles flous sont connues, par exemple ils sont fournis par un expert du domaine.** L'auteur choisit la multiplication comme t-norme.

Définition (itemset fréquents)

Un itemset (*X*, *A*) est dit fréquent si et seulement si sont facteur de signification est supérieure où égale à un seuil défini par l'utilisateur. Le facteur de signification est défini par la formule suivante :

$$S_{(X,A)} = \frac{\sum_{t_i \in T} \prod_{x_j \in X} \{\alpha_{\alpha_j}(t_i[x_j])}{total(T)} \qquad \text{(III.16)}$$

Afin de ne considérer que les attributs significatifs, l'auteur introduit un seuil minimum d'appartenance ω, en dessous duquel l'auteur considère que la transaction ne contient pas le couple (X, A). Il utilise à cet effet une α-coupe de seuil ω des fonctions d'appartenance :

$$\alpha_{\alpha_j}(t|x_j|) = \begin{cases} \mu_\alpha(t|x|) & \text{si } \mu_\alpha(t|x|) > \omega \\ 0 & \text{sinon} \end{cases} \qquad \text{(III.17)}$$

Le facteur de certitude d'une règle d'association $(X, A) \to (Y, B)$

La règle d'association floue est évaluée par l'utilisation d'un facteur de certitude, l'auteur propose deux méthodes pour calculer le facteur de certitude :

La première méthode ; le facteur de certitude est basée sur le facteur de signification, comme la confiance est basée sur le support, pour calculer le facteur de certitude, l'auteur propose la formule suivante :

$$C_{\langle\langle X,A\rangle,\langle Y,B\rangle\rangle} = \frac{\sum_{t_i \in T} \prod_{x_k \in Z} \{\alpha_{c_k}(t_i[Z_k])\}}{\sum_{t_i \in T} \prod_{x_k \in Z} \{\alpha_{\alpha_j}(t_i[x_j])\}} \qquad \text{(III.18)}$$

Avec :

$$\alpha_{c_k}(t|Z_k|) = \begin{cases} m_{c_k \in C}(t_i[Z_k]) & \text{Si } m_{c_k} > \omega \\ 0 & \text{Sinon} \end{cases} \qquad \text{(III.19)}$$

$$Z = X \cup Y, C = A \cup B$$

La deuxième méthode ; l'auteur utilise la corrélation pour calculer le facteur de certitude, il utilise la formule suivante :

$$C_{\langle\langle X,A\rangle,\langle Y,B\rangle\rangle} = \frac{Cov(X,Y)}{\sqrt{Var(X)\times Var(Y)}} \quad \text{(III.20)}$$

B) ***Travaux de TP Hong***

L'approche par agrégation algébrique des valeurs floues proposé par TP Hong [55] intègre les concepts de la théorie des ensembles flous pour trouver les règles d'association intéressantent à partir des données quantitatives. **L'auteur considère que les fonctions d'appartenances sont données d'une manière empirique**.

Mesure de support et de confiance

Pour calculer le support et la confiance l'auteur propose d'agréger de manière algébrique (somme algébrique) les cardinalités.

Support d'un itemset

Le support d'un *k*-itemset $s = (s_1, s_2, ..., s_k)$ est calculé à partir des degrés d'appartenance flous f_{is}, le support de l'itemset est calculé par la formule :

$$count_s = \sum_{i=1}^{n} f_{is} \quad \text{(III.21)}$$

Où *n* le nombre de transactions

Confiance d'une règle d'association

$$conf = \frac{\sum_{i=1}^{n} f_{is}}{\sum_{i=1}^{n} \left(f_{is_1} \wedge ... \wedge f_{is_{k-1}} \wedge f_{is_{k+1}} \wedge ... \wedge f_{is_q} \right)} \quad \text{(III.22)}$$

L'opérateur "min" est choisi comme une t-norme

Exemple :

Afin d'illustrer le travail de TP Hong ainsi présentés, nous présentons l'exemple suivant :

Soit la base de transaction donnée dans le tableau III.4. On pose $\alpha = 1.5$ et $\lambda = 0.7$. Chaque item de la transaction est défini par un tuple (item, quantité) par exemple pour la 4^{eme} transaction on a 9 unités de pain et 10 unités de T-shirt. Chaque item est présenté par un symbole, par exemple *A* est l'item lait, *B* est l'item jus.

TID	Items
1	(lait, 3) (pain, 4) (T-shirt, 2)
2	(jus, 3) (pain, 7) (jacket, 7)
3	(jus, 2) (pain, 10) (T-shirt, 5)
4	(pain, 9) (T-shirt, 10)
5	(lait, 7) (jacket, 8)
6	(jus, 2) (pain, 8) (jacket, 10)

Table III.4 : *une base T de six transactions*

Pour cet exemple, les quantités des items sont représentées par trois région floues : *peu*, *beaucoup*, *moyen*, la partition est la même pour les différents attributs considérés. Cette partition est donnée dans la figure III.9

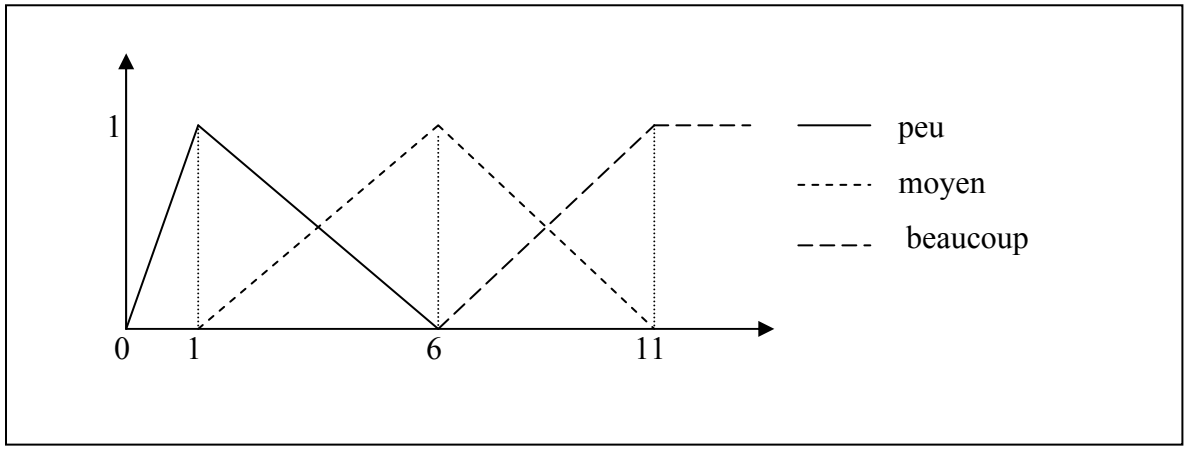


Figure III.9 : *Représentation des trois régions floues utilisée dans l'exemple*

TID	Items
1	(*A*,3) (*C*,4) (*E*,2)
2	(*B*,3) (*C*,7) (*D*,7)
3	(*B*,2) (*C*,10) (*E*,5)
4	(*C*,9) (*E*,10)
5	(*A*,7) (*D*,8)
6	(*B*,2) (*C*,8) (*D*,10)

Table III.5 : *la base T avec les items ajoutées*

Etape 1 : trouver les valeurs floues, par exemple le 1er item de la 4eme transaction, la valeur 9 de *C* est transformer à un ensemble flou (0.4/ *C*.moyen+0.6/ *C*. beaucoup). Table III.6.

TID	ensemble flou
1	(0.6/ *A*.peu+0.4/ *A*. moyen)(0.4/ *C*.peu+0.6/*C*.moyen)(0.8/*E*.peu+0.2/*E*.moyen)
2	(0.6/ *B*.peu+0.4/ *B*. moyen)(0.8/ *C*.moyen+0.2/ *C*. beaucoup)
3	(0.8/ *B*.peu+0.2/ *B*. moyen)(0.2/ *C*.moyen+0.8/ *C*. beaucoup)(0.2/ *E*. peu+0.8/ *E*.moyen)
4	(0.4/ *C*.moyen+0.6/ *C*. beaucoup)(0.2/ *E*.moyen+0.8/ *E*. beaucoup)
5	(0.8/ *A*. moyen+0.2/ *A*. beaucoup)(0.6/ *D*.moyen+0.4/ *D*. beaucoup)
6	(0.8/ *B*.peu+0.2/ *B*. moyen)(0.6/ *C*.moyen+0.4/ *C*. beaucoup)(0.2/ *D*.moyen+0.8/ *D*. beaucoup)

Table III.6 : *les ensemble flous*

Etape 2 : la cardinalité de chaque région floue dans la transaction est calculé, par exemple la cardinalité de *C*. beaucoup= 0+0.2+0.8+0.6+0+0.4=2. Table III.7

Item	count	Item	Count	Item	count
A.peu	0.6	*C*. peu	0.4	*E*. peu	1.0
A. moyen	1.2	*C*.moyen	2.6	*E*.moyen	1.2
A. beaucoup	0.2	*C*. beaucoup	2.0	*E*. beaucoup	0.8
B.peu	2.2	*D*. peu	0.0		
B. moyen	0.8	*D*.moyen	1.6		
B. beaucoup	0.0	*D*. beaucoup	1.4		

Table III.7 : *le support de chaque région flou*

Etape 3 : la valeur maximale des trois région flou pour chaque item est trouvé, par exemple pour l'item *A*, sa cardinalité pour peu est 0.6, *A.* moyen=1.2, *A.* beaucoup=0.2, la valeur maximale de ces trois région est 1.2 donc on garde l'item *A.* moyen.

Etape 4 : les valeurs trouvées à l'étape 3 sont comparés au support minimal, les itemset qui ont une valeur $< \alpha$ sont supprimés. Table III.8.

Itemset	Count
B.peu	2.2
C.moyen	2.6
D.moyen	1.6

Table III.8 :*l'ensemble de 1-itemset fréquent*

Etape 5 : l'ensemble de candidat est généré à partir de L_1.

Etape 6 : pour chaque 2-itemsets de C_2, le degré d'appartenance à chaque transaction est calculé, table III.9. Puis nous calculons le support de chaque 2-itemset de C_2, table III.10. Les supports sont testés par rapport au support minimum α, les itemsets qui ont un support $\geq \alpha$ sont utilisés pour générer L_2.

TID	*B*.peu	*C*.moyen	*B*.peu $\cap$ *C*.moyen
1	0.0	0.6	0.0
2	0.6	0.8	0.6
3	0.8	0.2	0.2
4	0.0	0.4	0.0
5	0.0	0.0	0.0
6	0.8	0.6	0.6

Table III.9 : *les degrés d'appartenances de* *B*.peu $\cap$ *C*.moyen

Itemset	count
(*B*.peu, *C*.moyen)	1.4
(*B*.peu, *D*.moyen)	0.8
(*C*.moyen, *D*.moyen)	1.0

Table III.10 : *le support des itemsets dans* C_2

Etape 7 : $L_2 \neq \phi$, alors exécuté l'étape suivante.

Etape 8 : r=2, r est utilisée pour garder la taille de l'itemset.

Etape 9 : générer C_3 à partir de L_2, $C_3 \neq \phi$ étape 12 est exécuté pour générer les règle d'association.

Etape 12 : les règles d'association sont générées à partir de chaque itemset fréquent et la confiance de chaque règle est calculé.

Etape 13 : supprimer les règles qui ont une confiance $< \lambda$.

Conclusion

Les deux approches algébriques utilisent la cardinalité scalaire pour le calcul du support et de la confiance. Les deux méthodes présentées considèrent que les ensembles flous sont donnés par un expert de domaine. Pour la recherche des règles d'association floues les deux méthodes appliquent un algorithme de recherche des règles d'association binaires tel que Apriori.

III.2.1.3 Approche sémantique

Dans l'approche proposée par Delgado [54], le concept de transaction floue a été défini comme un sous ensemble flou $\tilde{\tau}$($\tilde{\tau} \subseteq I$, où I est l'ensemble d'items) [54]. Un modèle général de découverte de règles d'association sur des transactions floues a été présenté. Ce type de règles d'association a été appelé règle d'association floue. L'auteur met l'accent sur le calcul du support et de la confiance. Pour cela, il propose une approche sémantique basée sur l'évaluation de formules quantifiées pour le calcul du support et de confiance en utilisant les

cardinalités floues [72]. Nous donnons d'abord certaines définitions utilisées par l'auteur et détaillons ensuite cette approche.

Définition (Transaction floue)

Une transaction floue (notée $\tilde{\tau}$) est un ensemble flou (non vide) de $I(\tilde{\tau} \subseteq I)$. Où I est un ensemble d'items. Pour tout $i \in I$, on note $\tilde{\tau}(i)$ le degré d'appartenance de l'item *i* à la transaction $\tilde{\tau}$. On note $\tilde{\tau}(I_0)$ le degré d'inclusion de l'itemset $I_0 \subseteq I$ à la transaction floue $\tilde{\tau}$ défini comme suit,

$$\tilde{\tau}(I_0) = \min_{i \in I_0} \tilde{\tau}(i) \qquad \text{(III.23)}$$

Exemple

Soit $I = \{i_1, i_2, i_3, i_4\}$ un ensemble d'items, le tableau III.11 présente six transactions définies dans *I.*

	i_1	i_2	i_3	i_4
$\tilde{\tau}_1$	0	0.6	0.7	0.9
$\tilde{\tau}_2$	0	1	0	1
$\tilde{\tau}_3$	1	0.5	0.75	1
$\tilde{\tau}_4$	1	0	0.1	1
$\tilde{\tau}_5$	0.5	1	0	1
$\tilde{\tau}_6$	1	0	0.75	1

Table III.11 : *ensemble de six transactions floues*

On a $\tilde{\tau}_1 = 0.6/i_2 + 0.7/i_3 + 0.9/i_4$ et $\tilde{\tau}_2 = 1/i_2 + 1/i_4$

$\tilde{\tau}_1(\{i_2, i_3, i_4\}) =$ min $\{0.6, 0.7, 0.9\} = 0.6$

Donc 0.6 représente le degré d'inclusion de l'itemset $\{i_2, i_3, i_4\}$ dans la transaction floue $\tilde{\tau}_1$.

L'auteur appelle *T-set* un ensemble de transaction ordinaire (non flou) et *FT-set* un ensemble de transaction floues. La table III.11 présente un *FT-set* $\{\tilde{\tau}_1,...,\tilde{\tau}_6\}$ de six transactions floues

Définition (règle d'association floue)

Soit I un ensemble d'items, T un *FT-set*, et A, C deux itemsets ($A,C \subseteq I$, $A,C \neq \phi$ et $A \cap C = \phi$). Une règle d'association floue $A \Rightarrow C$ est valide si : $\tilde{\tau}(A) \leq \tilde{\tau}(C) \quad \forall \tilde{\tau} \in T$

C'est-à-dire le degré d'inclusion de C est supérieur au degré d'inclusion de A pour chaque transaction floue $\tilde{\tau}$.

Support et confiance d'une règle d'association floue

L'auteur emploie une approche sémantique basée sur l'évaluation de formules quantifiées. L'évaluation sémantique de formules quantifiées a été présentée par l'auteur dans un papier précédent [72]. Une formule quantifiée est une expression de la forme "Q de F sont G ", ou F et G deux ensembles flous de l'ensemble X finie, et Q un quantificateur relative, les quantificateurs relatives comme "peu", "beaucoup", "moyen", sont des étiquettes linguistiques qui permettent de représenter le pourcentage flou d'un ensemble flou dans l'intervalle [0,1]. Parmi les quantificateurs relatifs, un quantificateur cohérent, est celui qui vérifie les propriétés suivantes :

- $Q(0) = 0$ et $Q(1) = 1$
- si $x < y$ alors $Q(x) \leq Q(y)$

Par exemple, dans l'expression "beaucoup de jeunes personnes sont grandes", *beaucoup* correspond au quantificateur Q, F et G sont les distribution de possibilité induire dans l'ensemble X = personnes par les termes imprécis "*jeune*" et "*grand*" respectivement. L'évaluation de formule quantifiée prend ses valeurs dans l'intervalle [0,1].

Définition (support d'un itemset flou)

Soit $I_0 \subseteq I$, le support de l'itemset I_0 dans T est l'évaluation de formules quantifiée :

$$Q \text{ de } T \text{ sont } \widetilde{\Gamma}_{I_0} \qquad \text{(III.24)}$$

Où $\widetilde{\Gamma}_{I_0}$ est l'ensemble flou de T défini par : $\widetilde{\Gamma}_{I_0}(\widetilde{\tau}) = \widetilde{\tau}(I_0) = \min_{i \in I_0} \widetilde{\tau}(i)$

Exemple

La table III.12 présente le support de différents itemset de la table III.11.

Itemset	Support
$\{i_1\}$	0.58
$\{i_4\}$	0.98
$\{i_2, i_3\}$	0.18
$\{i_1, i_3, i_4\}$	0.26

Table III.12 : *Support des itemsets*

Définition (support d'une règle d'association floue)

Le support d'une règle d'association floue $A \Rightarrow C$ dans l'ensemble de transactions floues T est $Supp(A \cup C)$ c'est à dire l'évaluation de la formule quantifiée "Q de T sont $\widetilde{\Gamma}_{A \cup C}$ ". On remarque en outre que :

$$Q \text{ de } T \text{ sont } \widetilde{\Gamma}_{A \cup C} = Q \text{ de } T \text{ sont } \left(\widetilde{\Gamma}_A \cap \widetilde{\Gamma}_C\right) \qquad \text{(III.25)}$$

Définition (**confiance d'une règle d'association floue**)

La confiance d'une règle d'association floue $A \Rightarrow C$ dans l'ensemble de transaction floue *T,* est l'évaluation de la formule quantifiée :

$$Q \text{ de } \widetilde{\Gamma}_A \text{ sont } \widetilde{\Gamma}_C$$

Les mesures de confiance et de support dépendent de la méthode d'évaluation et de quantificateur choisir. Sachant que l'auteur à défini l'évaluation de l'expression " Q de F sont G " comme suit [72] :

$$GD_Q\left(\frac{G}{F}\right)=\sum_{\alpha_i\in\Delta(G/F)}(\alpha_i-\alpha_{i+1})Q\left(\frac{\left|(G\cap F)_{\alpha_i}\right|}{\left|F_{\alpha_i}\right|}\right) \qquad \text{(III.26)}$$

Où $\Delta(G/F)=\Lambda(G/F))\cup\Lambda(F)$, $\Lambda(F)$ est le niveau de l'ensemble de F, et $\Delta(G/F)=\{\alpha_1,...,\alpha_p\}$ avec $\alpha_i>\alpha_{i+1}$ pour tout $i\in\{1,...,p\}$.

Choix du quantificateur

Le choix du quantificateur, permet de changer la sémantique d'une valeur dans le contexte linguistique.

$Supp(A\Rightarrow C)$ peut être interprété comme la preuve que le pourcentage de transactions dans $\widetilde{\Gamma}_{A\cup C}$ est Q, et $conf(A\Rightarrow C)$ est interprété comme la preuve que le pourcentage de transactions dans $\widetilde{\Gamma}_A$ sont aussi dans $\widetilde{\Gamma}_C$ est Q.

Plusieurs méthodes d'évaluation, et différents quantificateurs peuvent être choisis pour caractériser et calculer le support et la confiance d'une règle d'association floue, mais la méthode et le quantificateur choisis doivent vérifier les quatre propriétés de mesures de la règle d'association ordinaire suivantes :

1- si $\widetilde{\Gamma}_A\subseteq\widetilde{\Gamma}_C$, alors $conf(A\Rightarrow C)=1$

2- si $\widetilde{\Gamma}_A\cap\widetilde{\Gamma}_C=\phi$, alors $Supp(A\Rightarrow C)=0$ et $conf(A\Rightarrow C)=0$

3- si $\widetilde{\Gamma}_A\subseteq\widetilde{\Gamma}_{A'}$ (particulièrement quand $A'\subseteq A$), alors $conf(A'\Rightarrow C)\le conf(A\Rightarrow C)$

4- si $\widetilde{\Gamma}_C\subseteq\widetilde{\Gamma}_{C'}$ (particulièrement quand $C'\subseteq C$), alors $conf(A\Rightarrow C)\le conf(A\Rightarrow C')$

Les quatre propriétés sont vérifies par GD_Q [72].

Delgado choisit d'utiliser le quantificateur Q_M (autrement dit l'identité), définie par $Q_M(x)=x$, les mesures obtenues par l'application de ce quantificateur pour le calcul du support et de la confiance sont les mesures ordinaires du support et de confiance pour le cas non flou [54], comme le montre les propositions suivantes.

Proposition

Soit $I_0 \subseteq I$ sachant que $\widetilde{\Gamma}_{I_0}$ est non flou, alors *supp*(I_0) évalué par la mesure *GD* et par le quantificateur Q_M est le support ordinaire de l'itemset

$$GD_Q\left(\frac{G}{F}\right) = Q\left(\frac{|F \cap G|}{|F|}\right)$$

$$Supp(I_0) = GD_{Q_M}\left(\frac{\widetilde{\Gamma}_{I_0}}{T}\right) = \frac{\left|\widetilde{\Gamma}_{I_0}\right|}{|T|} \qquad \text{(III.27)}$$

Proposition

Soit $A \Rightarrow C$, une règles d'association ordinaire dans *T*. $Supp(A \Rightarrow C)$, mesuré par *GD* avec Q_M, est le support ordinaire d'une règle d'association.

$$\begin{aligned} Supp(A \Rightarrow C) &= GD_{Q_M}\left(\frac{\widetilde{\Gamma}_{A\cup C}}{T}\right) \\ &= \frac{\left|\widetilde{\Gamma}_{A\cup C}\right|}{|T|} = Supp(A \cup C) \end{aligned} \qquad \text{(III.28)}$$

Proposition

Soit $A \Rightarrow C$ une règle d'association ordinaire dans *T*. $conf(A \Rightarrow C)$, mesuré par *GD* avec Q_M, est la confiance ordinaire d'une règle d'association.

$$conf(A \Rightarrow C) = GD_{Q_M}\left(\frac{\widetilde{\Gamma}_{A\cap C}}{\widetilde{\Gamma}_A}\right) = \frac{\left|\widetilde{\Gamma}_A \cap \widetilde{\Gamma}_C\right|}{\left|\widetilde{\Gamma}_A\right|} \qquad \text{(III.29)}$$

Pour les propositions précédentes, $|\ |$ représente la cardinalité de l'ensemble flou, cette cardinalité correspondant à la cardinalité scalaire (cf. III.1.4.1)

Exemple

La table III.13 illustre le concept introduit par les propositions.

règle	support	confiance
$\{i_2\} \Rightarrow \{i_3\}$	0.18	0.28
$\{i_1, i_3\} \Rightarrow \{i_4\}$	0.26	1
$\{i_1, i_4\} \Rightarrow \{i_3\}$	0.26	0.44

Table III.13 : Support et confiance de trois règles d'association

Ainsi nous remarquons que, $Conf(\{i_1, i_3\} \Rightarrow \{i_4\}) = GD_{Q_M}\left(\frac{\widetilde{\Gamma}_{\{i_4\}}}{\widetilde{\Gamma}_{\{i_1,i_3\}}}\right) = 1$ puisque $\widetilde{\Gamma}_{\{i_1,i_3\}} \subseteq \widetilde{\Gamma}_{\{i_4\}}$.

III.2.2 Approche logique (Dubois et Prade)

Dans leur proposition, Dubois et Prade [71] [70] [30] [22], **considèrent les règles d'association floues comme des implications logiques**. L'évaluation de telle implication logique est alors effectuée en choisissant un opérateur d'implication présenté dans la section III.1.6. Avant de présenter cette approche, nous donnons les notations utilisées.

Soit $R(A, B, C, ...)$ un schéma relationnel, les variables $A, B, C....$ correspondent aux attributs de la relation *R*. Le domaine d'un attribut *A* est l'ensemble $D_A = \{a_1, ..., a_{|D_A|}\}$.

D'autre part, l'auteur estime que traiter un schéma relationnel de *n* attributs revient à traiter un schéma relationnel de deux attributs. Pour cela, l'auteur considère uniquement des schémas relationnels de type $R(A, B)$. Sur un tel schéma, un tuple est noté $t(a, b)$.

Le sous ensemble A de D_A définir un sous ensemble $R(A)$ sur R. La valeur d'appartenance de tuple $t = (a, b)$ dans $R(A)$ est $A(a)$, la valeur d'attribut *a* dans l'ensemble flou *A*.

Définition d'une règle d'association floue

Une règle d'association floue est interprétée comme une implication "si X est A alors Y est B". Par exemple, dans la règle d'association floue, "si age est jeune, alors le salaire est bas", les ensembles "*age*" et "*salaire*" sont modélisé par des ensembles classiques, par contre "*jeune*" est "*bas*" sont modélisés par des ensembles flous.

Dans [30], [70] [75], Dubois et Prade ont identifié deux types de règles floues, les règles graduelles (*gradual rules*), les règles à certitude (*certainty rules*), qui sont distinguées par leurs sémantiques et formellement par le type d'opérateur d'implication qui définit la relation . Les règles à certitude ont l'interprétation suivante "plus X est A, plus il est certain que Y soit B". L'opérateur d'implication utilisée est l'opérateur Reichenbach, où Dienes (cf. III.1.6). L'utilisation de ces opérateurs d'implications indique que les tuples (a,b) pertinents, supportent complètement la règle si " $A(a)=1$ et $B(b)=1$ ", dans le cas ou la conséquence est certaine.

L'interprétation des règles graduelles est "plus X est A, plus Y est B ", l'opérateur d'implication utilisée à cet effet est l'opérateur Lukasiewicz (cf. III.1.6).

Confiance d'une règle d'association floue

Pour évaluer la confiance d'une règle d'association floue, l'approche logique adopte l'interprétation de l'implication. Pour l'interprétation de l'implication deux cas sont à considérer :

- le premier cas, considère le tuple $t(a,b)$ totalement pertinent pour la règle d'association $A \Rightarrow B$ si " $A(\alpha)>0$ ". Par cette interprétation le tuple $t(a,b)$ à le degré $\top(\delta(A(a)), I(A(a),B(b)))$ où δ est la fonction indicatrice.

$$\delta(x)=\begin{cases}1 & \text{si} \quad x>0 \\ 0 & \text{sinon}\end{cases} \qquad \text{(III.30)}$$

La confiance est définir par :

$$imp_f(A \Rightarrow B) = \frac{\sum_{k=1}^{|R|} \mathrm{T}\left(\delta(A(a_{ik})), I\left(A(a_{ik}), B(b_{jk})\right)\right)}{\sum_{k=1}^{|R|} \delta(A(a_{ik}))} \quad \text{(III.31)}$$

L'indice *f* est l'abréviation de mot totalement (*full en anglais*).

- le deuxième cas, l'interprétation de pertinence considère le tuple $t(a,b)$ pertinent que pour le degré $A(a)$

Dans ce cas la confiance est définit comme suit :

$$imp_p(A \Rightarrow B) = \frac{\sum_{k=1}^{|R|} \mathrm{T}\left(A(a_{ik}), I\left(A(a_{ik}), B(b_{jk})\right)\right)}{\sum_{k=1}^{|R|} A(a_{ik})} \quad \text{(III.32)}$$

L'indice *p* indique que les tuples sont considéré partiellement pertinent, basée sur le degré de satisfaction de l'antécédent.

III.3 Constat

Tout processus de découverte de règles d'association floues repose obligatoirement sur le partitionnement flou du domaine d'un attribut quantitatif. A travers notre état de l'art sur les règles d'association floues, nous avons constaté que toutes les approches proposées et existantes à ce jour nécessitent l'intervention d'un expert du domaine pour obtenir les différentes régions floues respectant la partition du domaine de l'attribut quantitatif ainsi considéré. Comme ce n'est pas toujours évident de trouver un expert du domaine, fort souvent les partitions floues sont décrites de manière **empirique.** D'ailleurs, même un expert utilise un processus mental empirique pour aboutir à ses résultats.

Une étude réalisée dans [28], propose d'utiliser un algorithme de segmentation, permet de trouver des partitions non floue d'attribut, puis les degrés d'appartenances sont

calculés par une formule mathématique, donc les degrés d'appartenances ne sont pas déterminés automatiquement.

Pour cela, nous proposons dans ce mémoire une approche originale qui utilise la segmentation floue afin d'obtenir automatiquement les partitions floues, et de déterminer les degrés d'appartenances de chaque élément aux partitions floues.

III.4 Conclusion

Dans ce chapitre nous avons introduit les notions des sous ensembles flous. Nous avons aussi présenté les différentes cardinalités, et les différents normes appliquer aux ensembles flous. Les cardinalités et les normes sont les opérations sur les quelles sont basés le calcul du support et de confiance d'une règle d'association floue.

Nous avons présenté les deux approches pour la découverte des règles d'association floues, l'approche ensembliste qui est la plus adapté pour la fouille de données et l'approche logique qui interprète une règle d'association floue comme une règle floue logique.

Toutes les approches présentées considèrent des partitions empiriques ou données par un expert du domaine, comme les problèmes de data mining considèrent de très gros volumes de données, ce qui rend le partitionnement impossible par un expert humain. Pour cela, nous proposons dans ce mémoire un partitionnement automatique et intelligent.

Dans le chapitre suivant nous présentons la segmentation floue.

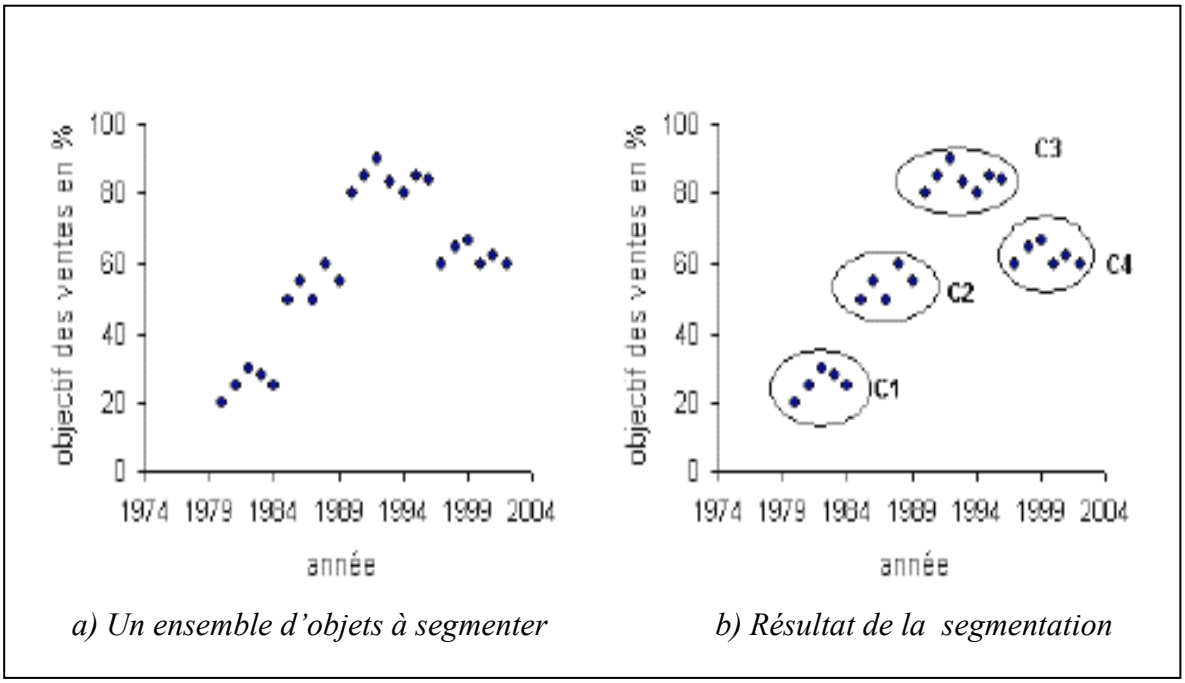
Afin de pallier aux limites et incohérences souvent engendrées par un partitionnement empirique nous proposons dans ce travail un partitionnement automatique basé sur le principe de la segmentation floue (fuzzy clustering).

Pour cela, nous présentons dans ce chapitre les principes de la segmentation non supervisée (clustering). Nous détaillerons plus particulièrement la méthode dite des Fuzzy C-Means sur laquelle est basée notre proposition.

IV.1 Principe fondamental de la segmentation

Le processus de segmentation (clustering en anglais) vise à construire des segments (groupe, clusters ou regroupement) d'objets similaires à partir d'un ensemble d'objets hétérogènes [79], [80], [74], [57]. Chaque groupe issu de ce processus doit vérifier les deux propriétés suivantes :

- La cohésion interne (les objets appartenant à ce regroupement doivent être les plus similaires possibles).
- L'isolation externe (les objets appartenant à des regroupements différents doivent être les plus distincts possibles).



a) Un ensemble d'objets à segmenter *b) Résultat de la segmentation*

Figure IV.1 : *Exemple de segmentation.*

La segmentation repose sur une mesure précise de la similarité /dissimilarité des objets que l'on veut regrouper. Cette mesure est appelée métrique. Beaucoup de travaux considèrent cette métrique comme une distance [97] [98] [99]. La figure (IV.1) donne un exemple de segmentation.

On peut formaliser la segmentation de la manière suivante :

Soit X un ensemble fini, soit $C = (C_1, ..., C_q)$ un ensemble de parties non vides de *X*. On dit que *C* est une partition si :

$$\forall i \neq j, C_i \cap C_j = \phi$$

$$\bigcup_i C_i = X$$

Dans un ensemble $X = \{x_1, ...x_n\}$ partitionné en q groupes, chaque élément de l'ensemble appartient à un et un seul regroupement. Une manière pratique de décrire la partition *C* consiste à utiliser une notation matricielle. Soit *U* la matrice caractéristique de la partition de *X*

$$U = \begin{pmatrix} \mu_{11} \ldots\ldots \mu_{1q} \\ \vdots \quad \ddots \quad \vdots \\ \mu_{ik} \\ \vdots \quad \ddots \quad \vdots \\ \mu_{n1} \ldots\ldots\ldots \mu_{nq} \end{pmatrix}$$

Où :

$$\mu_{ik} = \begin{cases} 1 & \text{si } x_i \in C_k \\ 0 & \text{sinon} \end{cases} \qquad \text{(IV.1)}$$

Nous pouvons remarquer que *U* vérifie les propriétés suivantes :

- La somme de chaque ligne est égale à 1 : $\sum_{k=1}^{q} \mu_{ik} = 1$.
- La somme de valeurs de la $k^{ième}$ colonne vaut n_k (le nombre d'éléments du regroupement C_k): $\sum_{i=1}^{n} \mu_{ik} = n_k$.
- On peut ainsi déduire que $\sum_{k=1}^{q} n_k = n$

IV.1.1 Différents domaines d'application de segmentation

A nos jours, la segmentation trouve application dans plusieurs domaines, comme la vision artificielle, la biologie, l'Internet, l'analyse de données et beaucoup d'autres. Dans ce contexte, nous allons citer quelques exemples d'application de la segmentation :

- L'analyse des données qui peuvent provenir des images satellites, équipement médical, systèmes d'information géographique, afin d'extraire les caractéristiques nécessaires pour accélérer le processus d'exploitation de ces données [56].

- La génération des hypothèses afin d'inférer des règles pour caractériser les données et suggérer des modèles, par exemple : l'établissement des diagnostics médicaux sur des bases de données de patients [104] [65].

- La réduction de la dimension des bases de données afin de conserver le maximum d'information utile dans un espace de dimension inférieure [105].

- La prospection du Web (Web Mining) et l'analyse des données textuelles (Text Mining) pour la recherche d'informations à partir de certains mots clés [106].

IV.1.2 Processus de segmentation

Etant donnée un ensemble d'objets $X = \{x_1, x_2, ..., x_n\}$ dans l'espace d'attributs $\Re^d$ avec, d : dimension de l'espace et n : le nombre d'objets. $x_i = \{x_{i1}, x_{i2}, ..x_{id}\}$ représente le $i^{\text{ème}}$ objet, et x_{ij} correspond à la valeur du $j^{\text{ème}}$ attribut pour le $i^{\text{ème}}$ objet. Le but principal de la segmentation est la recherche des groupes similaires dans l'espace d'objets $\Re^d$.

Ce problème a été abordé dans plusieurs travaux [76]. A travers cette littérature, on constate que toutes les méthodes de segmentation suivent le même principe général qui consiste à maximiser la similarité des objets à l'intérieur d'un groupe, et minimiser la

similarité des objets appartenant à des groupes différents. La figure (IV.2) illustre les différentes étapes d'un processus de segmentation [76].

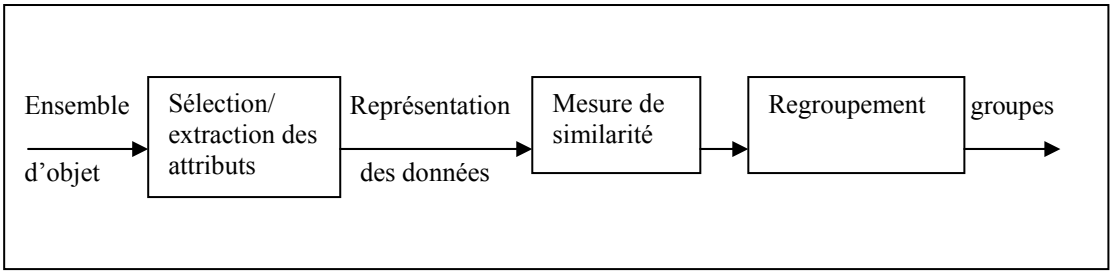


***Figure IV.2** : Différentes étapes d'un processus de segmentation*

1. La sélection / extraction des attributs correspond à l'utilisation d'une ou plusieurs transformations des attributs fournis en entrée afin de sélectionner le sous-ensemble d'attributs le plus efficace à utiliser pour la segmentation. Plusieurs méthodes qui traitent ce problème ont été proposées dans la littérature [76] [77] [78] [46].

2. La représentation des données se réfère à la spécification du nombre de données, ainsi que la dimension et le type des attributs disponibles pour l'algorithme de segmentation.

3. La mesure de similarité consiste à définir une métrique appropriée au domaine des données. Différentes mesures de similarité ont été utilisées dans la segmentation. La distance euclidienne est l'une des métriques les plus utilisées [58].

4. Le regroupement consiste à la construction des groupes similaires, qui représentent le résultat du processus de segmentation. Ce résultat peut être dur "hard" (partition des objets en groupes distincts), ou flou "fuzzy" (chaque objet a un degré variable d'appartenance à chacun des groupes ainsi formés).

IV.2 Méthodes de segmentation

Il existe différentes méthodes de segmentation. Ces méthodes sont généralement clairement distinguées dans la littérature. A ce titre, la figure IV.3 illustre les distinctions entre les différentes méthodes existes [62] [76].

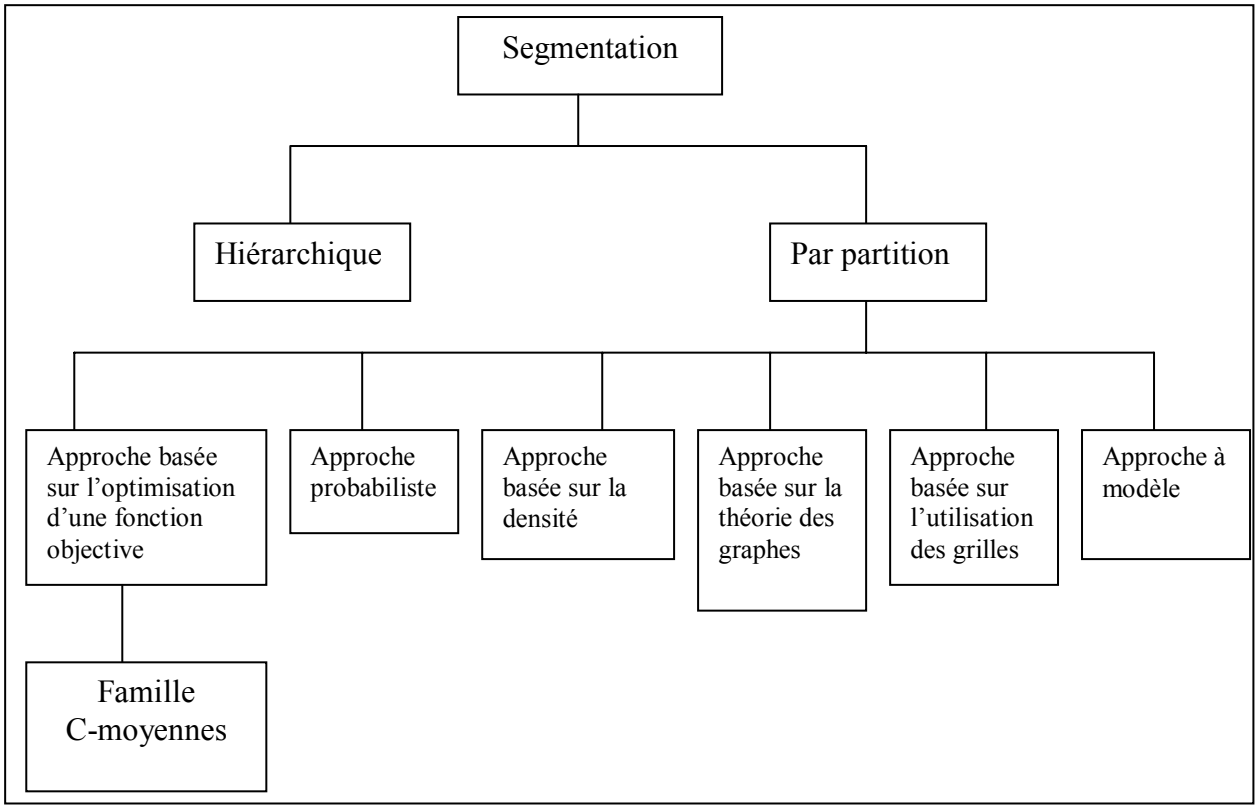


Figure IV.3 *: Différentes approches de segmentation.*

1. ***la segmentation hiérarchique*** : le but est de former une hiérarchie de classes, de telle sorte que plus on descend dans la hiérarchie, plus les regroupements sont spécifiques à un certain nombre d'objets considérés comme similaires [63].

2. ***La segmentation par partition*** : dont le but est de former plusieurs partitions dans l'espace des objets, de telle sorte que chaque partition représente un regroupement.

Selon le schéma établi dans la figure (IV.3), il existe plusieurs approches qui se distinguent fortement dans cette catégorie.

Quelle que soit l'approche envisagée, toutes reposent sur un même fondement : l'utilisation d'une mesure de similarité (ou de dissimilarité) qui permet de déterminer si deux individus de la base et par extension si deux sous-ensembles d'individus (deux regroupements) se ressemblent.

Dans les espaces de représentations numériques, les mesures de dissimilarité les plus courantes reposent sur l'utilisation d'une métrique. Le choix de celle-ci est un élément très important pour le bon fonctionnement des algorithmes de la segmentation non supervisée, il influe très fortement sur la forme des regroupements trouvés ainsi que sur les propriétés de segmentation qui en découle.

IV.2.1 La segmentation hiérarchique

Différents méthodes ou approches de segmentation hiérarchique ont été proposés dans la littérature [94] [81]. Toutes ces méthodes partagent une caractéristique importante : ils ne produisent pas une seule partition mais une hiérarchie de partitions emboîtées. Ici, un regroupement est défini comme un noeud d'arbre, auquel est associé l'ensemble des objets qui le composent, ainsi leurs caractéristiques. Il existe généralement deux grandes catégories de méthodes hiérarchiques : *les méthodes ascendantes* et *les méthodes descendantes*.

IV.2.1.1 Méthodes ascendantes ou agglomératives

Dans les méthodes ascendantes, la partition initiale contient autant de regroupement que d'objets ($q = n$). A chaque étape, on cherche un couple (C_a,C_b) de classe candidats à la fusion qui maximise (resp. minimise) une certaine mesure de similarité (resp. de dissimilarité). On réitère ce processus jusqu'à n'obtenir qu'un regroupement contenant tous les éléments. Afin de déterminer le nombre de regroupement, on coupe la hiérarchie à un certain niveau de détail. Concernant ce choix, Benfield et Raftery [96] ont proposé une heuristique appelée " AWE ". La figure (IV.4) illustre cette hiérarchie de partitions sous forme appelée dendogramme.

Il existe plusieurs méthodes permettant de déterminer quels regroupements il faut fusionner. Beaucoup sont fondées sur l'utilisation d'une mesure de similarité.

Ainsi, si $\{C_1,...,C_q\}$ représente l'ensemble des regroupements qui ont déjà été établis à l'itération courante et que x, y sont les exemples de la base, on peut utiliser les critères suivants pour fusionner deux regroupements i et j :

- *La distance minimum entre les regroupements (single-link)*

$$d_{\min}(C_i, C_j) = \min_{x \in C_1, y \in C_j} d(x, y)$$

$d(x, y)$: la distance entre un objet x de C_i et un objet y de C_j.

- *La distance maximum entre les regroupements (complete-link)*

$$d_{\max}(C_i, C_j) = \max_{x \in C_i, y \in C_j} d(x, y)$$

$d(x, y)$: la distance entre un objet x de C_i et un objet y de C_j.

- *La distance moyenne entre les regroupements (average-link*)

$$d_{moy}(C_i, C_j) = \frac{1}{|C_i| . |C_j|} \sum_{x \in C_i} \sum_{y \in C_j} d(x, y)$$

$d(x, y)$: la distance entre un objet x de C_i et un objet y de C_j.

$|C_i|$ (respectivement $|C_j|$) représente le nombre d'échantillons appartenant à C_i (respectivement C_j).

- *La distance entre les centres des regroupements*

$$d_{centre}(C_i, C_j) = d(m_i, m_j)$$

Où m_i et m_j sont les centres des regroupements C_i et C_j

L'un des avantages des techniques de la segmentation hiérarchique est de fournir, via le dendogramme, une interprétation naturelle du comportement de l'algorithme. A l'opposé, on est généralement confronté à une grande complexité en temps et surtout en espace.

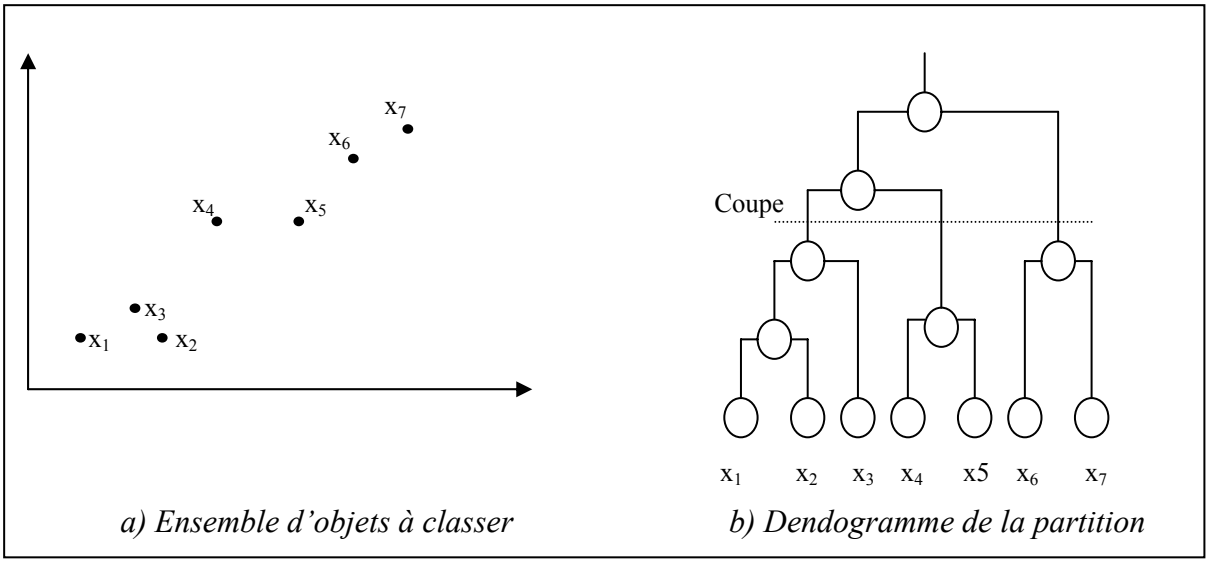


Figure IV.4 *: Principe de la segmentation hiérarchique*

IV.2.1.2 Méthodes descendantes

Les algorithmes de *méthodes descendantes*, fonctionnent à l'inverse des algorithmes précédents : le point de départ est un regroupement unique constitué de l'ensemble des données de la base. Deux divisions sont ensuite faites à chaque pas de l'itération. Ces divisions peuvent par exemple s'opérer sur les regroupements de plus grandes tailles ce qui favorise les partitions équilibrées mais ne reflète pas nécessairement la réalité. Utiliser la distance intra-classe au sein de chaque regroupement comme critère de division, donne alors de meilleurs résultats lorsque les regroupements naturels sont de taille variée.

IV.2.2 La segmentation par partition

Contrairement à la segmentation hiérarchique, la segmentation par partition a pour but de trouver une et une seule partition de l'espace d'objet, de telle sorte qu'elle soit la plus pertinente pour la formation du regroupement. Différentes méthodes de segmentation par partition sont détaillées dans cette section.

IV.2.2.1 Méthode basé sur la densité

Le but ici est de chercher à former des groupes denses, de telle sorte que chaque groupe représente une région homogène de haute densité, entourée par des régions de faible densité. Pour cela, deux paramètres qui contrôlent la densité sont utilisés :

- Le rayon maximum du voisinage : E_{ps}
- Le nombre minimum de points qui doivent être contenus dans ce voisinage : *MinPts*

Le voisinage d'un objet est défini comme suit :

$$V_{E_{ps}}(x_i) = \{x_j \in X / dist(x_i, x_j) \leq E_{ps}\}$$

L'algorithme DENCULE présenté dans [32] et l'algorithme DBSCAN présenté dans [33], sont des exemples d'algorithmes appartenant à cette catégorie. La figure (IV.5) illustre un exemple de segmentation basé sur la densité, dont le principe général est décrit comme suit :

1. Sélectionner aléatoirement un objet x_i. (par exemple p)

2. Vérifier si son voisinage respecte le critère de densité ; c'est-à-dire s'il y a au moins *MinPts* points dans la sphère de centre x_i et de rayon E_{ps}.(dans l'exemple *MinPts=2, q* vérifient le critère de densité).

3. Si le critère de densité est respecté, intégrer les objets correspondant dans le segment, et répéter le procédé avec ces objets. (dans l'exemple q est le prochain objet).

4. Sinon, aller à 1 (la sélection aléatoire se fait sur les objets non encore classés).

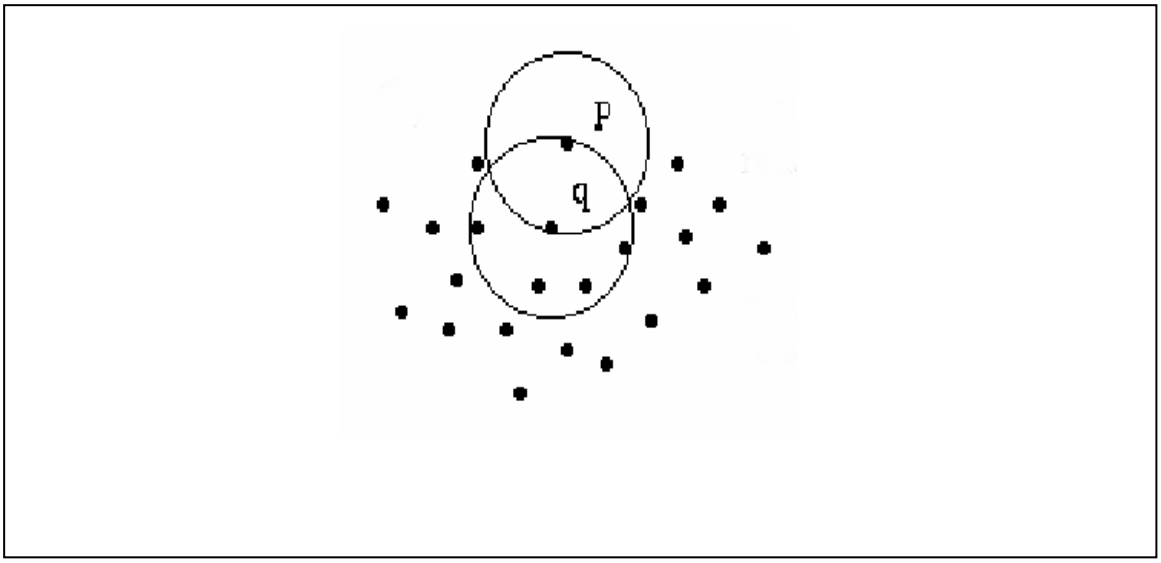


***Figure IV.5** : Exemple de segmentation basé sur la densité*

IV.2.2.2 Méthode basée sur les grilles

Le principe de base de cette méthode est d'utiliser une grille pour diviser l'espace en un ensemble de cellules, ensuite identifier les ensembles de cellules denses connectées pour former les groupes. Un groupes est vu donc comme un ensemble de cellules denses et connectées. STING "Statistical Information Grid-based method" [34] et WaveCluster [35] sont des exemples d'algorithmes appartenant à cette catégorie.

Il existe deux types de méthodes pour identifier un groupe :

- Les méthodes qui calculent la densité de chaque cellule, puis fusionnent les cellules pour que la résultante soit suffisamment dense et uniforme. La figure (IV.6) est une illustration graphique de ces méthodes.

- Les méthodes qui se basent sur la détection des limites des groupes. Le principe de base ici est la détection des limites entre les zones de haute densité et les zones de faible densité, ensuite la reconstitution des groupes à partir de ces limites.

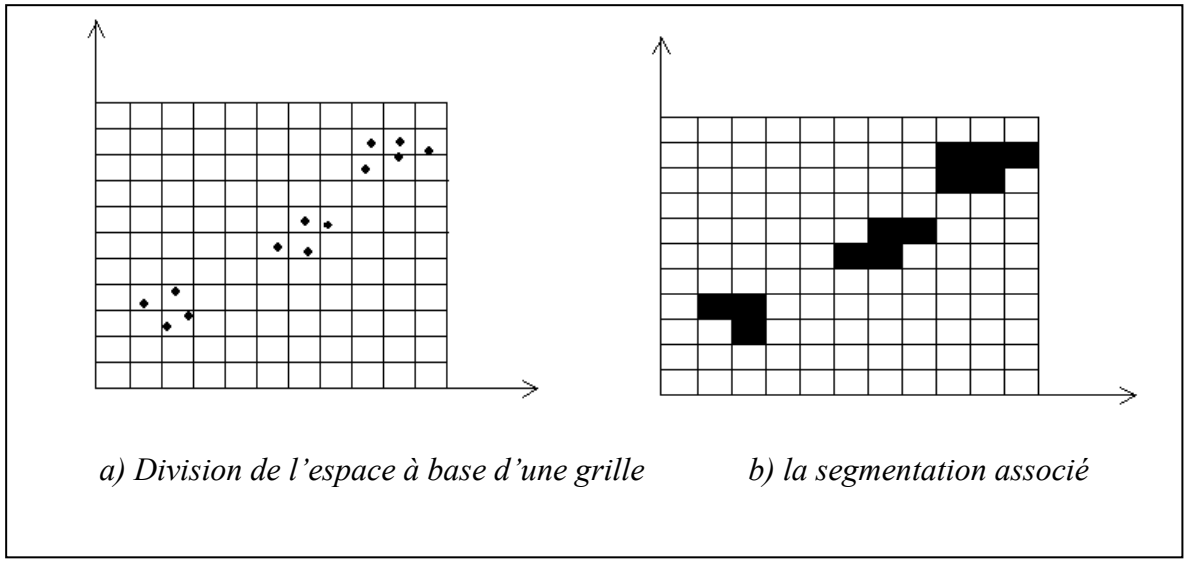


Figure IV.6 : S*egmentation basée sur les grilles*

La majorité des algorithmes appartenant à cette catégorie souffrent d'une problématique importante, qui est le choix de la taille des cellules. Les cellules de trop petite taille, amènent à une estimation bruitée de la densité (problème du "sur partitionnement"). A l'inverse, les cellules de taille importante, amènent à une estimation trop faible de la densité (problème du "sous partitionnement").

IV.2.2.3 Méthodes basés sur la théorie des graphes

Le principe de cette méthode consiste en la recherche des arcs à conserver dans un graphe qui connecte les différents objets entre eux afin de former des groupes. Ici un groupe est défini comme un ensemble de noeuds connectés dans un graphe. La figure (IV.7) est une illustration graphique d'un exemple de segmentation basé sur la théorie des graphes, dont le principe général est décrit comme suit [36] :

1. Construction d'un MST " Minimal Spanning Tree " de données, cela revient à définir un graphe connexe en joignant tous les objets de la base, dont la somme des valeurs des étiquettes associées aux arcs est minimale.

2. Suppression des arcs les plus longs pour la création des groupes.

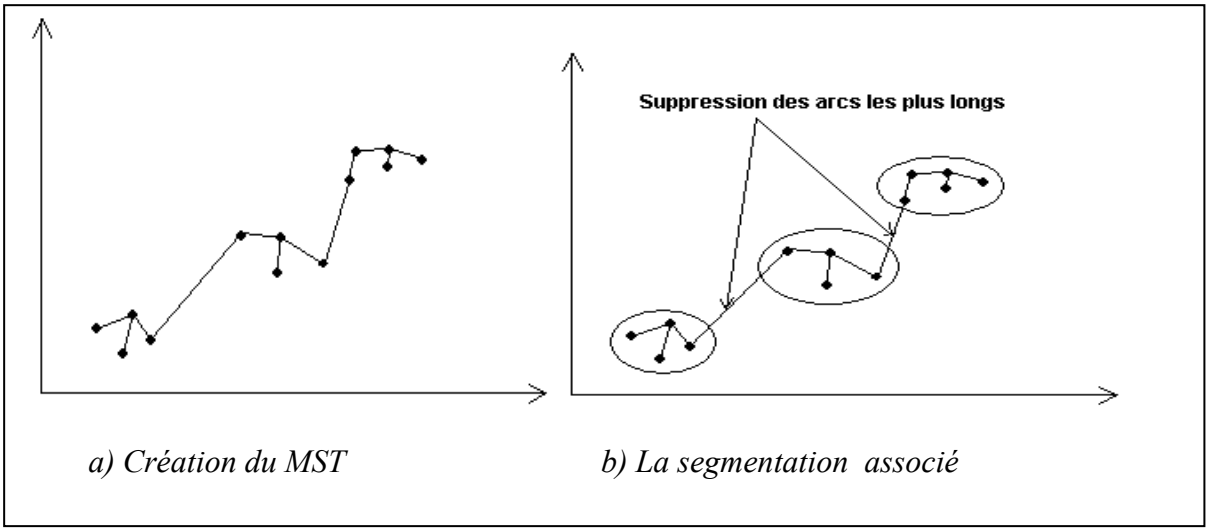


Figure IV.7 : S*egmentation basé sur la théorie des graphes.*

Une autre possibilité proposée dans [37] consiste à conserver les liens entre les objets séparés par une distance inférieure à un certain seuil, les groupes étant alors l'ensemble des objets connectés. D'autres algorithmes ont été proposés dans la littérature. Comme exemple, on peut citer l'approche proposée dans [38] basé sur " Relative Neighborhood Graph " (RNG). Alors que dans [39] une autre méthode est présentée, spécialement dédiée à des partitions avec recouvrement entre les groupes.

IV.2.2.4 Méthodes basés sur la minimisation d'une fonction objective

Parmi les différentes méthodes de segmentation présentées, celles basées sur l'optimisation d'une fonction objective et spécialement, celles appartenant à la famille des C-moyennes (C-Means en anglais), représentent l'une des techniques les plus robustes et les plus utilisées en segmentation.

Dans la famille des C-moyennes nous distinguons trois types d'algorithmes suivant les contraintes imposées sur les degrés d'appartenance :

- *Les C-moyennes dures " Hard C-Means : HCM "*

$$\forall i, \forall k \ \mu_{ik} \in \{0,1\} \text{ et } \sum_{k=1}^{q} \mu_{ik} = 1$$

Pour Les C-moyennes dures, le degré d'appartenance d'un objet à un groupe est soit "0" qui indique qu'un objet n'appartient pas du tout à un groupe et le degré "1" indique qu'un objet appartient totalement à un groupe. Alors un objet ne peut appartenir qu'à un seul groupe.

- *Les C-moyennes floues " Fuzzy C-Means : FCM "*

$$\forall i \, \forall k \; \mu_{i,k} \in [0,1] \text{ et } \sum_{k=1}^{q} \mu_{ik} = 1$$

Un objet peut appartenir plus ou moins à un groupe. Le degré d'appartenance d'un objet à un groupe est une valeur numérique comprise dans l'intervalle réel [0,1].

- *Les C-moyennes possibilistes " Possibilistic C-Means : PCM "*

$$\forall i \, \forall k \;\; \mu_{ik} \in [0,1]$$

Pour les C-moyennes possibilistes, le degré d'appartenance d'un objet à un groupe ne dépend pas de celui des autres groupes.

Ces méthodes sont fondées sur la minimisation d'une fonction objective commune :

$$J_m(U,V,W\;) = \sum_{k=1}^{q}\sum_{i=1}^{n} \mu_{ik}^m d_{ik}^2 + \sum_{k=1}^{q} w_k \sum_{i=1}^{n} (1-\mu_{ik})^m$$

U : la matrice de partition floue d'éléments u_{ik} respectant les contraintes relatives à l'algorithme utilisé.

$V = \{v_1,...,v_q\}$ l'ensemble des centres des groupes.

$W = \{w_1,...,w_q\}$ l'ensemble des termes de pénalité de C-moyennes possibilistes des données atypiques associés à chacun des groupes (égale à zéro dans le cas de C-moyennes dures et C-moyennes floues).

Nous présentons plus en détail la segmentation floue et la méthode des C-moyennes flous dans la section suivante.

IV.3 Segmentation floue

Les approches classiques de segmentation non supervisée (par exemple hiérarchiques) ont un inconvénient majeur : les données ne peuvent appartenir qu'à un seul regroupement. Cette vision est en fait assez éloignée de la réalité. En effet dans la plupart des cas les regroupements recherchés ne possèdent pas de frontières bien déterminées. Il arrive aussi qu'ils se chevauchent. Il est alors plus correct de dire qu'un individu situé prés d'une frontière peut appartenir aux différents groupes concernés plutôt que de l'affecter arbitrairement un seul regroupement. Ce comportement est gênant non seulement pour la classification de nouveaux individus mais aussi pour la détermination de la partition si celle-ci s'effectue de manière itérative.

Après l'introduction par Zadeh [95] du concept d'ensemble flou, la notion de segment (i.e. groupe, cluster) flou trouve son cadre d'expression naturel. La segmentation floue, développée au début des années 1970, généralise une approche classique en segmentation en élargissant la notion d'appartenance à un ensemble. En effet l'appartenance d'un élément à un ensemble n'est plus une valeur vrai ou fausse, mais elle est caractérisée par un réel compris entre 0 et 1 appelé degré d'appartenance u_{ik}. Ainsi un élément peut appartenir à plusieurs ensembles avec différents degrés d'appartenance.

IV.3.1 Algorithme de C-moyennes floues (CMF)

L'algorithme des C-moyennes floues CMF (en anglais Fuzzy C-means) introduit par Dunn [97] et défini par Bezdek [98], se base sur une fonction objective pour optimiser une partition initiale des données. Cette partition est caractérisée par un ensemble de prototypes correspondant à des centres de groupes.

Le principe de fonctionnement de l'algorithme ainsi que ses avantages et inconvénients sont d'écrits ci-dessous.

Soit $X = \{x_j / j = 1...n\}$ l'ensemble des données définies dans $\Re^d$ que l'on souhaite partitionner en q regroupements $C_1,...,C_k,...,C_q$. Soit $V = \{v_1,...,v_q\}$ l'ensemble des centres

des groupes (appelés centroïdes) caractérisant ces regroupements et $d(x_j, v_c)$ la métrique employée pour calculer la similarité entre une donnée x_j et le centroïde v_c .

Le choix de la métrique $d(x_j, v_c)$ dépend de la forme des groupes recherchés. La distance euclidienne est généralement la métrique adoptée par l'algorithme de C-moyenne floue. On notera : $d(x_j, v_c) = |x_j - v_c|$.

L'utilisation de cette métrique impose une uniformité dans les formes des groupes recherchés. Par conséquent, les groupes recherchés par l'algorithme de C-moyenne floue seront sous forme sphérique. La figure IV.8 illustre la forme des groupes obtenus par l'application de CMF.

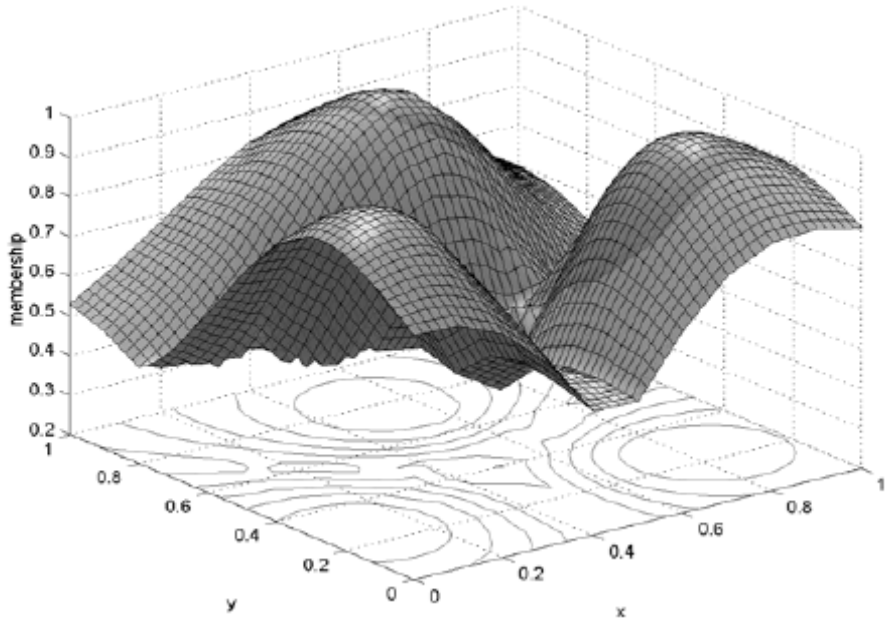


Figure IV.8 **:** *Résultat de segmentation avec l'application de CMF*

Chaque individu x_j de X est caractérisé par un degré d'appartenance μ_{jq} à chacun des regroupements C_q . Le but de l'algorithme consiste alors à minimiser la fonction objective suivante qui représente la distance moyenne intra-group "floue" au sens des moindres carrés :

$$J_{V,U,X} = \sum_{c=1}^{q} \sum_{j=1}^{n} \left(\mu_{jc}\right)^{m} d^2(x_j, v_c) \qquad \text{(IV.1)}$$

Dans cette fonction :

- U représente la matrice $C \times N$ de partitionnement flou. Elle est constituée des éléments μ_{jq} tels que.

$$\forall j \in \{1,...,n\}, \forall k \in \{1,...q\}, \quad \mu_{jk} \in [0,1]$$

$$\forall j \ \sum_{c=1}^{q} \mu_{jc} = 1 \quad \text{et} \quad \forall k \ \ 0 < \sum_{j=1}^{n} \mu_{jk} < n \qquad \text{(IV.2)}$$

La contrainte imposée sur les degrés d'appartenance (leur somme vaut 1) empêche d'obtenir $\mu_{jc} = 0$, $\forall j, c$ comme solution triviale pour la minimisation de $J_{V,U,X}$.

- Le paramètre *m* défini dans $[1,\infty[$ représente le degré de flou de la partition. Il permet de jouer sur le degré de chevauchement autorisé entre les regroupements [67]. Quand $m \to 1$, la partition tend vers une partition nette (c'est-à-dire non floue), au contraire, quand $m \to \infty$, la partition devient de plus en plus floue. m=2, choix qui est très souvent fait en pratique.

De plus la minimisation de la fonction objective nécessite de déterminer les différents centroïdes V_c et les degrés d'appartenance comme suit :

- la détermination des centroïdes V_c [98] :

$$V_c = \frac{\sum_{j=1}^{n} (\mu_{jc})^m x_j}{\sum_{j=1}^{n} (\mu_{jc})^m} \qquad \text{(IV.3)}$$

- détermination des fonctions d'appartenance [98] :

$$\left.\begin{array}{ll} \mu_{jk} = \dfrac{1}{\sum_{l=1}^{q}\left(\dfrac{d^2(x_j, v_k)}{d^2(x_j, v_l)}\right)^{\frac{1}{m-1}}} & \text{si } J_j = \phi \\ \mu_{jk} = 0 \quad \text{si } k \notin J_j & \\ \sum_{k \in Jj} \mu_{jk} = 1 \quad \text{si } k \in J_j & \text{si } J_j \neq \phi \end{array}\right\} \qquad \text{(IV.4)}$$

Avec $J_j = \{k / 1 \leq k \leq q, d^2(x_j, v_k) = 0\}$

Nous détaillons maintenant l'algorithme des C-moyennes floues. Celui-ci se déroule selon les étapes suivantes :

1. étape 01 : initialisation

   - fixer le nombre de regroupements q,
   - fixer le degré de flou m, généralement m=2,
   - adopter une métrique, généralement la métrique adoptée est la distance Euclidienne,
   - initialiser la matrice V des centres de groupes de manière aléatoire,
   - t=1. t représente le nombre d'itération.

2. étape 02 : calcule des degrés d'appartenances

   - calculer le degré d'appartenance de chaque objet x_i aux centres de groupes par les équations (IV.4),
   - calculer la valeur de la fonction objective J_t par l'équation (IV.1). Sinon,
   - si $t \neq 1$ alors, si $J_t - J_{t+1} < \xi$ alors fin de l'algorithme, sinon aller à l'étape3.

3. étape 03 : mise à jour des centres des groupes

   - mémoriser U dans U_{svgd} et recalculer V par l'équation (IV .3),
   - recalculer J_{t+1} par l'équation (IV.1).

4. étape 04 : test d'arrêt

   - si $J_{t+1} - J_t < \xi$ alors fin de l'algorithme, sinon,
   - $t = t + 1$, aller à l'étape 2.

Remarque 1 *:*

L'initialisation peut être faite en déterminant la matrice des degrés d'appartenances initiale, dans ce cas l'algorithme des C-moyennes floues se déroule selon les étapes suivantes :

1. étape 01 : initialisation

   - fixer le nombre de regroupements q,
   - fixer le degré de flou m, généralement m=2,
   - adopter une métrique, généralement la métrique adoptée est la distance Euclidienne,
   - initialisée la matrice U des degrés d'appartenances aléatoirement .en respectant les contraintes (IV.2),
   - t=1. t représente le nombre d'itération.

2. étape 02 : calcul des centres de groupes

   - calculer les centres de groupes par l'équation (IV.3),
   - calculer la valeur de la fonction objective J_t par l'équation (IV.1).
   - si $t \neq 1$ alors si $J_t - J_{t+1} < \xi$ alors fin de l'algorithme, sinon aller à l'étape3.

3. étape 03 : mise à jour de la matrice U des degrés d'appartenances

   - mémoriser U dans U_{svgd} et recalculer U par l'équation (IV .4),
   - calculer J_{t+1}.

4. étape 04 : test d'arrêt

   - si $J_{t+1} - J_t < \xi$ alors fin de l'algorithme, sinon,
   - $t = t + 1$, aller à l'étape 2.

***Remarque 2**:*

- Il existe différents critères d'arrêt soit $\|U - U_{svgd}\| < \xi$ ou soit $\|V - V_{svgd}\| < \xi$

  A titre indicatif [107] [99] utilise $\xi = 0.001$ pour définir le critère d'arrêt.

Dans la suite de ce document nous utilisons CMF pour désigner l'algorithme de C-Moyenne Flou.

IV.3.2 Avantages et inconvénients de l'algorithme (CMF)

L'avantage principal de l'algorithme CMF est le fait d'obtenir des regroupements tels que l'appartenance d'un objet à un regroupement devient une notion graduelle. Cette notion est généralement matérialisée par des degrés d'appartenance μ_{jk}. Ces degrés d'appartenance rendent le processus itératif beaucoup plus robuste notamment en permettant de prendre en compte les recouvrements entre les regroupements. Il permet ainsi d'obtenir des partitions plus pertinentes et plus proches de la réalité.

Parmi les avantages nous pouvons aussi constater que sa complexité algorithmique est relativement réduite par rapport à d'autres algorithmes de segmentation non supervisée. Cela le rend plus facilement exploitable pour traiter des problèmes de taille importante (avec beaucoup de données), comme indiquée en [9], si *n* est le nombre de valeur d'un attribut, *q* le nombre de groupes, *l* le nombre d'itération. Le tableau IV.1 indique les complexités en temps et en espace mémoire.

Malgré tout l'algorithme CMF possède aussi quelques inconvénients. On peut citer par exemple le problème de la sensibilité à l'initialisation (différentes initialisations peuvent aboutir à différentes partitions), la nécessité d'imposer le nombre de regroupements *q* à priori ou encore le manque de flexibilité sur la forme des regroupements qu'il peut détecter.

L'algorithme	Complexité en temps	Complexité en espace
Average link	$O(n^2)$	$O(n^2)$
Complete link	$O(n^2 \log n)$	$O(n^2)$
Simple link	$O(n^2 \log n)$	$O(n^2)$
CMF	$O(nlq)$	$O(q)$

***Table IV.1** : complexité algorithmique*

IV.3.3 Les algorithmes dérivés de l'algorithme CMF

Les CMF ont été très largement utilisées dans de nombreux domaines [31] [40] [41] [59]. Différentes adaptations ont aussi été faites notamment pour pouvoir traiter les problèmes dans lesquels les regroupements ont des formes variées. Ces adaptations se fondent essentiellement sur une modification de la métrique utilisée [66].

Pour pouvoir effectuer des regroupements avec des formes variées une solution simple consiste à modifier la métrique utilisée. Rappelons que dans le cas des CMF la distance utilisée est la distance Euclidienne. Les regroupements trouvés possèdent donc une forme hyper sphérique (cf. figure IV.8).

IV.3.3.1 Algorithme de Gustafson et kessel

Cet algorithme [45] consiste à adapter la distance de manière à pouvoir obtenir des regroupements de formes hyperellispoïdales quelconques. Pour cela la distance Euclidienne est remplacée par une distance de Mahalanobis et en définissant une par regroupement.

$$d^2(x_j, v_c) = [\det(\mathrm{cov}_c)]^{1/2} (x_j - v_c)^t \, \mathrm{cov}_c^{-1} (x_j - v_c) \qquad \text{(IV.5)}$$

Avec n la dimension de l'espace des données et Cov_c la matrice de covariance floue associée au groupe c Celle-ci est calculée à chaque pas de l'itération par :

$$\mathrm{cov}_c = \frac{1}{\sum_{j=1}^{n} \mu_{jc}^m} \sum_{j=1}^{n} \mu_{jc}^m (x_j - v_c)(x_j - v_c)^t \qquad \text{(IV.6)}$$

où n : le nombre d'objet,

m :le degré de flou,

μ_{jc} : le degré d'appartenance de l'objet j au groupe c.

Dans cet algorithme les groupes ne sont plus uniquement déterminés par les centres v_c mais aussi par la matrice de covariance Cov_c associée.

IV.3.3.2 Algorithme de Gath et Geva (FMLE)

La mesure de dissimilarité [99] entre une donnée x_j et un regroupement Cc est considérée comme étant inversement proportionnelle à la probabilité a priori que x_j appartienne à C_c. Pour cela la distance utilisée est adaptée de la manière suivante :

$$d^2\left(x_j, v_c\right) = \frac{\left[\det\left(\mathrm{cov}_c\right)\right]^{1/2}}{p_c} \exp\left(\frac{\left(x_j - v_c\right)\mathrm{cov}_c^{-1}\left(x_j - v_c\right)^t}{2}\right) \qquad \text{(IV.7)}$$

Où p_c est la probabilité a priori qu'une donnée appartienne à C_c. Celle-ci est estimée à chaque pas de l'itération par :

$$p_c = \frac{\sum_{j=1}^{n} \mu_{jc}^m}{N} \qquad \text{(IV.8)}$$

N : le nombre d'itération.

Cette mesure de dissimilarité permet d'obtenir des regroupements hyperellipsoïdaux de formes et de densités variées. Dans ce contexte les prototypes sont définis à la fois par les centres v_c, les matrices de covariance Cov_c et la probabilité p_c. La figure (IV.9) illustre la forme des groupes par l'application de l'algorithme FMLE.

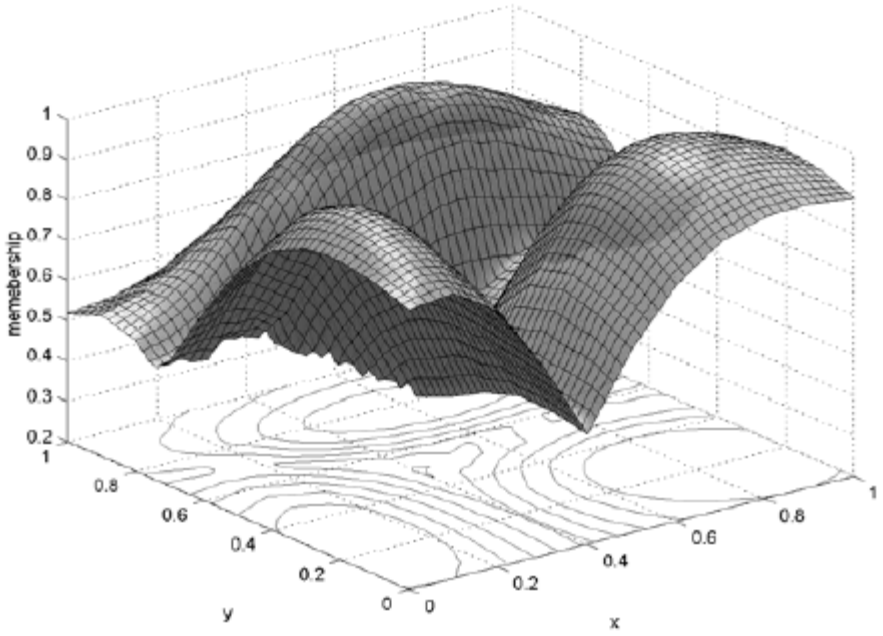


Figure IV.9 **:** *Résultat de segmentation avec l'application de FMLE*

IV.4 Le nombre de groupes et les indices de validités

En segmentation non supervisée, les groupes possibles ne sont pas connus à l'avance et les exemples d'attributs disponibles sont non étiquetés. Le but est de regrouper dans un même groupe les objets considérés comme similaires selon une métrique. Cependant, la majorité des algorithmes de segmentation souffrent du problème de détermination du nombre de groupes qui est souvent laissé à l'utilisateur. A ce problème, plusieurs fonctions appelées indices de validité ont été proposées [64].

IV.4.1 Le nombre de groupes

Le choix du bon nombre de regroupement q consiste en une condition essentielle pour que la méthode CMF produise une segmentation efficace. Par "bon nombre de regroupement" on veut dire que chacun de ces regroupements doit faire apparaître des propriétés intrinsèques aux données et ainsi permettre de faciliter leur interprétation. Chaque regroupement doit avoir une signification pour l'expert du domaine. Typiquement deux situations peuvent se présenter:

- *Trop de regroupement* : cette situation peut entraîner une grande confusion car certains regroupements sont " artificiels ", c'est-à-dire qu'ils ne représentent aucune réalité du domaine concerné.
- *Pas assez de regroupement* : cet autre cas peut cacher des aspects importants présents dans les données. Par exemple, on peut séparer un ensemble de patients en deux groupes : les patients sains et malades. Mais il peut être plus intéressant pour le médecin d'utiliser une structure en trois groupe faisant ressortir les patients sains, malades et à risque.

A ce problème, l'approche la plus utilisée consiste à :

1) Exécuter les algorithmes de segmentation avec différents nombres de regroupement.
2) Evaluer leurs résultats, et ce à partir d'une comparaison entre ces derniers.

L'évaluation des résultats est basée essentiellement sur l'utilisation des indices de validité. Plus formellement, un indice de validité est une fonction qui mesure la qualité du résultat final d'un algorithme de segmentation. Afin de trouver le nombre de regroupements qui optimise (la plus petite ou la plus grande valeur) l'indice de validité en question, l'algorithme proposé utilise un processus itératif qui consiste à exécuter un algorithme de segmentation avec différents nombres de regroupement. Cependant, un problème crucial émerge, il concerne le choix d'un intervalle dans lequel le processus itératif de recherche doit être exécuté.

Dans ce contexte, si nous faisons l'hypothèse initiale que chaque élément de l'ensemble de données X constitue un regroupement, nous aurons un problème de taille énorme et un processus de recherche très compliqué en terme d'espace mémoire et en terme de temps d'exécution. En plus de ça, dans la majorité des cas réels, le nombre de regroupement est nettement inférieur aux nombre d'objet : ($q << n$).

L'hypothèse de limiter la recherche sur un intervalle bien défini, semble plus raisonnable et mieux adapté à ce genre de situation. Généralement, le processus de recherche s'effectue entre les deux quantités C_{min} et C_{max} où C_{min} (resp. C_{max}) correspond au nombre minimum (resp. maximum) de regroupements.

Dans la majorité des cas $C_{min} \geq 2$, alors que pour le choix C_{max} il n'y a aucune règle formelle. Quelques auteurs (Bezdek [100]) proposent de choisir $C_{\max} = \sqrt{n}$.

La stratégie de recherche du nombre de regroupements q qui optimise un indice de validité $Vd(c)$ est la suivante :

Entrée : $X = \{x_1,...,x_n\}$ ensemble de données,

C_{min} le nombre minimum de regroupements

C_{max} le nombre maximum de regroupements.

Sortie : le nombre de regroupements qui optimise un indice de validité.

1. Pour $c = C_{min}$, jusqu'à C_{max} .
a- Appliquez l'algorithme de segmentation.
b- Calculer la valeur de l'indice de validité.
2. Determiner q_f de telle sorte que l'indice $Vd(q_f)$ soit optimal.
3. Fin.

Algorithme IV.1 : *Stratégie de recherche de nombre optimal de regroupemenst*

Dans le paragraphe suivant on présente quelques indices de validité.

IV.4.2 Les indices de validités dédies à la segmentation floue

Valider les résultats d'un algorithme de segmentation est une étape cruciale pour l'évaluation de la qualité de la partition résultante [60]. En général, on peut grouper les indices de validité dédiés à la segmentation floue non supervisée en trois catégories. Ceux de la première catégorie utilisent les propriétés des degrés d'appartenance u_{ik} pour évaluer une partition. Ceux de la deuxième catégorie combinent les propriétés des degrés d'appartenance u_{ik} et l'ensemble de données X. Alors que les indices appartenant à la troisième catégorie sont basés sur le concept de l'hypervolume et de la densité.

- Première catégorie des indices de validité

Les fonctions de validité noté V_{PC} "Partition Coefficient" [101] et V_{PE} "Partition Entropy" [102] sont les premiers indices de validité dédiés à la classification floue non supervisée proposés par Bezdek.

Partition Coefficient V_{PC} :

Si la valeur de V_{PC} tend vers son maximum 1, pour un certain nombre de groupe *c*, on aura une partition qui est constituée de groupes bien séparés. Si la partition en question ne contient aucune structure de groupes, V_{PC} atteint sa valeur minimale 0. Il est clair que le nombre de groupe *q* qui maximise V_{PC} indique le nombre optimal de groupe.

$$V_{PC}(U) = \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{q}\mu_{ik}^{2} \qquad \text{(IV.9)}$$

Sous la contraintes que $\mu_{ik} \in [0,1]$ $\sum_{k=1}^{q}\mu_{ik} = 1$ on a donc : $\frac{1}{n} \le V_{PC} \le 1$

Partition Entropy V_{PE}

Si la valeur de V_{PE} tend vers sont minimum 0, on aura une partition constituée de groupes bien séparés. Et s'il atteint sa valeur maximale $log_a(c)$ la partition en question n'a aucune structure de groupes. Alors le nombre de groupes qui minimise V_{PE} indique le nombre optimal de groupes.

$$V_{PE}(U) = \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{q}\mu_{ik}\log_a(\mu_{ik}) \qquad \text{(IV.10)}$$

Où $a \in (0,\infty)$représente la base logarithmique. La valeur de V_{PE} est comprise entre: $0 \le V_{PE} \le \log_a(q)$.

- Deuxième catégorie des indices de validité

L'évaluation du résultat d'un algorithme de classification par l'intermédiaire des indices de validité appartenant à cette catégorie est basée essentiellement sur deux facteurs:

- La cohésion interne ou " compactness " : pour une partition aussi pertinente que possible, les objets appartenant au même groupe doivent être les plus proches que possible les uns des autres afin de former des structures compactes. L'idée ici est de maximiser la similarité entre les objets de même groupe.

- L'isolation externe ou séparation : l'objectif ici est de maximiser la distance entre les points représentant les groupes (un groupe est représenté par sont prototype).

Parmi les indices appartenant à cette catégorie nous citons :

L'indice de Fukayama et Sugeno V_{FS} :

V_{FS} [103] est une combinaison linéaire entre la cohésion interne globale "compactness" d'un ensemble de données et la fonction de séparation. Cet indice est défini comme suit :

$$V_{FS}(U,V,X) = J_m(U,V,X) - K_m(U,V,X) \qquad \text{(IV.11)}$$

Où *Jm* mesure la cohésion interne globale: $J_m(U,V,X) = \sum_{i=1}^{n}\sum_{k=1}^{q}\left[\left(\mu_{ik}^m\right)\|x_i - v_k\|^2\right]$

Plus une partition contient des groupe compact plus la valeur de J_m devient de plus en plus petite.

K_m est la fonction de séparation: $K_m(U,V,X) = \sum_{i=1}^{n}\sum_{k=1}^{q}\left[\left(\mu_{ik}^m\right)\|v_k - \bar{v}\|^2\right]$

K_m mesure la séparation entre les centres de groupe et le centre moyen de tout l'ensemble de donnée *v*, avec: $\bar{v} = \frac{1}{n}\sum_{i=1}^{n} x_i$.

Le nombre de groupes qui minimise V_{FS} correspond au meilleur résultat d'un algorithme de segmentation.

L'indice de H. Sun, S. Wang et Q. Jiang V_{WSJ}

H. Sun, S. Wang et Q. Jiang [69] ont proposé un indice de validité basé sur une combinaison linéaire entre la cohésion interne et la séparation. Il a la forme générale suivante:

$$V_{wsJ}(U,V,X) = Scat + \frac{Separation(q)}{Separation(q_{\max})} \qquad \text{(IV.12)}$$

Scat (*q*) est définie par : $Scat(q) = \frac{\frac{1}{q}\sum_{k=1}^{q}\|\sigma(v_k)\|}{\|\sigma(X)\|}$

Afin d'achever un meilleur équilibre entre *Scat*(q) et *Separation*(q), la séparation est définie comme suit:

$$Separation(q) = \frac{D_{\max}^2}{D_{\min}^2} \sum_{k=1}^{q} \left(\sum_{j=1}^{q} \left\| v_k - v_j \right\|^2 \right)^{-1}$$

Avec $D_{\min} = \min_{k \neq j} \left\| v_k - v_j \right\| \quad (k, j \in [1, q]), \qquad D_{\max} = \max_{k,j} \left\| v_k - v_j \right\| \quad (k, j \in [1, q]).$

Le nombre de groupe q qui minimise V_{WSJ} est retenu comme le nombre de groupe optimal.

- Troisième catégorie des indices de validité

Les indices présentés ici sont ceux de Gath et Geva [99]. Ils ont proposé trois indices basés essentiellement sur deux critères, hypervolume et densité. Ces trois indices sont :

Fuzzy Hypervolume V_{FH}

V_{FH} calcule la somme des volumes de tous les groupes présents dans la partition à évaluer. Le minimum de V_{FH} indique que la partition en question contient des structures denses et compactes, ce qui correspond à la partition optimale.

$$V_{fh}(Cov) = \sum_{k=1}^{q} \left[\sqrt{\det(Cov_k)} \right] \qquad \text{(IV.13)}$$

Cov_k est la matrice de covariance: $Cov_k = \dfrac{\sum_{i=1}^{n} \mu_{ik} (x_i - v_k)(x_i - v_k)^t}{\sum_{i=1}^{n} \mu_{ik}}$

Partition Density V_{PD}

V_{PD} correspond à l'interprétation physique de la notion de la densité (nombre de point par volume). Il doit être maximisé, car les groupes appartenant à la partition à évaluer doivent correspondre à des accumulations de points distincts.

$$V_{PD}(U, Cov) = \frac{\sum_{k=1}^{q} S_k}{V_{FH}(Cov)} \qquad \text{(IV.14)}$$

S_k est la somme des éléments répartis autour du centre. Elle s'énonce comme suit:

$$S_k = \sum_{l \in w_k} \mu_{ik}, \quad w_k = \left\{ t \in N \leq n : \left[(x_i - v_k)^T Cov_k^{-1} (x_i - v_k) \right] < 1 \right\}$$

On peut définir w_k comme un ensemble de points répartis dans une région spécifique autour du centre du $k^{ème}$ groupe. Cette région peut être vue comme une boule ouverte avec un centre v_k et un rayon égal à 1. Chaque élément x_i appartenant à cet espace doit vérifier la norme de Mahalanobis.

Average Partition Density V_{APD}

V_{APD} calcule la densité total moyenne de chaque groupe. Il s'énonce comme suit:

$$V_{APD}(U, Cov) = \frac{1}{q} \sum_{k=1}^{q} \frac{S_k}{\sqrt{\det(\text{cov}_k)}} \qquad \text{(IV.15)}$$

Plus il y a des points dans la région définie par w_k, plus la valeur de V_{APD} devient grande. Ce la revient à dire que le nombre de groupe qui maximise V_{APD} est retenu comme le nombre de groupes optimal.

IV.5 Conclusion

Dans ce chapitre, nous avons présenté le principe fondamental de segmentation. Différentes méthodes de segmentation ont été présentées. Un accent particulier a été mis sur les méthodes de segmentation par partition, plus particulièrement les algorithmes de segmentation floue non supervisée. Comme toutes autres approches non supervisées, ces algorithmes souffrent du problème du choix du bon nombre de groupes qui est souvent laissé à l'utilisateur. Dans ce contexte, nous avons présenté les différents indices de validité qui permettent d'optimiser le nombre de groupes.

Nous présentons dans ce chapitre notre proposition globale. Nous considérons que celle-ci s'étale sur trois volets. Le premier volet concerne le partitionnement automatique par la segmentation floue. A cette fin, nous proposerons trois approches différentes pour déterminer le nombre de groupes. A travers le second volet, nous présentons l'approche utilisée pour la recherche des règles d'association floues. Et afin d'évaluer la pertinence des règles d'association floues trouvées, nous proposons d'utiliser les mesures de comparaisons floues.

V.1 Principe général

Nous présentons dans ce travail de partitionner le domaine de valeurs d'un attribut quantitatif en partitions floues (groupes flous, ensemble flous) par un algorithme de segmentation floue non supervisé (FCM).

Le problème majeur des algorithmes de segmentations est de déterminer le nombre de partition (groupes, ensemble) pour chaque domaine d'attributs. Pour pallier à ce problème, nous proposons d'appliquer la méthode d'Agrawal et al. introduite dans [21], et nous proposons d'utiliser deux indices de validités, puis nous comparons les résultats obtenu par l'application de la méthode d'agrawal et al. et les résultats obtenu par l'application des indices de validités.

Après que les partitions floues (groupes flous, ensemble flous) seront obtenues nous cherchons les règles d'association floues, l'approche adoptée est l'approche ensembliste (cf. III.2.1).

Pour réaliser ces étapes nous proposons la procédure suivante, qui se compose de quatre étapes :

Etape 1 : trouver le nombre optimal de partition (groupes, ensemble) pour chaque attribut quantitatif.

Etape 2 : trouver les partitions floues (groupes flous, ensemble flous).

Etape 3 : trouver les règles d'association floues.

Etape 4 : évaluer la pertinence des règles d'association floues.

Les étapes suivantes sont illustrées par la figure figure V.1.

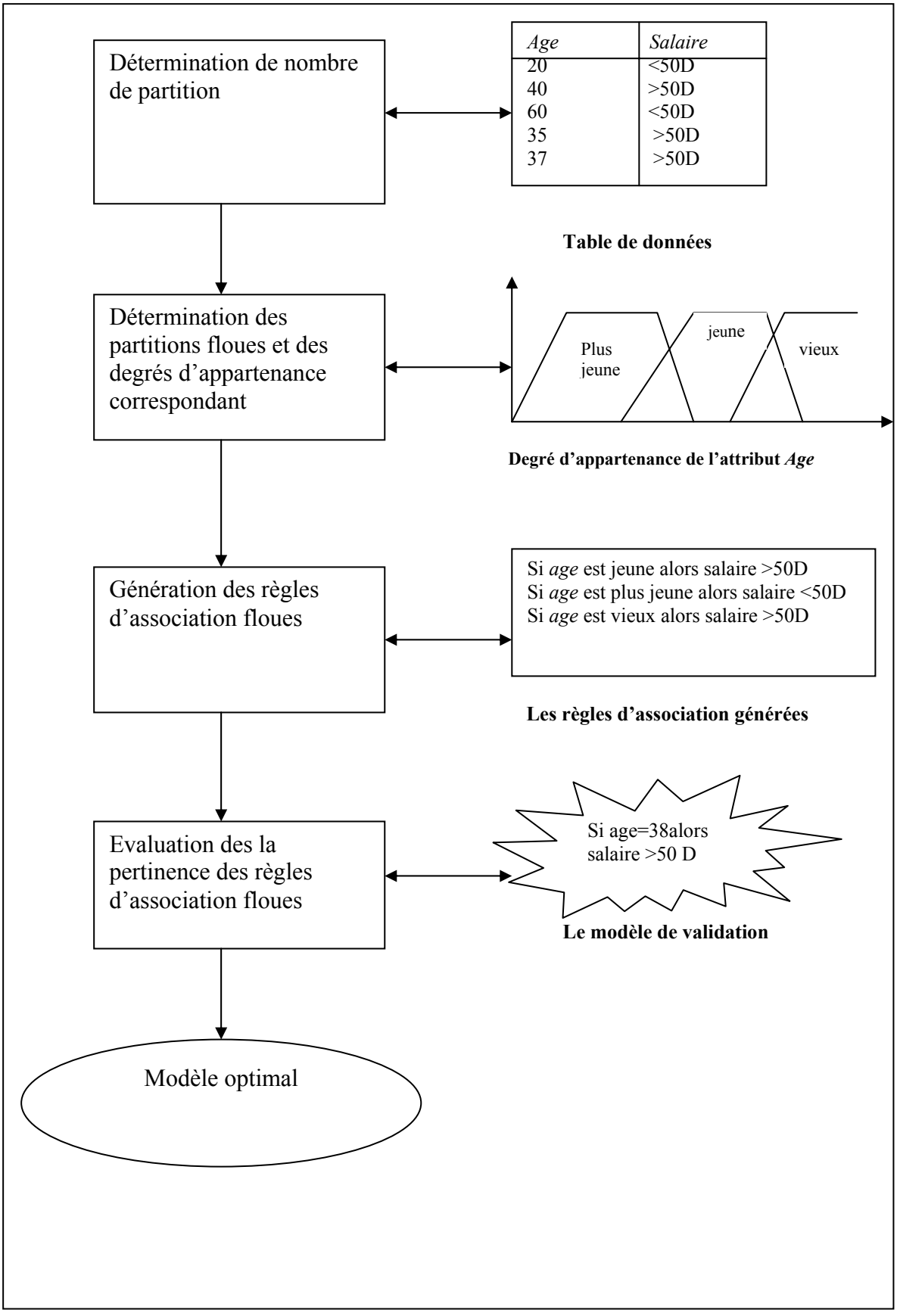


Figure V.1 **:** *La procédure générale proposée*

V.1.1 Adaptation de l'Algorithme CMF

Nous avons déjà présenté dans le paragraphe (§ IV.2) le principe algorithmique de la segmentation floue (algorithme CMF). Nous détaillons maintenant notre algorithme de partitionnement automatique qui est inspiré (ou adoptée) de l'algorithme classique de segmentation floue.

L'algorithme utilisé est le suivant :

2. étape 01 : initialisation

- déterminer le nombre de regroupements q : afin de déterminer le nombre de regroupement, nous proposons d'utiliser trois méthodes : la méthode d'Agrawal et al., et deux méthodes qui utilisent les indices de validités (V_{PC}, V_{FS}).

- le degré de flou m : plusieurs littératures [96] [97] [98], proposent d'initialiser m à 2. A notre tour, nous retiendrons cette valeur,

- la métrique utilisée : nous proposons d'utiliser la distance Euclidienne comme métrique.

$$d(x,y)=\sqrt{\sum_{k=1}^{n}\left|x_k-y_k\right|^2} \qquad \text{(V.1)}$$

  Pour le cas de notre étude, x représente le centre de groupe, est y représente une valeur de l'attribut. Par l'application de cette distance, on trouve des groupes de forme sphérique,

- la matrice V : l'initialisation de la matrice V est faite d'une manière aléatoire,

- t=1, t représente le nombre d'itération.

3. étape 02 : calcul des degrés d'appartenance

- nous proposons de déterminer le degré d'appartenance de chaque objet aux regroupements par les équations suivantes :

$$\left.\begin{array}{ll} \mu_{jk} = \dfrac{1}{\sum_{l=1}^{q}\left(\dfrac{d^{2}(x_j, v_k)}{d^{2}(x_j, v_l)}\right)^{\frac{1}{m-1}}} & \text{si } J_j = \phi \\ \left.\begin{array}{ll} \mu_{jk} = 0 & \text{si } k \notin J_j \\ \sum_{k \in Jj} \mu_{jk} = 1 & \text{si } k \in J_j \end{array}\right\} & \text{si } J_j \neq \phi \end{array}\right\} \quad \text{(V.2)}$$

$$J_j = \{k / 1 \le k \le q, d^2(x_j, v_k) = 0\}.$$

- calculer la fonction objective $J^t_{V,U,X}$ par l'équation

$$J_{V,U,X} = \sum_{i=1}^{q}\sum_{j=1}^{N}(\mu_{ji})^m d^2(x_j, v_i) \quad \text{(V.3)}$$

- si $t \neq 1$, alors si $J^t_{V,U,X} - J^{t+1}_{V,U,X} < \xi$ alors
- mémoriser V dans V_{svgd}
- mémoriser U dans U_{svgd}
- fin de l'algorithme
- sinon aller a étape3.

4.étape 03 : mise à jour des centres des groupes

- recalculer V par l'équation

$$V_c = \frac{\sum_{j=1}^{N}(\mu_{jc})^m x_j}{\sum_{j=1}^{N}(\mu_{jc})^m} \quad \text{(V.4)}$$

- recalculer la fonction objective $J^{t+1}_{V,U,X}$ par l'équation (V.3).

5.étape 04 : test d'arrêt

- si $J^{t}_{V,U,X} - J^{t+1}_{V,U,X} < \xi$ alors
- mémoriser V dans V_{svgd}
- mémoriser U dans U_{svgd}
- fin de l'algorithme
- sinon, incrémenter t : t=t+1.

➢ Aller a étape2.

Remarques

- Pour l'étape d'initialisation de la matrice V des centroïdes, les valeurs de V sont choisies aléatoirement sur l'ensemble de valeurs de l'attribut.
- Les matrices des degrés d'appartenance et les matrices des centroïdes, sont sauvegardées pour les utilisées aux étapes suivantes de l'application proposée.

Dans ce qui suit nous présentons le formalisme de la méthode CMF que nous avons implémenter.

Formalisation de l'algorithme CMF :

Entrée :

- $\{X_1,...X_n\}$ ensemble de données,
- q le nombre de groupes,
- $V^{(0)}$ l'ensembles des centres de groupes initiaux,
- d une métrique,
- m=2 le coefficient de flou

Sortie :

- La matrice des degrés d'appartenance U,
- La matrice des centres de groupes V.

Début

1- $t = 1$, // le nombre d'itération.

2- Mise à jour des degrés d'appartenance par l'équation (V.2).

3- Calcul de la fonction objective J_t par l'équation (V.3).

4- Si $t \neq 1$, alors si $J^t_{V,U,X} - J^{t+1}_{V,U,X} < \xi$ alors

1. mémoriser U_t dans U_{svgd}
2. mémoriser V_t dans V_{svgd}
3. fin de l'algorithme

sinon

1. aller a étape5.

5- Mise à jour des centres des classes par l'équation (V.4).

6- Recalcule de la fonction objective J_{t+1} par l'équation (V.3).

7- Test de convergence: si $J^t_{V,U,X} - J^{t+1}_{V,U,X} < \xi$ alors

1. mémoriser U_t dans U_{svgd}
2. mémoriser V_t dans V_{svgd}
3. fin de l'algorithme

sinon

1. *t=t+1 ;*
2. aller a étape 2*}*

Fin.

Algorithme V.1 : *L'algorithme des C-moyennes floues " CMF ".*

V.1.2 Justification du choix de l'algorithme CMF

Le choix de l'algorithme CMF pour notre étude est basé sur les critères suivants :

1- le choix d'un algorithme de segmentation non supervisée basé sur l'optimisation d'une fonction objectif est du à la complexité des algorithmes hiérarchiques [9].

2- le choix de CMF par rapport à d'autres algorithmes de segmentation non supervisé basés sur l'optimisation d'une fonction objectif est du à l'utilisation de concept des sous ensembles flous ce qui permet de trouver les partition floues de chaque attribut quantitatif, mais aussi le degré d'appartenance des valeurs de l'attribut à chaque partition floue (groupes flous, ensembles flous) trouvé. La flexibilité associée aux degrés d'appartenance fournit de très nombreux avantages [73][42], elle permet de représenter des catégories aux limites mal définies, en évitant l'arbitraire d'une valeur frontière entre deux propriétés, ainsi que des valeurs approximatives.

3- L'existence de différentes formes de fonction d'appartenance, engendre plusieurs modélisations pour les attributs quantitatifs. Par exemple pour la présentation de l'attribut *Age*, nous pouvons avoir plusieurs fonctions d'appartenance selon le nombre de partition, la forme de la fonction d'appartenance, et le choix des bornes (inférieur et supérieur). par exemple pour *Age*=15, plusieurs degrés d'appartenance sont possibles à données pour cette valeur (cf. figure V.2).

Par contre, l'utilisation d'une méthode déterministe (CMF), nous permette de donnée qu'un seul degré d'appartenance à une valeur d'une partition floue.

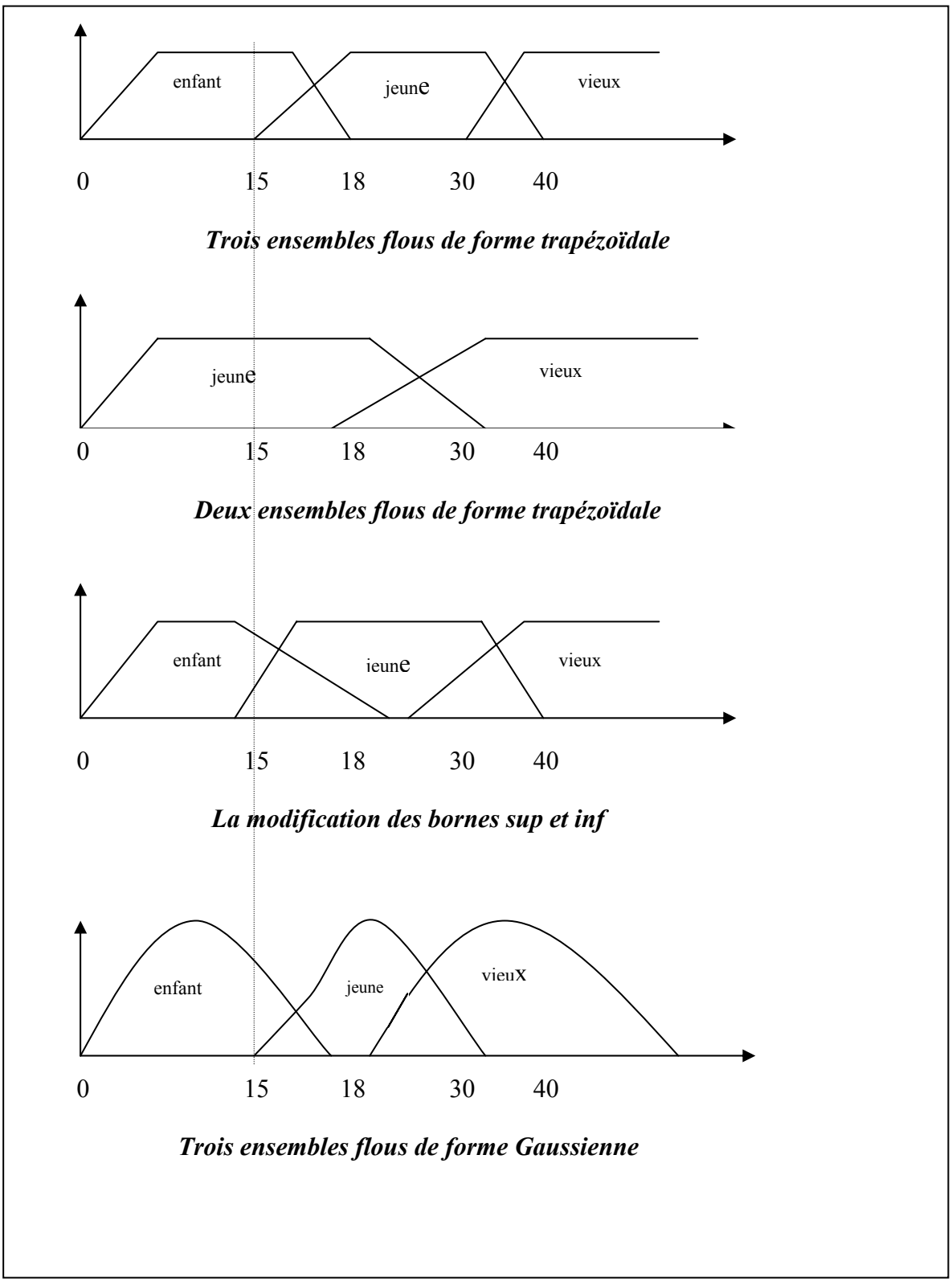


Figure V.2 **:** *L'influence d'une fonction d'appartenance sur la modélisation d'un attribut*

V.2 Détermination du nombre optimal de groupes

Pour déterminer le nombre optimal de groupes, nous proposons d'utiliser trois méthodes. La première méthode s'inspire de la notion de support, les deux autres méthodes sont basées sur les indices de validités. Il est à noter que ces méthodes sont fondamentalement différentes. En effet, si les méthodes utilisant les indices de validité sont issues de travaux de recherche sur la segmentation, la méthode utilisant le support est inspirée de travaux de recherche sur les règles d'association quantitatives.

Nous présentons plus en détail ces méthodes dans les paragraphes suivants.

V.2.1 Proposition basée sur le support

Le nombre de partition (groupes, ensemble) est calculé par la méthode introduite par Agrawal et al. [21]. Dans [21], l'auteur propose d'utiliser la formule suivante pour calculer le nombre de partition.

$$\text{Nb\_intervalle}=\frac{2\times n}{m\times(k-1)} \qquad \text{(V.5)}$$

n : nombre d'attribut quantitatif,

m : support minimum,

k : mesure de complétude partielle (partial completeness).

Pour la mesure de complétude partielle, l'auteur propose d'utiliser la formule suivante :

$$k=1+\frac{2\times n\times s}{m} \qquad \text{(V.6)}$$

n : nombre d'attribut quantitatif,

m : support minimum,

s : support maximum d'une partition.

Pour cette méthode l'auteur propose de partitionner le domaine d'attribut a des intervalles de base (correspond à des partitions initiales). Par exemple nous pouvons partitionner le domaine d'attribut en m intervalle de largeur uniforme "*equi-wedth*", c'est-à-

dire que chaque intervalle prend la même proportion du domaine. Puis l'auteur cherche la partition qui a le plus grand support, et il calcule la valeur de *k.*

Nous remarquons que si on remplace *k* par sa formule dans (V.5), on obtient :

$$\text{Nb\_intervalle}=\frac{1}{s} \qquad \text{(V.7)}$$

Pour le cas de notre étude, on considère que les partitions initiales ont une largeur qui égale à cinq. C'est-à-dire que chaque partition contient cinq éléments et on calcule le support de chaque partition, on sauvegarde le support maximum *s*, puis on calcule le nombre d'intervalle pour chaque attribut quantitatif par la formule (V.5). Le nombre d'intervalle indique le nombre de partition.

Formalisation de l'algorithme :

Entrée :

- la base de transaction *T*

Sortie :

- *C* : tableau pour enregistrer le nombre de groupe pour chaque attribut

---

Debut

Suppmax=0 ;

Pour *t* =0 ;*t*<nb_transaction ;*t*++{

Pour chaque valeur d'attribut d'un intervalle

1. calculer sa fréquence (*supp*)
2. If (*supp*> *Suppmax)*

Suppmax= supp;

}

C[*i*]=1/*Suppmax;*

}

Fin

Algorithme V.2: *Algorithme nbgroupe*

V.2.2 Proposition utilisant l'indice de validité V_{PC}

Pour notre étude nous utilisons l'indice de validité V_{PC} proposée par Bezdek et al. [101] [102]. Cet indice appartient à la première catégorie des indices de validité dédiés à la segmentation floue. Il est défini comme suit :

$$V_{PC} = \frac{1}{n}\sum_{k=1}^{n}\sum_{i=1}^{q}\mu_{ki}^{2} \qquad \text{(V.8)}$$

n : représente le nombre de valeur d'un attribut donnée

q : représente le nombre de groupe

μ_{ki} : représente le degré d'appartenance de l'élément k au groupe i

Si la valeur de V_{PC} tend vers son maximum 1, pour un certain nombre de groupes k, on aura une partition qui est constituée de groupes bien séparés. Si la partition en question ne contient aucune structure de groupes, V_{PC} atteint sa valeur minimale 0. Le nombre de groupes q qui maximise V_{PC} indique le nombre optimal de groupes.

V.2.3 Proposition utilisant l'indice de validité V_{FS}

Cet indice proposé par Fukayama et Sugeno [103] appartient à la deuxième catégorie d'indices de validité dédiés à la segmentation floue. Cet indice est défini comme suit :

$$V_{FS}(U,V,X) = J_m(U,V,X) - K_m(U,V,X) \qquad \text{(V.9)}$$

Où J_m mesure la cohésion interne globale et K_m est la fonction de séparation

$$J_m(U,V,X) = \sum_{i=1}^{n}\sum_{k=1}^{q}\left[\left(\mu_{ik}^{m}\right)\left\|x_i - v_k\right\|^2\right] \qquad \text{(V.10)}$$

$$K_m(U,V,X) = \sum_{i=1}^{n}\sum_{k=1}^{q}\left[\left(\mu_{ik}^{m}\right)\left\|v_k - \bar{v}\right\|^2\right] \qquad \text{(V.11)}$$

où $\bar{v}$ est la moyenne des centres de classes.

V : la matrice de centre de classe.

U : la matrice des degrés d'appartenance

Le nombre de groupes qui minimise V_{FS} correspond au meilleur résultat d'un algorithme de segmentation.

Formalisation de l'algorithme :

Afin de calculer le nombre de groupes par l'application d'une méthode qui utilise un indice de validité, nous appliquons l'algorithme V.3

Entrée :

- la base de transaction *T,*
- C_{min}=2, C_{max}=10.

Sortie :

- q : le nombre optimal de groupe pour chaque attribut

Debut

Pour j= C_{min} jusqu'à C_{max} {

1- Appliquer l'algorithme FCM

2- Calculer l'indice de validité V_d

si $V_d(j)$ est optimale alors $q=j$

}

Fin

Algorithme V.3 : *Détermination de nombre de groupe par un indice de validité*

Le nombre de groupes est calculé pour chaque attribut. Lorsque la valeur de V_d est optimale pour un nombre de groupe q alors, q représente la valeur finale choisie.

Pour le choix de C_{min} et C_{max}, dans [107], ces valeurs sont initialisées comme suit C_{min} =2 et C_{max}=10. A notre tour, nous retiendrons ces valeurs.

V.3 Proposition pour la découverte des règles d'association floues

Nous proposons d'opter pour l'approche ensembliste afin d'induire les règles d'association floues. Rappelons que cette approche pose le problème de la definition d'une mesure de cardinalité et la définition d'une $\top$-norme, pour le calcul de support et de confiance d'une règle d'association floues.

$$Supp(A \Rightarrow B) = \text{ card } (\top (A(x_i), B(x_i))) / \text{ card } (T)$$

$$conf(A \Rightarrow B) = \text{ card } (\top (A(x_i), B(x_i))) / \text{ card } (A(x_i))$$

Avons de donner nous propositions sur la question, nous allons décrire la sémantique des règles d'association floues.

V.3.1 Sémantique des règles d'association floues

Une règle d'association floue se présente sous la forme " Si X est A alors Y est B ", cette règle se note $(X, A) \Rightarrow (Y, B)$ telle que (X, A) est la condition de la règle, et (Y, B) est la conclusion de la règle. $X = \{x_1, ..., x_p\}$ et $Y = \{y_1, ..., y_q\}$ sont deux itemsets disjoints, X et Y sont deux sous-ensemble de I. $A = \{f_{x_1}, f_{x_2}, ..., f_{x_p}\}$ et $B = \{f_{y_1}, f_{y_2}, ..., f_{y_q}\}$ sont les ensembles des sous ensemble flous associer aux éléments de X et Y, par exemple l'item x_k de X est associer à l'ensemble flou f_{x_k} de A.

La sémantique d'une règle d'association floue est, lorsque "X est A" est satisfaite, alors que "Y est B" et aussi satisfaite. C'est-à-dire qu'il y a un nombre suffisant de transactions de T supportant les paires [attribut x_k, sous-ensemble floue f_{x_k}] et [attribut y_c, sous-ensemble floue f_{y_c}].

Pour satisfaire ces conditions nous adoptons les règles d'association floues de type graduel. Ce type exprime une synergie entre les attributs flous dans la partie condition et dans la partie conclusion. Le sens général d'une règle graduelle est :

"**plus *X* est *A*, plus *Y* est *B***". Une règle d'association floue sera satisfaite si $\mu(X) \leq \mu(Y)$. Cette dernière contrainte sera appliquée dans notre processus de découverte.

V.3.2 Proposition d'une mesure de cardinalité floue

Nous proposons d'utiliser les α-coupes (α-cuts) afin de ramener le problème de calcul de la cardinalité d'un ensemble flou à celui du calcul de la cardinalité d'un ensemble classique (non flou) [115].

En effet, à travers la littérature existante et consultée sur la cardinalité des ensembles flous (cf. III.1.3) aucune des cardinalités floues proposée ne semble adaptée à des problématique de fouille de donnée (très gros volume de données). Aussi, nous avons préféré l'utilisation de α-coupes.

V.3.3 Proposition d'une t-norme

Pour le choix d'une t-norme, plusieurs littératures [54] [70] [55], proposent d'utiliser l'opérateur "min" comme une t-norm. A notre tour, nous retiendrons l'opérateur "min" comme t-norme.

V.3.4 Présentation détaillée de l'algorithme de découverte des règles d'association floues

L'évaluation d'une règle d'association floue est basée sur la mesure de support et de la confiance. Pour notre cas d'étude le support et la confiance d'un sous-ensemble I_0 sont déterminés par l'utilisation d'une cardinalité floue basée sur les α-coupes [115], et de l'opérateur "min" pris comme une t-norme.

Avant de présenter l'algorithme proposé nous présentons la notation utilisée,

Soit I un ensemble d'items $(I = \{x_1,...,x_m\})$, T un ensemble de transactions $(T = \{t_1,...t_n\})$.

D_j :	Domaine de l'item x_j
h_j :	Nombre de sous-ensembles flous pour l'item x_j
v_{ij} :	Valeur de l'item x_j dans la transaction t_i
f_{ij} :	La valeur floue transformer à partir de v_{ij}

Avant de chercher les règles d'association floue, l'ensemble de transactions T est transformé en un ensemble de transaction floues FT (cf. III.2.1.3). Pour chaque transaction non floue, la valeur v_{ij} est transformée en valeur floue, qui indique le degré d'appartenance de la valeur v_j à la région floue correspondante.

Sur l'ensemble FT des transactions floues, nous définissons une coupe de niveau α notée $FT_{\tilde{I}_0/\alpha}$ sur un sous-ensemble flou $\tilde{I}_0 \subseteq \tilde{I}$ tel que $\tilde{I}_0 = \{i_1,...,i_p\}$ par :

$$FT_{\tilde{I}_0/\alpha} = \{\tilde{t}_l, l = 1,n / \min_{j=1,p}(\mu(i_j, \tilde{t}_l)) \geq \alpha\} \qquad \text{(V.12)}$$

où $\mu(i_j, \tilde{t}_l)$ correspond au degrés d'appartenance de l'item i_j dans la transaction floue $\tilde{t}_l$.

- ***Support d'un itemset flou***

  Nous proposons le calcul du support d'un itemset I_0 comme suit :

$$\text{supp}(\tilde{I}_0) = \sum_{\alpha_i \in \Delta} (\alpha_i - \alpha_{i+1}) \frac{|FT_{\tilde{I}_0/\alpha_i}|}{|FT|} \qquad \text{(V.13)}$$

  où $\Delta = \{\alpha_1,...,\alpha_k\}$ avec $\alpha_i > \alpha_{i+1}$ pour chaque $i \in \{1,...,k\}$.

- ***Confiance d'une règle d'association floue***

  La confiance d'une règle d'association floue se déduit à partir de la formule (V.13) :

$$conf(X \Rightarrow Y) = \frac{\text{supp}(X \cap Y)}{\text{supp}(X)} \qquad \text{(V.14)}$$

L'algorithme de découverte des règles d'association floues proposé est décrit comme suit :

Entrée :

- Un ensemble $\widetilde{T}$ de transaction floue,
- Un seuil minimal de support λ,
- Un seuil minimal de confiance γ,
- Une partition floue pour chaque domaine d'attribut.

Sortie :

- Ensemble de règles d'association floues.

Début

Etape 01 :

- Déterminer les valeurs floue de chaque item.

Etape 02 :

- Déterminer l'ensemble L_1 de 1-itemset fréquent.

$$L_1 = \{i_j / \text{supp}(i_j) \geq \lambda\}$$

- Si $L_1 = \phi$ terminer l'algorithme.
- Sinon r=2 {r calcule la taille de l'itemset}.

Etape 03 :

- Générer l'ensemble des candidats C_r à partir de L_{r-1} selon la méthode proposée dans [7].

Etape 04 :

- Pour chaque k-itemset $S \in C_r$, calculer le support des S $\text{supp}(S)$ selon (V.13)
- Si support de S $\text{supp}(S) \geq \lambda$ alors $L_r = L_r \cup \{S\}$.

Etape 05 :

- Si $L_r \neq \phi$, $r = r + 1$, allez a l'étape 03.

Etape 06 :

- Générer toutes les règles d'association de la forme $X \Rightarrow Y$ tel que $X \cap Y = \phi$ et $X, Y \neq \phi$

Etape 07 :

- Calculer la confiance $conf(X \Rightarrow Y)$ de chaque règle.
- Si $conf(X \Rightarrow Y) \geq \gamma$, générer la règle $X \Rightarrow Y$ en sortie.

Fin

***Algorithme V.4*:** *Recherche des règles d'association floues*

V.3.5 Evaluation des règles d'association floues

La recherche des règles d'association floues se fait après l'étape de découverte des partitions floues (groupes flous, ensembles flous), pour laquelle nous avons proposé trois méthodes différentes. Conséquemment nous obtiendrons trois ensembles de règles d'association floues (selon la méthode utilisée pour trouver les partitions floues). Pour pouvoir comparer ces trois ensembles de règles d'association par rapport à un modèle de règle d'association, nous proposons d'utiliser la notion de mesure de similarité floue.

Les mesures de similarité sont divisées en deux catégories [111] : les mesures géométriques et les mesures entre ensembles. Pour les mesures entre ensemble classique (non flou), Tversky [113] a proposé une approche basée sur la comparaison entre ensemble de caractéristiques. Dans le cadre proposé, les objets comparés sont décrits par leurs ensembles de caractéristiques binaires. Pour comparer deux objets *a* et *b*, décrits par leurs ensembles de caractéristiques *A* et *B*, les mesures de Tversky mettent en rapport trois composantes : les caractéristiques communes à *a* et *b* $(A \cap B)$, et les caractéristiques propres à *a* $(A - B)$et propres à *b* $(B - A)$. Une mesure de similarité *s* est ainsi définie par une fonction réelle *F* de trois variables $S(a,b) = F(A \cap B, A - B, B - A)$. Les travaux de Tversky aboutissent à la proposition d'un modèle de mesure. Avec une mesure *f* permettant de mesurer les trois ensembles de caractéristiques communes et distinctives.

$$S(A,B) = \frac{f(A \cap B)}{f(A \cap B) + \alpha . f(B - A) + \beta . f(A - B)} \qquad \text{(V.15)}$$

où α et β, deux paramètres réels positifs, permettent de pondérer l'importance accordée aux deux ensembles distinctifs de caractéristiques.

A cause de la restriction des mesures de Tversky aux caractéristiques binaires (les ensembles *A* et *B* sont classiques), une extension de ces mesures a été proposée [112] [114], pour permettre la comparaison entre ensembles graduels (ou flous). Dans ce qui suit nous présentons en détaille les mesure de comparaisons des ensembles flous.

V.4 Comparaison des ensembles flous

Pour Ω un ensemble d'élément, on a $F(\Omega)$ l'ensemble des sous ensembles flous de Ω, μ_A la fonction d'appartenance de tout ensemble flou A de $F(\Omega)$. La comparaison de deux ensembles flous A et B définis sur un univers de référence Ω, prend en considération les éléments de A et B. Les mesures de comparaison mettent en rapport trois composantes : les caractéristiques communes $(A \cap B)$, et les caractéristiques propres à A et non a B et propres à B et non a A, et aussi elle prend en considération les degrés d'appartenance des éléments aux ensembles A et B.

Pour identifier les caractéristiques propres à A et non a B et propres à B et non a A, il été nécessaire d'introduit un opérateur de différence [112] (voir Annexe1).

Définition : (*M*-mesure de comparaison)

Une *M*-mesure de comparaison *S* sur Ω est une fonction $S : F(\Omega) \times F(\Omega) \to [0,1]$ telle que :

$$S(A,B) = f_s(M(A \cap B), M(B-A), M(A-B)) \qquad \text{(V.16)}$$

où f_x est une fonction $f_x : \Re^3 \to [0,1]$ et M une mesure d'ensemble flous sur $F(\Omega)$.

Les mesures de similarité floues couvrent différentes familles de mesure (cf. figure V.3). Dans le cas de notre étude, nous nous intéressons principalement aux mesures dites mesure de ressemblance. Nous détaillerons ces mesures dans le paragraphe suivant.

Dans le paragraphe suivant nous utilisons les notations suivantes :

$X = M(A \cap B)$
$Y = M(B-A)$
$Z = M(A-B)$

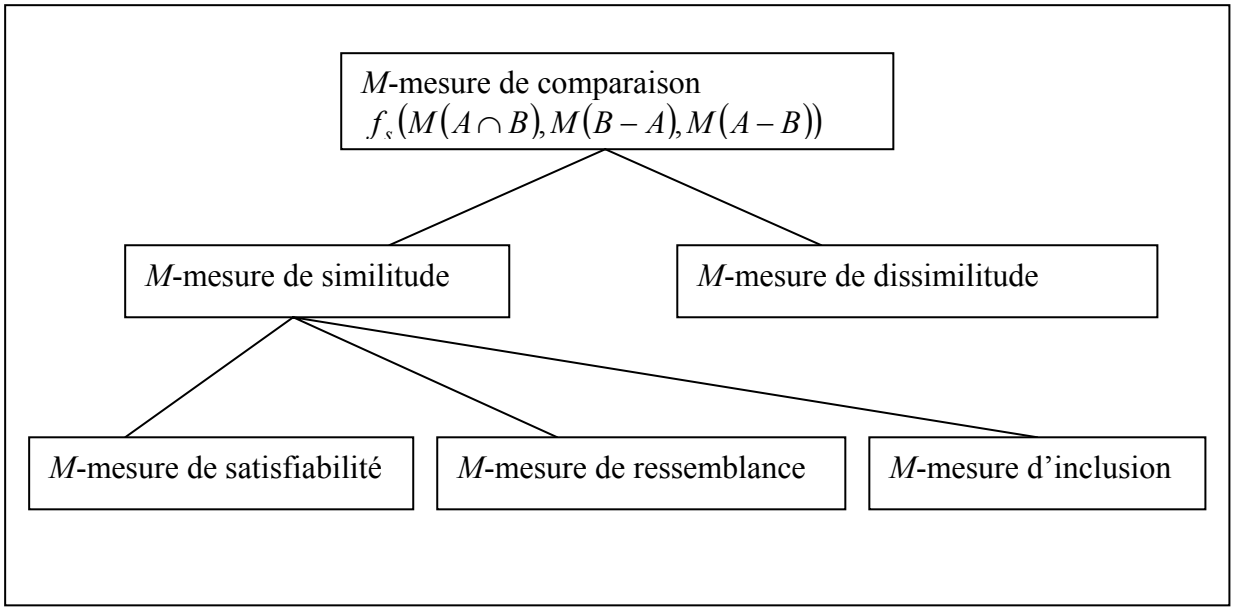


***Figure V.3* :** *Différentes mesures de comparaison*

V.4.1 Mesure de ressemblance

Les mesure de ressemblance sont utilisées pour comparer deux objet de même niveau de généralité, afin d'extraire les caractéristiques communes [114].

Définition : (*M*-mesure de ressemblance)

Une *M*-mesure de ressemblance *S* sur Ω est une *M*-mesure de similitude *S* telle que :

- $\forall X \neq 0 \;\; f_s(X,0,.) = 1$ (réflixivité),
- $\forall X\, Y\, Z \; f_s(X,Y,Z) = f_s(X,Z,Y)$ (symétrie).

Exemple de M-mesure de ressemblance :

- $S(A,B) = \dfrac{M(A \cap B)}{M(A \cup B)}$ avec M_1 et M_3 les mesures floues (voir Annexe1).
- $S(A,B) = 1 - \dfrac{1}{|\Omega|}\sum_x |\mu_A(x) - \mu_B(x)| = 1 - \dfrac{1}{|\Omega|}(M_3(A-B) + M_3(B-A))$ et comme opérateur de différence l'opérateur (Annexe1.4).

Pour le cas de notre étude, pour évaluer la pertinence des règles d'association générées à partir de chaque méthode, nous proposons de les comparées à un modèle de validation de

règle d'association générer à partir de la base de transaction initiale (sans transformation à une base de transaction floue).

V.4.2 L'agrégation des mesures de ressemblances

Soit Ω l'ensemble d'objets qui sont décrit par l'utilisation des attributs définis dans les ensembles $\Omega_1, \Omega_2, ..., \Omega_n$. Un objet O est associé aux valeurs $A_1, A_2, ..., A_n$ des attributs, respectivement définit comme des ensembles flous de $\Omega_1, \Omega_2, ..., \Omega_n$. Nous considérons un autre objet O' associé aux valeurs $B_1, B_2, ..., B_n$.

- Un objet O est inclus dans l'objet O' si est seulement si $\forall i$ A_i est inclus dans B_i ,

- L'intersection de O et O' est définit comme un objet $O \cap O'$ avec les valeurs d'attributs $A_i \cap B_i$,

- La différence entre un objet O et un objet O' est définit comme un objet $O - O'$ avec les valeurs d'attributs $A_i - B_i$.

Si on veut donnée une valeur général de degré de ressemblance S définit sur Ω, qui satisfait les propriétés des mesure de ressemblance, [111] [112] [114] propose d'utiliser une t-norme comme opérateur d'agrégation.

Pour le cas de notre étude nous utiliserons l'opérateur min comme opérateur d'agrégation.

V.5 Conclusion

Nous avons présenté dans ce chapitre nos differentes propositions. Notre approche globale est constituée de quatre étapes, la première étape consiste à la détermination de nombre de groupe pour cette effet nous avons proposé trois approches, la première approche consiste à appliquer la méthode présenté dans [21], les deux autres approches, sont basées par l'utilisation d'un indice de validité pour cette effet nous avons utilisé deux types d'indice de validité. La deuxième étape consiste à générer les partitions floues, et puis nous avons

présenté un algorithme de découverte des règles d'association floues, et afin de valider la pertinence des règles d'association floues générées nous proposons d'utiliser les mesures de ressemblance floues, pour comparer le modèle de règles obtenues par rapport à un modèle de validation. Ce dernier modèle étant obtenu à partir d'une base d'apprentissage.

Dans le chapitre suivant nous donnons les résultats expérimentaux de nos propositions.

Le présent chapitre présente les différents résultats expérimentaux. Nous avons expérimenté notre proposition sur deux différentes Base de données. A cet effet nous présentons d'abord la description des bases de données. Nous nous intéressons aux règles d'association floues produites pour chaque base de données par les différentes partitions floues trouvées à partir des différentes méthodes proposées. Afin de comparer les différents modèles de règles d'association trouvés, nous utilisons les mesures de ressemblance.

VI.1 Présentation des bases de données

VI.1.1 La base de données *KDD'99*

Au courant de l'année 1999 et en marge de la conférence mondiale sur la découverte de connaissances dans les bases de données *KDD*, s'est tenue une compétition réservée pour les systèmes de détection d'intrusions. La compétition *KDD'99* Cup [110] a enregistré la participation de 24 algorithmes et IDS réseaux basés sur des approches en rapport avec le thème de la conférence (fouille de données, apprentissage automatique/classification, etc.).

Pour les besoins de cette compétition, les bases de données d'apprentissage et de test utilisées sont une version synthétisée de la base de donnée DARPA'98.

La base d'apprentissage DARPA'98 totalise à elle seule 4 gigaoctets de données binaires compressées générées par TCPDump. Dans la base de données *KDD*'99, on s'est restreint à l'utilisation des données d'audit réseau de DARPA'98 après avoir été traitées.

On a obtenu de la base de données DARPA'98 après traitement, l'équivalent de 5 millions(4898430) de connexions pour la base d'apprentissage et 0.3 millions(311029) pour la base de test.

Attributs d'une connexion dans la base de données KDD'99

Une connexion dans *KDD*'99 est caractérisée par 41 attributs dont certains sont exactement les mêmes que dans DARPA'98 alors que d'autres, appelés attributs experts ou de haut niveau sont élaborés et calculés ou dérivent de DARPA'98 et ce dans l'objectif de mieux

distinguer ou "discriminer" une connexion normale d'une connexion faisant partie d'une attaque. Ces attributs supplémentaires ont trait à :

- l'aspect temporel du trafic réseau [time-based traffic features],
- trafic par rapport à une machine hôte particulière [host-based traffic features],
- contenu (données utiles) des paquets[1] [content features].

Chaque connexion est décrite par 9 attributs de bases (intrinsèques) et 32 attributs de haut niveau. Les attributs de base décrivent les données au niveau paquet. Les attributs supplémentaires sont obtenus par des techniques de fouille de données ou suggérés par la connaissance experte du domaine et calculés à partir des connexions constituant la base de données DARPA'98. Ci après la description détaillée de l'espace d'attributs et de classes

Nom de l'attribut	Type
duration	Numérique (entier positif)
protocol_type	Symbolique {tcp,udp,icmp}
Service	Symbolique{http,smtp,mtp,domain,domain_u,auth,finger,telnet,eco_i,ftp,ntp_u,ecr_i,other,urp_i,private,pop_3, p_data,netstat,daytime,ssh,echo,name,whois,gopher,remote_job,rje,ctf,supdup,link,systat,iscard,X1shell,login, imap4,nntp,uucp,pm_dump,IRC,Z39_50,netbios_dgm,ldap,sunrpc,courier,exec,bgp,csne_ns,http_443,klogin, printer,netbios_ssn,pop_2,nnsp,efs,hostnames,uucp_path,sql_net,vmne,iso_tsap,netbios_ns,kshell,urh_i,http_2784 ,harvest, aol,tftp_u,http_8001,tim_i,red_i,time}
flag	Symbolique {SF,RSTO,REJ,S0,RSTR,SH,S3,S2,S1,RSTOS0,OTH}
src_bytes	Numérique (entier positif)
dst_bytes	Numérique (entier positif)
land	Logique {0,1}
wrong_fragment	Numérique (entier positif)
urgent	Numérique (entier positif)
hot	Numérique (entier positif)
num_failed_logins	Numérique (entier positif)
logged_in	logique{0,1}
num_compromised	Numérique (entier positif)
root_shell	logique{0,1}
su_attempted	Logique {0,1}
num_root	Numérique (entier positif)
num_file_creations	Numérique (entier positif)
num_shells	Numérique (entier positif)
num_access_files	Numérique (entier positif)
num_outbound_cmds	Numérique (entier positif)
is_host_login	Logique {0,1}
is_guest_login	Logique {0,1}
count	Numérique (entier positif)

srv_count	Numérique (entier positif)
serror_rate	Numérique [0,1]
srv_serror_rate	Numérique [0,1]
rerror_rate	Numérique [0,1]
srv_rerror_rate	Numérique [0,1]
same_srv_rate	Numérique [0,1]
diff_srv_rate	Numérique [0,1]
srv_diff_host_rate	Numérique [0,1]
dst_host_count	Numérique (entier positif)
dst_host_srv_count	Numérique (entier positif)
dst_host_same_srv_rate	Numérique [0,1]
dst_host_diff_srv_rate	Numérique [0,1]
dst_host_same_src_port_rate	Numérique [0,1]
dst_host_srv_diff_host_rate	Numérique [0,1]
dst_host_serror_rate	Numérique [0,1]
dst_host_srv_serror_rate	Numérique [0,1]
dst_host_rerror_rate	Numérique [0,1]
dst_host_srv_rerror_rate	Numérique [0,1]
attack_name (variable de classe)	Symbolique{apache2,back,buffer_overflow,ftp_write,guess_passwd,httptunnel,imap,ipsweep,land,ladmodule, mailbomb,mscan,multihop,named,neptune,nmap,normal,perl,phf,pod,portsweep,processtable,ps,rootkit,saint,satn, sendmail,smurf,snmpgetattack,snmpguess,spy,sqlattack,teardrop,udpstorm,warezclient,warezmaster,worm,xlock, xsnoop,xterm}

Table VI.1**:** *description des attributs de la base KDD'99.*

Attributs retenus pour le cas de notre étude

Pour le cas de notre étude, et dans le but d'extraire des règles d'association floues nous avons intéressé à un seul type d'attaque, nous avons choisi l'attaque de type Apache2, et pour ce cas nous avons choisi que quelque attribut qui présente une variation de valeurs. Pour les autres attributs nous avons constaté, qu'ils ont qu'une seule valeur pour ce type d'attaque. Dans ce qui suit nous présentons les attributs retenus pour notre étude.

Nom de l'attribut	Type
duration	Numérique (entier positif)
flag	Symbolique {SF,RSTO,REJ,S0,RSTR,SH,S3,S2,S1,RSTOS0,OTH}
src_bytes	Numérique (entier positif)
logged_in	Logique{0,1}
count	Numérique (entier positif)
srv_count	Numérique (entier positif)
serror_rate	Numérique [0,1]
srv_serror_rate	Numérique [0,1]
rerror_rate	Numérique [0,1]
srv_rerror_rate	Numérique [0,1]
srv_diff_host_rate	Numérique [0,1]
dst_host_count	Numérique (entier positif)
dst_host_srv_count	Numérique (entier positif)
dst_host_same_srv_rate	Numérique [0,1]
dst_host_diff_srv_rate	Numérique [0,1]
dst_host_serror_rate	Numérique [0,1]
dst_host_srv_serror_rate	Numérique [0,1]
dst_host_rerror_rate	Numérique [0,1]
dst_host_srv_rerror_rate	Numérique [0,1]
attack_name	apache2

Table VI.2**:** *description des attributs de base retenues pour le cas de notre étude*

VI.1.2 La base de données *adult*[1]

La base de donnée *adult*[1] a été donnée [68] pour indiquer les attributs qui peuvent prévoir un salaire inférieur ou égal à 50,000 Dollars. Dans ce qui suit nous présentons les attributs retenus pour le cas de notre étude.

1 :http//www.kdd.ics.edu/databases/adult

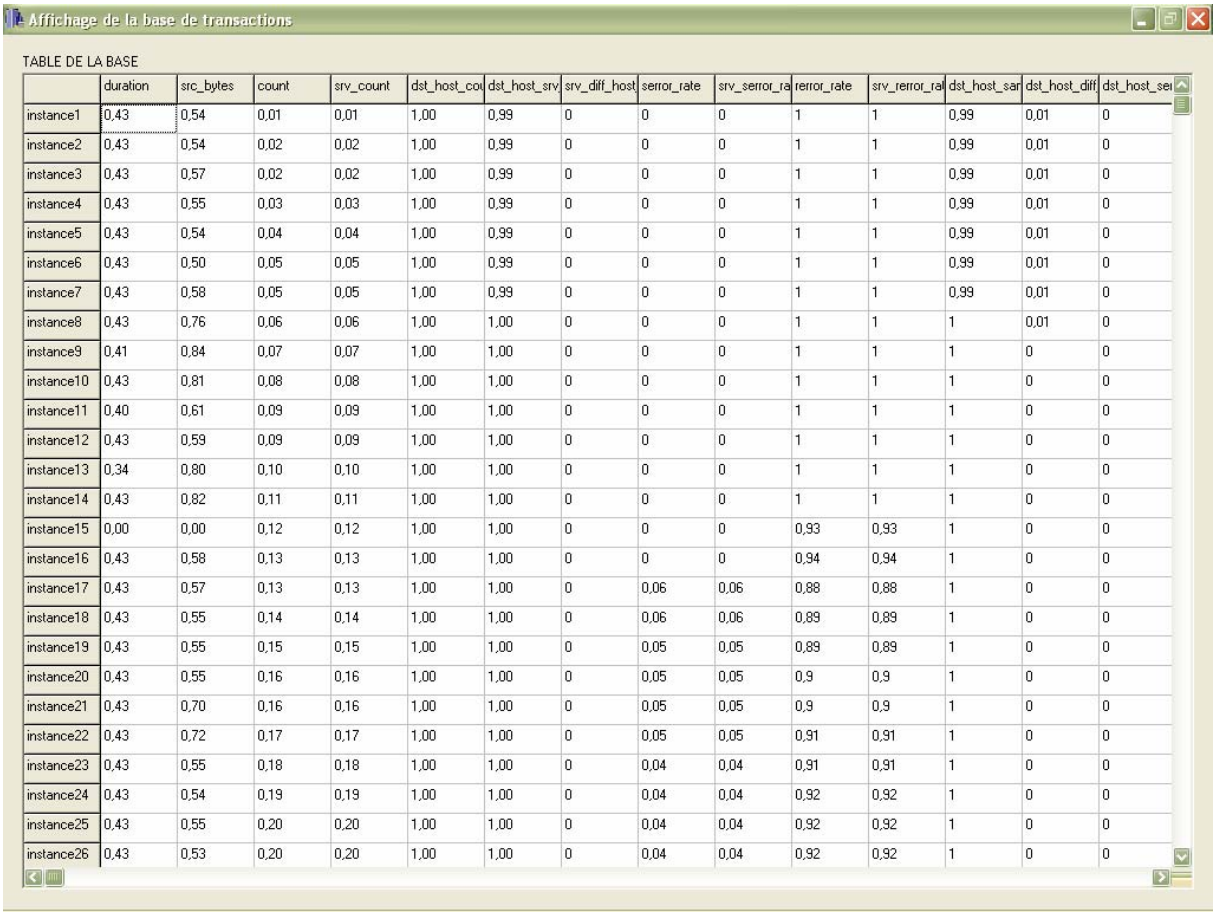
Nom de l'attribut	Type
Age	Numérique (entier positif)
fnlwgt	Numérique (entier positif)
education-num	Numérique (entier positif)
capital-gain	Numérique (entier positif)
capital-loss	Numérique (entier positif)
hours-per-week	Numérique (entier positif)
Nature de travail	Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked.
sex	Female, Male.
Type d'éducation	Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool.
marital-status	Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse
native-country	United-States, Cambodia, England, Puerto-Rico, Canada, Germany, Outlying-US(Guam-USVI-etc), India, Japan, Greece, South, China, Cuba, Iran, Honduras, Philippines, Italy, Poland, Jamaica, Vietnam, Mexico, Portugal, Ireland, France, Dominican-Republic, Laos, Ecuador, Taiwan, Haiti, Columbia, Hungary, Guatemala, Nicaragua, Scotland, Thailand, Yugoslavia, ElSalvador,Trinadad&Tobago, Peru, Hong, Holand-Netherlands.
class	<=50,000

Table VI.3**:** *description des attributs de la base adult*

Remarques

Il y a lieu de souligner que les bases de données sont converties au format *txt* car c'est ce format de données que nous avons adopté pour notre application. Ainsi que les valeurs des attributs sont séparées par des points virgules, la fin de chaque transaction est indiquée par un point. La première ligne indique les différents attributs de la base.

Dans notre application nous permettons à l'utilisateur d'afficher la base de transactions pour pouvoir la consulter, comme nous donnons les informations concernant le nombre de transaction, et le nombre d'attribut, et on indique le nombre d'item différent pour chaque attribut (cf. figure VI.1).



	duration	src_bytes	count	srv_count	dst_host_cou	dst_host_srv	srv_diff_host	serror_rate	srv_serror_ra	rerror_rate	srv_rerror_rat	dst_host_sar	dst_host_diff	dst_host_ser
instance1	0,43	0,54	0,01	0,01	1,00	0,99	0	0	0	1	1	0,99	0,01	0
instance2	0,43	0,54	0,02	0,02	1,00	0,99	0	0	0	1	1	0,99	0,01	0
instance3	0,43	0,57	0,02	0,02	1,00	0,99	0	0	0	1	1	0,99	0,01	0
instance4	0,43	0,55	0,03	0,03	1,00	0,99	0	0	0	1	1	0,99	0,01	0
instance5	0,43	0,54	0,04	0,04	1,00	0,99	0	0	0	1	1	0,99	0,01	0
instance6	0,43	0,50	0,05	0,05	1,00	0,99	0	0	0	1	1	0,99	0,01	0
instance7	0,43	0,58	0,05	0,05	1,00	0,99	0	0	0	1	1	0,99	0,01	0
instance8	0,43	0,76	0,06	0,06	1,00	1,00	0	0	0	1	1	1	0,01	0
instance9	0,41	0,84	0,07	0,07	1,00	1,00	0	0	0	1	1	1	0	0
instance10	0,43	0,81	0,08	0,08	1,00	1,00	0	0	0	1	1	1	0	0
instance11	0,40	0,61	0,09	0,09	1,00	1,00	0	0	0	1	1	1	0	0
instance12	0,43	0,59	0,09	0,09	1,00	1,00	0	0	0	1	1	1	0	0
instance13	0,34	0,80	0,10	0,10	1,00	1,00	0	0	0	1	1	1	0	0
instance14	0,43	0,82	0,11	0,11	1,00	1,00	0	0	0	1	1	1	0	0
instance15	0,00	0,00	0,12	0,12	1,00	1,00	0	0	0	0,93	0,93	1	0	0
instance16	0,43	0,58	0,13	0,13	1,00	1,00	0	0	0	0,94	0,94	1	0	0
instance17	0,43	0,57	0,13	0,13	1,00	1,00	0	0,06	0,06	0,88	0,88	1	0	0
instance18	0,43	0,55	0,14	0,14	1,00	1,00	0	0,06	0,06	0,89	0,89	1	0	0
instance19	0,43	0,55	0,15	0,15	1,00	1,00	0	0,05	0,05	0,89	0,89	1	0	0
instance20	0,43	0,55	0,16	0,16	1,00	1,00	0	0,05	0,05	0,9	0,9	1	0	0
instance21	0,43	0,70	0,16	0,16	1,00	1,00	0	0,05	0,05	0,9	0,9	1	0	0
instance22	0,43	0,72	0,17	0,17	1,00	1,00	0	0,05	0,05	0,91	0,91	1	0	0
instance23	0,43	0,55	0,18	0,18	1,00	1,00	0	0,04	0,04	0,91	0,91	1	0	0
instance24	0,43	0,54	0,19	0,19	1,00	1,00	0	0,04	0,04	0,92	0,92	1	0	0
instance25	0,43	0,55	0,20	0,20	1,00	1,00	0	0,04	0,04	0,92	0,92	1	0	0
instance26	0,43	0,53	0,20	0,20	1,00	1,00	0	0,04	0,04	0,92	0,92	1	0	0

Figure VI.1 **:** *L'affichage de la base de transactions KDD'99*

VI.2 Résultats obtenus

L'application a été réalisée sur une machine de type Pentium 4 de 512Mo de RAM, et 3.4Gh de microprocesseur. Le langage de programmation retenu est le langage C^{++} (environnement Builder 5.0).

Notre approche proposée se compose de quatre étapes principales :

- La première étape, consiste à trouver le nombre de partition pour chaque attribut quantitatif de la base de transaction, pour cet effet nous avons proposé trois méthodes, deux méthodes qui se basent sur l'utilisation des indices de validités, et une méthode de fouille de données proposée par [21]. Dans notre application nous donnons à l'utilisateur le choix de choisir le modèle de partitionnement. Ainsi nous affichons le nombre de partitions trouvées pour chaque attribut quantitatif (cf. figure VI.2).

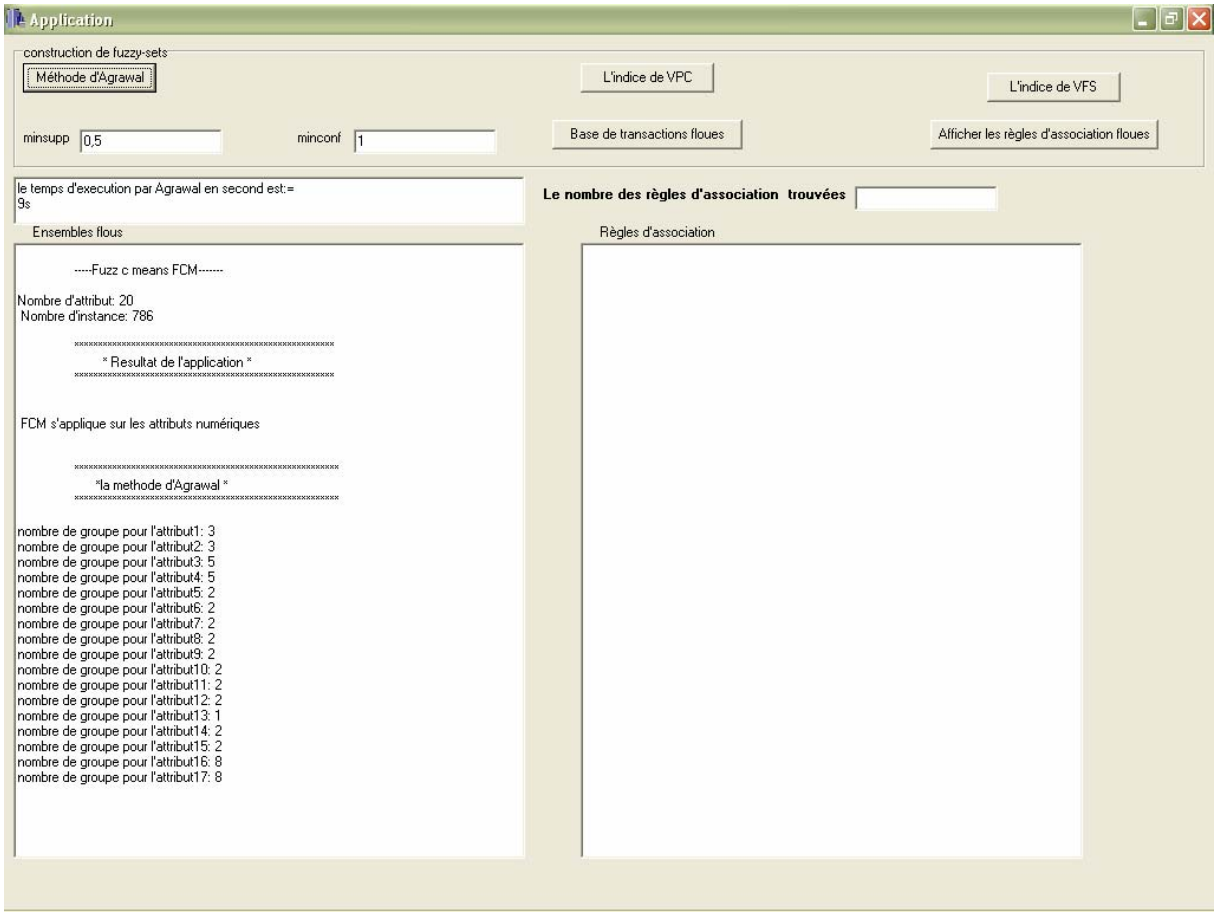


Figure VI.2 **:** *L'affichage d'une fenêtre de l'application*

- La deuxième étape, consiste à construire les différentes partitions floues (ensembles flous). Et parce que certaines valeurs des attributs sont de l'ordre de 10^{2} à 10^{3} , une étape de normalisation à été nécessaire pour ne pas avoir un problème de dépassement de capacité (over-flow).

La normalisation des valeurs d'attributs est réalisée par la fonction suivante :

$$x = \frac{x^{'} - x^{'}_{\min}}{x^{'}_{\max} - x^{'}_{\min}} \qquad \text{(VI.1)}$$

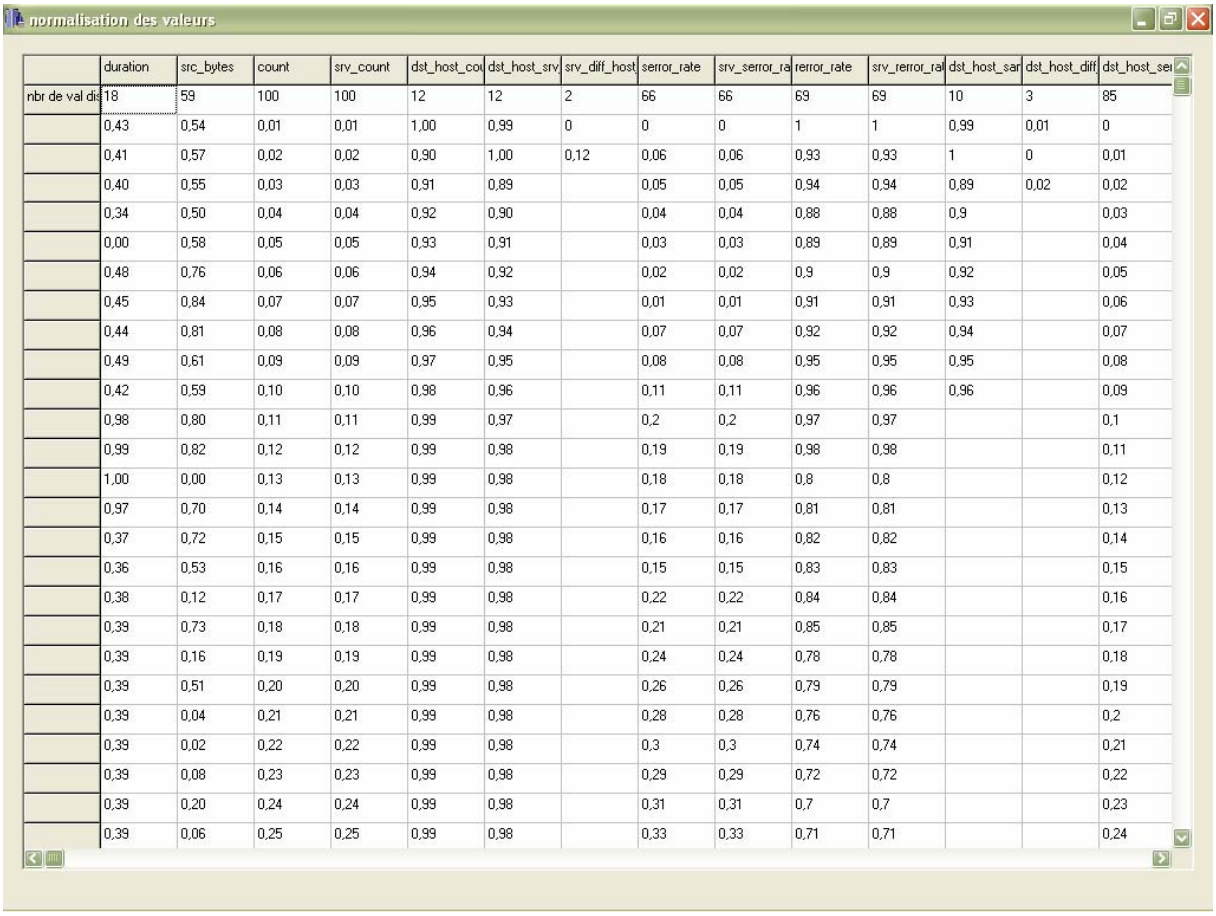
$x^{'}$: la valeur originale de l'attribut

$x^{'}_{\min}$: la valeur minimum de toutes les valeurs possibles de l'attribut

$x^{'}_{mx}$: la valeur maximum de toutes les valeurs possibles de l'attribut

x : la valeur normalisée $\in [0,1]$.

Pour les valeurs des attributs qui appartiennent déjà à l'intervalle $[0,1]$, on n'applique pas la fonction de normalisation (cf. figure VI.3).



	duration	src_bytes	count	srv_count	dst_host_cou	dst_host_srv	srv_diff_host	serror_rate	srv_serror_ra	rerror_rate	srv_rerror_rat	dst_host_sar	dst_host_diff	dst_host_ser
nbr de val di	18	59	100	100	12	12	2	66	66	69	69	10	3	85
	0,43	0,54	0,01	0,01	1,00	0,99	0	0	0	1	1	0,99	0,01	0
	0,41	0,57	0,02	0,02	0,90	1,00	0,12	0,06	0,06	0,93	0,93	1	0	0,01
	0,40	0,55	0,03	0,03	0,91	0,89		0,05	0,05	0,94	0,94	0,89	0,02	0,02
	0,34	0,50	0,04	0,04	0,92	0,90		0,04	0,04	0,88	0,88	0,9		0,03
	0,00	0,58	0,05	0,05	0,93	0,91		0,03	0,03	0,89	0,89	0,91		0,04
	0,48	0,76	0,06	0,06	0,94	0,92		0,02	0,02	0,9	0,9	0,92		0,05
	0,45	0,84	0,07	0,07	0,95	0,93		0,01	0,01	0,91	0,91	0,93		0,06
	0,44	0,81	0,08	0,08	0,96	0,94		0,07	0,07	0,92	0,92	0,94		0,07
	0,49	0,61	0,09	0,09	0,97	0,95		0,08	0,08	0,95	0,95	0,95		0,08
	0,42	0,59	0,10	0,10	0,98	0,96		0,11	0,11	0,96	0,96	0,96		0,09
	0,98	0,80	0,11	0,11	0,99	0,97		0,2	0,2	0,97	0,97			0,1
	0,99	0,82	0,12	0,12	0,99	0,98		0,19	0,19	0,98	0,98			0,11
	1,00	0,00	0,13	0,13	0,99	0,98		0,18	0,18	0,8	0,8			0,12
	0,97	0,70	0,14	0,14	0,99	0,98		0,17	0,17	0,81	0,81			0,13
	0,37	0,72	0,15	0,15	0,99	0,98		0,16	0,16	0,82	0,82			0,14
	0,36	0,53	0,16	0,16	0,99	0,98		0,15	0,15	0,83	0,83			0,15
	0,38	0,12	0,17	0,17	0,99	0,98		0,22	0,22	0,84	0,84			0,16
	0,39	0,73	0,18	0,18	0,99	0,98		0,21	0,21	0,85	0,85			0,17
	0,39	0,16	0,19	0,19	0,99	0,98		0,24	0,24	0,78	0,78			0,18
	0,39	0,51	0,20	0,20	0,99	0,98		0,26	0,26	0,79	0,79			0,19
	0,39	0,04	0,21	0,21	0,99	0,98		0,28	0,28	0,76	0,76			0,2
	0,39	0,02	0,22	0,22	0,99	0,98		0,3	0,3	0,74	0,74			0,21
	0,39	0,08	0,23	0,23	0,99	0,98		0,29	0,29	0,72	0,72			0,22
	0,39	0,20	0,24	0,24	0,99	0,98		0,31	0,31	0,7	0,7			0,23
	0,39	0,06	0,25	0,25	0,99	0,98		0,33	0,33	0,71	0,71			0,24

***FigureVI.3* :** *L'affichage de nombre de valeurs distingues pour chaque attribut, et la normalisation*

L'algorithme de CMF est appliqué sur les attributs quantitatifs, nous ne prenons pas en considération les attributs binaires. Car le but de CMF et de construire les ensembles flous c'est-à-dire trouver les degrés d'appartenance de chaque item d'un attribut dans sa partition correspondante.

- La troisième étape consiste à trouver les règles d'association floues. Avant l'extraction des règles d'association floues, une étape de transformation de la base de transactions initiale à une base de transactions floues est nécessaire. Pour faire la transformation floue, nous attribuons a chaque valeur de chaque attribut son degré d'appartenance aux différents ensemble flous trouvés correspondant à la partition de l'attribut. Pour notre application les colonnes

présentent les transactions et les lignes présentent les différentes partitions floues trouvées par l'application de CMF.

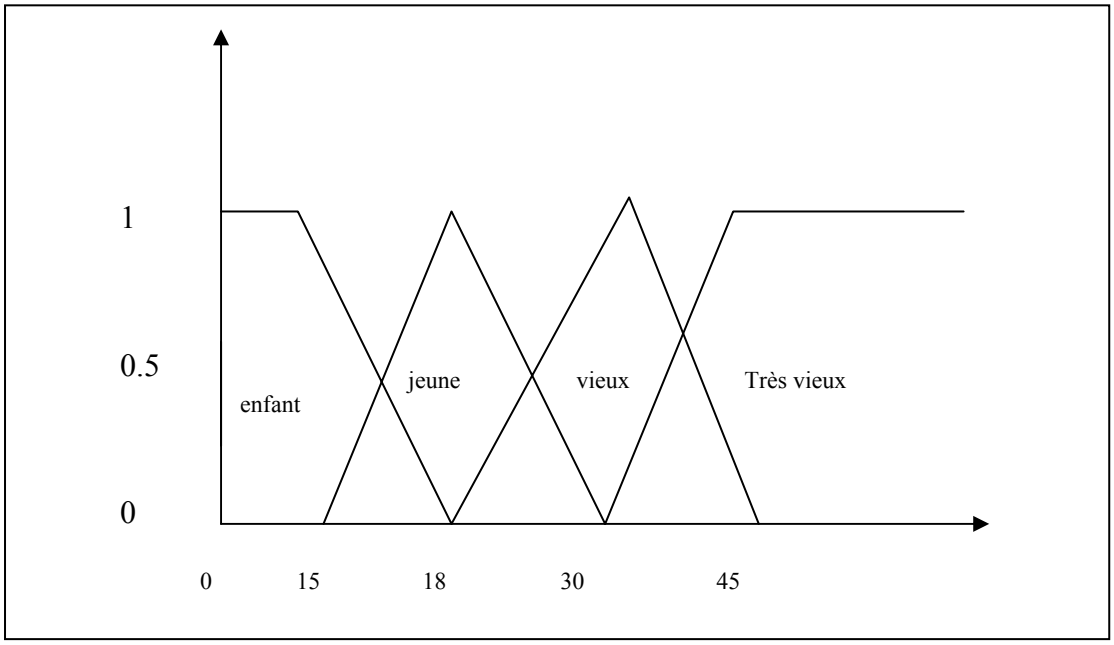
Exemple :

Afin d'illustrer la manière de transformer la base de transaction initiale en base de transaction floue nous présentons l'exemple suivant :

Nous considérons deux attribut : *Age*, *sexe*, et nous considérons la base de transaction initiale suivante :

	age	sexe
Transaction1	15	femelle
Transaction2	20	male
Transaction3	30	femelle
Transaction4	50	femelle
Transaction5	45	male
Transaction6	60	femelle
Transaction7	75	male
Transaction8	35	femelle

Table VI.4*: *exemple de transaction

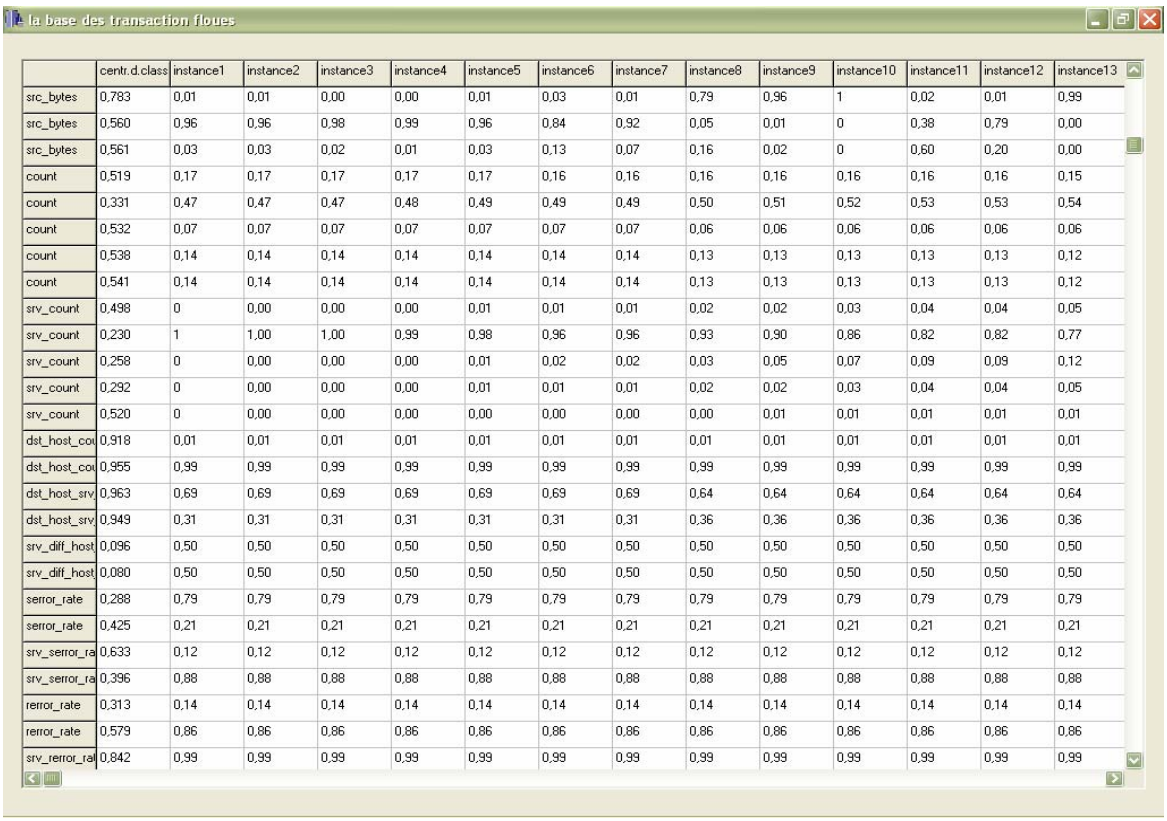


FigureVI.4* : *Exemple de partition floue pour Age

Pour la construction de la base de transaction floue nous associons à chaque valeur de l'attribut *Age* de la base initiale sa valeur floue correspond au différent partition floue de l'attribut *Age*. L'intersection d'une ligne et d'une colonne représente le degré d'appartenance de la valeur quantitative dans la partition floue.

	Trans1	Trans2	Trans3	Trans4	Trans5	Trans6	Trans7	Trans8
Age.enfant	0.7	0	0	0	0	0	0	0
Age.jeune	0.3	0.5	0.4	0	0	0	0	0
Age.vieux	0	0.4	0.6	0	0.3	0	0	0.6
Age.trés vieux	0	0	0	1	0.6	1	1	0
Sexe.femmelle	1	0	1	1	0	1	0	1
Sexe.male	0	1	0	0	1	0	1	0

Table VI.5**:** *exemple de base de transaction floue*



	centr.d.class	instance1	instance2	instance3	instance4	instance5	instance6	instance7	instance8	instance9	instance10	instance11	instance12	instance13
src_bytes	0,783	0,01	0,01	0,00	0,00	0,01	0,03	0,01	0,79	0,96	1	0,02	0,01	0,99
src_bytes	0,560	0,96	0,96	0,98	0,99	0,96	0,84	0,92	0,05	0,01	0	0,38	0,79	0,00
src_bytes	0,561	0,03	0,03	0,02	0,01	0,03	0,13	0,07	0,16	0,02	0	0,60	0,20	0,00
count	0,519	0,17	0,17	0,17	0,17	0,17	0,16	0,16	0,16	0,16	0,16	0,16	0,16	0,15
count	0,331	0,47	0,47	0,47	0,48	0,49	0,49	0,49	0,50	0,51	0,52	0,53	0,53	0,54
count	0,532	0,07	0,07	0,07	0,07	0,07	0,07	0,07	0,06	0,06	0,06	0,06	0,06	0,06
count	0,538	0,14	0,14	0,14	0,14	0,14	0,14	0,14	0,13	0,13	0,13	0,13	0,13	0,12
count	0,541	0,14	0,14	0,14	0,14	0,14	0,14	0,14	0,13	0,13	0,13	0,13	0,13	0,12
srv_count	0,498	0	0,00	0,00	0,00	0,01	0,01	0,01	0,02	0,02	0,03	0,04	0,04	0,05
srv_count	0,230	1	1,00	1,00	0,99	0,98	0,96	0,96	0,93	0,90	0,86	0,82	0,82	0,77
srv_count	0,258	0	0,00	0,00	0,00	0,01	0,02	0,02	0,03	0,05	0,07	0,09	0,09	0,12
srv_count	0,292	0	0,00	0,00	0,00	0,01	0,01	0,01	0,02	0,02	0,03	0,04	0,04	0,05
srv_count	0,520	0	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,01	0,01	0,01	0,01	0,01
dst_host_cou	0,918	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01
dst_host_cou	0,955	0,99	0,99	0,99	0,99	0,99	0,99	0,99	0,99	0,99	0,99	0,99	0,99	0,99
dst_host_srv	0,963	0,69	0,69	0,69	0,69	0,69	0,69	0,69	0,64	0,64	0,64	0,64	0,64	0,64
dst_host_srv	0,949	0,31	0,31	0,31	0,31	0,31	0,31	0,31	0,36	0,36	0,36	0,36	0,36	0,36
srv_diff_host	0,096	0,50	0,50	0,50	0,50	0,50	0,50	0,50	0,50	0,50	0,50	0,50	0,50	0,50
srv_diff_host	0,080	0,50	0,50	0,50	0,50	0,50	0,50	0,50	0,50	0,50	0,50	0,50	0,50	0,50
serror_rate	0,288	0,79	0,79	0,79	0,79	0,79	0,79	0,79	0,79	0,79	0,79	0,79	0,79	0,79
serror_rate	0,425	0,21	0,21	0,21	0,21	0,21	0,21	0,21	0,21	0,21	0,21	0,21	0,21	0,21
srv_serror_ra	0,633	0,12	0,12	0,12	0,12	0,12	0,12	0,12	0,12	0,12	0,12	0,12	0,12	0,12
srv_serror_ra	0,396	0,88	0,88	0,88	0,88	0,88	0,88	0,88	0,88	0,88	0,88	0,88	0,88	0,88
rerror_rate	0,313	0,14	0,14	0,14	0,14	0,14	0,14	0,14	0,14	0,14	0,14	0,14	0,14	0,14
rerror_rate	0,579	0,86	0,86	0,86	0,86	0,86	0,86	0,86	0,86	0,86	0,86	0,86	0,86	0,86
srv_rerror_rat	0,842	0,99	0,99	0,99	0,99	0,99	0,99	0,99	0,99	0,99	0,99	0,99	0,99	0,99

FigureVI.5 **:** *La base de transactions floues*

Remarque

Après la construction des ensembles flous, normalement il fallait attribuer à chaque partition trouvée un label (étiquette linguistique) qui décrit la partition. Dans le cas de notre étude, nous avons trouvé des difficultés pour le faire, car quand le nombre de partition dépasse le nombre de trois, il est délicat d'attribuer à chaque partition un label. Pour remédier à ce problème nous avons identifié chaque partition trouvée par son centre de classe.

Pour le cas de notre application, nous avons choisi d'afficher le centre de classe de chaque partition au niveau de la première colonne, (cf. FigureVI.5).

- La quatrième étape, consiste à évaluer les règles d'association floues générées par chaque méthode. Nous comparons les règles ainsi générées par rapport à un modèle de validation obtenu par apprentissage. Cette étape n'a pas été implémentée.

Dans ce qui suit nous présentons les différents résultas de l'application pour chaque base de transaction.

Remarque

Afin de faciliter l'affichage de résultat nous attribuons à chaque attribut quantitatif un entier positif.

Nom de l'attribut	Entier positif associé
duration	1
src_bytes	2
count	3
srv_count	4
dst_host_count	5
dst_host_srv_count	6
srv_diff_host_rate	7
serror_rate	8
srv_serror_rate	9
rerror_rate	10
srv_rerror_rate	11
dst_host_same_srv_rate	12
dst_host_diff_srv_rate	13
dst_host_serror_rate	14

dst_host_srv_serror_rate	15
dst_host_rerror_rate	16
dst_host_srv_rerror_rate	17
logged_in	18
flag	19
attack_name	20

Table VI.6*: *table de correspondance de la base KDD'99

Nom de l'attribut	Entier positif associé
Age	1
fnlwgt	2
education-num	3
capital-gain	4
capital-loss	5
hours-per-week	6
Nature de travail	7
sex	8
Type d'éducation	9
marital-status	10
native-country	11
class	12

Table VI.7*: *table de correspondance de la base Adult

VI.2.1 Le nombre de partitions trouvées par chaque méthode

Le nombre de partition floues (ensembles flous, groupes flous) trouvées pour chaque attribut quantitatif, varie d'une méthode à l'autre. Dans ce qui suit nous présentons les différents nombres de partitions trouvées par chaque méthode et cela pour les deux bases de données. Nous présentons aussi le nombre de valeurs distinctes pour chaque attribut.

- **Base *KDD'99***

attribut	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Nombre de val distinct	110	80	128	128	27	30	2	66	66	69	69	10	3	85	85	74	75
Agrawal	3	3	5	5	2	2	2	2	2	2	2	2	1	2	2	8	8
V_{PC}	3	2	2	2	3	2	3	3	3	3	4	2	3	3	3	2	2
V_{FS}	3	3	4	5	2	3	3	3	6	3	4	3	4	3	4	5	2

Table VI.8*: *nombre de partitions pour les attributs de la base KDD'99

- **Base *Adult***

attribut	1	2	3	4	5	6
Nombre de val distinct	67	1961	16	42	34	66
Agrawal	7	4	3	2	2	2
V_{PC}	3	2	2	3	2	2
V_{FS}	4	6	3	2	3	4

Table VI.9*: *nombre de partitions pour les attributs de la base adult

Afin d'évaluer la pertinence des trois méthodes proposées au niveau de nombre de partitions floues trouvées nous proposons de générer les règles d'association floues. Dans ce qui suit nous présentons les règles d'association floues générées pour chaque base de transaction.

VI.2.2 Génération des règles d'association floues

Après l'application des trois méthodes proposées pour trouver le nombre de partition, nous aurons trois modèles différents de règles d'association. La recherche des règles d'association telle qu'elle est connue consiste tout d'abord à trouver l'ensemble des itemset fréquents.

Pour l'ensemble de 1-itemset fréquent, aucune règle ne peut être générée, la recherche commence donc à partir de l'ensemble de 2-itemset fréquent. Pour la génération des règles d'association floues on n'aura pas de problème de redondance car nous nous sommes restreint aux règles d'association qui ont un seul attribut comme conclusion, il s'agit de l'attribut *attack_name* pour la base de données *KDD'99* qui indique le nom de l'attaque, et *class* pour la base de données *adult* qui indique un salaire inférieure ou égal à 50,000 Dollars.

Pour la valeur de support minimum et la valeur de la confiance, l'utilisateur a le choix de les fixer. Pour le cas de notre étude, pour générer les règles d'association floues nous avons choisi la valeur de **support minimum=0.5**, et **la confiance minimum=1**.

VI.2.2.1 Règles d'association générées à partir de *KDD'99*

Nous avons retenu la prédiction de l'attaque *Appache2* uniquement. Vu que les autres attaque ne présentaient pas une distribution uniforme quant aux valeurs des differents attributs.

Pour la découverte des règles d'association floues nous utilisons une base de 20 attributs dont 17 sont quantitatifs et 3 sont qualitatifs, et 786 transactions. Le modèle général pour l'affichage des attributs des règles d'association est le suivant :

(Le nom de l'item : le centre de classe)

- **La méthode d'Agrawal**.

Par la méthode d'Agrawal, notre système génère 31 règles (cf. figure VI.6).

Exemple de règle d'association:

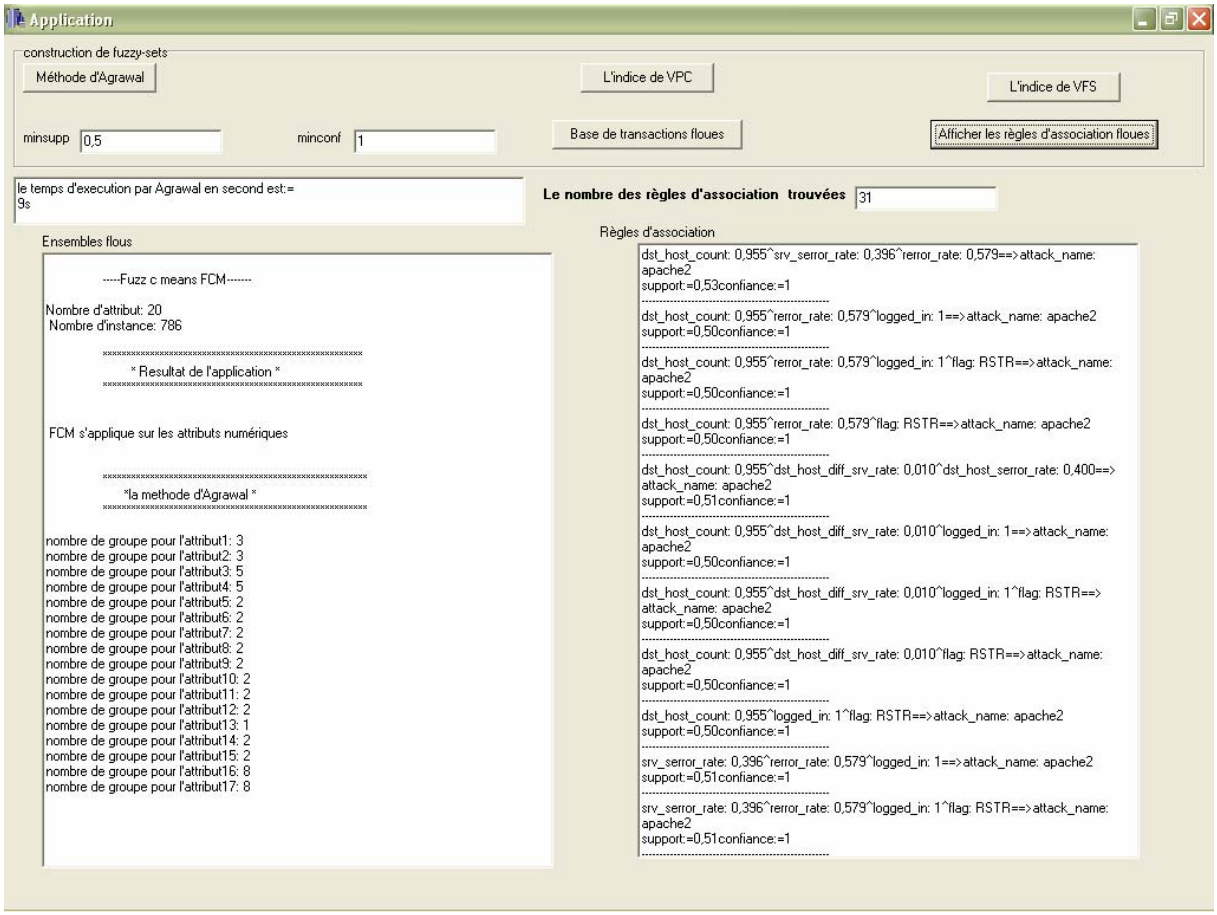
1. [srv_rerror_rate: 0,842] ∧ [dst_host_diff_srv_rate: 0,010] ∧ [logged_in: 1] ∧ [flag: RSTR] ==>apache2[2]
Avec **support=0,55 et confiance=1.**

2. [rerror_rate: 0,579] ∧ [srv_rerror_rate: 0,842] ∧ [flag:RSTR]==> apache2
Avec **support=0,51 et confiance=1.**

3. [dst_host_count: 0,955] ∧ [dst_host_diff_srv_rate: 0,010] ∧ [logged_in: 1] ∧ [flag: RSTR]==>apache2
Avec **support=0,50 et confiance=1.**

4. [srv_serror_rate: 0,396] ∧ [logged_in: 1] ∧ [flag:RSTR]==>apache2
Avec **support=0,51 et confiance=1.**

[2] Appache2 : signifie qu'il s'agit de l'attaque de type Appache 2



***FigureVI.6* :** *Les règles d'association floues générées par la méthode d'Agrawal*

- **La méthode d'indice de validité V_{PC}**

Par l'indice de validité V_{PC} , notre système génère 62 règles d'association floues.

Exemple:

1. [duration: 0,39]∧[dst_host_count: 0,94]∧[srv_diff_host_rate: 0,06]==> apache2

Avec **support=0,56 et confiance=1.**

2. [dst_host_count:0,94]∧[srv_diff_host_rate:0,06]∧[dst_host_serror_rate:0,42]∧[dst_host_srv_serror_rate: 0,43] ==> apache2

Avec **support=0,50 et confiance=1.**

3. [srv_diff_host_rate: 0,06]∧[srv_rerror_rate: 0,63]∧[dst_host_serror_rate: 0,42]∧[dst_host_srv_serror_rate: 0,43]∧[logged_in: 1]==> apache2

Avec **support=0,51 et confiance=1.**

4. [srv_rerror_rate: 0,63]∧[dst_host_serror_rate: 0,42]∧[dst_host_srv_serror_rate: 0,43]∧[logged_in: 1]∧[flag: RSTR]==> apache2

Avec **support=0,51 et confiance=1.**

- **La méthode d'indice de validité V_{FS}**

Par l'indice de validité V_{FS} , notre système génère six règles d'association floues.

Exemple:

1. [duration:0,39]∧[dst_host_count:0,98]∧ [srv_diff_host_rate:0]==> apache2

Avec **support:=0,60 et confiance=1.**

2. [dst_host_count:0,98]∧[srv_diff_host_rate:0,00]∧[logged_in:1] ==> apache2

Avec **support=0,62 et confiance=1.**

3. [dst_host_count:0,98] ∧ [srv_diff_host_rate:0,00] ∧ [logged_in:1] ∧ [flag: RSTR] ==> apache2

Avec **support=0,62 et confiance=1.**

4. [dst_host_count:0,98] ∧ [srv_diff_host_rate:0,00] ∧ [flag:RSTR] ==> apache2

Avec **support=0,62 et confiance=1.**

VI.2.2.2 Règles d'association floues générées à partir de *Adult*

Pour la découverte des règles d'association floues nous utilisons une base de 12 attributs dont 6 sont quantitatifs et 6 sont qualitatifs, et 2000 transactions. Le modèle général pour l'affichage des attributs des règles d'association est le suivant :
(Le nom de l'item : le centre de classe).

- **La méthode d'agrawal**

Par la méthode d'agrawal, notre système génère deux règles d'association floues.

Exemple :

1. [education-num:0,61] ∧ [capital-gain:0,07] ∧ [native-country:United-States] ==> <=50K[3]

Avec **support=0,54 et confiance=1.**

2. [capital-gain:0,07] ∧ [hours-per-week: 0,38] ∧ [native-country:United-States]==> <=50K

Avec **support=0,54 et confiance=1.**

- **La méthode d'indice de validité V_{PC}**

Par l'indice de validité V_{PC} , notre système génère 3 règles d'association floues (cf. figure VI.7).

Exemple :

[3] <=50K signifie que le salaire est <=50.000$

1. **[**fnlwgt:0,31**]**∧**[**education-num:0,62**]**∧**[**native-country: United-States**]** ==> <=50K

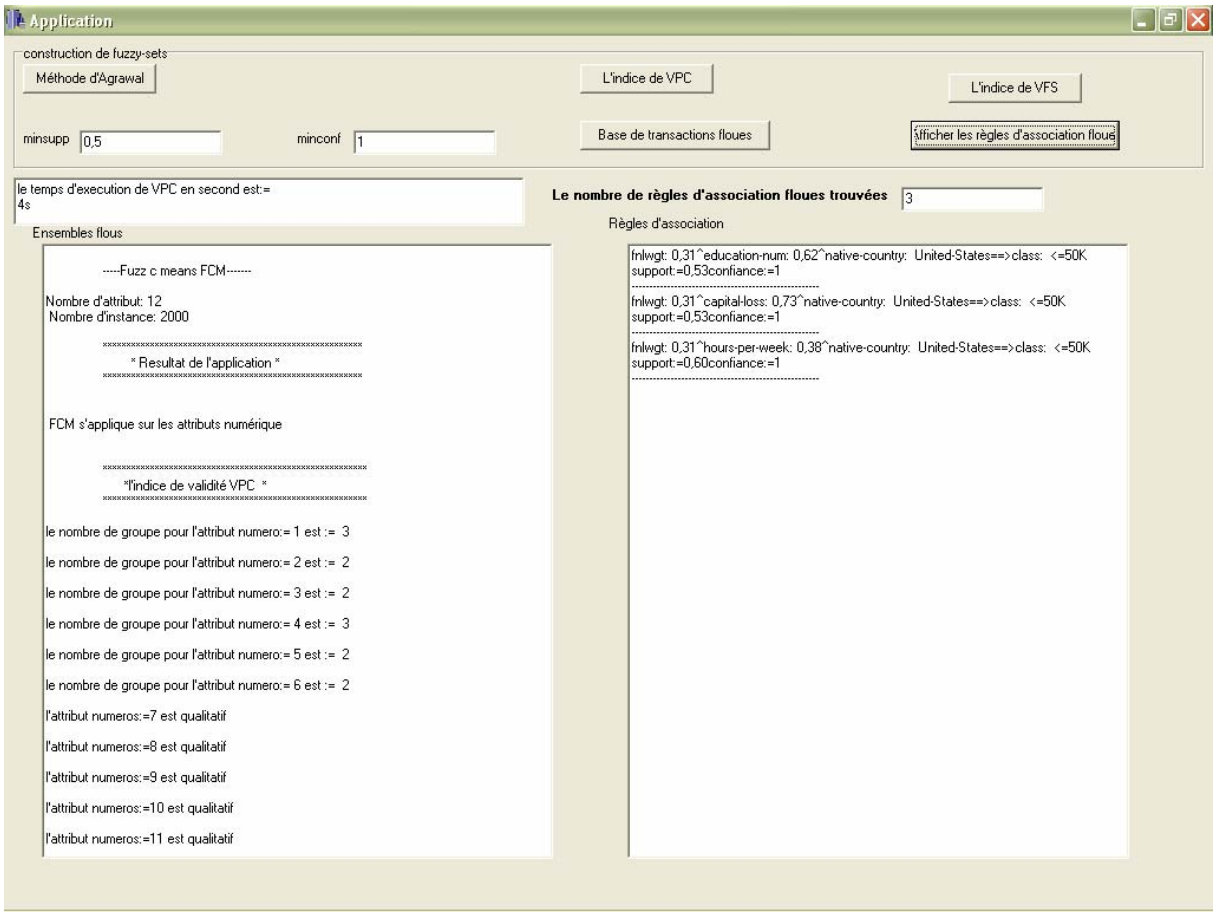
Avec **support=0,53 et confiance=1.**

2. **[**fnlwgt: 0,31**]**∧**[**capital-loss: 0,73**]**∧**[**native-country: United-States**]**==> <=50K

Avec **support=0,53 et confiance=1**

3. **[**fnlwgt: 0,31**]**∧**[**hours-per-week: 0,38**]**∧**[**native-country: United-States**]** ==> <=50K

Avec **support=0,60 et confiance=1.**



FigureVI.7*: *Les règles d'association floues générées par la méthode V_{PC}

- **La méthode d'indice de validité V_{FS}**

Par l'indice de validité V_{FS}, notre système génère trois règles d'association floues :

Exemple:

1. **[**capital-gain: 0,07**]**∧**[**capital-loss: 0,69**]**∧**[**native-country: United-States**]**==> <=50K

Avec **support=0,67 et confiance=1.**

2. **[**capital-gain: 0,07**]**∧**[**hours-per-week: 0,32**]**∧**[**native-country: United-States**]**==> <=50K

Avec **support=0,55 et confiance=1.**

3. **[**capital-loss: 0,69**]**∧**[**hours-per-week: 0,32**]**∧**[**native-country: United-States**]**⇒ <=50K

Avec **support=0,55 et confiance=1.**

VI.3 Comparaison des règles d'association floues

Chaque méthode proposée pour trouver le nombre de partitions floues, fournit à l'étape de construction de ces ensembles flous, des ensembles flous différents, ce qui influence sur les règles d'association floues générées. Afin de pouvoir comparer les trois modèles de règles d'association, nous proposons de comparer chaque modèle ainsi généré par rapport à un modèle de validation. Nous décrivons ci après le principe de construction de ce modèle de validation et présentons aussi notre méthode de comparaison.

VI.3.1 Construction du modèle de validation

Pour retenir la valeur de l'attribut qui participe au modèle de validation, nous considérons que la valeur moyenne ou la valeur modale est la valeur caractéristique de domaine d'attribut. Ces valeurs seront calculées comme suit :

- Les attributs quantitatifs

Pour les attributs quantitatifs, nous calculons la valeur moyenne de chaque attribut par la formule suivante :

$$\text{moyenne} = \frac{\sum_{i=1}^{n} x_i}{n} \qquad \text{(VI.2)}$$

n : Le nombre de valeur distincte de chaque attribut

x_i : Valeur *i* de l'attribu

- Les attributs qualitatifs

Pour les attributs qualitatifs, la valeur modale correspond à la valeur de la variable qui apparaît le plus souvent dans la distribution.

Par le calcul des valeurs moyennes et des valeurs modales de chaque base nous aurons les modèles de validation de chaque base.

a. Construction du modèle de validation pour la base KDD'99

attribut	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Moyenne	789.03	39118.90	42.96	42.98	254.55	250.38	0	0.32	0.32	0.68	0.68	0.98	0.01	0.13	0.13	0.4	0.4

Table VI.10*: *la valeur moyenne de chaque attribut quantitatif (base KDD'99)

attribut	18	19	20
Valeur modale	1	RSTR	Appach2

Table VI.11*: *la valeur modale de chaque attribut qualitatif (base KDD'99)

A partir de la table VI.10 et de la table VI.11, nous pouvons déduire le modèle de validation M suivant :

M :([duration=789.03]∧[src_bytes=39118.90]∧[count=42.96]∧
[srv_count=42.98]∧[dst_host_count=254.55]∧[dst_host_srv_count
=250.38] ∧[srv_diff_host_rate=0] ∧[serror_rate=0.32]
∧[srv_serror_rate=0.32]∧[rerror_rate=0.68]
∧[srv_rerror_rate=0.68]∧[dst_host_same_srv_rate=0.98]
∧[dst_host_diff_srv_rate=0.01]∧[dst_host_serror_rate=0.13]∧[d
st_host_srv_serror_rate=0.13]∧[dst_host_rerror_rate=0.4]∧
[dst_host_srv_rerror_rate=0.4]∧[logged_in=1]∧[flag=RSTR]⇒
=Appach2).

b. Construction du modèle de validation pour la base Adult

attribut	1	2	3	4	5	6
Moyenne	37,061	190308,34	9,65	145,19	61,75	38,84

Table VI.12**:** *la valeur moyenne de chaque attribut quantitatif (base adult*

attribut	7	8	9	10	11	12
Valeur modale	Private	male	HS-grad	Married-civ-epouse	United-States	<=50

Table VI.13 : *la valeur modale de chaque attribut qualitatif (base adult)*

A partir de la table VI.12 et de la table VI.13, nous pouvons déduire le modèle de validation M suivant :

M : ([*age*=37.061] ∧ [*fnlwgt*=190308,34] ∧ [*education-num*=9.65] ∧ [*capital-gain*=145.19] ∧ [*capital-loss*=61.75] ∧ [*hours-per-week*=38.84] ∧ [*workclass= private*] ∧ [*sex=male*] ∧ [*education= HS_grad*] ∧ [*marital-status= Married-civ-epouse*] ∧ [native-country=United-States] ⇒ <=50).

VI.3.2 Principe de comparaison par rapport au modèle de validation

La comparaison d'un modèle de règle d'association généré par rapport au modèle de validation repose sur le principe suivant :

Soit une règle générée $\varphi : A_1, A_2, ..., A_m \Rightarrow A_{m+1}$.

Soit le modèle de validation $M : V_1, V_2, ..., V_m \Rightarrow V_{m+1}$.

Le principe de comparaison consiste à comparer chaque attribut de modèle de validation à son adéquat dans la règle d'association générée, et cela ce fait par le choix d'une mesure de ressemblance et d'un opérateur d'agrégation. C'est l'attribut n'est pas présent dans la règle générée alors nous ne le considère pas dans la comparaison.

a. ***Mesure de ressemblance***

La mesure de ressemblance adoptée est la suivante :

$$S(V_i, A_i) = \frac{M_3(V_i \cap A_i)}{M_3(V_i \cup A_i)} \quad \text{(VI.3)}$$

avec M_3 la mesure floue (voir annexe1).

b. ***Mesure d'agrégation***

Pour comparer une règle d'association par rapport au modèle de validation, il faut comparer tout l'ensemble d'attribut qui constitué le modèle de validation ainsi la règle d'association. A cet effet l'utilisation d'un opérateur d'agrégation est nécessaire.

$$S(M, \varphi) = \text{ag}_{i=1,..,m+1}(S(V_i, A_i)) \quad \text{(VI.4)}$$

où $m+1$: Le nombre d'attribut

Pour le cas de notre étude, nous choisissons de considérer l'opérateur *min* comme opérateur d'agrégation.

VI.3.3 Evaluation des règle générées à partir de la base *KDD'99*

Pour comparer les règles d'association floues trouvées par chaque méthode nous choisissons de prendre en considération la règle d'association floue qui a le plus grand nombre d'attribut et le plus grand support. Et pour calculer le degré de ressemblance de la règle d'association générée par rapport au modèle de validation, nous calculons le degré de ressemblance de chaque attribut de la règle généré à l'attribut qui lui correspond dans le modèle de validation.

- ***Le modèle de validation***

M :([duration=789.03]∧[src_bytes=39118.90]∧[count=42.96]∧
[srv_count=42.98]∧[dst_host_count=254.55]∧[dst_host_srv_count
=250.38] ∧[srv_diff_host_rate=0] ∧[serror_rate=0.32]
∧[srv_serror_rate=0.32]∧[rerror_rate=0.68]
∧[srv_rerror_rate=0.68]∧[dst_host_same_srv_rate=0.98]
∧[dst_host_diff_srv_rate=0.01]∧[dst_host_serror_rate=0.13]∧

[dst_host_srv_serror_rate=0.13] ∧ [dst_host_rerror_rate=0.4] ∧ [dst_host_srv_rerror_rate=0.4] ∧ [logged_in=1] ∧ [flag=RSTR] ⇒ =Appach2).

Dans la suite nous présentons pour chaque méthode les différents résultats de comparaison, nous présentons un exemple de comparaison, les détails de notre méthode sont donnés dans l'annexe 2.

- **La méthode d'agrawal**

$\varphi \equiv$ [srv_rerror_rate:0,842] ∧ [dst_host_diff_srv_rate:0,010] ∧ [logged_in: 1] ∧ [flag: RSTR] ==> apache2.

Exemple : Soit l'attribut : *srv_rerror_rate*

$$S(srv\_rerror\_rate = 0.68 \wedge srv\_rerror\_rate = 0.842) = ...$$
$$..... = \frac{M_3((srv\_rerror\_rate = 0.68 \wedge srv\_rerror\_rate = 0.842))}{M_3((srv\_rerror\_rate = 0.68 \wedge srv\_rerror\_rate = 0.842))}$$

d'où

$$S(srv\_rerror\_rate = 0.68 \wedge srv\_rerror\_rate = 0.842) = \frac{0.46}{1} = 0.46$$

Attribut	M	φ	$\min(S(\varphi, M))$
duration	789.03	/	**0.46**
src_bytes	39118.90	/	
count	42.96	/	
srv_count	42.98	/	
dst_host_count	254.55	/	
dst_host_srv_count	250.38	/	
srv_diff_host_rate	0	/	
serror_rate	0.32	/	
srv_serror_rate	0.32	/	
rerror_rate	0.68	/	
srv_rerror_rate	0.68	0.842	
dst_host_same_srv_rate	0.98	/	
dst_host_diff_srv_rate	0.01	0.010	
dst_host_serror_rate	0.13	/	
dst_host_srv_serror_rate	0.13	/	
dst_host_rerror_rate	0.4	/	
dst_host_srv_rerror_rate	0.4	/	
logged_in	1	1	
flag	RSTR	RSTR	
attack_name	Appach2	Appach2	

Table VI.14 : *table de degré de comparaison pour la base KDD'99 (par la méthode d'Agrawal)*

- **La méthode V_{PC}**

$\varphi \equiv$ [srv_diff_host_rate:0,06] $\wedge$ [srv_rerror_rate:0,63] $\wedge$ [dst_host_serror_rate:0,42] $\wedge$ [dst_host_srv_serror_rate:0,43] $\wedge$ [logged_in:1] $\wedge$ [flag:RSTR] ==> apache2.

Exemple : Soit l'attribut : *dst_host_srv_serror_rate*

$$S(dst\_host\_srv\_serror\_rate = 0.13, dst\_host\_srv\_serror\_rate = 0.43) = ...$$

$$..... = \frac{M_3(dst\_host\_srv\_serror\_rate = 0.13 \wedge dst\_host\_srv\_serror\_rate = 0.43)}{M_3(dst\_host\_srv\_serror\_rate = 0.13 \wedge dst\_host\_srv\_serror\_rate = 0.43)}$$

d'où

$$S(dst\_host\_srv\_serror\_rate = 0.13, dst\_host\_srv\_serror\_rate = 0.43) = \frac{0.25}{1} = 0.25$$

Attribut	M	φ	$\min(S(\varphi, M))$
duration	789.03	/	
src_bytes	39118.90	/	
count	42.96	/	
srv_count	42.98	/	
dst_host_count	254.55	/	
dst_host_srv_count	250.38	/	
srv_diff_host_rate	0	0.06	
serror_rate	0.32	/	
srv_serror_rate	0.32	/	
rerror_rate	0.68	/	
srv_rerror_rate	0.68	0.63	
dst_host_same_srv_rate	0.98	/	**0.25**
dst_host_diff_srv_rate	0.01	/	
dst_host_serror_rate	0.13	0.42	
dst_host_srv_serror_rate	0.13	0.43	
dst_host_rerror_rate	0.4	/	
dst_host_srv_rerror_rate	0.4	/	
logged_in	1	1	
flag	RSTR	RSTR	
attack_name	Appach2	Appach2	

Table VI.15 *table de degré de comparaison pour la base KDD'99 (par la méthode de l'indice V_{PC})*

- **La méthode V_{FS}**

$\varphi \equiv$ [dst_host_count:0,98] $\wedge$ [srv_diff_host_rate:0,00] $\wedge$ [logged_in:1] $\wedge$ [flag: RSTR] ==> apache2.

Exemple : Soit l'attribut : *dst_host_count*

$$S(dst\_host\_count = 254.55, dst\_host\_count = 0.98) = ...$$
$$..... = \frac{M_3(dst\_host\_count = 254.55 \cap dst\_host\_count = 0.98)}{M_3(dst\_host\_count = 254.55 \cup dst\_host\_count = 0.98)}$$

d'où

$$S(dst\_host\_count = 254.55, dst\_host\_count = 0.98) = \frac{0.54}{1} = 0.54$$

Attribut	M	φ	$\min(S(\varphi, M))$
duration	789.03	/	
src_bytes	39118.90	/	
count	42.96	/	
srv_count	42.98	/	
dst_host_count	254.55	0.98	
dst_host_srv_count	250.38	/	
srv_diff_host_rate	0	0.00	
serror_rate	0.32	/	
srv_serror_rate	0.32	/	
rerror_rate	0.68	/	**0.50**
srv_rerror_rate	0.68	/	
dst_host_same_srv_rate	0.98	/	
dst_host_diff_srv_rate	0.01	/	
dst_host_serror_rate	0.13	/	
dst_host_srv_serror_rate	0.13	/	
dst_host_rerror_rate	0.4	/	
dst_host_srv_rerror_rate	0.4	/	
logged_in	1	1	
flag	RSTR	RSTR	
attack_name	Appach2	Appach2	

Table VI.16 : *table de degré de comparaison pour la base KDD'99 (par la méthode de l'indice V_{Fs})*

VI.3.4 Evaluation des règles générée à partir de la base *adult*

Pour comparer les règles d'association floues trouvées par chaque méthode nous choisissons de prendre en considération la règle d'association floue qui a le plus grand nombre d'attribut et le plus grand support. Et pour calculer le degré de ressemblance de la règle d'association générée par rapport au modèle de validation, nous calculons le degré de ressemblance de chaque attribut de la règle généré à l'attribut qui lui correspond dans le modèle de validation.

➢ ***Le modèle de validation***

M : ([*age*=37.061] $\wedge$ [*fnlwgt*=190308,34] $\wedge$ [*education-num*=9.65] $\wedge$ [*capital-gain*=145.19] $\wedge$ [*capital-loss*=61.75] $\wedge$ [*hours-per-week*=38.84] $\wedge$ [*workclass= private*] $\wedge$ [*sex=male*] $\wedge$ [*education= HS_grad*] $\wedge$ [*marital-status= Married-civ-epouse*] $\wedge$ [native-country=United-States] $\Rightarrow$ <=50).

Dans la suite nous présentons pour chaque méthode les différents résultats de comparaison, nous présentons un exemple de comparaison, les détails de notre méthode sont donnés dans l'annexe 2.

➢ **La méthode d'agrawal**

Par cette méthode, le système génère deux règles d'association floues qui ont le même nombre d'attribut, et le même support. Alors nous calculons le degré de ressemblance des deux règles d'association par rapport au modèle de validation.

La première règle d'association floue :

$\varphi \equiv$ [education-num:0,61] $\wedge$ [capital-gain:0,07] $\wedge$ [native-country:United-States] ==> < =50K

Exemple : Soit l'attribut : *education-num*

$$S(education-nim=9.65, education-num=0.61)= \ldots\ldots$$

$$\ldots\ldots = \frac{M_3(education-num=9.65 \cap education-num=0.61)}{M_3(education-num=9.65 \cup education-num=0.61)}$$

d'où

$$S(education-num=9.65, education-num=0.61)=\frac{0.49}{1}=0.49$$

Attribut	M	φ	$\min(S(\varphi, \mathrm{M}))$
Age	37.061	/	**0.45**
fnlwgt	190308.34	/	
education-num	9.65	0.61	
capital-gain	145.19	0.07	
capital-loss	61.75	/	
hours-per-week	38.84	/	
Workclass	private	/	
sex	Male	/	
Type d'éducation	HS_grade	/	
marital-status	Married-civ-epose	/	
native-country	United-States	United-States	
class	<=50k	<=50	

Table VI.17 : *table de degré de comparaison pour la base (par la méthode d'Agrawal)*

La deuxième règle d'association floue :

$\varphi \equiv$ `[capital-gain:0,07]` $\wedge$ `[hours-per-week: 0,38]` $\wedge$
`[native-country: United-States]==> <=50K`

Exemple : Soit l'attribut : *capital-gain*

$$S(capital-gain = 145.19, capital-gain = 0.07) = ...$$
$$..... = \frac{M_3(capital-gain = 145.19 \cap capital-gain = 0.07)}{M_3((capital-gain = 145.19 \cup capital-gain = 0.07))}$$

d'où

$$S(capital-gain = 145.19, capital-gain = 0.07) = \frac{0.45}{1} = 0.45$$

Attribut	M	φ	$\min(S(\varphi, \mathrm{M}))$
Age	37.061	/	**0.45**
fnlwgt	190308.34	/	
education-num	9.65	/	
capital-gain	145.19	0.07	
capital-loss	61.75	/	
hours-per-week	38.84	0.38	
Workclass	private	/	
sex	Male	/	
Type d'éducation	HS_grade	/	
marital-status	Married-civ-epose	/	
native-country	United-States	United-States	
class	<=50k	<=50	

Table VI.18 : *table de degré de comparaison pour la base Adult (par la méthode d'Agrawal)*

➢ **La méthode V_{PC}**

Par l'indice de validité V_{PC} le système génère trois règles d'association floues, avec le même nombre d'attribut, dans ce cas nous choisissons celle qui a le support maximum.

$\varphi \equiv$]fnlwgt: 0,31] $\wedge$ [hours-per-week: 0,38] $\wedge$ [native-country: United-States]==>< =50K

Exemple : Soit l'attribut : *hours-per-week*

$$S(hours-per-week = 38.84, hours-per-week = 0.38) = ...$$

$$..... = \frac{M_3(hours-per-week \cap hours-per-week = 0.38)}{M_3((hours-per-week = 38.84 \cup hours-per-week = 0.38))}$$

d'où

$$S(hours-per-week = 38.84, hours-per-week = 0.38) = \frac{0.61}{1} = 0.61$$

Attribut	M	φ	$\min(S(\varphi, M))$
Age	37.061	/	**0.61**
fnlwgt	190308.34	0.31	
education-num	9.65	/	
capital-gain	145.19	/	
capital-loss	61.75	/	
hours-per-week	38.84	0.38	
Workclass	private	/	
sex	Male	/	
Type d'éducation	HS_grade	/	
marital-status	Married-civ-epose	/	
native-country	United-States	United-States	
class	<=50k	<=50	

Table VI.19 : *table de degré de comparaison pour la base Adult (par la méthode de l'indice V_{PC})*

➢ **La méthode V_{FS}**

Par cette méthode on aura trios règles d'association qui ont le même nombre d'attribut nous choisissons celle qui a le plus grand support.

$\varphi \equiv$ [capital-gain: 0,07] $\wedge$ [capital-loss: 0,69] $\wedge$ [native-country: United-States]==>class: <=50K

Exemple : Soit l'attribut : *capital-gain*

$$S(capital-gain = 145.19, capital-gain = 0.07) = ...$$

$$..... = \frac{M_3(capital-gain = 145.19 \cap capital-gain = 0.07)}{M_3((capital-gain = 145.19 \cup capital-gain = 0.07))}$$

d'où

$$S(capital-gain = 9.65, capital-gain = 0.07) = \frac{0.39}{1} = 0.39$$

Attribut	M	φ	$\min(S(\varphi, M))$
Age	37.061	/	**0.09**
fnlwgt	190308.34	/	
education-num	9.65	/	
capital-gain	145.19	0.07	
capital-loss	61.75	0.69	
hours-per-week	38.84	/	
Workclass	private	/	
sex	Male	/	
Type d'éducation	HS_grade	/	
marital-status	Married-civ-epose	/	
native-country	United-States	United-States	
class	<=50k	<=50	

Table VI.20 : *table de degré de comparaison pour la base Adult (par la méthode de l'indice V_{FS})*

VI.3.5 Interprétation des résultats

	Degré de ressemblance		Taille de prémisse	
	KDD'99	*Adult*	*KDD'99*	*Adult*
Agrawal	0.46	0.45	4	3
L'indice V_{PC}	0.25	0.61	6	3
L'indice V_{FS}	0.5	0.09	4	3

Table VI.21 : *table de degré de comparaison pour les deux base de données*

- La méthode V_{FS} peut donner, de manière assez inexpliquée, de mauvais résultats (0.09 de degré de ressemblance pour une prémisse de taille 3 semble peu satisfaisant).
- La méthode V_{PC} peut donner les meilleurs résultats en terme de degré de ressemblance et se présente comme assez stable par rapport à cet aspect. La taille de prémisse peut cependant changer assez sensiblement d'une base à une autre.
- La méthode d'Agrawal présente une meilleure stabilité en terme de degré de ressemblance par rapport au modèle de validité, en effet, le degré de ressemblance change sensiblement peu d'une base à une autre.

VI.4 Le temps d'exécution

Nous intéressons aussi au temps d'exécution de l'application, à cet effet nous calculons le temps pour la recherche du nombre de partition, la construction de ses partitions floues, jusqu'à la découverte des règles d'association. Et parce que la base *Adult* est celle qui contient le plus grand nombre de transactions, nous utilisons la base *adult*. La table VI.22 et la figure VI.8, présentent les différents résultats.

Nb transactio / méthode	500	1000	1500	2000	2500	3000	4000	5000
Agrawal	1	3	5	6	11	11	16	23
V_{PC}	1	2	4	4	5	7	15	19
V_{FS}	1	2	3	4	5	7	10	14

Table VI.22 **:** *la variation de temps d'exécution par rapport au nombre de transactions*

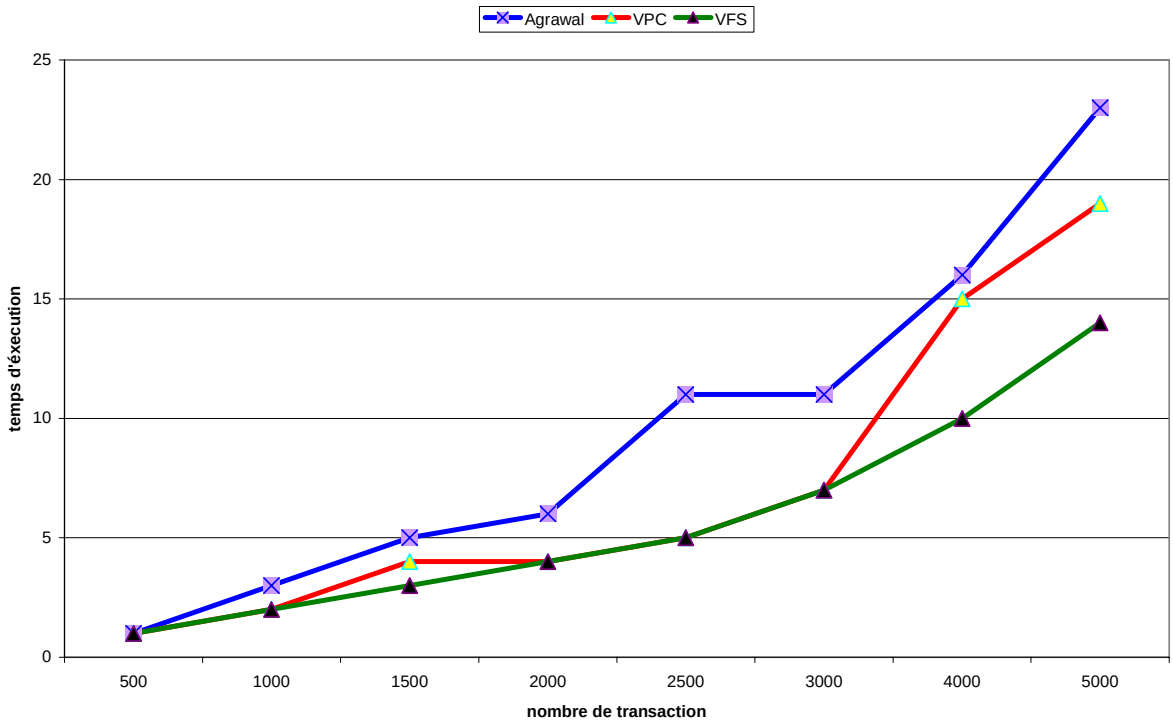


Figure VI.8 **:** *Variation de temps d'exécution par rapport au nombre de transaction*

Nous constatons que le temps d'exécution augmente linéairement en fonction du nombre de transaction, et il varie entre 1 à 23 secondes, pour une base varie de 500 à 5000

transactions. De là nous déterminons que l'algorithme est très efficace pour une large base de données.

VI.5 Conclusion

Nous venons de présenter dans ce chapitre les différents résultas expérimentaux de notre proposition, la proposition a été appliquée sur deux bases de données. Nous avons considéré parallèlement trois approches pour la construction des partitions floues. Une évaluation des règles d'association floues ainsi générées a été aussi expérimentée manuellement. Cette évaluation repose sur la construction d'un modèle de validation et l'utilisation d'une fonction de ressemblance. Une synthèse et interprétation des résultats expérimentaux à été aussi donnée.

Conclusion

Dans notre travail, nous avons proposé une nouvelle approche pour la découverte de règles d'association floues.

Beaucoup de travaux ont été proposé pour la découverte de telles règles, mais tous les travaux proposés considèrent que les ensembles flous sont fournis par un expert du domaine ou de manière empirique. Dans notre travail, nous avons proposé de déterminer automatiquement les partitions floues par une méthode de segmentation floue non supervisée. Notre proposition est théoriquement fondée.

Les méthodes de segmentation floue non supervisées soufrent du problème du choix du bon nombre de partitions. Pour palier à ce problème nous avons proposé trois approches pour trouver le nombre optimal de partitions.

Après la construction des partitions floues de chaque domaine d'attribut, nous avons proposé de transformer la base de transaction à une base de transaction floue, puis nous cherchons les règles d'association floues. La recherche des règles d'association floues est basée sur les mesures de support et de confiance, ces mesures sont basées sur l'utilisation d'une cardinalité floue. Beaucoup de travaux existent sur la cardinalité des ensembles flous mais aucune de ces cardinalités floues ne semble adaptée à des problématique de fouille de donnée (très gros volume de données). A cet effet nous proposons d'utiliser une cardinalité floue basée sur les coupes de niveau α (α-coupes).

Considérant trois approches pour trouver le nombre optimal de partition, nous aurons trois partitions floues différentes, en conséquence des degrés d'appartenances différents, ceci va nous générer trois types de règles d'association floues. Afin d'indiquer la meilleure méthode de partitionnement nous proposons de calculer le degré de ressemblance de différentes règle générées par rapport à un modèle de validation généré à partir de la base de transaction initiale.

Pour l'approche que nous avons proposée nous avons eu de grande difficulté lors d'attribution des labels linguistiques aux différents partitions floues trouvées, ce qui nous à amené à proposer d'associer à chaque partition son centre de classe. Par le calcul du temps

d'exécution, nous avons constaté que notre approche proposée est bien adaptée à de larges bases de données.

Et comme perspective de notre travail, nous envisageons d'étendre notre travail pour la prise en compte des valeurs manquantes (missing values), car ce cas est très largement rencontré dans des bases de données réelles. Nous proposons aussi d'étendre ce travail afin de considérer une hiérarchie de clusters ce qui permettrait de formaliser les partitions et leurs affecter des labels linguistiques.

Ce travail a été enrichissant du moment qu'il m'a permis de m'approfondir sur trois domaines : Data Mining, Théorie des ensembles flous, Segmentation.

Dans tous les domaines de l'informatique dans lesquels on désire analyser un ensemble de données il est nécessaire de disposer d'un opérateur capable d'évaluer précisément les ressemblances ou les relations qui existent entre les informations manipulées. Pour qualifier cet opérateur nous utiliserons les mesures de similarités.

Définition : (*M*-mesure floue)

M une mesure floue est une fonction de $F(\Omega)$ dans $\Re$ telle que pour tout A et B de $F(\Omega)$:

- $M(\phi)=0$,
- si $B \subseteq A$, alors $M(B) \leq M(A)$.

Les mesures floues suivantes sont utilisées comme mesure de comparaison :

$$M_1 = \int_{\Omega} \mu_A(x)dx \qquad \text{(Annexe1.1)}$$

$$M_2(A) = \sup_{x\in\Omega} \mu_A(x) \qquad \text{(Annexe1.2)}$$

$$M_3(A) = \sum_{x\in\Omega} \mu_A(x) \qquad \text{(Annexe1.3)}$$

Définition : (différence sur $F(\Omega)$)

La différence entre les ensembles flous est définie comme le complément d'un opérateur d'implication. Une opération de différence est noté $(-)$ doit vérifier pour tout sous-ensemble flou *A* et *B* les propriétés suivante :

- Si $A \subseteq B$, alors $A - B = \phi$
- $A - B \subseteq A - (A \cap B)$
- $B \subset B' \Rightarrow (B - A \subseteq B' - A)$

Exemple d'un opérateur de différence

$$\mu_{A-B}(x) = \max(0, \mu_A(x) - \mu_B(x)) \qquad \text{(Annexe1.4)}$$

$$\mu_{A-B}(x) = \begin{cases} \mu_A(x) & \text{si } \mu_B(x) = 0 \\ 0 & \text{si } \mu_B(x) > 0 \end{cases} \qquad \text{(Annexe1.5)}$$

Similitude des ensembles flous

Définition : (*M*-mesure de similitude)

Une *M*-mesure de similitude S sur Ω est une mesure de comparaison telle que $S(A,B) = f_s(X,Y,Z)$

Est :

- monotone non décroissante selon X (les composantes communes),
- monotone non croissante selon Y et Z (les composantes propres).

Il existe différente famille de mesure de similarité, pour pouvoir choisir parmi ces mesures, [112] considère les propriétés suivantes et qui sont satisfait par les *M*-mesure de similitude.

- réflexivité $(S(A,A) = 1)$ qui signifier que $\forall X \neq 0\ f_s(X,0,0) = 1$
- exclusivité $(S(A,B) = 0$ if $(A \cap B) = \phi)$ qui signifier que
  $\forall Y \neq 0\, \forall Z \neq 0\ f_s(0,Y,Z) = 0$.
- symétrie $(S(A,B) = S(B,A))$ qui signifier que
  $\forall X\ Y\ Z\ f_s(X,Y,Z) = f_s(X,Z,Y)$

Mesure de satisfiabilité

Pour comparer en se référant a un objet, afin de décider si un objet nouveau et compatible avec l'objet de référence ou satisfait l'objet de référence.

Définition : (*M*-mesure de satisfiabilité)

Une *M*-mesure de satisfiabilité S sur Ω est une *M*-mesure de similitude S telle que :

- $\forall\ Y\ Z\ \ f_s(0,Y,Z) = 0$ (exclusivité),
- $f_s(X,Y,Z)$ indépendant de Z, c'est-à-dire $S(A,B) = f_s(X,Y)$
- $\forall X \neq 0\ \ f_s(X,0,.) = 1$ (réflexivité).

Exemple de M-mesure de satisfiabilité :

- $S(A,B)=\frac{M(A\cap B)}{M(B)}$ avec M_1 et M_3 les mesures floue et comme opérateur de différence l'opérateur (Annexe1.4).
- $S(A,B)=\inf_x \min(1-\mu_B(x)+\mu_A(x),1)=1-M_2(B-A)$ et comme opérateur de différence l'opérateur (Annexe1.4).

Mesure d'inclusion

Pour comparer en se référant a un objet, afin de décider si un objet nouveau et compatible avec l'objet de référence ou satisfait l'objet de référence. Il prend en considération la notion d'inclusion.

Définition : (*M*-mesure d'inclusion)

Une *M*-mesure d'inclusion *S* sur Ω est une *M*-mesure de similitude *S* telle que :

- $\forall Y\ Z\ \ f_s(0,Y,Z)=0$ (exclusivité),
- $f_s(X,Y,Z)$ indépendant de Y, c'est-à-dire $S(A,B)=f_s(X,Z)$
- $\forall X\neq 0\ \ f_s(X,0,.)=1$ (réflexivité).

Exemple de M-mesure de satisfiabilité :

- $S(A,B)=\frac{|(A\cap B)|}{|(A)|}=\frac{M_3(A\cap B)}{(M_3(A\cap B)+M_3(A-B))}$ avec l'opérateur (Annexe1.4) comme opérateur de différence.
- $S(A,B)=\inf_x \min(1-\mu_A(x)+\mu_B(x),1)=1-M_2(A-B)$ et comme opérateur de différence l'opérateur (Annexe1.4).

Les différentes valeurs de degrés de ressemblance

La base *KDD'99*

L'évaluation d'une règle d'association générée par la méthode d'agrawal

$$S(srv\_rerror\_rate = 0.68, srv\_rerror\_rate = 0.842) = ...$$
$$..... = \frac{M_3((srv\_rerror\_rate = 0.68 \cap srv\_rerror\_rate = 0.842))}{M_3((srv\_rerror\_rate = 0.68 \cup srv\_rerror\_rate = 0.842))}$$
$$S(srv\_rerror\_rate = 0.68, srv\_rerror\_rate = 0.842) = \frac{0.46}{1} = 0.46$$

$$S(dst\_host\_diff\_srv\_rate = 0.01, dst\_host\_diff\_srv\_rate = 0{,}01) = ...$$
$$..... = \frac{M_3((dst\_host\_diff\_srv\_rate = 0.01 \cap dst\_host\_diff\_srv\_rate = 0{,}01))}{M_3((dst\_host\_diff\_srv\_rate = 0.01 \cup dst\_host\_diff\_srv\_rate = 0{,}01))}$$
$$S(dst\_host\_diff\_srv\_rate = 0.01, st\_host\_diff\_srv\_rate = 0{,}01) = \frac{1}{1} = 1$$

$$S(\log ged\_in = 1, \log ged\_in = 1) = 1$$

$$S(flag = RSTR, flag = RSTR) = 1$$

$$S(attack\_name = Appach2, attack\_name = Appach2) = 1$$

$$\bigcap_{i=1}^{5} S(\mathrm{M}_i, \varphi_i) = 0.46$$

Donc:

$S(\mathrm{M}, \varphi) = 0.46$.

L'évaluation d'une règle d'association générée par la méthode de l'indice V_{PC}

$$S(srv\_diff\_host\_rate = 254.55, srv\_diff\_host\_rate = 0.06) = ...$$
$$..... = \frac{M_3((srv\_diff\_host\_rate = 254.55 \cap srv\_diff\_host\_rate = 0.06))}{M_3((srv\_diff\_host\_rate = 254.55 \cup srv\_diff\_host\_rate = 0.06))}$$
$$S(srv\_diff\_host\_rate = 254.55, srv\_diff\_host\_rate = 0.06) = \frac{0.50}{1} = 0.50$$

$$S(srv\_rerror\_rate = 0.68, srv\_rerror\_rate = 0.842) = ...$$

$$..... = \frac{M_3((srv\_rerror\_rate = 0.68 \cap srv\_rerror\_rate = 0.842))}{M_3((srv\_rerror\_rate = 0.68 \cup srv\_rerror\_rate = 0.842))}$$

$$S(srv\_rerror\_rate = 0.68, srv\_rerror\_rate = 0.842) = \frac{0.45}{1} = 0.45$$

$$S(dst\_host\_diff\_srv\_rate = 0.01, dst\_host\_diff\_srv\_rate = 0{,}01) = ...$$

$$..... = \frac{M_3((dst\_host\_diff\_srv\_rate = 0.01 \cap dst\_host\_diff\_srv\_rate = 0{,}01))}{M_3((dst\_host\_diff\_srv\_rate = 0.01 \cup dst\_host\_diff\_srv\_rate = 0{,}01))}$$

$$S(dst\_host\_diff\_srv\_rate = 0.01, dst\_host\_diff\_srv\_rate = 0{,}01) = \frac{1}{1} = 1$$

$$S(dst\_host\_serror\_rate = 0.13, dst\_host\_serror\_rate = 0.42) = ...$$

$$..... = \frac{M_3(dst\_host\_serror\_rate = 0.13 \cap dst\_host\_serror\_rate = 0.42)}{M_3(dst\_host\_serror\_rate = 0.13 \cup dst\_host\_serror\_rate = 0.42)}$$

$$S(dst\_host\_serror\_rate = 0.13, dst\_host\_serror\_rate = 0.42) = \frac{0.25}{1} = 0.25$$

$$S(dst\_host\_srv\_serror\_rate = 0.13, dst\_host\_srv\_serror\_rate = 0.43) = ...$$

$$..... = \frac{M_3(dst\_host\_srv\_serror\_rate = 0.13 \cap dst\_host\_srv\_serror\_rate = 0.43)}{M_3(dst\_host\_srv\_serror\_rate = 0.13 \cup dst\_host\_srv\_serror\_rate = 0.43)}$$

$$S(dst\_host\_srv\_serror\_rate = 0.13, dst\_host\_srv\_serror\_rate = 0.43) = \frac{0.25}{1} = 0.25$$

$$S(\log ged\_in = 1, \log ged\_in = 1) = 1$$

$$S(flag = RSTR, flag = RSTR) = 1$$

$$S(attack\_name = Appach2, attack\_name = Appach2) = 1$$

$$\bigcap_{i=1}^{8} S(\mathrm{M}_i, \varphi_i) = 0.25$$

Donc:

$S(\mathrm{M}, \varphi) = 0.25$.

L'évaluation d'une règle d'association générée par la méthode d'indice de validité V_{FS}

$$S(dst\_host\_count = 254.55, dst\_host\_count = 098) = ...$$

$$..... = \frac{M_3(dst\_host\_count = 254.55 \cap dst\_host\_count = 098)}{M_3(dst\_host\_count = 254.55 \cup dst\_host\_count = 098)}$$

$$S(dst\_host\_count = 254.55, dst\_host\_count = 098) = \frac{0.54}{1} = 0.54$$

$$S(srv\_diff\_host\_rate = 254.55, srv\_diff\_host\_rate = 0.00) = ...$$

$$..... = \frac{M_3((srv\_diff\_host\_rate = 254.55 \cap srv\_diff\_host\_rate = 0.00))}{M_3((srv\_diff\_host\_rate = 254.55 \cup srv\_diff\_host\_rate = 0.00))}$$

$$S(srv\_diff\_host\_rate = 254.55, srv\_diff\_host\_rate = 0.00) = \frac{0.50}{1} = 0.50$$

$$S(\log ged\_in = 1, \log ged\_in = 1) = 1$$

$$S(flag = RSTR, flag = RSTR) = 1$$

$$S(attack\_name = Appach2, attack\_name = Appach2) = 1$$

$$\bigcap_{i=1}^{5} S(\mathrm{M}_i, \varphi_i) = 0.50$$

Donc:

$S(\mathrm{M}, \varphi) = 0.50$.

Base *Adult*

L'évaluation de la première règle d'association générée par la méthode d'agrawal

$$S(education-num=9.65, education-num=0.61)=.....$$

$$.....=\frac{M_3(education-num=9.65\cap education-num=0.61)}{M_3((education-num=9.65\cup education-num=0.61))}$$

$$S(education-num=9.65, education-num=0.61)=\frac{0.49}{1}=0.49$$

$$S(capital-gain=145.19, capital-gain=0.07)=.....$$

$$.....=\frac{M_3(capital-gain=145.19\cap capital-gain=0.07)}{M_3((capital-gain=145.19\cup capital-gain=0.07))}$$

$$S(capital-gain=145.19, capital-gain=0.07)=\frac{0.45}{1}=0.45$$

$$S(nativ-country=united-statee, nativ-country=united-statee)=1$$

$$S(class=<=50, class=<=50)=1$$

$$\bigcap_{i=1}^{4} S(\mathrm{M}_i, \varphi_i)=0.45$$

Donc:

$S(\mathrm{M}, \varphi\ )=0.45$.

L'évaluation de la deuxième règle d'association générée par la méthode d'agrawal

$$S(capital-gain=145.19, capital-gain=0.07)=.....$$

$$.....=\frac{M_3(capital-gain=145.19\cap capital-gain=0.07)}{M_3((capital-gain=145.19\cup capital-gain=0.07))}$$

$$S(capital-gain=145.19, capital-gain=0.07)=\frac{0.45}{1}=0.45$$

$$S(hours-per-week=38.84, hours-per-week=0.38)=.....$$

$$.....\frac{M_3(hours-per-week=38.84\cap hours-per-week=0.38)}{M_3(hours-per-week=38.84\cup hours-per-week=0.38)}$$

$$S(hours-per-week=38.84, hours-per-week=0.38)=\frac{0.53}{1}=0.53$$

$$S(nativ-country=united-statee, nativ-country=united-statee)=1$$

$$S(class=<=50, class=<=50)=1$$

$$\bigcap_{i=1}^{4} S(\mathrm{M}_i, \varphi_i)=0.45$$

Donc:

$S(\mathrm{M}, \varphi\ )=0.45$.

L'évaluation de la règle d'association générée par la méthode de l'indice de validité V_{PC}

$$S(fnlwgt=190308.34, fnlwgt=0.31)=.....$$

$$.....=\frac{M_3(fnlwgt=190308.34\cap fnlwgt=0.31)}{M_3(fnlwgt=190308.34\cup fnlwgt=0.31)}$$

$$S(fnlwgt=190308.34, fnlwgt=0.31)=\frac{0.83}{1}=0.83$$

$$S(hours-per-week=38.84, hours-per-week=0.38)=.....$$

$$.....=\frac{M_3(hours-per-week=38.84\cap hours-per-week=0.38)}{M_3(hours-per-week=38.84\cup hours-per-week=0.38)}$$

$$S(hours-per-week=38.84, hours-per-week=0.38)=\frac{0.61}{1}=0.61$$

$$S(nativ-country=united-statee, nativ-country=united-statee)=1$$

$$S(class =<= 50, class =<= 50) = 1$$

$$\bigcap_{i=1}^{4} S(\mathrm{M}_i, \varphi_i) = 0.61$$

Donc:

$S(\mathrm{M}, \varphi\ ) = 0.61$.

L'évaluation de la règle d'association générée par la méthode de l'indice de validité V_{FS}

$$S(capital - gain = 145.19, capital - gain = 0{,}07) = ...$$

$$..... = \frac{M_3(capital - gain = 145.19 \cap capital - gain = 0{,}07)}{M_3(capital - gain = 145.19 \cup capital - gain = 0{,}07)}$$

$$S(capital - gain = 145.19, capital - gain = 0{,}07) = \frac{0.39}{1} = 0.39$$

$$S(capital - loss = 61.75, capital - loss = 0{,}69) = ...$$

$$..... = \frac{M_3(capital - loss = 145.19 \cap capital - loss = 0{,}69)}{M_3(capital - loss = 145.19 \cup capital - loss = 0{,}69)}$$

$$S(capital - loss = 61.75, capital - loss = 0{,}69) = \frac{0.09}{1} = 0.09$$

$$S(nativ - country = united - states, nativ - country = united - states) = 1$$

$$S(class =<= 50, class =<= 50) = 1$$

$$\bigcap_{i=1}^{4} S(\mathrm{M}_i, \varphi_i) = 0.09$$

Donc:

$S(\mathrm{M}, \varphi\ ) = 0.09$.

BIBLIOGRAPHIE

[1] Fayyad U., Piatetsky-Shapiro G., Smyth P., "*From Data Mining to Knowledge Discovery in Databases*", Dans aimag KDD overview pp 1-34, 1996.

[2] Zighed D.A., kodratoff Y., Napoli A. "*Extraction de connaissance à partir d'une base de donnée*" Bulletin AFIA'01,2001.

[3] Fayyad U., Piatetsky-Shapiro G., Smyth P., "*From Data Mining to Knowledge Discovery in Databases*", Advices in Knowledge Discovery and Data Mining, MIT Press, 1: pp 1-36, 1998.

[4] Zighed D.A., Duru G., Auray J.P., "*Sipina, méthode et logiciel*", A. Lacassagne 2000.

[5] Agrawal R., Imielinski T., Swami A., "*Mining Association rules between sets of items in large database*", Proceedings of the ACM SIGMOD International Conference on Management of Data, Washington, DC, pp 207-216, May 26-28, 1993.

[6] morgon.univ-lyon2.fr/Introduction_au_datamining_cours.htm

[7] Agrawal R., Srikant A., "*Fast algorithms for mining association Rules*", IBM Research Report RJ9839, IBM Almaden Research Center, San Jose, CA, June 1994.

[8] Agrawal R., "*Parallel mining of association rules*", IEEE Transaction on knowledge and Data Engineering, 8 (6), Decembre 1996.

[9] Jourdan L., " *Methaheuristiques pour l'extraction de connaissance : Application a la Génomique*", Thèse de doctorat, Université des sciences et technologies de Lille, Novembre 2003.

[10] Azé J., "*Une nouvelle mesure de qualité pour l'extraction de pépites de connaissances*", Extraction des connaissances et apprentissage ECA, 17(1), pp171-182, 2003.

[11] Han J., Fu Y., "*Discovery of multiple level association rules from databases*" Procceding of the 21st international conference on very Large Data Bases (VLDB'95), Morgan Kaufman, pp 407-419, September 1995.

[12] Bayarado R.J., Agrawal R., "*Mining the most Interesting Rules*", Proceeding of the 5th ACMSIGKDD Int'l Conference on knowledge Discovery and Data Mining KDD'99, pp 145-154, 1999.

[13] Dong G., Li J., "*Intersting of discoverd association rules in terms of neighborhood-based unexpectedness*", proceedings of the 2nd Pacific-Asia international conference on research and development in knowledge Discovery and Data Mining PAKDD'98 vol 1394, pp 72-86, Avril 1998.

[14] Kandel A., "*Fuzzy expert system*", CRC Press, Boca Raton, FL, pp. 8-19, 1992.

[15] Silberschatz A., tuzhilincs A., "*on subjective measures of interestingness in Knowledge discovery*", Menlo Park, California, USA, AAAI Press, pp 275-281, 1995.

[16] Silberschatz A., tuzhilincs A., "*User-Assisted Knowledge discovery: How Much Should the user be Involved*", Montréal, Quebec, Canada, 1996.

[17] Pasquier N., "*Data mining: Algorithme d'extraction et de reduction des règles d'association dans les bases de* données", thèse d'université, université de Clermont Ferrand II, 2000.

[18] Klemettinen M., Manilla H., Ronkainen P., Toivonen H., Verkamo A.I., "*Finding intersting rules from large sets of discoverd association rules*", proceeding of international conference on information and knowledge Management CIKM'94, pp 404-407, November 1994.

[19] Cooley R., Mobasher B., Srivastava J., "*Data preparation for mining World Wide Web Browsing Patterns*", Knowledge and information systems, 1(1):pp 5-32, 1999.

[20] Baccini A., Besse P., "*Data mining I Exploration statistique*", Publication du Laboratoire de Statistique et Probabilités Université Paul Sabatier, septembre 2005.

[21] Srikant R., Agrawal R., "*Mining quantitative association rules in large relational tables*", proccedings of ACM SIGMOD, pp 1-12, 1996.

[22] Dubois D., Prade H., "*Measuring properties of fuzzy sets: a general technique and its use in fuzzy query evaluation*", Fuzzy Sets and Systems 38, pp.137-152, 1989.

[23] Piatsky-Schapiro G., Frawly W.J., "*Knowledge Discovery in Databases*", AAAI Presse, The MIT Press, Menlo Park, California, 1991.

[24] Bastide Y., "*Data Mining: algorithms par niveau, techniques d'implémentation et application*", Thèse de doctorat, Université Blaise Pascal, Clermont-Ferrand II, 2000.

[25] Pëi J., Han J., Mao R., "*closet: an efficient Algorithme for mining frequent closed itemsets*", ACM SIGMOD work shop on research Issues In Data mining and Knowledge Discovery, pp 21-30, 2000.

[26] Zaki M.J., Ho C.T., "*Large scale parallel data mining*", Vol. 1759. LNAI, Berlin, Germany, 2000.

[27] Zaki M.J., "*Mining Non-redundant Association Rules*. *Data Mining and Knowledge* discovery", An international journal, 9, pp 223-248, 2004.

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

[29] Quinlan J.R., "*C4.5: Programs for Machine Learning*", Morgan Kaufmann, 1993.

[30] Dubois D., Prade H., "*What are fuzzy rules and how to use them. Fuzzy Sets and Systems*", n_84, pp 169-185.

[31] Ng R., Han J., "*Efficient and effective clustering methods for spatial data mining*", Proceedings of the 20th VLDB conference, 1994.

[32] Hinneburg A., Keim D.A., "*An Efficient Approach to Clustering in Large Multimedia Databases with Noise*", Proceedings of the 4th International Conference on Knowledge Discovery in Databases (KDD'98), pp 58–65, New York, USA, 1998.

[33] Ester M., Kriegel H.P., Sander J., Xu X., "*A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise*", Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, pp 226-231, Portland, Oregon, 1996.

[34] Wang W., Yang J., Muntz R., "*STING: A Statistical Information Grid Approach to Spatial Data Mining*", Proceedings of the 23rd International Conference on Very Large Data Bases, (VLDB), pp 186–195, Athens, Greece, 1997.

[35] Sheikholeslami G., Chatterjee S., Zhang A., "*WaveCluster: A Multi-Resolution Clustering Approach for Very Large Spatial Databases*", Proceedings of the 24rd International Conference on Very Large Data Bases, (VLDB), pp 428–439, New York, USA, 1998.

[36] Zahn C.T., "*Graph- Theoretical Methods for Detecting and Describing Gestalt Clusters*", IEEE Transactions on Computers, 20(1), pp 68-86, 1971.

[37] Hinneburg A., Keim D.A., "*Clustering Techniques for Large Data Sets: From the Past to the Future*", Proceedings of the International Conference on Knowledge Discovery in Databases*(KDD'99)*, San Diego, CA, USA, 1999.

[38] Toussaint G.T., "*The Relative Neighbourhood Graph of a Finite Planar Set*", Pattern Recognition, 12(4), pp 261-268, 1980.

[39] Ozawa K., "*A Stratification Overlapping Cluster Scheme*", Pattern Recognition, 81(3-4), pp 279-286, 1985.

[40] Dempster A.P., Laird N.M., Rubin D.B., "*Maximum Likelihood from Incomplete Data Via the EM Algorithm*", Journal of the Royal Statistical Society, Series B, 39(1), pp 1–38, 1977.

[41] Hartley H., "*Maximum Likelihood Estimation from Incomplete Data. Biometrics*", 14, pp 174–194, 1958.

[42] Kuok C.M., Fu A.W.C., Wong M.H., "*Mining Fuzzy Association Rules in Databases*", SIGMOD Record, 27(1), pp 41-46, 1998.

[43] KODRATOFF Y., "*techniques et outils de l'extraction de connaissances à partir des données*", Signaux n°92 pp 38-43, Mars 1998.

[44] Raynaud O., "*Le projet logiciel E.C.D.Sagitta : Un état des lieux*", Research Report LIMOS/RR-06-05, 22 mai 2006.

[45] Gustafson D. E., Kessel W.C., "*Fuzzy clustering with a fuzzy covariance matrix*", in Proc. IEEE Conf. Decision Contr., San Diego, CA, 1979.

[46] Jain A., Zongker D., "*Feature Selection: Evaluation, Application, and Small Sample Performance*", IEEE Transactions on pattern analysis and machine intelligence, Vol. 19, No. 2, FEBRUARY 1997.

[47] Wang W., "*Data Mining: Concepts, Algorithms, and Applications*", COMP pp 290-090, 2003.

[48] Ben Yahia S., Nguifo E.M., "*Approches d'extraction de règle d'association basées sur la correspondance de Galois*", RSTI - ISI – 9, Motifs dans les bases de données, pp 23-55, 2004.

[49] Rioult F., "*Fouille de données orientée motifs, méthodes et usages*", GREYC - Équipe Données-Documents-Langues CNRS UMR 6072 Université de Caen Basse-Normandie France

[50] Bastide Y., Taouil R., Pasquier N., StummeG., Lakhal L., "*Pascal : un algorithme d'extraction des motifs fréquents*", Technique et science informatiques. Vol. 21, n◦ 1, pp 65-95, 2002.

[51] Fiot C., "*Données imparfaites et motifs séquentiels*", Séminaire LIRMM, Juin 2006.

[52] Fiot C., "*Quelques techniques de fouille de données*", Master Pro., 2005/06.

[53] Au W.H., Chan K.C.C., "*FARM: A data mining system for discovering fuzzy association rules*", Proceedings 8th IEEE Internat. Conf. Fuzzy systems, Seoul, Korea, pp 1217-1222, 1999.

[54] Delgado M., Marin N., Sànchez D., Vila M. A., "*Fuzzy association rules: General model and application*", IEEE Transactions on Fuzzy Systems 11(2), pp. 214-225, 2003.

[55] Hong T.P., Lin K.Y., Wang S.L., "*Fuzzy data mining for interesting generalized association rules*", Fuzzy Sets and Systems 138, pp 255-269, 2003.

[56] Ray S., Turi R. H., "*Determination of Number of Clusters in k-means Clustering and Application in Color Image Segmentation*". In Proceedings of the 4th International Conference on Advances in Pattern Recognition and Digital Techniques, Calcutta, India, pp. 137–143, 1999.

[57] A.k.Jain, M.N.Marty, et P.J.flynn., "*Data clustering*". Dans A Review, volume 31, Chapitre 6, pages 263-323. ACM, Sept. 1999.

[58] Chuanjun L.B, Si-Qing Z.P., "*Similarity measure for multi-attribute data*", No. 0237954

[59] HúsekD., Pokorný J., Řezanková H., Snášel V., "*Data clustering: from documents to the web*", work supported by the project 1ET100300419 of the Program Information Society of the Thematic Program II of the National Research Program of the Czech Republic and the project 201/05/0079 of the Grant Agency of the Czech Republic.

[60] Liao T.W. et al., "*A fuzzy c-means variant for the generation of fuzzy term sets*", Elsevier Science / Fuzzy Sets and Systems 135, pp 241–257, 2003.

[61] Dick S., Meeks A., Last M., Bunke H., Kandel A., "*Data mining in software metrics databases*", Elsevier Science / Fuzzy Sets and Systems 145 pp 81–110, 2004.

[62] Leski J., "*Towards a robust fuzzy clustering*", *Elsevier* Science / Fuzzy Sets and Systems 137, pp 215–233, 2003.

[63] Paviot G., "*Etude comparative de la classification ascendante hiérarchique et de la classification floue pour identifier cinq famille de voitures*", Document de recherche N°1997-4, laboratoire Orleanais de gestion.

[64] Pakhira M.K. et al., "*Astudy of some fuzzy cluster validity indices, genetic clustering and application to pixel classification*", Elsevier Science / Fuzzy Sets and Systems 155 pp 191–214, 2005.

[65] Shihab A.I., "*Fuzzy clustering algorithms and their application to medical image analysis*", these de doctorat, université de London, Decembre 2000.

[66] Kaymak U., Setnes M., "*Extended Fuzzy Clustering Algorithms*" publication de ERIM. ERS-2000-51-LIS, Novembre 2000

[67] Klawonn F., höppner f., "*What is fuzzy about fuzzy clustering, understanding and improving the concepte of the fuzzifier*", publication de departement d'informatique, Germany.

[68] http// www.ics.uci.edu/databases/Adulte

[69] Sun H., Wang S., Jiang Q., "*FCM-Based Model Selection Algorithms for Determining the Number of Clusters*", Pattern Recognition 37, pp 2027 – 2037, 2004

[70] Dubois D., Prade H., H¨ullermeier E., "*A Systematic Approach to the Assessment of Fuzzy Association Rules*", paper presented at the 10th International Fuzzy Systems Association World Congress, Istambul, 2003.

[71] Dubois D., Prade H., Sudkamp T., "*On the representation, measurement, and discovery of fuzzy associations*", IEEE transactions on fuzzy systems, Vol. 13, NO. 2, APRIL 2005.

[72] Delgado M., Sanchez D., Vila M. A., "*Fuzzy cardinality based evaluation of quantified sentences*", International Journal of Approximate Reasoning, vol. 23, pp. 23-66, 2000.

[73] Bouchon-Meunier B., "*La logique floue. Que sais-je ?*", Presses Universitaires de France, num. 2702, 2ème édition, 1994.

[74] Leray P., "*Le Clustering en 3 leçons ”*publication de laboratoire PSI. pp.1/65

[75] Prade H., Richard G., Serrurier M., "*Enriching inductive logic programming with fuzzy predicates*", International Conference on Inductive Logic Programming Publication IRIT UMR5505, pp. 399-410, 2003.

[76] Jain A.K., Dubes R.C., "*Algorithms for Clustering Data*", Prentice-Hall, Englewood Cliffs, NJ, 1988.

[77] Roberto J., Bayardo Jr D., "*Efficient Mining Long Patterns From Databases*", ACM-SIGMOD Int. conf. on Management of data, pp. 85–93, 1998.

[78] Sun H., Wang S., Mei M., "*A Fuzzy Clustering Based Algorithm for Feature Selection*", Proceedings of the 1st IEEE International Conference on Machine Learning and Cybernetics ICMLC, Vol. 4, pp 1993 – 1998, Beijing, China, 2002.

[79] Everitt B.S., "*Cluster Analysis*", Halstead Press, London, 1974.

[80] Anderberg M.R., "*Cluster Analysis for Applications*", Academic Press, New York, 1973.

[81] Zhang T., Ramakrishnan R., Livny M., "*BIRCH: An Efficient Data Clustering Method for Very Large Data Bases*", Proceedings of the ACM SIGMOD International Conference on Management of Data, ACM Press. pp 103–114, Montréal, Québec, Canada, 1996.

[82] http://www.stat.stanford.edu/%7Ejhf/ftp/dm-stat.ps

[83] http://www. wikipedia.org/wiki/Entrepôt_de_données

[84] Klemettinen M., Mannila H., Toivonen H., " *A Data Mining Methodology an dits application to semi-automatic knowledge acquisition*", DEXA’97, proceedings of the 8th International work shop on database and expert systems applications, pp. 670 Washington, DC, USA, 1997.

[85] Cooley R., Srivastava J., Mobasher B., "*Web Mining: information and pattern Discovery on the World Wide Web*", Proceedings of the 9th IEEE International conference on tools with artificial intelligence (ICTAI’97), new port Beach, CA, USA, pp. 558-567, 1997.

[86] Wang K., Zhow S., Yeung J.-M.-S., Yang Q., "*Mining customer value: from association rules to direct marketing*", proceedings of the 19th International conference data Engineering (ICDE’03), Bangalore, India, pp. 738-740, 2003.

[87] Ordonnez C., Omiecinski E., "*Image mining: a new approach for data mining*", Technical report, Georgia Institue of technology, Atlanta, USA, 1998.

[88] Zaki M.-J., Parthasonathy S., Ogihara M., Li W., "New algorithms for fast Discovery of Association Rules", Proceedings of the 3rd International conference on knowledge discovery and data Mining, pp. 283-296, 1997.

[89] Toivonen H., "*Sampling Large Databases for Association Rules*", Proceedings of the 22nd International conference on Very Large databases (VLDB'96), pp. 134-145, 1996.

[90] Savasere A., Omiecinski E., Navathe S.-B., "*An efficient Algorithm for Mining Association Rules in Large Databases*", Proceedings of the 21st International conference on Very Large databases (VLDB'95), pp. 432-444, 1995.

[91] Toivonen H., Klemettinen M., Ronkainen P., Hatonen K., Manila H., "*Pruning and grouping of discovered association rules*", In Workshop Notes of the ECML-95 Workshop on Statistics, Machine learning, and knowledge discovery in databases, pp.47-52, Heraklion, Greece, April 1995.

[92] Srikant R., Vu Q., Agrawal R., "*Mining Association Rules with Item constraints*", Proceedings of the 3rd International conference knowledge discovery and data Mining, pp. 67-73, 1997.

[93] Goetlas B., Zaki M.-J., "*FIMI'03: Workshop on Frequent Itemset Mining Implementation*", FIMI'03 Workshop on frequent Itemset Mining Implementations, 2003.

[94] Diday E., Simon J.C., "*Clustering Analysis*", In Fu. K. S., editor, *In Digital Pattern Recognition*, volume 10, New York, USA, pp. 47–94, 1976.

[95] Zadeh L., "*Fuzzy Sets*", Information and control, 8, pp. 338-353, 1965.

[96] Banfield J., Raftery A., "*Model-Based Gaussian and non-Gaussian Clustering*", Biometrics, 49(3): pp.803–821, 1993.

[97] Dunn J.C., "*A Fuzzy Relative of the ISODATA Process and its Use in Detecting Compact, Well-Separated Clusters*", Journal of Cybernetics, 3(3): pp.32–57, 1973.

[98] Bezdek J.C., "*Pattern Recognition with Fuzzy Objective Function Algorithms*", Plenum, New York, 1981.

[99] Gath I., Geva A.B., "*Unsupervised Optimal Fuzzy Clustering*", IEEE Transactions on Pattern Analysis and Machine Intelligence, 11(7): pp.773–781, 1989.

[100] N.R. Pal and J.C. Bezdek., "*On Cluster Validity for the Fuzzy C-Means Model*", IEEE Transactions on Fuzzy Systems, 3(3):pp.370–379, 1995.

[101] Bezdek J.C., "*Cluster Validity With Fuzzy Sets*", J. Cybernet., 3(3): pp.58–72, 1974.

[102] Bezdek J.C., "*Mathematical Models for Systematics and Taxonomy*", In Proceedings of the 8th International Conference on Numerical Taxonomy, San Franscisco, CA, USA, pp. 143-166, 1975.

[103] Fukuyama Y., Sugeno M., "*A New Method of Choosing the Number of Clusters for the Fuzzy C-means Method*", In 5th Fuzzy Systems Symposium, pp. 247–250, 1989.

[104] Fink E., Kokku P.K., Nikiforou S., Hall L.O., Goldgof D.B., Krischer J.P., "*Selection of Patients for Clinical Trials: An Interactive Web-Based System. Artificial Intelligence in Medicine*", 31(3): pp. 241–254, 2004.

[105] Eschrich S., Ke J., Hall L.O., Goldgof D.B., "*Fast Accurate Fuzzy Clustering through Data Reduction*", IEEE Transactions on Fuzzy Systems, 11(2): pp.262–270, 2003.

[106] Chen Y., Qiu L., Chen W., Nguyen L., Katz R., "*Clustering Web Content for Efficient Replication*", In Proceedings of the 10th IEEE International Conference on Network Protocols (ICNP02), Paris, France, pp. 165–174, 2002.

[107] Bouguessa M., Wang S., Sun H., "*An objective approach to cluster validation*", Patern Recognition Letters, 27, pp. 1419-1430, 2006.

[108] De Luca A., Termini S., "*A definition of non-probabilistic entropy in the setting of fuzzy sets theory*", Information and Control, 20: pp. 301-312, 1972.

[109] Zadeh L., "*A theory of approximate reasoning*", In: R.R. Yager et al., eds., Fuzzy Sets and Applications. Selected Papers by Zadeh L. (John Wiley and Sons), pp. 367-412, 1987.

[110] http// www.kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[111] Omahover J.-F, Bouchon-Meunier B.,"*Equivalence entre mesure de similarité floues : application à la recherche d'images par le contenu*",6ème Congrès Européen de Science des systèmes, 19-22 septembre, 2005.

[112] Bouchon-Meunier B., Rifqi M., "*Towards general measures of comparison of objects*", Fuzzy sets and systems, vol.84 (2), pp. 143-153, 1996.

[113] Tversky A., "*Features of similitary*", Psychological Review, vol. 84, pp. 327-352, 1977.

[114] Rifqi M., Berger V., Bouchon-Meunier B., "*Discrimination power of measures of comparison*", Fuzzy sets and systems, 110, pp. 189-196, 2000.

[115] Djouadi Y., Radaoui S., "*Découverte de règles d'association*: *Application aux Données imprécises*", Logique floue et ses applications, Toulouse, France, pp. 195-202, 19-20 octobre, 2006.