N° d'ordre : 375/21

UNIVERSITE SULTAN MOULAY SLIMANE´
FACULTE DES SCIENCES ET TECHNIQUES´
Béni Mellal

Centre des Études Doctorales : Sciences et Techniques.
Formation doctorale : Mathématique et Physique appliquées.

THESE

Présentée par
Mouhcine EL Hassani

Pour l'obtention du grade de

DOCTEUR

Discipline : INFORMATIQUE
Spécialité : Informatique

Titre de la thèse : Contribution aux sciences de données via des modèles Hybrides de Clustering.

Soutenue le Samedi 13 Novembre 2021 à 10h devant la commission d'examen :

Pr Benachir ELHADADI	: Professeur à Faculté Polydisciplinaire, Université Sultan Moulay Slimane, Béni Mellal	Président du jury
Pr Karim EL MOUTAOUAKIL	: Professeur à Faculté Polydisciplinaire, Université Sidi Mohamed Ben Abdellah, Taza	Rapporteur
Pr Rachid EL AYACHI	: Professeur Habilité, Faculté des Sciences et Techniques, Université Sultan Moulay Slimane, Béni Mellal	Rapporteur
Pr Mohamed BASLAM	: Professeur Habilité, Faculté des Sciences et Techniques, Université Sultan Moulay Slimane, Béni Mellal	Rapporteur
Pr Said SAFI	: Professeur à la Faculté Polydisciplinaire, Université Sultan Moulay Slimane, Béni Mellal	Examinateur
Pr. Belaid BOUIKHALENE	: Professeur à la FP de Béni Mellal	Co-Directeur de thèse
Pr. Noureddine FALIH	: Professeur à la FP de Béni Mellal	Directeur de thèse

Dédicace

Cette thèse est le fruit d'effort conjugué de la part de ma famille entière et à leur tête ma mère et mon père, tant bien aimés. Je tiens à la leur dédier ainsi qu'à toutes les personnes qui y ont apporté bénéfiquement de près ou de loin.

Remerciements

Je tiens à exprimer ma gratitude à Monsieur MOHAMED NAIMI, professeur de l'enseignement supérieur, à l'Université Sultan Moulay Slimane (FST Béni-Mellal) et directeur du Laboratoire de Modélisation des Ecoulement et des Transferts (LAMET), pour m'avoir accepté au sein de son entité de recherche, diriger cette thèse, tout en ayant l'obligeance de consacrer de son temps pour prodiguer des conseils très précieux, pour son soutien constant tout au long de ce travail et pour la finalisation du présent mémoire. Que dieu bénisse son âme et qu'elle repose en paix.

Mes remerciements s'adressent également à mon co-directeur de thèse, Monsieur BELAID BOUIKHALENE et mon encadrant Monsieur NOUREDDINE FALIH professeurs de l'enseignement supérieur à l'Université Sultan Moulay Slimane (FP Béni-Mellal), pour leurs aides et leurs présences constantes chaque fois que je les sollicite pour une telle cause. Leurs nombreuses remarques et suggestions ont été bénéfiques pour avancer dans mes investigations. Qu'ils trouvent ici l'expression de mon entière reconnaissance.

Je remercie mes rapporteurs M. Karim EL MOUTAOUAKIL Professeur à Faculté Polydisciplinaire, Université Sidi Mohamed Ben Abdellah, Taza, M. Rachid EL AYACHI et M. Mohamed BASLAM Professeurs Habilités à la Faculté des Sciences et Techniques, Université Sultan Moulay Slimane, Béni Mellal pour avoir accepté de consacrer de leur temps pour juger ce travail et participer au jury de soutenance.

Mes remerciements vont également à M. Said SAFI Professeur à la Faculté Polydisciplinaire, Université Sultan Moulay Slimane, Béni Mellal, pour l'honneur qu'il m'a fait en acceptant d'examiner le présent mémoire et en porter un jugement.

Une pensée particulière va à messieux, Benachir El Hadadi, doyen de la FP Beni-Mellal qui m' a honnorer en acceptant d'etre le président du jury, CHEKLEKBIRE MALAININE, chef du département d'économie et de gestion à la FP Béni-Mellal, Monsieur RACHID HASNAOUI et Monsieur Bakhate Mohcine, professeurs habilités à la FP Béni-Mellal, qui m'ont supporté et assisté tout au long de ce travail, tant bien moralement que scientifiquement.

Enfin, je n'oublie pas de remercier profondément l'administration de la FST Béni-Mellal, à sa tête son doyen Monsieur Said MELLIANI, pour m'avoir offert la possibilité de m'inscrire au sein de l'institution dont il assure la responsabilité, chose qui ne m'a pas été facile ailleurs.

Résumé

Le besoin d'extraire et de traiter les informations à partir de données brutes, est l'un des objectifs les plus connus en analyse de données pour l'aide à la décision.

Dans ce contexte, nous abordons les différents aspects techniques liés aux algorithmes utilisés pour la classification et l'extraction de connaissances utiles. Dans notre thèse, nous exposons les critères de mesure, de classifications et d'évaluations de données. Toutefois, face au nombre énorme d'informations et de leurs types, il nous a paru intéressant de mettre en évidence les processus de datamining et de Clustering.

Nous commençons par analyser les méthodes exploitant les règles d'associations et la recherche de motifs fréquents. Nous invoquons les relations et les concepts formels ainsi que les méthodes de recherches sélectives et la programmation logique inductive sans oublier les graphes comme support d'information.

Nous mettons en relief les différents types de Clustering envisageables suivant les critères et les contraintes existantes. Ainsi nous identifions les différents paramètres de mesure de similarité et d'évaluations .Par la suite, nous proposons une nouvelle approche qui vise à automatiser l'extraction de la connaissance à partir du texte et de données numériques. Et en comparant quatre modèles, nous mettons en évidence l'importance de l'intervention humaine pour une bonne classification de données textuelles. Nous finissons notre approche par l'étude de la classification à l'aide des algorithmes DBSCAN et DENCLUE, et à travers une application réalisée par nous-même nous faisons des simulations sur des bases de données pour montrer les limites et les paramètres agissant sur la qualité et le choix de la méthode de Clustering.

Mots-clés : Clustering, Data mining, extraction de connaissance, aide à la décision, classification.

Abstract

The need to extract and process information from raw data is one of the most well known goals in data analysis for decision support.

In this context, the researcher discusses various technical aspects related to the algorithms used for the classification and the extraction of useful knowledge. In the present thesis, the researcher exposes the criteria for measuring, classifying and evaluating data. However, given the huge amount of information and its types, it seemed interesting to us to highlight the processes of data mining and clustering.

The study starts by analyzing the methods exploiting rules of associations and the search for frequent patterns. It invokes formal relations and concepts as well as selective research methods and inductive logic programming without forgetting graphs as information support.

In addition to that, it highlights the different types of Clustering that can be envisaged according to the criteria and existing constraints. Thus, the researcher identifies different parameters for measuring similarity and evaluations. Subsequently, I propose a new approach that aims to automate the extraction of knowledge from text and digital data. Furthermore, by comparing four models, I shed light on the importance of human intervention for a good classification of textual data. I end up my approach by studying the classification using the DBSCAN and DENCLUE algorithms, and through an application of mine, I came up with simulations on databases to show the limits and the parameters acting on quality and choice of Clustering method.

Keywords: Clustering, Data mining, knowledge extraction, decision support, classification.

Sommaire

Liste des figures

Liste des tableaux

Liste des algorithmes

Sigles et abréviations

Sigle	Abréviation
ACF	l'analyse de concepts formels
ACP	Analyse en Composantes Principales
ACR	Analyse de concept relationnel
AD	Analyse de Données
AS	Apprentissage Symbolique
CA	Connaissance Actionnable
CAH	Classification Ascendante Hiérarchique
CAP	Classification Ascendante Pyramidale
CDFKM	Cluster Displacement Fuzzy K-means
CLIQUE	Clustering In QUest
CN	Nœud de Concept
CNN	Concepts of Nearest Neighbors
COBWEB	Modèle de la toile d'araignée
Confmin.	confiance minimale
CPCC	Coefficient De Corrélation Cophénétique
CU	Utilite de Categorie
DBI	Davies-Bouldin index
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DENCLUE	DENsity-based CLUstEring
DENCLUE	DENsity-based CLUstEring
DIANA	DIvisive ANAlysis
DNF	Forme Normale Disjonctive
DR	Data Reduction
DT	Delaunay Triangulation
ECD	Extraction de Connaissances à partir de Données
ECT	Extraction de Connaissance à partir de Texte
EM	Espérance-Maximisation
FCM	k-moyennes flou hiérarchique
Foil	First-Order Inductive Learner
freqmin	fréquence minimale
FTP	File Transfer Protocol
GG	Gabriel Graph
GVR	Graphiques de Voisinage dont ceux Relatifs
HFCM	Hierarchical Fuzzy-k-Means».
HFCM	Hierarchical Fuzzy-k-Means
ILP	Inductive Logical Programming
KDD	Knowledge Discovery in Databases
k-means	k-moyennes
k-NN	Plus Proches Voisins
MASK	Méthode d'Analyse et Structuration des Connaissances
MST	Minimum Spanning Tree
MUC	Message Understanding Conferences
NNG	Nearest Neighbor Graph
NTIC	Nouvelles Technologies de l'Information et de la Communication
PLI	Programmation Logique Inductive
PPV	Plus Proche Voisin
PRE	Proportion de Réduction d'Erreur
RF	Reconnaissance de Formes

RNG	Relative Neighborhood Graph
SAHN	Sequential Agglomerative Hierarchical and Nonoverlapping
SAHN	Sequential Agglomerative Hierarchical and Nonoverlapping
SC	Sciences Cognitives
SQL	Structured Query Language
STING	Statistical Information Grid
TIC	Top down Induction of Clustering trees
TILDE	Top- down Induction Logical Decision tree
TM	Texte Mining
UML	Langage de Modélisation Unifié
WaveCluster	Wavelet-Based Clustering

Introduction générale

Le domaine de la recherche opérationnelle et de l'optimisation a enregistré de sérieux progrès dans la deuxième moitié du XX^{ème} siècle à la fois grâce à l'élaboration de procédés numériques, et à la montée fulgurante de performances techniques des ordinateurs.

L'envie de mieux adopter des techniques de recherches opérationnelles dans le domaine de la décision organisationnelle a poussé les chercheurs à adopter une nouvelle conception de l'intervention scientifique, c'est à dire l'aide à la décision. Celle-ci a pour but d'aller au-delà de la simple prise en compte de données "objectives" en essayant d'intégrer les aspects particuliers qui entrent dans un processus de décision. En conséquence, l'objectif de l'aide à la décision se définie comme le suivi du décideur dans la construction d'une décision acceptable. Cela ne se limite pas uniquement à l'utilisation d'une méthode mais exige aussi une aide à la compréhension, à la structuration du problème et la communication entre les différents intervenants.

Dans un tel contexte, les questions qui viennent souvent à l'esprit se présentent comme suit :

- peut-on considérer que la méthodologie à adopter pour le traitement et l'évaluation d'une information, est généralement variable ?
- aussi, quand les données l'admettent « données homogènes, échelles facilement mesurables … etc.», serait-il possible de les ramener à une simple information pouvant constituer le résultat de l'évaluation ? certes, l'adoption d'une telle approche reste douteuse lorsque les aspects à prendre en compte sont irréguliers, surtout quand ils sont opposés et mal recueillis. Si tel est le cas, peut-on étendre l'analyse à toutes les situations ?

Ainsi , il devient possible de reformuler la problématique qui consiste à développer systématiquement des groupes d'informations dans les divers domaines d'intérêt ;à savoir ,la santé pour ordonner suivant des critères spécifiques les maladies, l'environnement pour identifier et prédire l'évolution météorologique, les produits financiers pour évaluer les fluctuations des actions et obligations, et bien d'autres .

Ce travail se veut une contribution à la recherche de techniques répondant à ce qui précède c'est à dire, pour être plus précis l'extraction de connaissances à partir de données réelles. A cet égard, le présent mémoire est organisé comme vient :

- le premier chapitre met en valeur, à travers un résumé bibliographique, les travaux antérieurs se rapportant à l'extraction de la connaissance utile à partir de données préexistantes.
- le second chapitre met le point sur différentes notions, commençant par celle des données, d'information et de connaissance par la suite.
- le chapitre troisième expose les techniques de mesure et de structuration des données à travers les processus de classification et de Clustering.
- le chapitre quatrième présente quelques algorithmes permettant la représentation des données sous forme de clusters dans le but de pouvoir appliquer ces techniques de fouilles de données à la recherche de connaissances exploitables. Les facteurs et les critères d'évaluation des différentes méthodes de Clustering y sont également traités.
- Le chapitre cinquième fait l'analyse des techniques d'extraction de connaissances à partir du texte.

- Le sixième chapitre traite la recherche d'informations numériques par évaluation de quatre modèles.
- Le chapitre septième étudie les limites d'exploitation des algorithmes DBSCAN et DENCLUE à travers une application informatique.
- Le huitième et dernier chapitre sanctionne tout ce qui précède par un modèle pratique sur les techniques d'extraction de connaissances à partir de données à travers une application opérationnelle sur les techniques de Clustering.
- Enfin nous clorons le manuscrit par une conclusion générale récapitulant l'essentiel des résultats obtenus suite aux investigations menées dans le cadre de ce travail de thèse et dans laquelle une partie « appelée perspective » est définie.

Sans oublier de signaler que , dans un contexte parfois difficile et limité par le temps , l'importance de la réalisation d'un système performant et crédible ne peut en aucun cas défier l'intervention humaine et son expérience, pour une bonne prise de décision.

En outre, cette expérience humaine est sensée être conservée et préconisée à travers l'intelligence artificielle. Étant donné que l'homme est un facteur incontrôlable, et dont les qualités peuvent se perdre à travers le temps, on a pensé à automatiser via ce travail de thèse les taches liées au traitement d'informations et aux exploitations des algorithmes de Clustering, les résultats sont lisibles dans le chapitre huit.

Chapitre I. Revue bibliographique sur les travaux antérieurs se rapportant à l'extraction de la connaissance utile à partir de données préexistantes.

I.1. Introduction : histoire de la théorie de l'information

Depuis l'antiquité le savoir et l'information étaient considérés comme secrets précieux pour le développement de l'humanité, à chaque obstacle rencontré, on cherche à trouver une solution en se basant sur l'expérience des ancêtres. A cet égard, nous allons rappeler les travaux réalisés par différents chercheurs pour parvenir à notre objectif principal, à savoir l'extraction et le traitement des informations à partir de données brutes.

Ainsi, Euler (1736) [1]a eu l'occasion de résoudre le problème des sept ponts liant différentes parties de la ville de Königsberg. L'idée a été de savoir s'il est possible, à partir d'un point de départ au choix, de passer une et une seule fois par chaque pont, et de revenir à son point de départ, et ç'a été à travers la théorie des graphes qu'il a pu trouver une solution . Cette théorie est la base de représentation d'informations sous forme de graphes pour pouvoir extraire la connaissance.

Figure I.1 : Représentation graphique des sept ponts de la ville de Königsberg au temps d'Euler.

I.2 Mesure de la similarité de données à travers les calculs mathématiques :

Deux siècles après, le botaniste Paul Jaccard (1901)[2] a repris, de façon développée, la notion d'indice de similitude, autrefois appelé coefficient de communauté. Celui-ci permet de comparer la similarité et la distance des échantillons dans différents domaines d'applications comme la biogéographie afin de les classer selon certains critères. Vers les années 1920, le même auteur s'était engagé dans une dispute avec le botaniste et phyto-géographe finlandais Alvar Palmgren sur l'interprétation du rapport espèces-genre, comme preuve d'exclusion compétitive, détenue par Jaccard ou attribuable à l'échantillonnage aléatoire.

Par la suite, la théorie des graphes fini et infini peut être attribuée à König(1936)[3] dans laquelle il a lancé les premiers concepts de base.

Juste après la deuxième guerre mondiale, Shannon et Weaver (1948)[4] ont préconisé une théorie mathématique de la communication, dans laquelle ils ont indiqué l'importance de l'extraction de la connaissance dans une information codée en temps de guerre. Cette recherche informatique simultanée confondait la cryptographie et l'intelligence : c'était la Crypto-

intelligence. Elle permet de récupérer des informations signifiantes et utiles à l'exploitation, à partir de communications apparemment aléatoires ou désordonnées.

Grace aux travaux publiés par Shannon, la "théorie de l'information" a pris naissance. Lors de ses recherches sur la cryptographie chez « Bell Labs », il a développé la "théorie mathématique" au cours de ses rencontres avec Turing en plein guerre. Les deux hommes ont souvent déjeuné ensemble et discuté des choses comme le cerveau humain et la machine de calcul. L'intérêt de Shannon pour une théorie de la communication a précédé la guerre, mais il a reconnu la cryptographie comme permettant et stimulant ce qu'il a appelé les «bons aspects» de la théorie de l'information .Il a développé des méthodes et des formules adoptées après par la linguistique computationnelle, la cryptographie moderne et l'informatique numérique. Shannon a montré comment les modèles, codes et informations, pourraient être extraits de bruit mécanique (chiffré), naturel ou autre. Le travail de Shannon a fourni alors d'autres éléments clés dans la crypto-intelligence.

I.3 Idée de classifier en faisant appel à la notion de voisinage.

En 1967 Cover & Hart [5] ont abordé le problème de la classification du modèle du plus proche voisin(PPV), la recherche de voisinage est utilisée dans de nombreux domaines, et c'est un problème de classification connu dans le monde. L'idée est que «les choses qui se ressemblent doivent être semblables ». Ils ont tenté de desseller quelques propriétés sans avoir recours à des distributions de cette règle de classification, c'est à dire des propriétés qui sont vraies indépendamment des distributions conjointes des catégories sous-jacentes et des observations.

Dix ans plus tard et dans le même cadre de l'analyse de données, Schön (1983) [6] a élaboré la théorie de «La réflexion-en-action et la réflexion-sur-l'action ». Il a définit la réflexion sur l'incident soit en exploitant ce dernier, soit en réfléchissant à la façon dont on ferait les choses différemment à l'avenir. C'est un outil utile dans les disciplines où le professionnel doit réagir à un événement au moment où il se produit - plutôt que de penser à ce qui s'est passé et de faire des changements plus tard. Sa description du processus est comme ceci : «Quand quelqu'un réfléchit dans l'action, il devient un chercheur dans le contexte de la pratique. Il ne dépend pas des catégories ou de la théorie et de la technique établies, mais construit une nouvelle théorie du cas unique ».

Naturellement, des incidents surprenants se passent parce que, dans une nouvelle condition, on n'exploite pas les connaissances acquises dans d'autres situations similaires, parce qu'elles ne conviennent pas à la situation actuelle. Ainsi, pour s'en sortir, plutôt que d'utiliser des idées préconçues sur ce qui devrait être fait dans une situation particulière, celui qui réfléchit décide ce qui fonctionne le mieux à ce moment-là pour cet événement ou cet incident unique.

Alternativement, la réflexion sur l'action suppose de penser sur la façon dont la pratique peut être développée après l'événement, c'est ce que Schön a dit «Nous réfléchissons à l'action, repensons à ce que nous avons fait pour découvrir comment notre savoir-être a pu contribuer à un résultat inattendu ». Ceci se traduit par le fait qu'on cherche une solution après l'événement sur la façon dont la connaissance d'événements similaires antérieurs a pu conduire à l'incident inattendu et à ce qu'on doit changer pour l'avenir.

I.4 Exploration de données via l'apprentissage inductif :

Dans la même année, Michalski (1983)[7] a élaboré une théorie et une méthodologie de l'apprentissage inductif. Il s'agit d'une recherche heuristique à travers un espace de descriptions symboliques, généré par l'application de certaines règles d'inférence aux énoncés d'observation

initiaux. Les règles d'inférence comprennent des règles de généralisation, qui effectuent des transformations généralisatrices sur les descriptions, et des règles déductives conventionnelles préservant la vérité (règles de spécialisation et de reformulation). L'application des règles d'inférence aux descriptions est limitée par la connaissance de l'arrière-plan du problème et guidée par des critères évaluant la qualité des assertions inductives générées. Sur la base de cette théorie, une méthodologie générale d'apprentissage des descriptions structurelles à partir d'exemples, appelée étoile, a été décrite et illustrée par un problème du domaine de l'analyse des données conceptuelles.

Fowlkes et Mallows (1983)[8], ont conçu une méthode pour comparer deux clusters hiérarchiques. C'est une évaluation externe qui détermine la similarité entre deux regroupements (les clusters obtenus après un algorithme de Clustering). Cette mesure de la similarité pourrait être soit entre deux groupements hiérarchiques ou une classification en clusters et celle de référence. Ainsi, une valeur plus élevée pour l'indice Fowlkes-Mallows indique une plus grande similarité entre les clusters et les classifications de référence.

En effectuant une analyse de clusters d'une série de données multi variées p-dimensionnelles, on mesure la similarité de deux ou plusieurs regroupements hiérarchiques du même ensemble d'objets. Par exemple, on peut étudier l'effet de l'utilisation de différentes mesures de similarité, ou de différents algorithmes de regroupement, ou de données provenant de deux sources différentes. Informellement, on inspecte les regroupements pour déterminer les groupes importants, la structure des groupes d'un arbre peut être comparée à celle des groupes d'un second arbre. Les correspondances indiquent une similarité entre les deux arbres. Cette méthode de comparaison de deux groupements peut être extrêmement laborieuse et longue et ne permet aucune mesure de l'efficacité des comparaisons. Fowlkes et Mallows proposent une méthode de comparaison de deux regroupements hiérarchiques qui donne une mesure numérique du degré de similarité. Non seulement la méthode fournit une comparaison entre deux regroupements, mais elle est également utile comme outil d'étude de la classification hiérarchique en général.

Breiman, Friedman et al. (1984)[9], ont proposé les arbres de classification et de régression, comme introduction aux arbres de décision deux ans après. Les arbres de classification incluent les modèles dans lesquels la variable dépendante (la variable prédite) est catégorique. Les arbres de régression incluent ceux dans lesquels le module « arbre » est continu. Au sein de ces types d'arbres, le module « arbre » peut utiliser des prédicteurs catégoriels ou continus, selon que l'instruction de tri inclut une partie ou la totalité des prédicteurs. Pour l'un des modèles, une variété de fonctions de perte est disponible. Chaque fonction de perte est exprimée en termes de qualité d'ajustement de la proportion de réduction d'erreur (PRE).

Dans sa publication « Induction of decision trees », Quinlan (1986)[9] a montré que la technologie de construction de systèmes basés sur la connaissance par inférence inductive à partir d'exemples, est efficace dans plusieurs applications pratiques. Il a traité une approche de synthèse des arbres de décision qui a été utilisée dans divers systèmes. Les résultats d'études récentes montrent comment la méthodologie peut être changée pour analyser des informations bruyantes et ou imprécises. Les arbres de décision à partir d'exemple sont puissants, leur exploitation dans les systèmes et outils commerciaux actuels est un succès remarquable même avec des données incomplètes. Les arbres de décision générés par ces systèmes sont d'exécution rapide et peuvent être très précis. Ils laissent beaucoup à désirer en tant que représentations de la connaissance. Les experts, à qui l'on montre de tels arbres pour des tâches de classification dans leur propre domaine, peuvent souvent identifier une petite utilisation familière. C'est ce manque de familiarité

(et peut-être un manque sous-jacent de modularité) qui constitue le principal obstacle à l'utilisation de l'induction pour construire de grands systèmes experts.

Fisher(1987)[10], dans son article « Machine Learning & Knowledge Acquisition, Improving Inference Through Conceptual Clustering » a montré que le regroupement conceptuel est très utile pour résumer et expliquer les données. Cependant, la formulation récente de ce paradigme a permis peu d'exploration du regroupement conceptuel comme moyen d'amélioration de la performance. De plus, les travaux antérieurs sur ce regroupement n'ont pas traité clairement des contraintes imposées par des types d'informations réelles. L'auteur a présenté COBWEB, comme un système de classification conceptuel qui organise les données de manière à maximiser la capacité d'inférence. COBWEB schématise les clusters sous forme d'une distribution de probabilité sur l'espace des valeurs d'attributs, générant alors un arbre de classification hiérarchique, dans lequel les nœuds intermédiaires définissent des sous-concepts. Ce processus est incrémentiel et économique en termes de calcul. Il peut donc être appliqué de manière flexible dans divers domaines. Alors que le système utilise une fonction d'évaluation cohérente avec les préférences dans le classement humain, il ne devrait pas être considéré comme un modèle cognitif, mais comme une méthode de regroupement d'usage général. COBWEB cherche explicitement à optimiser la classification sans montrer comment ces résultats apparaissent en évolution. Cependant, Fisher a décrit une progéniture de ce système qui peut être considérée comme un modèle cognitif et qui explique certains phénomènes psychologiques, y compris les effets de niveau de base et de typicalité (Capacité d'un produit ou d'un service à représenter une catégorie).

Deux ans après, Fisher et al. (1989)[11] ont sorti un article « Models of Incremental Concept Formation », dans lequel ils ont traité trois types de modèles de formation de concepts incrémentaux. Ainsi, le premier permettait la reconnaissance et l'acquisition de concepts composites impliquant plusieurs parties, le deuxième générait des plans en utilisant de l'analyse des moyennes-finies et en acquérant l'expertise du plan à partir de traces de solution réussies et le troisième traitait de l'exécution, de l'acquisition et du perfectionnement des compétences motrices. Tous les trois utilisent un algorithme appelé CLASSIT de Gennari et al. (1989)[12], comme sous-programme pour retrouver et acquérir des concepts probabilistes. Les travaux futurs se concentreront sur l'amélioration et l'intégration de ces composantes, en incorporant des pulsions pour la génération de nouveaux concepts, et en développant un mécanisme de groupement.

Dans leur étude se rapportant à la généralisation des arbres hiérarchiques, BERTRAND et DIDAY (1990)[13], ont montré que à l'instar les arbres hiérarchiques, les représentations pyramidales constituent une extension naturelle, ce sont des ensembles de parties (appelées classes ou paliers) de l'ensemble des objets à classer. Cependant la classification pyramidale permet d'obtenir des relations plus complexes entre les classes. En particulier deux classes non disjointes ne sont pas nécessairement assemblées, comme c'est le cas pour une hiérarchie. Une propriété intéressante des représentations pyramidales est leur capacité à produire un petit nombre d'ordres qui respectent les contraintes de proximité entre les objets à classer.

I.5 Histoire des données textes pour l'extraction de la connaissance

Pour extraire la connaissance à partir de texte, Lehnert et al. (1991)[14] ont décrit l'outil « CIRCUS », l'analyseur conceptuel qui produit des représentations de cadres de cas sémantiques pour des phrases d'entrée. Bien que l'espace ne permette pas de donner une description technique complète de CIRCUS, ils ont essayé de transmettre un certain sens de l'analyse de la phrase via ce dernier. Un tel outil n'utilise aucune grammaire syntaxique et ne produit aucun arbre d'analyse,

car en analysant d'une phrase, il utilise plutôt des connaissances syntaxiques indexées lexicalement pour segmenter le texte entrant en syntagmes nominaux, phrases prépositionnelles et phrases verbales. Ces constituants sont stockés dans des tampons globaux qui suivent les sujets, les verbes, les objets directs et les segments prépositionnels d'une phrase. Les auteurs ont limité le contenu du tampon à des constituants simples avec un sens très local de la phrase, les composants de plus grande taille ne sont pas explicitement stockés par la composante syntaxique de CIRCUS. Puisque les tampons syntaxiques sont liés à des fragments de phrase, un mécanisme pour gérer la sémantique prédictive responsable de l'établissement des attributions de rôles. Les cadres de cas sémantiques sont activés par des définitions de nœud de concept (CN), et chaque définition CN peut être déclenchée par un ou plusieurs éléments lexicaux. Associé à chaque slot dans un CN sont à la fois des contraintes dures et douces. Une contrainte dure est un prédicat qui doit être satisfait, tandis qu'une contrainte souple définit une référence plutôt qu'une exigence absolue. Lorsqu'une instanciation CN répond à certains critères établis par la définition CN, CIRCUS gèle cette trame de cas et la transmet comme sortie de l'analyseur de phrases. Une seule phrase peut générer un nombre arbitraire d'instanciations de cadres de cas en fonction de la complexité conceptuelle de la phrase et de la disponibilité des définitions CN pertinentes dans le dictionnaire.

En procédant autrement, Rakesh Agrawal et Ramakrishnan Srikant (1994)[15], ont examiné le problème de la découverte de règles d'association entre les éléments d'une grande base de données de transactions de vente. Ils ont présenté deux nouveaux algorithmes pour résoudre ce problème. L'évaluation empirique montre que ces algorithmes surpassent les algorithmes connus par des facteurs allant de trois pour les petits problèmes à plus d'un ordre de grandeur pour les grands problèmes. Ils ont montré également comment les meilleures caractéristiques des deux algorithmes proposés peuvent être combinées en un algorithme hybride, appelé « AprioriHybrid ». Les expériences de mise à l'échelle montrent parfaitement que cet algorithme s'étend linéairement avec le nombre de transactions. AprioriHybrid possède également des propriétés de mise à l'échelle en ce qui concerne la taille de la transaction et le nombre d'éléments dans la base de données.

Par ailleurs, Lavrac et Džeroski (1996)[16], ont montré l'importance de la programmation logique inductive (PLI) dans le déploiement des techniques d'exploitation de données relationnelles et surtout celles stockées dans plusieurs tables, bien que l'apprentissage des règles récursives et l'utilisation des prédicats a acquis beaucoup d'intérêt au sein de la communauté (PLI), peu de résultats pratiques existent jusqu'à présent.

Au même titre que ces derniers auteurs, Nada Matta et al. (1996)[17], ont inventé la Méthode d'Analyse et Structuration des Connaissances (MASK). C'est un processus global de gestion des connaissances dans l'entreprise. Il comprend toutes les étapes qui permettent la capitalisation, le partage et l'évolution du capital de l'organisation... Les techniques d'ingénierie des connaissances sont souvent utilisées pour capitaliser les connaissances. Ces techniques devraient être adaptées et évoluées en considérant les principaux objectifs de la gestion des connaissances, qui sont la construction et l'utilisation de la mémoire organisationnelle.

Parallèlement, Usama Fayyad (1996) [18] a traité les problèmes liés à la réduction de données de grandes tailles tout en préservant l'intérêt des techniques de classification. Dans l'exploration de données, leur réduction est considéré comme une tâche principale, il a proposé une approche générale algébrique et par la suite a développé un système de réduction de données appelé (DR- Data Reduction) réduisant ainsi les ensembles de données mais épargnant les structures de classification qui présentent un intérêt. Le choix d'attributs et la discrétisation de

ceux continus pouvaient être représentés en tant que sous-produit de la réduction des données, le système pouvait traiter aisément les valeurs manquantes ou incomplètes.

Juste après, Blockeel et DeRaedt(1997)[19], ont donné une approche de Clustering qui adapte l'induction descendante aux classifications utilisant les arbres de décision. Ils ont utilisé la notion d'instance implémentée dans le système TIC (Top down Induction of Clustering trees) qui donne une représentation arborescente logique de premier ordre de la programmation logique inductive.

L'algorithme CLIQUE revient à Agrawal et al. (1997)[20]. Il identifie des clusters denses dans des sous-espaces de dimensionnalité maximale, leur génère des descriptions sous la forme d'expressions DNF (Forme Normale Disjonctive) qui sont minimisées pour faciliter la compréhension. Il produit des résultats identiques quel que soit l'ordre dans lequel les enregistrements d'entrée sont présentés et ne présume aucune forme mathématique spécifique pour la distribution des données.

Dans la même année, Grishman (1997)[21], a abordé l'extraction de l'information à partir de grands volumes de texte. Sa méthode se base sur la récupération de documents, l'étiquetage de termes particuliers et la création d'une représentation structurée (telle qu'une base de données) d'informations à partir du texte.

Pour comprendre les messages, au cours des dernières décennies, grâce à une série de conférences ces auteurs ont utilisé des versions simplifiées des tâches d'extraction d'informations tout au long de documents, comme les événements terroristes. Pour chaque événement, ils ont analysé et conçu un système permettant de déterminer le type d'attaque (bombardement, incendie criminel, etc…), la date, le lieu, l'auteur (si indiqué), les cibles et les effets sur les cibles. Et c'était leur principale motivation vue le risque répandu du terrorisme dans le monde. Sachant que l'extraction d'informations est une tâche limitée par rapport à la "compréhension du texte intégral". Dans cette dernière, on essaie de montrer de manière explicite toutes les informations contenues dans un texte, par contre, dans l'extraction d'information, on délimite d'avance, dans le cadre de la spécification de la tâche, la gamme sémantique de la sortie : les relations qu'on va représenter et les charges admissibles dans chaque tranche d'une relation.

Cette année-là, Kumar Singh (1997)[22] est arrivé avec l'algorithme STING (Statistical Information Grid), il quantifie l'espace objet en un nombre fini de cellules formant ainsi une structure de grille sur laquelle toutes les opérations de Clustering s'exécutent, de même la structure de STING répond efficacement à diverses requêtes à partir de base de données et même si les informations statistiques ne sont pas suffisantes, un ensemble de réponses possibles peuvent être générées.

Un tel algorithme a également été analysé par Wei Wang et all (1997)[23], ils ont proposé une approche hiérarchique pour l'exploration de données spatiales, qui se basait sur une grille d'informations statistiques afin de réduire davantage les coûts. Le concept est de capturer des informations statistiques associées à des cellules spatiales de telle manière que des classes entières de requêtes et de problèmes de regroupement puissent être résolues sans voir les objets individuellement. En théorie, toute en étant confirmée par des études empiriques, cette approche a montré son efficacité, surtout quand l'ensemble de données est très grand.

En 1998, Brooking [24], dans son livre a de son côté attiré l'intention sur l'importante de la connaissance et du savoir acquis à partir des informations de l'entreprise. Pour preuve, le nombre d'entreprises, ayant licencié des employés à la suite de la restructuration pour les recruter comme consultants, a augmenté. La raison en est que les connaissances, le savoir-faire, l'expérience et les compétences qu'ils possèdent sont très précieux. Elle a mis en valeur la mémoire de l'entreprise comme facteur stratégique pour la gestion des connaissances en

montrant au gestionnaire innovant comment explorer l'actif incorporel de son entreprise, comment identifier les connaissances au sein de sa culture organisationnelle et regarder vers le partage des connaissances, et ceux, grâce à l'utilisations d'outils et systèmes informatiques variés pour le stockage et le traitement objectifs de données dans le but de répondre aux besoins des entreprises .

I.6 Modélisation de données à partir de graphique de voisinage

En 1998, Jerzy et al.[25] ont traité de façon détaillée les algorithmes pour les graphiques de voisinage dont ceux relatifs (GVR) tout en s'intéressant aussi autres membres de cette famille de graphiques. De nombreuses questions intéressantes sont restées ouvertes. Parmi elles, le problème de limites serrées pour le nombre d'arêtes du (GVR) dans R3. La limite supérieure la mieux établie est super linéaire. Aussi, seule une borne inférieure linéaire triviale est connue. Dans ce contexte, il a été intéressant de développer des algorithmes optimaux ou sensibles à la sortie pour le GVR à 3 dimensions ainsi que pour celui à 2 dimensions. Il a été constaté qu'étant donné la complexité des 0-squelettes en trois dimensions, plus de recherche est méritée. Un autre domaine qui a nécessité un complément d'étude a été la question de la reconnaissance des graphes de proximité, en d'autre terme, pour une classe de graphes de proximité et un graphe G, G appartient-il à cette classe ? Les résultats déjà connus concernaient la triangulation de Delaunay et les facteurs f des ensembles de points dans le plan.

Compte tenu des applications généralisées des graphes de voisinage à la morphologie computationnelle, à l'analyse géographique et à l'analyse de formes, la conception d'algorithmes robustes est une tâche particulièrement importante. En l'occurrence, pour les graphes de voisinage, il reste difficile d'avoir des algorithmes forts et stables lord de leur implémentation numérique.

Ganter et al. (1999)[26] ont développé la mise à l'échelle conceptuelle dans le cadre de l'analyse conceptuelle formelle, c'est une théorie basée sur une mathématisation des hiérarchies conceptuelles. Dans leur étude, ils ont effectué une enquête introductive sur l'échelle conceptuelle qui se concentre sur les échelles de type ordinal. Dans un premier temps, ils ont fait un rappel des notions de base, les résultats de l'analyse conceptuelle formelle et la démonstration par un exemple. Ensuite, ils se sont focalisés sur la mesurabilité par des échelles standardisées de type ordinal, sachant que la mesure conceptuelle est souvent discutée. Ces idées ont été appliquées à la mise à l'échelle des contextes de données pour trouver des hiérarchies conceptuelles pour les données, afin d'introduire et d'étudier une notion générale de dépendance entre attributs qui couvre des notions particulières comme la dépendance fonctionnelle et linéaire.

Peu de temps après, Weiss et Jordan (1999)[27] se sont intéressés au problème du regroupement automatique et de la segmentation des images propres à la vision par ordinateur. Certains auteurs se sont distingués dans ce domaine grâce à l'utilisation de méthodes basées sur les vecteurs propres de la matrice d'affinité. Ces approches sont extrêmement attrayantes dans la mesure où elles sont basées sur des algorithmes de décomposition simple à stabilité bien contrôlée. Néanmoins, l'utilisation des compositions propres dans le contexte de la segmentation est loin d'être bien comprise. Les auteurs ont fait alors un traitement unifié de ces algorithmes, et ont montré les liens étroits entre eux tout en mettant en évidence leurs caractéristiques distinctives. Ils ont validé après leur résultats sur des vecteurs propres de matrices de blocs qui ont permis d'analyser les performances de ces algorithmes pour des paramètres de regroupement simples. Ils ont illustré leur analyse avec des résultats sur des images réelles et synthétiques. L'homme qui perçoit une scène peut souvent facilement la diviser en segments ou groupes cohérents. Il y a eu énormément d'efforts déployé pour atteindre le même niveau de performance

en vision par ordinateur. Dans de nombreux cas, ceci est réalisé en associant à chaque pixel un vecteur de caractéristiques (par exemple, couleur, mouvement, texture, position…) et en utilisant un algorithme de regroupement ou de regroupement sur ces vecteurs de caractéristiques.

En l'an 2000 KÄRKKÄINEN et al. [28] se sont penchés sur le problème de la classification en faisant appel à l'indice Davies-Bouldin comme critère d'optimisation. Le problème a été de partitionner un ensemble de données de N vecteurs en M groupes de sorte que la valeur de l'indice de Davies-Bouldin est minimisée. L'indice diffère de l'erreur quadratique moyenne en ce sens qu'il prend également en compte la distance entre les vecteurs de code. Cela mène au problème que la définition des vecteurs de code en tant que centroïde des clusters ne donne pas le placement optimal pour minimiser l'indice de Davies-Bouldin. Ils ont dérivé une formule pour optimiser l'emplacement des vecteurs de code en utilisant l'indice de Davies-Bouldin. La formule a ensuite été appliquée dans le Clustering lors de la résolution du nombre correct de clusters. Les résultats étaient théoriquement bien argumentés mais leur impact dans l'application pratique n'était pas significatif. En expérimentant, ils ont pu tirer les conclusions provisoires suivantes :

- il semblait que le DBI (Davies-Bouldin index) fonctionne bien lorsque les clusters sont clairement séparables. La sous-optimalité de l'emplacement du centroïde ne semble pas être un problème sérieux lorsqu'ils ont utilisé la stratégie de recherche par force brute dans le Clustering. Le problème ne se pose que dans des situations extrêmes où les clusters se chevauchent beaucoup.
- Deuxièmement, il est possible que la formule donnant la localisation localement optimale d'un seul vecteur de code ne soit pas suffisante. Au lieu de cela, un placement globalement optimal des vecteurs de code pourrait résoudre le problème mais reste non réalisable. De sur croît, il n'est pas certain que le DBI lui-même soit la mesure correcte car il comporte des éléments heuristiques. Par exemple, il considère pour chaque cluster uniquement le cluster le plus proche dans la mesure. D'autres études seraient donc nécessaires pour parvenir à une réponse concluante au problème.

Lord de de la 14ème conférence annuelle sur les processus d'information des systèmes neuronaux, Dietterich et al. (2001)[29] ont présenté les résultats de travaux réalisés par des groupes d'étudiants dans le domaine. L'évènement couvre un large éventail de sujets relatifs au calcul neuronal, y compris les sciences cognitives, théorie d'apprentissage, algorithmes et architectures, implémentations, traitement de la parole et du signal, traitement visuel, applications, contrôle et navigation (qui inclut l'apprentissage par renforcement). Cette dimension d'actualité a reçu le soutien de collaborateurs ayant des racines intellectuelles dans divers domaines : neurosciences, sciences cognitives, statistiques, mathématiques, ingénierie, informatique, psychologie, finance et physique. Ce fut l'avènement des réseaux de neurones.

Kramer et al. (2001)[30] ont réussi à transformer une représentation relationnelle d'un problème d'apprentissage en une représentation propositionnelle (basée sur des caractéristiques, une valeur d'attribut…). Ce type de changement de représentation est connu sous le nom de propositionalisation. Considérant une telle approche, la construction de caractéristiques peut être découplée de la construction du modèle. Il a été démontré que dans de nombreuses applications d'exploration de données relationnelles, ceci peut se faire sans perte de performance prédictive. Après avoir passé en revue les approches de propositionnalisation, générales et dépendantes de la littérature, ils ont préconisé une extension de celle Linus, permettant ainsi au système de traiter des variables locales non-déterminées.

Dans sa thèse de doctorat, Berasaluce (2002) [31] a présenté les résultats des différentes expérimentations d'extraction de connaissances dans les bases de données de réactions chimiques.

Elle a appliqué les techniques de fouille à des données très particulières qui sont d'origine structurale. Elle a pu constater à quel point les transformations des données effectuées induisent la perte de l'information structurale. La modélisation des méthodes de construction de cycles a été particulièrement bénéfique pour la compréhension et la prise en compte des cas particuliers rencontrés. Elle a mis en valeur l'importance de l'utilisation des méthodes de fouille de données appliquée à l'exploitation de données biologiques et génétiques dans les interactions génotype-phénotype intermédiaire et maladies cardiovasculaires.

Pour leur part, Vlachos et al. (2002)[32] ont mis en œuvre un algorithme modifié dans le but de réaliser le Clustering de séries chronologiques. Il a été remarqué que, bien qu'il y ait eu beaucoup de recherches sur le Clustering en général, la plupart des algorithmes classiques d'apprentissage automatique et d'exploration de données fonctionnent mal pour les séries temporelles en raison de leur structure unique. En particulier, la forte dimensionnalité, la corrélation très élevée des caractéristiques et généralement la grande quantité de bruit qui caractérise les données de séries chronologiques constituent un grand défi. Ils ont affronté celui-ci en introduisant une nouvelle version de l'algorithme de Clustering k-Means pour les séries chronologiques. Celui-ci fonctionne en exploitant la propriété multi-résolution des ondelettes. En particulier, un regroupement initial est effectué avec une représentation très grossière des données. Les résultats obtenus à partir de ce Clustering sont utilisés pour initialiser un Clustering à un niveau d'approximation un peu plus fin. Ce processus est répété jusqu'à ce que les résultats de regroupement se stabilisent ou de sorte que l'approximation coïncide avec les données brutes. Per ailleurs, leur approche parait posséder deux autres propriétés très peu intuitives. La qualité de la mise en clusters est souvent meilleure que celle de l'algorithme par lots, et même si l'algorithme est exécuté jusqu'à la fin, le temps d'exécution consacré est généralement beaucoup plus court que celui de l'algorithme original. Ils ont démontré empiriquement l'applicabilité de ces propriétés, remarquables, sur des expériences complètes effectuées sur plusieurs ensembles de données réelles accessibles au public.

En 2004, Nakache et al. [33] ont abordé la classification comme une branche de l'analyse statistique multidimensionnelle descriptive, qui a fait l'objet de nombreuses publications. Ils ont analysé une large gamme de méthodes de classification, des plus classiques aux plus récentes. Cela est très utile pour les praticiens confrontés, à des données multidimensionnelles importantes et exerçant dans de nombreux domaines (sciences sociales, psychologie, médecine, météorologie, industrie, marketing, documentation…), et pour les enseignants chercheurs, ingénieurs et étudiants comme support de cours dans les universités et les grandes écoles.

La notion de k-means flou appartient, elle, à CHANG1 et al. (2011)[34], qui l'ont utilisé pour connaitre le déplacement du centre de cluster entre des processus itératifs successifs, afin réduire la complexité inhérente au calcul de l'algorithme de Clustering k-means classique. La méthode proposée, appelée CDFKM (Cluster Displacement Fuzzy K-means), classe d'abord les centres de clusters en groupes actifs et stables. Aussi, ignore-t-elle les calculs de distance pour les clusters stables dans le processus itératif. Pour accélérer la convergence de CDFKM, ils ont développé un algorithme pour déterminer les centres de cluster initiaux pour CDFKM. Comparé à l'algorithme de Clustering k-means classique, la méthode proposée présente l'avantage de réduire le temps de calcul d'un facteur de 3,2 à 6,5 en utilisant les ensembles de données générés à partir de la séquence de Gauss Markov. Leur algorithme peut réduire de 38,9% à 86,5% le nombre de calculs de distance de celui classique en utilisant les mêmes ensembles de données.

Vue le besoin pour les méthodes automatisées pour apprendre les caractéristiques générales des interactions d'une classe de ligands avec son ensemble divers de récepteurs

protéiques, MUGGLETON (2012) [35] a usé d'un nouveau système ILP (Inductive Logical Programming), appelé ProGolem, et a démontré sa performance sur les caractéristiques d'apprentissage des interactions entre les protéines et les hexoses. Il a exploité une approche d'apprentissage automatique appropriée et la programmation logique inductive (ILP), qui génère automatiquement des règles compréhensibles en plus de la prédiction. Le développement de systèmes ILP capables d'apprendre les règles de complexité requises pour les études sur la structure des protéines reste un défi. Comme résultats, les règles induites par ProGolem détectent les interactions induites par les aromatiques et par les résidus polaires planaires, en plus de caractéristiques moins communes telles que le sandwich aromatique. Il a montré dans son étude que la programmation logique inductive implémentée dans ProGolem peut dériver des règles donnant des caractéristiques structurelles des interactions protéine / ligand. Plusieurs de ces règles sont conformes aux descriptions de la littérature. D'autre part, le modèle de ProGolem a une précision prédictive croisée 10 fois supérieure et dispose d'un niveau de confiance de 95% par rapport à un autre système ILP précédemment utilisé dans l'étude des interactions protéine / hexose.

Au début de 2013, Bernardo et al. [36] ont repris l'étude des graphes de voisinage déjà effectuée par Jerzy (1998). Les auteurs ont défini de la règle de proximité spécifique de plusieurs graphes avec l'ensemble de sommets U, dans lequel deux sommets sont adjacents, comme étant la distance du chemin le plus court en un graphe, G, générant certains des graphiques de proximité les plus courants dans les espaces euclidiens. Ils ont ainsi démontré les propriétés de base des graphes en fournissant des algorithmes pour faire le calcul.

Conjointement, Kriege et al. (2013)[37] ont présenté un nouvel algorithme de Clustering SAHN heuristique qui se sert des propriétés de mesures de distances métriques arbitraires dans un espace linéaire et supportant la liaison médiane et centroïde, afin de réduire, le mieux possible , le temps d'exécution dans la limite d'une complexité de l'ordre de O (n log n) pour n entrées. Cet algorithme s'est avéré adéquat pour des mesures de distances coûteuses et n'a besoin que d'un nombre linéaire d'opérations de calcul de distance exacte.

Deux années plumard, Chabanet et al. (2015) [38] ont fait l'analyse des données aberrantes et extrêmes ainsi que des outliers, ils se sont intéressés aux méthodes paramétriques les plus connues comme les tests, l'analyse de variance ou la régression linéaire. Ils ont pu exploiter les graphiques pour la détection des points extrêmes, la représentation dépend de la structure des données et du nombre d'observations, ils ont fait des transformations et des tests statistiques pour s'assurer de la normalité des distributions et avoir une bonne appréciation graphique.

Récemment, RAKOTOMALALA(2016) [39], a entrepris une étude d'algorithme des k-médoïdes pour la classification automatique utilisant l'utilitaire Tanagra, il a explicité les techniques de classification par partition exploitant la distance euclidienne, la distance de Manhattan et autre, afin de réduire énormément l'impact des points atypiques. La visualisation à l'aide de cet utilitaire est intéressante pour mieux expliquer le déroulement de l'algorithme et la visualisation des clusters résultants.

Tout récemment, Platoš (2017)[40] a présenté une analyse de Clustering à base de densité. L'idée a été d'identifier les régions denses à grain fin dans les données, leur regroupement produit des clusters de forme arbitraire. C'est un algorithme de classification hiérarchique appelé DBSCAN (Density-Based Spatial Clustering of Applications with Noise) basé sur la densité. Les groupes sont formés de régions de grilles denses de connectivité adjacente, puisqu'elles partagent un côté et un coin communs. Les régions connectées peuvent être trouvées en traversant en premier ou en profondeur à l'aide d'un modèle basé sur un graphique. L'algorithme donne de

bonnes représentations avec les points de bruit. Le contour des clusters est plus lisse, alors que les régions rectangulaires sont substituées par une zone sphérique identifiée par le rayon.

I.7 Conclusion

Le désire de classer pour réduire et mieux contrôler s'est développé progressivement vers l'ambition d'automatiser la classification pour concevoir et, pourquoi pas, prévoir l'avenir. Cette vision humaine poussait la communauté scientifique dans le domaine, depuis des années, à améliorer et à trouver de nouvelles solutions permettant de dégager les connaissances utiles à travers des applications puissantes et autonomes qui concernent la classification automatique, la simulation et la visualisation en deux ou trois dimensions d'évènements pouvant arriver dans l'avenir. C'est dans ce cadre que se situe notre contribution.

Chapitre II. Etat de l’art sur la connaissance et les processus d’indexation de données

II.1 Introduction

Dans ce chapitre on se focalise sur la notion d'information qui est liée à la connaissance de l'incertain et, dès lors, est indissociable de celle d'événement aléatoire.

Une description, même très concise, de la théorie de l'information ne saurait donc, sans sacrifier à la généralité et à la rigueur du raisonnement, se préserver d'une certaine abstraction. Pourtant, le point de départ est intuitif et, historiquement, la notion d'information est née de l'étude de problèmes pratiques, notamment ceux que pose la transmission par fil des informations.

II.2 Aperçu sur la notion de l’information.

II.2.1 Le commerce de l’information.

Un immense marché s’offre aux entreprises spécialisées dans la vente de l’information .Selon Antoine Lefébure (1979) [41], cette dernière peut être considérée comme une marchandise « Cette denrée se préparera sous des formes de plus en plus maniables ou comestibles ; elle se distribuera à une clientèle de plus en plus nombreuse, elle deviendra chose de commerce, chose qui s’exporte, chose enfin qui s’imite et se produit un peu partout. »

Les connaissances s’exploitent, se transforment et se traitent. L’accès au savoir se vend, comme n’importe quel autre produit industriel. Les impératifs de rentabilité et la concurrence annoncent une guerre commerciale où risquent de s’engloutir les notions de service public et de souveraineté de l’Etat.

L’information n’a jamais été gratis. Livres et revues se vendent comme n’importe quel produit. Quand on assimile un livre ou une revue à une marchandise, on ne peut dissocier l’information de son support papier et encre. A un nombre de pages égal, un livre « intéressant » et bourré d’informations coûte aussi cher que le livre « creux ». Seules certaines publications, les lettres d’informations confidentielles sur abonnement, les études réservées à un ou à quelques commanditaires font payer cher la valeur des informations rassemblées, et non le prix du papier.

L’existence d’un secteur public de données (librairies, services universitaires, centres d’information) représente une poche de gratuité dans un ensemble soumis aux lois du marché. Même les services documentaires d’entreprise échappent à la logique marchande : s’ils respectent les critères de productivité, ils n’en sont pas moins perçus comme frais de fonctionnement internes à l’entreprise, jamais comme achat d’une prestation.

Par rapport à l’information stockée sur papier, dispersée, inaccessible à distance, les données extraites et transmises en temps réel représentent une information d’un ordre supérieur. On n’achète pas une certaine quantité de papier, une certaine qualité d’impression ou de brochage,

mais l'information elle-même, référence ou donnée brute. En fait, cette information n'est pas non plus dissociable du support informatique qui la restitue, a fortiori quand la restitution se complique d'un traitement graphique ou mathématique.

Ce sont, à la fois, un service et un produit qui sont vendus et suivant les cas, incorporés au produit final, plus ou moins de temps machine, plus ou moins de temps humain. Dans une banque de données, on peut acquérir une donnée brute par exemple combien de tonnes de café ont été produites au Brésil en 1976, mais aussi une série statistique sur dix ans, un agrégat de production agro-alimentaire ou une estimation prospective liée au nombre de tonnes ...etc.

II.2.2 La guerre des données.

Dans les conflits économiques et industriels mondiaux, surgit un nouveau front : celui de l'information. Ligne difficile à cerner, car elle se rapporte aussi bien à l'information comme production immatérielle - de la transaction bancaire à la distribution par correspondance, en passant par la gestion ou l'enseignement - qu'à l'information comme ressource, clé de toute stratégie, prévision ou décision.

Pigeons voyageurs et bibliothèques furent longtemps tout l'arsenal des Etats, des militaires et des marchands dans leurs efforts millénaires pour collecter et transmettre l'information. L'imagination déployée à cet effet suivait trois axes relativement linéaires : collecte (le renseignement) ; exhaustivité (l'inventaire) ; synthèse (le rapport). Subitement, cet ordre rassurant se trouve dispersé par le rythme rapide des mutations scientifiques et techniques, le volume gigantesque des informations à traiter, la brusque réduction des délais, pour la prise de décision et l'exécution.

La notion d'extraction de connaissance a pris naissance dans le secteur militaire .Ainsi, confrontée au déficit mentionné, l'armée américaine, fut la première à faire appel à des systèmes automatiques, ouvrant ainsi à l'industrie l'accès à un nouveau et vaste domaine. Dans les banques et bases de données, dans les réseaux télématiques, l'information s'émancipe des supports traditionnels (livres, revues, journaux). Convertie techniquement en unités élémentaires mesurables.

II.3 Notion de connaissance : l'information et la donnée à travers la connaissance.

La définition de la notion de connaissance peut viser deux points dans l'analyse : l'information et la donnée .Cependant, la nature et la connaissance de l'information au cours de sa transformation peut ouvrir un champ très vaste dans l'analyse et aboutir à du savoir au sein des entreprises.

II.3.1 Nature de la connaissance.

Certains auteurs ont classifiés la nature de la connaissance en trois notions : la donnée, l'information et la connaissance, or ce n'est pas toujours le cas pour tous, mais le plus important dans ces définitions c'est qu'on peut exploiter cela pour avancer dans l'analyse et la conception dans les études des cas.

- La donnée.

Une donnée est une image, un acte, nombre sans contexte. Pour d'autres [42] c'est un fait de base, qui apparait au cours de la réalisation d'une tâche comme symbole chiffre schéma …

La donnée est considérée comme l'élément de base qui mène à l'information, à la connaissance et au savoir, elle peut être mesurable quantitativement par des nombres ou qualitativement par des textes ou des images …

- 1'information.

L'information[1] « Élément de connaissance susceptible d'être codé ou représenté à l'aide de conventions pour être conservé, traité ou communiqué ».

L'information est une donnée triée et arrangée pour un objectif déterminé. C'est une donnée liée aux formules et moyens d'utilisation.

L'information [43] en mathématique est une quantité mesurée à l'aide d'une formule qui est sensiblement la même que celle utilisée par le physicien Ludwig Boltzmann à la fin du XIXe siècle pour mesurer l'entropie des gaz.

Dans la théorie de Shannon on parle de la structure de l'information, la fonction de l'information qui est le traitement adopté à la notion d'informatique.

Pour récapituler, l'information a un niveau plus haut dans la conception que la donnée, elle est souvent associée à l'exploitation dans des applications, au traitement informatique et au stockage sur des supports.

- La connaissance.

La connaissance est une information [44] en contexte, associée à une compréhension de son mode d'utilisation, comme par exemple : la connaissance au sujet du drainage de l'eau dans une rue, déduite de l'observation d'un schéma et la compréhension des influences de l'emplacement des maisons d'habitation sur ces drainages.

La connaissance [45], à l'inverse de l'information, repose sur un engagement des systèmes de valeurs et de croyances, sur l'intention. La connaissance est bâtie à partir de l'information pour faire quelque chose, pour agir…

La connaissance est donc située dans un niveau plus grand que l'apprentissage d'un savoir.

Une donnée après son analyse devient information et après l'application d'une conception, elle va devenir une connaissance exploitable dans l'entreprise.

II.3.2 Types de connaissances pour les entreprises.

Deux types de connaissances [46] :

- Connaissances implicites regroupant la connaissance pratique reçue par l'expérience et l'imitation, et les compétences ou culture d'entreprise visant à connaitre l'individu par certains indices et habitudes.

[1] www.larousse.fr, consulté le 12/12/2017 à 14h30

- Connaissances explicites : informations liées à un contexte donné, elles peuvent être gardées, stockées et transmises dans des CD-Rom, livres, voies électroniques …

Ces deux types sont trop importants dans la gestion des connaissances car ils nous permettent de bien concevoir les bases de données et les applications permettant un traitement et une exploitation efficace de l'information.

II.3.3 Gestion de connaissance :

La gestion des connaissances [47] : réaliser un système de gestion de flux cognitifs assurant l'enrichissement des connaissances de l'entreprise en localisant, en partageant en mettant en valeur tout savoir stratégique et de prise de décision au sein de l'entreprise.

La gestion des connaissances [48] est un moyen qui valorise les aptitudes, l'expérience de chacun là où il sera le mieux placé, qui fait circuler l'information utile et qui aide à trouver au bon moment celle dont on a besoin dans l'agissement.

La gestion de connaissance [49] « Knowledge management » :

- est une approche qui tente de manager des items aussi divers que pensées, idées, intuitions, pratiques, expériences, émis par des gens dans l'exercice de leur profession ;
- est un processus de création, d'enrichissement, de capitalisation et de diffusion des savoirs qui implique tous les acteurs et l'organisation, en tant que consommateurs et producteurs ;
- suppose que la connaissance soit capturée là où elle est créée, partagée par les hommes et finalement appliquée à un processus de l'entreprise.

L'intelligence artificielle était là à exploiter une gestion structuré de données à travers des robots autonomes qui traitent de l'information à grande échelle via les réseaux informatiques et de télécommunication ainsi ils permettent de partager les connaissances dynamiquement entre les êtres humains afin de bien prédire et d'une bonne prise de décision dans les domaines d'activités .d'où l'apparition de Systèmes de gestions des bases de données puissants qui facilitent le contrôle, l'enregistrement et l'accès rapides aux données .

II.3.4 Connaissance actionnable.

Connue par le nom « Actionable knowledge » [50] a pour but de faire la différence entre la connaissance et l'action liée à cette connaissance, en d'autre terme séparer entre le savoir et le savoir-faire et agir, afin de valoriser cette connaissance dans la prise de décision.

Il s'agit de créer des connaissances représentant des réflexions, des expériences, des actions…

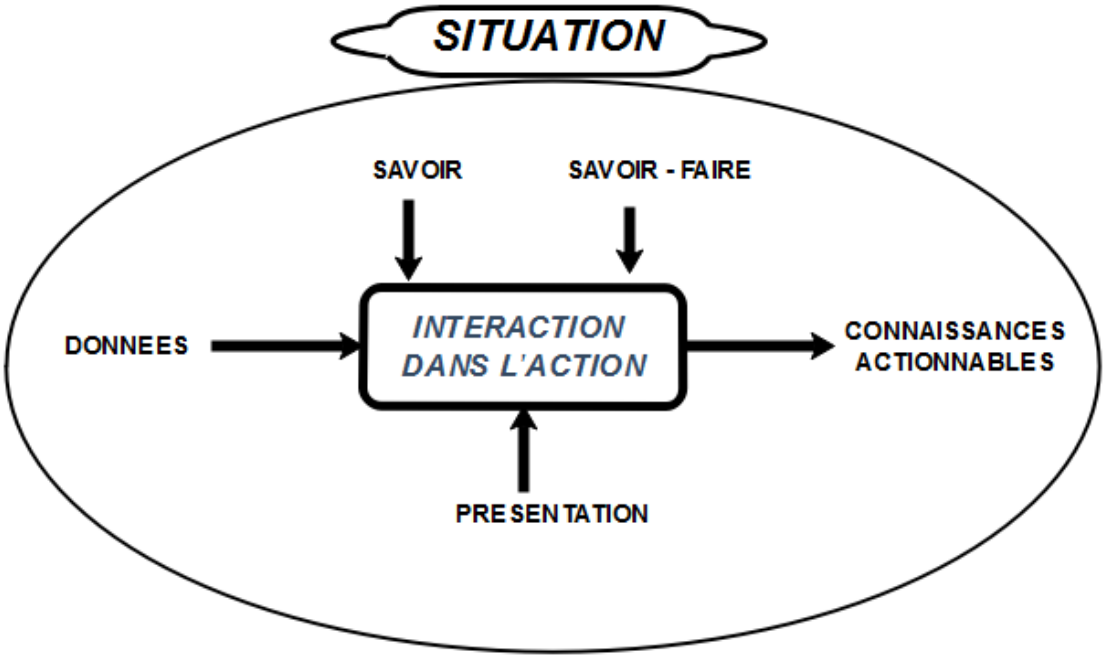


Figure II. 1: Connaissance Actionnable (CA) (réalisée par nos soins)

Du schéma ci-dessus on conclut que la gestion de connaissance actionnable est le processus de partage dynamiquement de connaissances nécessaires aux actions.

II.3.5 Processus d'indexation.

Afin de bien rendre accessible l'information contenue dans une mémoire, on doit la trier et l'organiser sous forme de répertoire : c'est le mécanisme ou processus d'indexation, il met un lien localisant chaque information textuelle à partir de son index.

« Indexer[2] un document consiste à construire la liste alphabétique des mots, des sujets, des noms apparaissant dans un ouvrage, une collection, etc. avec les références permettant de les retrouver ».

L'indexation peut être réalisée soit automatiquement ou manuellement :

II.3.6 Indexation manuelle :

Elle est faite par l'être humain qui peut organiser un document texte selon sa façon de voir, dans le but d'y accéder rapidement en cas de besoin.

II.3.7 Indexation automatique

Celle-ci est faite par la machine, où l'on définit au préalable un modèle géré par des règles et on l'applique sur un document.

[2] www.larousse.fr, consulté le 12/12/2017 à 14h30

II.3.8 Comparaison des deux processus

Automatiser le processus d'indexation c'est une question de rapidité et d'efficacité et surtout lorsqu'on est sensé de traiter des connaissances de grande taille et de les enregistrer dans une base de données. Par contre ce type d'indexation est moins précis puisqu'il écarte le facteur humain pour certaines données dont on a besoin d'une décision humaine.

II.4 Conclusion

Ce chapitre donne un aperçu sur la théorie de l'information développée par Shannon .0n s'y est intéressé à la construction et à l'étude de modèles mathématiques à l'aide essentiellement de la théorie des probabilités.

Depuis son premier exposé publié en 1948, la théorie de l'information s'est faite de plus en plus précise et est devenue aujourd'hui incontournable dans la conception de tout système de communication au sens le plus large de ce terme.

Chapitre III. L'extraction de la connaissance : une prolifération de méthodes.

III.1 Introduction

L'extraction des connaissances à partir de données se base sur les techniques de fouilles de données .A cet égard, les graphes sont souvent exploités dans les recherches car ils peuvent schématiser une grande partie de phénomènes naturels.

Dans ce chapitre nous allons décrire ce qui résulte exactement de la fouille de graphes dans un contexte global de « Data Mining » (forage des données relationnelles). Outre, nous allons énumérer les raisons qui poussent à faire le choix de la fouille de graphes comme solutions aux problèmes de la fouille des bases de données relationnelles.

Après avoir procéder à un examen générale de la notion de « Data Mining », nous allons présenter certaines notions et méthodes propres à la fouille des données tabulaires de type objets × attributs qui sont susceptibles de subir une conversion pour être réutilisées dans le forage graphique.

Nous comptons parler, d'abord, des règles d'association et la recherche des motifs fréquents pour deux raisons essentielles suivantes :

- les algorithmes de recherche des motifs fréquents étaient considérés comme le point de départ des méthodes de fouille de graphes ;
- les premières [51] fouilles des bases de données relationnelles se basent sur l'extraction des règles d'association fréquentes.

Par la suite, nous projetons aborder l'analyse de concepts formels (ACF).

L'ACF,qui est une technique destinée pour traiter les propriétés des motifs d'attributs et dont ne disposent pas les graphes , ce qui permet ainsi de mettre plus aisément au point les limites de l'analogie pouvant se faire entre motifs d'attributs et motifs de graphes.

Depuis plus de vingt ans, l'outil informatique a permis de produire et de stocker des données numériques de taille élevée. Celles-ci contiennent de nombreuses indications sur les objets qui y sont décrits même si en général, elles sont collectées pour un service donné ou répondre à une question précise.

Cependant ces éléments ne peuvent être analysés par l'être humain, vue la grande taille et la complexité des informations qui y trouvent.

Pour remédier à ce problème, une nouvelle discipline scientifique vient d'être instaurée par la [52] communauté des bases de données appelée « Fouille de données ».

Ainsi selon Fayyad [53] (1996), l'extraction de connaissances à partir de données se définit comme « l'extraction automatique de connaissances nouvelles, utiles et valides à partir de grandes quantités de données ». En termes scientifiques, la fouille de données vise l'informatique et les statistiques.

De telles méthodes trouvent leurs applications en statistiques et en méthodes d'apprentissage automatique, qui sont dédiées pour la conception d'algorithmes capables de trouver la solution d'un problème à partir d'exemples de solutions.

Le but c'est de dégager des modèles qui soient significatifs des données tout en étant robustes aux erreurs, et à la fois à la conception d'algorithmes sûrs. Nous visons le traitement de grandes

quantités de données et à prendre en compte des questions plus ouvertes que celles traitées dans l'analyse statistique de données. Par ailleurs contrairement à cette dernière, l'objectif de la fouille de données n'est pas de juger des hypothèses mais d'accroître les domaines de connaissances à travers tous les moyens informatiques et modèles de conception disponibles.

La fouille de données regroupe ainsi des techniques autant variées que l'extraction de règles associatives, les méthodes de régression de fonctions ou de « Clustering » d'objets de caractéristiques similaires.

L'opération de fouiller des données n'a aucune importance si les résultats de cette fouille ne sont pas explicables. Donc La fouille de données est l'une des étapes d'un processus d'extraction de connaissances plus étendu qui commence de la préparation des données jusqu'à l'interprétation des résultats. La fouille de données mène à des connexions vers d'autres disciplines préexistantes telles que les bases de à l'intelligence artificielle. Ci-dessous, la fouille des données au sein du données, les outils de visualisation, les modèles de visualisation de la connaissance et plus profondément processus d'extraction de connaissances tel que décrit dans Fayyad (1996)[53] :

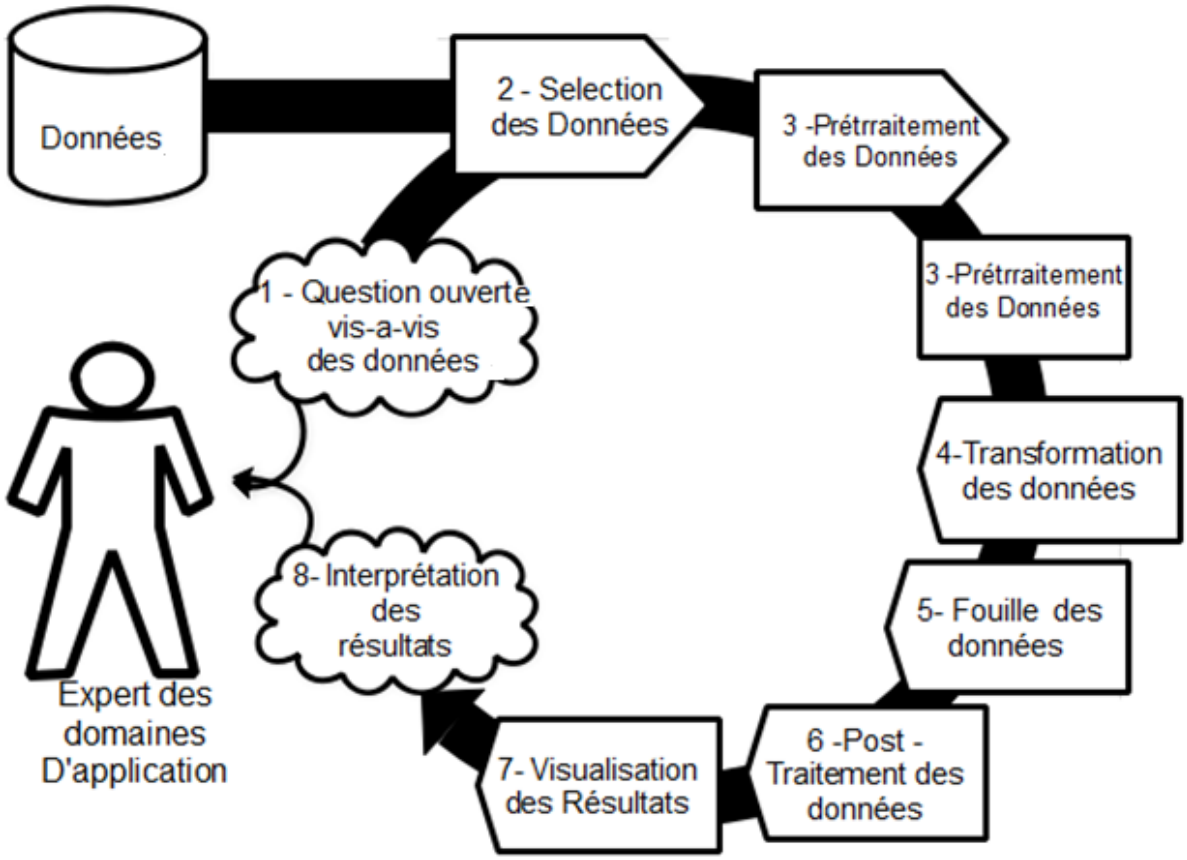


Figure III.1: Le processus d'extraction de connaissances à partir de données. (réalisée par nos soins)

Dans ce processus l'expert système de fouille des données applique la fouille pour avoir une réponse aux questions. Cette étape de fouille des données est l'étape capitale la plus claire du point de vue de l'utilisateur. Fayyad (1996) souligne notamment le point itératif et interactif du processus : l'expert adapte ses questions selon les réponses que lui renvoie le système de fouille de données.

Au fur et à mesure du traitement, le cycle d'extraction des connaissances devient plus visible à travers la prédiction des liaisons formables et leur analyse. L'auteur montre l'importance de la phase de prétraitement pour le filtrage et la transformation des données, pour une utilisation idéale de la méthode de fouille de données.

III.2 L'extraction des connaissances à partir de données.

III.2.1 La recherche des motifs fréquents et l'extraction des règles d'association.

Agrawal [54] (1993) fut le premier à résoudre le problème de la recherche des motifs fréquents à l'aide de son algorithme « apriori ». A travers son article il a réalisé des travaux d'apprentissage symbolique qui traitent des problèmes de classifications supervisées ou non supervisées.

A l'inverse des méthodes de régression qui décrivent les données de manière globale, sa méthode cherche à définir des règles d'association « locales »pour donner des descriptions à un sous ensemble de données réduit.

Finalement à l'inverse des systèmes existants de classification à base de règles (Breiman, 1984 ; Piatetsky-Shapiro, 1991 ; Han, 1992 ; Quinlan, 1993 ; Fayyad, 1993), la méthode présente un algorithme fini pour extraire un nombre important de règles significatives tout en manipulant de grandes quantités de données.

La recherche des motifs d'attributs fréquents : dans son formalisme initial, considère un ensemble O de n objets (ou transactions) décrits par un ensemble A de m attributs selon une relation binaire $R \subseteq O \times A$. Un exemple de la table de la figure ci-dessous montre des relations binaires entre attributs et objets :

Les colonnes représentent les attributs de A à D et les lignes représentent les objets d'o_1 à o_5. La croix montre quels sont les attributs qui sont en relation avec un objet.

R	A	B	C	D
o_1	×	×	×	×
o_2	×	×	×	
o_3			×	×
o_4	×		×	
o_5			×	

Tableau III.1 : Un exemple de relation binaire

L'ensemble des objets présentant un ensemble donné d'attributs est identifié par la donnée de cette relation binaire.

Le but de cette recherche et de repérer la présence simultanée fréquente et répétitive d'attributs dans les données.

Pour mieux illustrer la technique de recherche des motifs d'attributs fréquents on donne les définitions suivantes :

Définitions.

1. Un motif est un ensemble d'attributs.
2. Un motif $M \subseteq A$ décrit ou couvre un objet de O si cet objet présente (i.e. est en relation avec) tous les attributs éléments de M.
3. Le support $support(M)$ du motif M est le nombre d'objets décrits par M : $support(M) = |o \epsilon O| \ \forall a \epsilon M, oRa|$[3] .Le support est appelé aussi fréquence absolue.
4. La fréquence relative d'un motif M notée $freq_r(M)$ est la fraction du support de M sur le nombre n d'objets et est donc comprise entre 0 et 1.

Le but recherché est de trouver la fréquence de l'ensemble des motifs fréquents possédant une fréquence supérieure ou égale à une fréquence minimale relative ou absoluef_{min}.

En observant la figure ci-dessous, pour le motif {A, C}, qu'on simplifie par AC, la fréquence de répétition des attributs dans les objets o_i peut être vue dans la valeur du support : support | $\{o_1, o_2, o_4\}$| = 3, la fréquence relative 3/5.

La fréquence est une fonction décroissante dans l'ordre des motifs ordonnés par la relation ⊆ d'inclusion ensembliste :

$$M_1 \subseteq M_2 \subseteq A \implies freq(M_1) \geq freq(M_2)$$

Le diagramme [55] ci-dessous appelé diagramme de Hasse représente l'ordre des motifs. On remarque la décroissance de la fréquence des motifs indiquée entre parenthèses, allant du sommet représenté par le vide et descendant vers le bas.

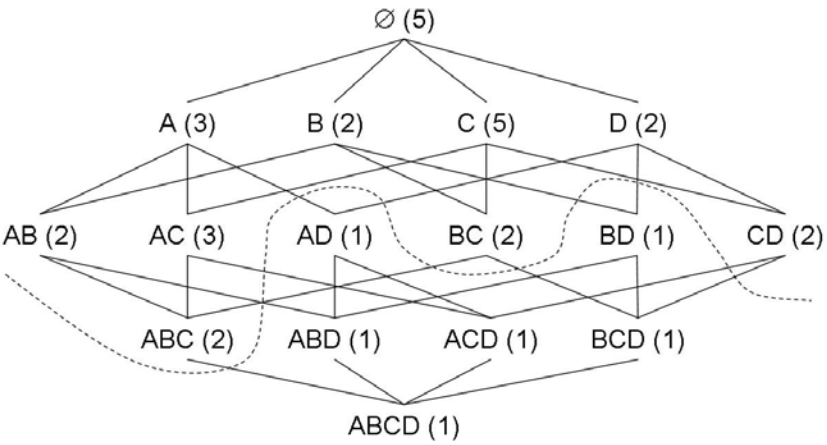


Figure III.2 : La décroissance des fréquences au sein de l'ordre des motifs(réalisée par nos soins)

Du diagramme on constate que l'ensemble de motifs {∅, A, B, C, D, AB, AC, BC, CD, ABC} ayant un support supérieur ou égale à 2 se trouve au-dessus de la ligne en pointillée.

[3] Cardinal d'ensemble.

Un motif est fréquent lorsque les motifs qui le composent le sont aussi, d'où la formule suivante :

$$M_1 \subseteq M_2 \subseteq A \ \ et\ freq(M_2) \geq f_{min} \Longrightarrow freq(M_1) \geq f_{min}$$

La recherche des motifs fréquents est un problème difficile car le nombre de motifs fréquents peut augmenter exponentiellement vis-à-vis du nombre de motifs traité.

L'étape d'examen de ces motifs fréquents est l'étape la plus coûteuse de l'extraction de règles d'association vu la taille exponentielle du domaine de recherche et du nombre élevé nécessaire de balayages complets du jeu de données.

Le choix de f_{min} peut rendre difficile la fouille des motifs fréquents, la complexité des algorithmes de recherche de motifs fréquents est exponentielle, le temps de recherche dépend alors de l'ajustement de f_{min} afin d'avoir des résultats acceptables en pratique.

Plus la longueur des motifs étudiés est courte, plus le nombre augmente et l'information apportée devient moins utile, d'où le choix de la fonction de score qui permet de bien sélectionner et de traiter un ensemble réduit afin de trouver les meilleures règles d'associations .

L'extraction des règles d'association fréquentes.

Une règle d'association H → C entre deux motifs H et C représente une relation de corrélation entre les attributs de H et de C.

Pour trouver les règles d'associations, on cherche des motifs fréquents dans les données. On dit qu'un motif est fréquent s'il a une présence minimale dans la base de données, cette présence est nommée support minimum qui peut être fixé d'avance.

Ainsi, si le support minimum est fixé à 10 %, les ensembles de motifs apparaissant dans moins de 10 % des transactions sont éliminés des motifs fréquents.

A partir de ces motifs et leurs supports, on calcule la valeur de confiance des règles que l'on peut en déduire.

Si par exemple le motif ABC est retenu comme motif fréquent, nous considérons alors les règles : AB → C, AC → B, BC → A, A → BC, B → AC et C → AB.

Définition : Formellement

1. Une règle d'association H → C est définie par un motif hypothèse H (ou prémisse) et un motif conclusion C disjoint de l'hypothèse. Elle exprime le degré de vraisemblance selon lequel un objet présentant les attributs du motif H présente aussi ceux de C.
2. Le support (H → C) de la règle H → C est le support de H ∪ C. Le support d'une règle permet d'évaluer sa représentativité dans les données.
3. La confiance (H → C) d'une règle H → C est le rapport du support de H ∪ C sur celui de H. La confiance représente le degré de vraisemblance ou précision de la règle.

Dans le cas de la figure 2.4 ci-dessus, la règle d'association A → BC a un support 2 et une confiance 2/3.Notre objectif est d'extraire des informations à partir de motifs fréquents possédant une fréquence relative et une confiance au-delà d'une fréquence minimale $freq_{min}$ et confiance minimale $Conf_{min}$.

Pour calculer la valeur de confiance de la règle AB → C, on calcule support(ABC)/support(AB) ce qui représente la probabilité conditionnelle de rencontrer C dans

une transaction sachant que celle-ci contient A et B. Le résultat diffère généralement sans doute pour celui de chacune des règles déduites d'un motif.

En conséquence, l'utilisateur peut encore filtrer les résultats obtenus sur la valeur de confiance d'une règle afin d'éliminer celles qui ne sont pas réellement significatives.

S'il s'agit de trouver tous les motifs de la base pour découvrir toutes les règles d'associations valables, il suffira de prendre pour support minimum 1. En conséquence, chaque motif se trouvant dans la base est considéré comme fréquent. Le principal problème pour trouver des règles d'associations consiste en la découverte des motifs fréquents. En effet, ceux-ci sont potentiellement au nombre de 2m pour une base contenant m motifs.

Le nombre des motifs fréquents est lié au choix du support minimum. Dans une base de données de grande taille, il peut ne pas exister de corrélation entre un motif et un autre même si le nombre de fréquence est important dans la base, à l'inverse, on peut trouver un motif peu représenté dans la base mais il implique la présence d'un autre .

Par exemple [56] :

- De nombreuses personnes achètent du pain et de nombreuses personnes achètent des piles (support important), cependant, on trouve une valeur de confiance peu élevée pour la règle (Pain) => (Piles).
- Peu de personnes achètent un appareil photo numérique, peu de personnes achètent des cartes mémoires (support faible), mais la valeur de confiance de la règle (Appareil photo numérique) => (Carte mémoire) est élevée.

La recherche de règles d'associations valables requiert une valeur de support minimum faible pour avoir plus de règles intéressantes. Plus le support minimum choisi est faible, plus les motifs fréquents sont nombreux et donc plus le traitement à sacrifier est important.

C'est la raison qui pousse à avoir plus de puissance de calcul et d'espace mémoire important pour la recherche des motifs fréquents.

De ce fait on est obligé d'utiliser des algorithmes parallèles, sachant que la taille des bases de données à traiter est grande, un calcul sur grille parait être une solution puissante de recherche des motifs fréquents.

Les algorithmes de recherche des motifs d'attributs fréquents.

Algorithme Apriori [57] : créé en 1994, par Rakesh Agrawal et Ramakrishnan Sikrant, pour l'apprentissage des règles d'association. Il permet de reconnaitre des propriétés des motifs fréquents dans une base de données et de les classer.

Principe de base de cet algorithme est le suivant :

- Si un ensemble est non fréquent, alors tous ses sous-ensembles ne sont pas fréquents.
- si {A} n'est pas fréquent alors {AB} ne peut pas l'être.
- si {AB} est fréquent alors {A} et {B} le sont.
- Itérativement, trouver les motifs fréquents dont la cardinalité varie de 1 à k (k-motif).
- Utiliser les motifs fréquents pour générer les règles d'association.

Pour l'algorithme d'Apriori :

- Rechercher les séquences de longueur 1 ayant un support supérieur à s, c'est l'ensemble des sous-ensembles fréquents.
- A partir des séquences trouvées dans l'étape précédente, construire les séquences de longueur 2 avec un support supérieur à s.
- Par itération, construire des séquences de longueur k avec un support supérieur à s à partir de celles trouvées pour une longueur k-1.

Algorithme [58] Apriori-gen(F)
Entrée : F : ensembles de motifs fréquents de cardinal k
Début

$$\boldsymbol{C \leftarrow \{c = f_1 \cup f_2\ tels\ que\ (f_1, f_2) \in F \times F, card(c) = k + 1\}}$$

Pour chaque $\boldsymbol{c \in C}$ faire
    Pour chaque $\boldsymbol{s \subset c, card(s) = k}$ faire
        Si $\boldsymbol{s \notin F}$ alors
            $\boldsymbol{C \leftarrow C \backslash \{c\}}$
        Retourner $\boldsymbol{C}$;
Fin.

Algorithme III.1 : Apriori-gen(F)

Algorithme Apriori [59] (T, *minsup*)
Entrée : T : corpus, *minsup* : entier.
Sortie :$\boldsymbol{U_k F_k}$.
Début

$$\boldsymbol{C_1 \leftarrow \{singletons\}}$$
$$\boldsymbol{k \leftarrow 1}$$

Tant que $\boldsymbol{C_k \neq \emptyset}$ faire
Pour chaque $\boldsymbol{c \in C_k}$ faire
    Pour chaque $\boldsymbol{t \in T}$ faire
        Si $\boldsymbol{c \subset t}$ alors
            $$\boldsymbol{support(c) \leftarrow support(c) + 1}$$
        $$\boldsymbol{F_k \leftarrow \{c \in C_k | support(c) \geq minsup\}}$$
        $$\boldsymbol{k \leftarrow k + 1}$$
        $$\boldsymbol{C_k \leftarrow Apriori - gen(F_{k-1})}$$
Retourner $\boldsymbol{U_k F_k}$
Fin.

Algorithme III.2 : Apriori (T, minsup)

Avec $\boldsymbol{C_k}$ est l'ensemble d'enregistrements contenant deux champs : le motif regroupant le sous ensemble d'éléments, et le champ « support » qui inclut la fréquence de cet ensemble dans la base de données. L'algorithme Apriori découvre les sous-ensembles de motifs fréquents en commençant par ceux de longueur est 1 et en incrémentant à chaque fois cette longueur.

L'algorithme apriori-gen est composé de deux étapes : dans la première on cherche tous les motifs possibles de longueur k à partir de l'ensemble **F : ensembles de motifs fréquents**, dans la deuxième, on élimine de $\boldsymbol{C_k}$ les motifs qui ne vérifient pas la propriété des sous ensemble fréquents.

Cet algorithme [60] est très compétitif, mais souffre si les ensembles d'éléments fréquents sont trop grands. De plus, scanner la base de données à la recherche d'un motif de façon répétée devient rapidement un frein aux performances sur de grosses bases de données.

III.2.2L'analyse de concepts formels du contexte relations, objets et attributs.

Birkhoff (1940) a développé la théorie mathématique des treillis ou treillis de Galois dont le but est de déduire et de représenter la connaissance à partir des motifs fréquents en passant par une analyse de concepts formels (ACF).

L'analyse de concepts formels [61] a vu le jour par Wille en 1982, elle définit un contexte (O, A, R) d'une relation binaire R entre un ensemble d'attributs A et un ensemble d'objets O.

Supposant un contexte $(\mathcal{O}, A, R)$, l'ACF fait appel à deux fonctions p et q qui font passer d'un motif d'attributs A aux objets O que ce motif décrit et réciproquement d'un ensemble d'objets au motif d'attributs commun à tous ces objets :

$$\boldsymbol{p}: \boldsymbol{M} \subseteq \boldsymbol{A} \longrightarrow \boldsymbol{p}(\boldsymbol{M}) = \{\boldsymbol{o} \in \mathcal{O} | \boldsymbol{a} \in \boldsymbol{M} \Rightarrow \boldsymbol{oRa}\} \; \boldsymbol{et} \; \boldsymbol{q}: \boldsymbol{O} \subseteq \mathcal{O} \longrightarrow \boldsymbol{q}(\boldsymbol{O}) = \{\boldsymbol{a} \in \boldsymbol{A} | \boldsymbol{o} \in \boldsymbol{O} \Rightarrow \boldsymbol{oRa}\}$$

Le couple (p, q) définit une correspondance de Galois qui exprimer la dualité existante entre les ordres $(2^A, \subseteq)$ et $(2^{\mathcal{O}}, \subseteq)$ des sous-ensembles d'attributs et d'objets :

Propriétés

- Le couple (p, q) de fonctions définit une correspondance de Galois :

1) p et q sont des fonctions décroissantes :

M1 ⊆ M2 ⊆ A ⇒ p(M1) ⊇ p(M2) et O1 ⊆ O2 ⊆ O ⇒ q(O1) ⊇ q(O2)

2) Les fonctions composées f = q ◦ p : $2^A \rightarrow 2^A$ et g = p ◦ q : $2^{\mathcal{O}} \rightarrow 2^{\mathcal{O}}$ sont extensives : ∀ M ⊆ A, f(M) ⊇ M et ∀O ⊆ O, g(O) ⊇ O

Cette correspondance de Galois rend les fonctions f et g comme opérateurs de fermeture :

- Les opérateurs f = q ◦ p et g = p ◦ q sont des opérateurs de fermeture, c'est-à-dire des fonctions croissantes, extensives et idempotentes :

(1) ∀M1 ⊆ A, ∀M_2 ∈ A, $M_1 \subseteq M_2$ ⇒ f(M1) ⊆ f(M_2) (croissance)

(2) $\forall M \subseteq A$, $f(M) \supseteq M$ (extensitivité)

(3) ∀M ⊆ A, f (f(M)) = f(M) (idempotence)

Les fermés liés à f sont alors les éléments stables de f (i.e. les éléments e tels que f(e) = e).

On décrit alors l'ensemble des concepts formel par les couples *(M, p(M))*, M est l'intention et *p(M)* est l'extension.

- C : l'ensemble des concepts formels est un treillis commutatif pour l'union et l'intersection : soient deux concepts formels *(M_1, O1) et (M_2, O2)*

1) (M_1, O1) ∨ (M_2, O2)= (M_1 ∩M_2, p (M_1 ∩M_2))
2) (M_1, O_1) ∧ (M_2, O_2) = (q (O_1 ∩ O_2), O_1 ∩ O_2).

D'après le diagramme de Hasse précèdent on arrive à représenter le treillis de concept de la figure ci-dessous représentant cinq concepts, on constate que pour chaque concept on présente une intention AC et une extension 124= {O_1, O_2, O_3}.

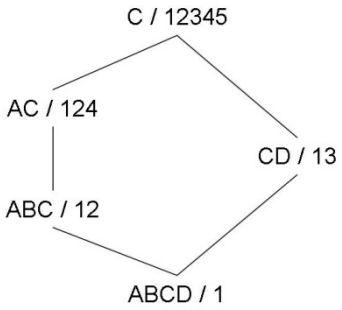


Figure III.3 : Exemple de Treillis de concept formel formé de cinq concepts (réalisée par nos soins)

De cet exemple on peut retrouver facilement la table des relations binaire vue précédemment : l'exemple du concept CD/13 signifie que les attributs C et D se trouvent dans les objets $o_1$et o_3.

III.2.3 Les méthodes de recherche sélective de motifs fréquents

L'analyse de concept formel a permis de réduire les règles d'association sans perdre la connaissance. Les opérateurs de fermeture f = q ◦ p ont permis d'identifier pour chaque classe d'équivalence une relation d'équivalence entre motifs, limités entre un élément minimum et un maximum.

Chaque classe d'équivalence C a un et un seul élément maximal qui est le motif fermé [62] f(M) pour tout motif M de C.

Le diagramme de Hasse de la figure 4 peut être représenté par les classes d'équivalence ci-dessous :

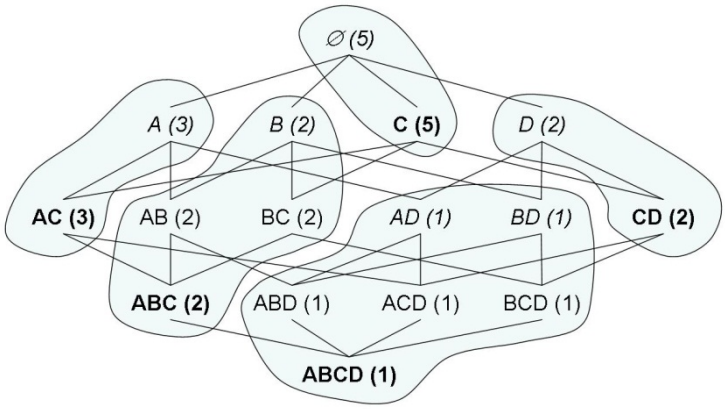


Figure III.4 : Classes d'équivalence de motifs (réalisée par nos soins).

On constate que chaque classe contient des motifs minimaux appelé générateurs en italique, et des motifs fermés en gras .L'exemple de concept (ABCD, 1) est lié à la classe d'équivalence {AD ; BD ; ABD ; ACD ; BCD ; ABCD} et possède le motif fermé ABCD et les motifs générateurs minimaux AD et BD. Cette représentation permet d'avoir une vue condensée de motif fréquents.

Une autre technique de recherche de motifs consiste à l'introduction des contraintes spécifiques dans le processus de data mining.

La réponse à une requête se traduit par un ensemble de motifs, dont on cherche à définir la relation avec un motif trouvé. Comme exemple, on préfère éviter la répétition entre motifs. Plus généralement, on désire souvent explorer des motifs de plus haut niveau aillant des

caractéristiques qui impliquent plusieurs motifs locaux et donnant à la fin un sens global à l'ensemble de motifs retourné. Pour cela, l'importance d'un motif doit obligatoirement dépendre des motifs qui l'environnent, afin de procéder à la comparaison de l'ensemble. Cette conception est déjà présente dans les représentations condensées de motifs, libres ou fermés.

Afin de saisir l'ensemble des motifs utiles retournés par une requête, il est évident de chercher à identifier des contraintes portant sur un ensemble de motifs et non plus sur un seul motif.

Zimmermann [63] a défini ces contraintes comme étant appliquées à des ensembles de motifs.

III.3 Techniques d'extraction de connaissances à partir de données relationnelles

Les techniques de recherche de motifs fréquents permettent de traiter un grand nombre de donnés. Toutefois, on est obligé de projeter les données dans des tables et d'y chercher les relations entre les attributs d'objets(O, R, A).

La meilleure façon de procéder consiste à examiner les relations entre classes au lieu de l'effectuer sur les objets, celles-ci peuvent être interprétées comme des attributs logiques liants les objets entre eux, et c'est donc le point fort des diagrammes de classes à travers un langage de modélisation unifié (UML).

Le but est alors de déduire de ces données une théorie logique la plus complète et la plus homogène possible.

III.3.1 L'extraction de connaissances à partir de relations.

Pour bien visualiser les modèles de relations (objets*attributs) dans l'analyse de concepts formels(ACF), on introduit des attributs multi-values dans le principe de « Conceptual Scaling» [64] : la mesure d'une propriété implique l'attribution de nombres aux systèmes pour représenter cette propriété, d'où l'étude la une notion générale de dépendance entre attributs couvrant en particulier la dépendance fonctionnelle, linéaire et bien d'autres.

Les attributs peuvent être liés par des relations binaires formant ainsi des graphes relationnels orientés où les sommets et les arcs symbolisent respectivement les attributs et les relations entre couples d'attributs.

La figue ci-dessous représente un ensemble d'objets dont les attributs sont liés par différentes relations binaires (représentées par des arcs de couleurs). L'objet o2 est décrit par un attribut a ayant une relation de type α avec l'attribut e, un attribut g en relation de type β avec h lui-même en relation avec g selon β et deux attributs isolés b et c. Si chaque objet est décrit par le même ensemble d'attributs et de relations, les objets (i.e. o1, o2 et o3) sont toutefois indépendants au sens où il n'existe pas de relations liant ces objets.

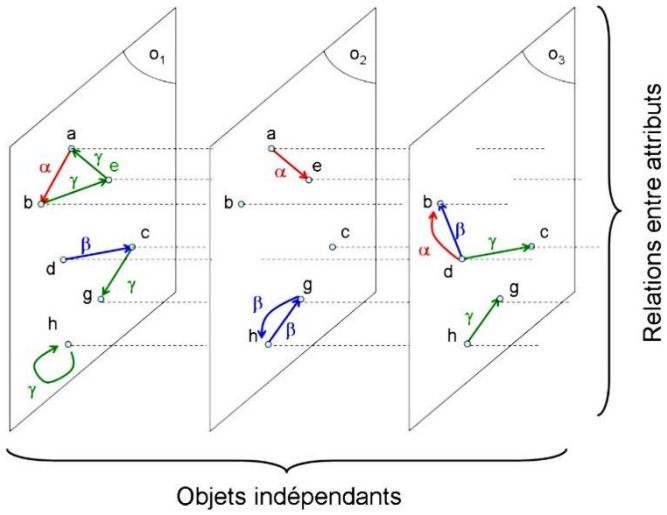


Figure III.5 : Un contexte de relations binaires entre attributs. (réalisée par nos soins)

Ce contexte de relations binaires entre attributs peut parfois être assimilé à des graphes relationnels jouant ainsi un rôle important dans divers applications comme les réseaux sociaux, les réseaux informatiques, comme Internet « WWW ».

Notons aussi que des objets du même contexte peuvent avoir des attributs indépendants. Ces attributs eux-mêmes peuvent être mis ou non en relation. Ainsi dans la figure ci-dessous, l'objet O_6 est décrit par l'attribut c et par les relations qu'il a avec o4 selon α (i.e. $O_4 \alpha\ O_6$) et avec O_1 selon β (i.e. $O_6\ \beta\ O_1$).

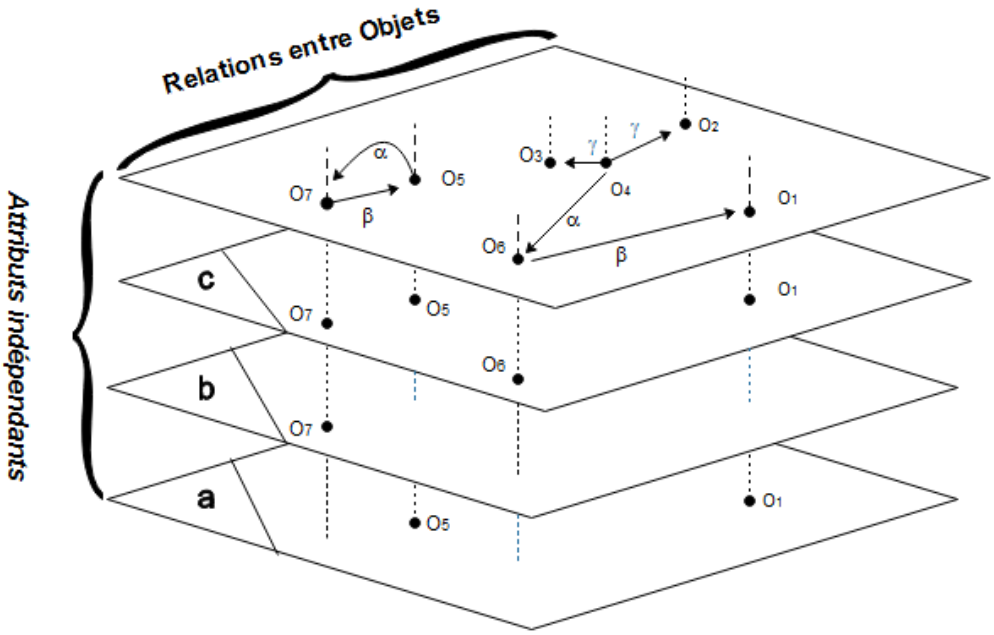


Figure III.6 : Un contexte de relations entre objets. (réalisée par nos soins)

Pourtant le traitement des relations entre objets n'est pas identique et reste difficile que celui entre attributs. À l'inverse dans l'ACF, les objets et les attributs n'ont pas la même fonction dans l'extraction de connaissances. Cela relève de l'évidence puisque les objets ne sont que des

vecteurs inconnus. Ils ne sont donc pas singuliers comme les attributs, qu'on peut en témoigner O6 et O7. Ces derniers sont liés aux autres objets O1 et O5 par la relation β.Quoi qu'il en soit, si O6 et O7 ont une relation commune, nous nous intéressons plutôt à leurs attributs qui peuvent être partagés entre les différents objets, ces attributs permettent d'avoir un motif significatif, définit par les deux attributs d'objets O6 et O1 et la relation β, cela nous permet alors d'identifier un concept formel lié à ce motif.

D'où l'apparition d'un nouveau concept relationnel (ACR : Analyse de concept relationnel) qui lie les attributs entres eux par des relations indépendamment aux objets là où ils se trouvent. L'ACR[65] est une extension de l'Analyse de Concepts Formels, une méthode de classification non supervisée d'objets sous forme de treillis de concepts. L'ARC supporte en plus la gestion de relations entre objets des différents contextes ce qui permet d'établir des liens entre les concepts des différents treillis.

Cette analyse (ACR) prend en compte non seulement les caractéristiques des objets, mais aussi les relations que les objets entretiennent entre eux. Elle applique itérativement un algorithme de l'ACF pour gérer les données relationnelles : les objets sont décrits par des attributs et par leurs relations vers d'autres objets. Les concepts découverts à une itération donnée sont retransmis le long des relations, pour permettre la découverte de nouveaux concepts à l'itération suivante.

III.3.2 La programmation logique inductive.

Muggleton[66] (1991) a nommé la "programmation logique inductive" (PLI) Elle regroupe l'apprentissage automatique et la programmation logique. À la différence de la programmation logique déductive, qui résulte des conséquences à partir des théories.

La programmation logique inductive cherche des hypothèses H à partir d'un ensemble d'observations E. Il s'agit de résumer de nouvelles connaissances à partir d'observations et d'une base de connaissances. Elle effectue le même travail que la fouille de données traditionnelle qui produit des hypothèses à partir de données.

La fouille de données manipule des données classées dans une table de la forme "attribut = valeur", mais la programmation logique inductive considère que les données en entrée et les modèles extraits sont exprimés en logique du premier ordre, appelée aussi logique des prédicats [67].

On définit la (PLI) de la façon suivante :

En entrées : on considère trois ensembles de clauses : B, P et N avec

- B : base de connaissances exprimées sous forme de clauses de Horn [68], (rappelons qu'une clause de Horn est toute clause ayant au plus un littéral positif B, ¬A, ¬Av¬C, ¬Av¬CvB sont des clauses de Horn mais AvB, ¬Av¬BvCvD n'en sont pas).
- P : exemples positifs exprimés sous forme de clauses de Horn.
- N : exemples négatifs exprimés sous forme de clauses de Horn.

En sortie : on cherche à trouver une hypothèse H sous forme d'un ensemble de clauses de Horn telle que les propriétés suivantes soient le plus possible respectées :

- Complétude : $\forall e \in P, H \cup B \models e$
- Consistance : $\forall e \in N, H \cup B$ (non $\models$) e

(Sachons que $x \vDash y$ signifie que x implique sémantiquement y)

On cherche donc à trouver une hypothèse H qui permet d'expliquer au mieux les exemples positifs, tout en restituant au maximum les exemples négatifs. Cette recherche se fait par

inversion du raisonnement déductif. Elle se base généralement sur la propositionnalisation [69] ou sur des méthodes de fouille de données adaptée à la logique de premier ordre et sur la substitution, la spécialisation, la généralisation, l'unification et la résolution de la programmation logique.

Notons que dans la programmation logique inductive, on trouve la méthode de TILDE (Top-down Induction Logical Decision tree), c'est une classification par arbre de décision fondée sur la logique de premier ordre.

Elle conçoit un arbre de décision binaire selon la définition d'un arbre de décision logique[70].

La méthode TILDE se base sur le même principe que les techniques classiques afin de construire un arbre de décision : elle subdivise une population d'apprentissage pour avoir des sous populations augmentant alors les éléments d'une des classes.

TILDE peut tenir en compte des relations entre tables, des règles d'experts ou des prédicats exprimant un groupement de prédicats simples.

Cette méthode produit des arbres moins profonds, la définition des étapes principales est décrite dans l'algorithme ci-dessous :

Algorithme TILDE [71] (Top- down Induction Logical DEcision tree)

En entrée : T : Arbre, E : ensemble d'exemples, B : base de connaissances,

Procédure Construire_Arbre (T, E, B, True) ;

/* La classe est un prédicat dans E. Initialement, T est vide est Q = true */

En sortie : T : Arbre de décision binaire

En entrée : N : nœud, E : Ensemble des exemples du nœud N, Q : prémisse du nœud

Début

Procédure Construire_Arbre

Si (E est suffisamment homogène) **alors**

1. K ← classe_majoritaire ; N : feuille (info (E)) ;

Sinon

2. L ← ensemble des spécialisations de Q dans E
3. Qb ← la meilleure condition qui segmente E

/*calculée suivant une heuristique : gain ratio */

4. Conj ← Qb ∧ Q ;
5. E1 = {e ∈ E/ e est vrai dans Conj} ; E2 = {e ∈ E/ e est faux dans Conj} ;
6. Construire_Arbre (gauche, E1, B, Qb} ;
7. Construire_Arbre (droit, E2, B, Q} ;
8. N = nœud (Conj, gauche, droit) ;

Fin si

Retourner l'Arbre T ;

Fin.

Algorithme III.3 : TILDE.

Au début, l'arbre est vide, la prémisse Q = true et toutes les observations E se trouvent dans le nœud racine. On vérifie si ce nœud racine est cohérent ou pas. Si oui, on déclare le nœud comme feuille (nœud saturé) et on récupère toutes les informations le concernant (ligne 1). Sinon, on calcule l'ensemble des spécialisations de la prémisse Q dans E (ligne 2). La spécialisation permet d'ajouter un littéral à la prémisse d'une clause ou à remplacer une variable par un terme. On retient parmi ces spécialisations celle qui retourne un meilleur fractionnement de E. Cette segmentation est sélectionnée selon le gain ratio (ligne 3). On ajoute cette spécialisation à la

prémisse de Q et on divise le nœud père en deux nœuds : fils gauche qui contient les observations qui vérifient la condition et fils droit qui contient les observations qui ne vérifient pas la condition (ligne 5). On réitère la procédure de construction de l'arbre pour chacun des fils gauche et droit (ligne 6 et 7) et on insère le nœud parent dans l'arbre (ligne 8). Le processus s'arrête lorsque tous les nœuds sont saturés.

III.3.3La programmation logique inductive et l'extraction de connaissances à partir de graphes.

La PLI peut dépasser certaines limites des formalismes de représentation des connaissances dans les systèmes d'apprentissage par l'utilisation de la logique du premier ordre. Cette modélisation met en valeur les relations entre objets.

La PLI met en œuvre l'induction en logique des prédicats qui est utilisée comme langage de représentations spatiales d'exemples et des hypothèses (clauses de Horn). La généralisation de nouvelles règles se fait par la recherche de clauses dans un espace organisé selon une relation de généralité.

Un des algorithmes de recherche descendante, comme Foil[72] « First-order inductive Logic », débutent d'une clause générale vers des clauses plus spécifiques en utilisant des opérations comme l'ajout de littéraux à la clause de démarrage ou l'application de remplacement pour convertir des variables en constantes ou pour unir plusieurs variables. Les clauses ainsi produites sont ensuite vérifiées sur les exemples de façon à généraliser le maximum d'exemples positifs et peu ou pas d'exemples négatifs.

En PLI, il est souvent indispensable de connaitre la taille pour mieux gérer le chemin parcouru pendant la recherche basé sur les heuristiques.

Un type d'heuristique très utilisé considère les fonctions d'évaluation, qui en mesure le rôle de chaque clause examinée. Le rôle d'une hypothèse peut se révéler par la quantité d'information apportée, par la réduction de la base de connaissance achevée ou par sa possibilité de discrimination comme l'heuristique Gain de Foil (First-Order Inductive Learner).

L'utilisation de la PLI pour les données spatiales permet une vision plus distincte que la structure de graphe multi-étiqueté. La représentation relationnelle est plus signifiante, elle peut exploiter des relations implicites, autres niveaux de granularité dans les relations.

La PLI a été utilisé dans des problématiques de classification supervisée de données à caractère spatial, l'algorithme S-TILDE [73] inspiré de TILDE (Top-down Induction Logical DEcision tree), c'est une méthode de classification par arbre de décision basée sur la logique du premier ordre, elle permet une représentation cartographie graphique des classes.

A travers cette PLI, on tente de trouver des solutions pour des problèmes complexes dans un espace très large. Toutefois, l'induction logique de la PLI n'est pas indispensable pour prendre en compte les phénomènes de cycles ou de multiplicité des relations non déterminées dans la fouille de relations. C'est pour cela qu'il devienne indispensable de trouver les motifs de graphes présents dans les données.

Ces approches de la fouille de relations et celle de la PLI, sont nommées par extraction de connaissances à partir de graphes. Ce sont des méthodes de fouille de graphes, dont le but est de concevoir des algorithmes efficaces de recherches de données se basant sur les graphes.

III.3.4 Les graphes comme support d'information.

Les graphes sont devenus de plus en plus importants et dynamiques en mathématique et en informatique. Cette progression monotone est due à l'utilisation par de nombreuses applications telles que la chimie, l'électronique, les mathématiques, l'informatique. . .

La théorie des graphes a vu le jour par le mathématicien Leonhard Euler (1736)[74] en essayant de résoudre le problème des ponts de Königsberg [74], cependant il a fallu deux cent ans pour aborder la théorie des graphes[75] .

Les graphes ont d'abord été considérés comme curiosité mathématique et comme moyens d'étude pour les jeux logiques dans l' exemple du déplacement du cavalier du jeu d'échec devant passer par chaque case de l'échiquier une fois tout en revenant à la case de départ. Ils sont devenus maintenant indispensable dans divers domaines humains. D'où l'intérêt porté aux graphes par plusieurs mathématiciens.

Cette théorie représente un ensemble d'objets complexes avec les relations entre leurs attributs. De nos jours la théorie des graphes est présente en algèbre, géométrie et algorithmique.

Un graphe est lié à plusieurs informations représentées soient par les sommets ou les arêtes entre deux sommets. La figure ci-dessous donne, selon Alain Bretto(2012)[76], la représentation simple d'un graphe :

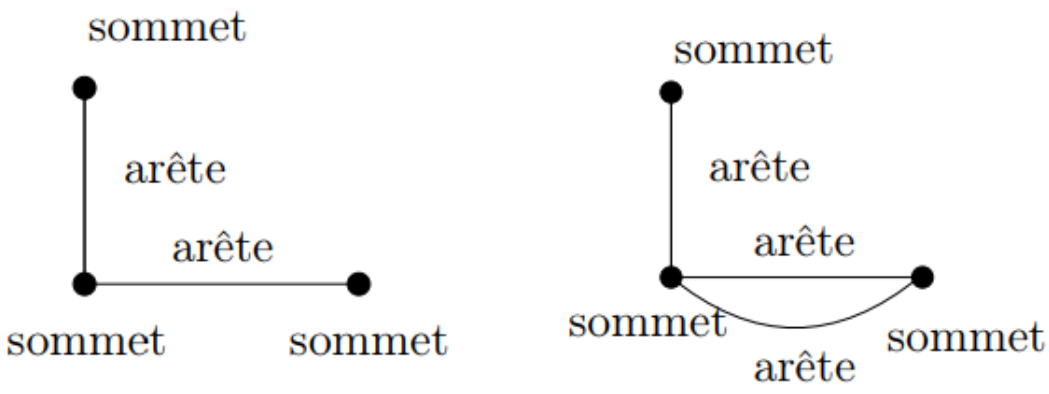


Figure III.7 : Exemple de graphe avec arête simple et arête multiple [76].

Un graphe est donc un triplet Γ= (V, E, N) où :

- V est l'ensemble des sommets du graphe ; il sera commode d'utiliser la notation V (Γ) pour désigner l'ensemble des sommets du graphe Γ.
- N est un ensemble qui sert à étiqueter les arêtes (par exemple N = {1, 2,..., p}, N = {bleu, rouge, vert, . . ., violet}, N = N. . .).
- E ⊂ P2(V) × N est l'ensemble des arêtes ; notation E = E(Γ).

(On note P2(V) l'ensemble des parties de V à 1 ou 2 éléments ({x} ou {x, y}).)

Une arête a ∈ E s'écrit a = ([x, y], n), x, y ∈ V, n ∈ N ; x et y sont les extrémités de a et n son étiquette ; a est incidente à x et y ; x et y sont dits adjacents ; si x = y, l'arête est une boucle.

Restriction : nous supposerons, dans les graphes que nous considérerons, que pour tous x, y ∈ V l'ensemble {a ∈ E : x et y incidents à a} est fini.

Pour représenter un graphe dans le plan ou dans l'espace, on matérialise généralement les arêtes par des segments ou des courbes.

Deux arêtes a et b sont adjacentes si elles ont (au moins) une extrémité commune.

La fonction d'incidence ε : E −→ P2(V) est définie par ε(a) = [x, y] si a = ([x, y], n).

Pour x, y fixés dans V l'ensemble {a ∈ E, ε(a)=[x, y]}, de cardinal p ≥ 1, est appelé p-arête ; c'est l'ensemble {([x, y], n1), ([x, y], n2),..., ([x, y], np)},

Où ni ∈ N, i = 1,..., p, sont les étiquettes de la p-arête. Si p = 1, on l'appelle également arête simple et on le note simplement {x, y} ; tandis que si p ≥ 2, on dit que c'est une multi-arête ou une arête multiple.

Une p-boucle est une p-arête dont les extrémités coïncident.

On prendra garde à bien distinguer la notion d'arête de celle de multi-arête, dans le sens où une multi-arête est généralement constituée de plusieurs arêtes.

L'exemple ci-dessous schématise un graphe comportant des multi-arêtes et une 2-boucle.

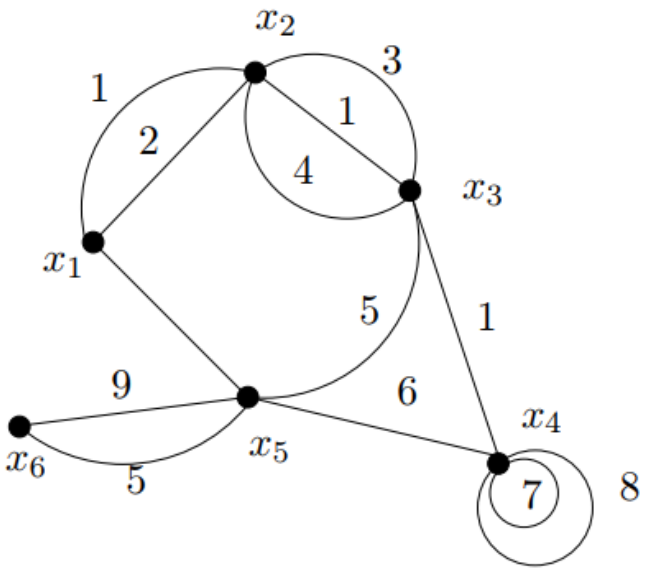


Figure III.8: Exemple de graphe avec arête simple et arête multiple [76].

Le graphe est définit par :

V = {x_1, x_2, x_3,..., x_6}, N = {1, 2,..., 9},

E = {([x_1, x_2], 1), ([x_1, x_2], 2), ([x_2, x_3], 3), ([x_2, x_3], 1), ([x_2, x_3], 4), ([x_3, x_5], 5),

([x_3, x_4], 1), ([x_4, x_5], 6), ([x_4, x_4], 7), ([x_4, x_4], 8), ([x_5, x_6], 9), ([x_5, x_6], 5)}.

On remarque que les étiquettes peuvent se répéter, de ce fait les arêtes ([x1, x2], 1) et ([x2, x3], 1) sont distinguées par le fait que x3 = x1. Ce graphe contient une 2-boucle (c'est-à-dire une 2-arête dont les extrémités coïncident), une 3-arête, deux 2-arête, les autres sont des arêtes simples.

- On dit que Γ est un graphe simple lorsqu'il ne contient que des arêtes simples.
- Un graphe est dit connexe si pour toute paire de sommets x, y ∈ V, il existe une chaîne entre x et y : on dit alors que les sommets x et y sont connectés.
- On définit aussi l'ordre d'un graphe par le nombre de sommets.

Le degré d'un sommet x ∈ V est le nombre d'arêtes incidentes à x : une boucle incidente à x contribue, par définition, deux fois dans le calcul du degré de x. Le degré de x sera noté dΓ(x) ou simplement d(x) et il correspond donc au nombre d'occurrences du sommet x comme extrémité d'arêtes a ∈ E :

d(x) = | {a ∈ E : ∃ y = x tel que ε(a)=[x, y]}| + 2| {a ∈ E : ε(a)=[x, x]}|, où |X| désigne le cardinal de X.

On définit un graphe orienté[77] par :

Un graphe orienté ou digraphe $\overrightarrow{\Gamma}$ (ou simplement Γ) est un triplet $\overrightarrow{\Gamma} = (V, \overrightarrow{E}; N)$ défini de la manière suivante :

- V est l'ensemble des sommets ; notation V = V ($\overrightarrow{\Gamma}$) ;
- $\overrightarrow{E} \subset V \times V \times N$ est l'ensemble des arcs ; notation $\overrightarrow{E} = \overrightarrow{E}(\Gamma)$;
- N est un ensemble servant à étiqueter les arcs.

Un arc $a \in \overrightarrow{E}$ sera noté a = ((x, y), n) : l'arc va de x vers y.

La figure ci-dessous représente un graphe orienté :

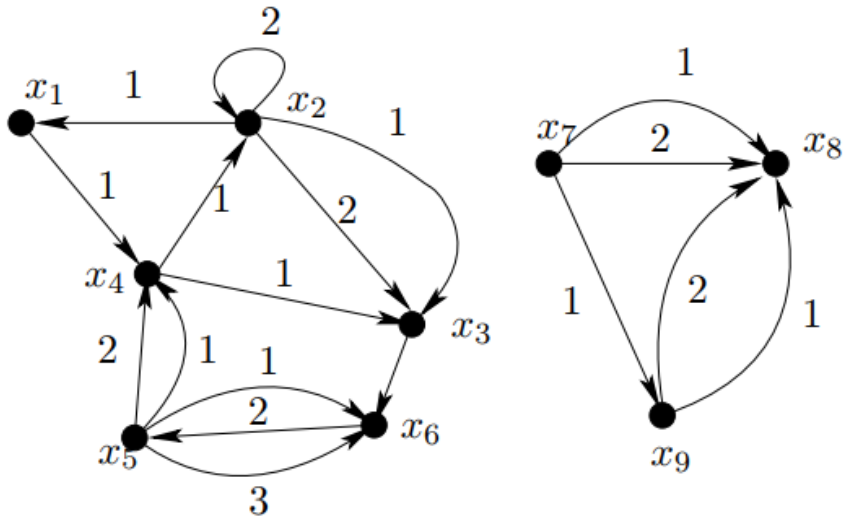


Figure III.9 : Exemple de graphe orienté [76].

Donc on peut dire que : un graphe connexe Γ= (V ; E ; N) est un graphe orientable, s'il existe une orientation des arêtes de telle sorte que le graphe orienté obtenu par orientation des arêtes, soit fortement connexe.

Les graphes interviennent dans plusieurs domaines de structuration d'informations, comme l'UML (Langage de Modélisation Unifié) qui utilise la notion de graphes orientées en diagramme de classe, de collaboration, d'états …

Ainsi on les trouve dans les bases de données relationnelles représentant les tables avec les relations, dans les réseaux informatiques, les réseaux de neurones et même dans la représentation d'algorithmes en ordinogrammes…

Le tableau de la figure ci-dessous met en valeur les utilisations des graphes :

Type de schémas
organigrammes
schémas de relations fonctionnelles
schémas de flux
schémas de réseaux opérationnels
arbre (ou hiérarchie)
schéma de processus
cartes et objets à l'échelle
diagramme de Venn
schéma d'influence
hiérarchie
schéma de causalité
schéma mathématique
diagramme de Gantt

Figure III.10: Exemple d'utilisation de graphes [78] dans la classification et la collecte de données quantitatives.

Un autre exemple de diagramme basé sur les graphes, c'est le diagramme d'état transition d'UML représentant les états d'un être humain liés au travail et à l'âge :

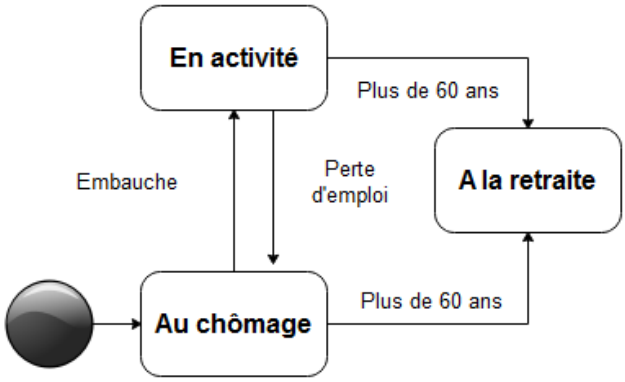


Figure III.11 : Exemple d'UML diagramme d'état transition (réalisée par nos soins).

En conséquence les graphes sont utilisés pour le calcul, la visualisation de l'information et la transformation de données d'une forme non significative à une forme plus exploitable et lisible afin d'en extraire la connaissance utile.

III.4. Conclusion

Il existe plusieurs méthodes de fouille, qui dépondent de la nature et de la quantité des données à fouillées et des questions auxquelles on cherche une réponse.

Certaines sont utilisées dans l'apprentissage automatique, surtout celles de Clustering comme par exemple les méthodes des plus proches voisins(PPV)[79], ou de classification supervisée, en particulier symbolique, comme les arbres de décision [80] et les méthodes de régression statistique.

La fouille des données, comme technique de recherche scientifique identifiée, ne s'est pas contentée de l'assemblage hybride de méthodes préexistantes, mais elle s'est développée rapidement avec l'apprentissage automatique et l'intelligence artificielle.

Chapitre IV. Analyse des algorithmes de Clustering pour le traitement de l'information.

IV.1 Introduction.

La classification de l'information est un domaine de recherche vague et difficile à explorer d'où l'apparition des techniques de regroupement, appelé souvent le Clustering .il convient de différentier entre une classification non supervisée de celle supervisée.

Dans l'analyse de données statistique, on associe un individu à une classe parmi plusieurs classes prédéfinies d'avance. Mais dans le cas d'une classification non supervisée, les classes ne sont pas connues d'avance, on regroupe alors les individus ou objets possédants des propriétés communes à partir d'un grand nombre de données, d'où la complexité du regroupement et d'identification du nombre de classes.

Une telle classification a pris naissance dans les analyses de données archéologiques (Classer les objets selon l'âge), médicales (Classer les malades selon l'âge, le poids, les symptômes…)…

Dès les années cinquantaines de telles techniques sont utilisées dans la taxonomie, ou taxinomie qui est une branche de science biologique qui a pour but de définir les organismes vivants et de les classer en entités nommées taxons afin de les reconnaître des clés de détermination dichotomiques.

Par la suite d'autres utilisations de la classification dans le traitement de données textes, de reconnaissances d'images d'extraction de la connaissance, ce qui a poussé les chercheurs à se concentrer dans la recherche des algorithmes et des techniques de Clustering au fur et à mesure du développement d'outils informatiques, d'où l'apparition d'une science visant les processus d'extraction de connaissances à partir de données (ECD).

Avant d'aborder le détail des techniques de Clustering (regroupement ou classification), on va commencer par donner quelques définitions de base sur lesquelles va se baser la suite de notre étude.

Définition formelle [81] du Clustering.

Soit un ensemble de n objets $X = \{x_1, ..., x_n\}$, et la matrice de dissimilarité D sur cet ensemble, telle que

- $d(x_i, x_j)$: la dissimilarité entre les deux objets x_i et x_j.
- D de taille : $n \times n$, contenant des valeurs dans [0, 1].

De telle définition est propre aux données relationnelles.

Actuellement les techniques de classification non-supervisée se basent sur la vision orientée objet, dans cette vision on définit des vecteurs d'objets, ce sont des données appartenant à l'ensemble V tel que :

- $V = \{v_1, ..., v_p\}$ de variables descriptives, telles que
- $v_j(x_i)$ désigne la valeur de l'objet $x_i \in X$ pour la variable $v_j \in V$.

Le Clustering génère un ensemble de t clusters C = {C1, ..., Ct} tel que :

- chaque cluster Ca est un sous-ensemble de X : $Ca \subset X$
- l'union des clusters couvre l'ensemble des objets de départ : $\bigcup^{t}_{a=1} C_a = X$.

Définition : Partition stricte[82].

C : est une partition de X si et seulement si C vérifie les propriétés suivantes :

1. $C_a \subset X$ pour tout $C_a \in C$
2. $\bigcup^{t}_{a=1} C_a = X$.
3. $C_a \cap C_b = \emptyset$ pour (a, b) tel que $a \neq b$.

Dans cette définition les clusters sont disjoints, un objet de X appartient à un seul cluster puisque l'intersection de ses partitions est vide.

Définition pseudo-partitions.

C : est une pseudo-partition de X si et seulement si C vérifie les propriétés suivantes :

1. $C_a \subset X$ pour tout $C_a \in C$
2. $\bigcup^{t}_{a=1} C_a = X$.
3. $C_a \subseteq C_b$ ssi a = b.

Les clusters peuvent avoir des intersections mais ne doivent pas avoir quelques-uns inclus dans d'autres.

Le processus de création de telle partition peut s'exprimer par l'expression de t fonctions de données binaires suivantes :

$$U_a : X \rightarrow \{0,1\},\ a = 1 \ldots t \text{ avec } U_a(xi) = \begin{cases} 1 \ si \ xi \ \in Ca \\ 0 \ \ sinon \end{cases}$$

Quand on a des variables de valeurs réelles on parle des partitions floues.

Définition partitions floues.

Une partition floue de X, notée $C = \{C_1, ..., C_t\}$, est définie par la donnée de t fonctions

$$U_a : X \rightarrow [0,1] \text{ avec } a = 1 \ldots t$$

$U_a(xi)$: c'est le degré d'appartenance de l'objet x_i au cluster C_a.

Arbre Hiérarchique :

Les dendrogrammes « arbres hiérarchiques » sont parmi les procédés de Clustering de données sous forme hiérarchique ou pseudo-hiérarchique existants.

Définition de l' Hiérarchies :

P est un ensemble de parties non vides sur X, P peut être une hiérarchie si on vérifie les propriétés suivantes :

1. $X \in P$
2. pour tout $x_i \in X$, $\{x_i\} \in P$
3. pour tout h, h' $\in P$, $h \cap h' \in \{\emptyset, h, h'\}$
4. pour tout $h \in P$, $\bigcup \{h' \in P : h' \subset h\} \in \{h, \emptyset\}$.

L'arbre est composé de l'ensemble X qui est la racine et les feuilles sont les $\{x_i\}$.

L'intersection de deux clusters est vide sinon un devrait contenir l'autre comme fils et lui serait considéré comme son père.

Définition pseudo-hiérarchies :

P est un ensemble de parties non vides sur X, P peut être une pseudo-hiérarchie si on vérifie les propriétés suivantes :

1. X ∈ P
2. pour tout xi ∈ X, {xi} ∈ P,
3. pour tout h, h' ∈ P, h ⋂ h' = ∅ ou h ⋂ h' ∈ P,
4. il existe un ordre (total) θ sur X compatible avec P.

Définition d'ordre :

θ est ordre compatible avec l'ensemble P de parties de X, si tout élément de h ∈ P est connexe selon θ.

Définition.

Soit h une partie, h peut être connexe selon l'ordre θ, avec x et y sont les bornes (i. e. le plus petit et le plus grand élément) de h selon θ, si la condition suivante est vérifiée :

{z compris entre x et y selon θ} <=> {z ∈ h}

Une pseudo-hiérarchie (pyramide) est définie de la façon suivante : chaque cluster peut avoir plusieurs prédécesseurs. Aussi, si un ordre θ sur X existe on peut alors visualiser cette pyramide.

Centroïdes et médoïdes.

Les algorithmes de Clustering représentent les clusters par des points, pour simplifier le traitement et ces points peuvent être soient des Centroïdes ou des médoïdes, d'où les définitions suivantes :

Définition : Centroïde

On dit x* est centroïde du cluster Ca s'il est point dans ν et respecte la propriété suivante :

$$\forall j = 1, \dots, p, \qquad v_j(x^*) = \frac{1}{C_a} \sum_{x_i \in C_a} v_j(x_i) \quad \text{(Equation IV.1)}$$

Les v_1,….. $v_p$sont des variables quantitatives mesurables et peuvent être soient discrètes ou continues .Le centroïde est donc le centre de gravité de cluster, c'est la valeur moyenne des objets de cluster pour chaque v_1, dans cette définition le centroïde peut ne pas être un élément du cluster.

Si les variables descriptives v_i sont qualitatives on cherche alors l'élément le plus significatif du cluster qu'on nomme le médoïde ou prototype.

Définition : médoïde

Considérons C_a un cluster d'objets sur ν, et possédant une mesure de dissimilarité sur ν, on définit l'objet x^* : médoïde du cluster C_a tel que :

$$x^* = argmin_{x_i \in C_a} \frac{1}{|C_a|} \sum_{x_j \in C_a} d(x_i, x_j)$$ (Equation IV.2)

Dans ce cas le médoïde est l'objet du cluster qui ressemble en moyenne le plus à tous les objets de ce cluster, donc la dissimilarité de ce médoïde avec tous les objets du cluster est minimale.

Format de Clusters

Dans certaines situations l'allure et le format des clusters peuvent être très utiles pour l'analyse et l'extraction de connaissance lors de la visualisation de données et du choix des clusters. On peut trouver des clusters imprégnés dans d'autres ou isolés, selon la représentation des données. Dans ce cas on introduit la notion de forme concave ou convexe à deux dimensions ou trois…

Figure IV. 1: Exemples de clusters à gauche concaves[83] et à droite convexes (réalisée par nos soins).

La notion de convexe signifie que les objets devraient être organisés autour d'un centre, c'est l'exemple typique de l'algorithme k-moyennes qui permet de construire des clusters entourant un centroïde ou médoïde, la visualisation des objets peut être claire selon la dimension de la représentation. Lorsque les clusters sont concaves, il est difficile d'analyser certains objets puisqu'ils peuvent être cachés par d'autres et l'extraction de la connaissance peut aboutir à des erreurs.

Notion d'outlier[84] :

Un outlier est une constatation rare qui ne respecte aucune règle, on peut douter de ce type d'observations non habituelles.

Il n'y a pas de définition exacte d'outlier mais on utilise «Distance-Based outlier», elle coïncide avec celle de Hawkins notée par DB (m, δ)-Outlier.

Définition : Outlier

On dit qu'un objet $x^* \in X$ est un *DB* (m, δ)-outlier «Distance-Based outlier» : s'il existe un sous-ensemble X' de X, constitué d'au moins m objets $x_{1'}, ..., x'_m$, tel que :

$$\forall x_i' \in X', \ d(x^*, x_i') > \delta$$

avec *d* est une mesure de dissimilarité définie sur X.

Définitions : rayon, diamètre et inertie de cluster

Considérons C_a un cluster d'objets de X.

Soit x_a^* *le* centre du cluster et d la distance sur X ou mesure de la dissimilarité.

Le rayon du cluster est tel que : $rayon\ (C_a) = \max_{\{\ x_i \in C_a\ \}} d(x_i, x_a^*)$

Le diamètre du cluster est tel que : $diam\ (C_a) = \max_{\{\ x_i\ , x_j \in C_a\ \}} d(x_i, x_j)$

L'inertie du cluster est la somme des carrés des distances au centre :

$$I_{intra}(C_a) = \sum_{x_i \in C_a} d(\ x_i, x_a^*\ )^2 \qquad \text{(Equation IV.3)}$$

Soit le schéma du Clustering $C = \{C_1, \dots, C_t\}$ et x_i^* le centre du cluster i. on définit l'inertie inter-clusters comme suit :

$$I_{inter}(C) = \sum_{i=2}^{t} \sum_{j<i} d(x_i^*\ , x_j^*)^2 \qquad \text{(Equation IV.4)}$$

La variance du cluster est alors la moyenne des carrés des distances au centre :

$$V(C_a) = \frac{1}{|C_a|} \sum_{x_i \in C_a} d(\ x_i, x_a^*)^2 \text{ (Equation IV.5)}$$

IV.2 Les étapes principales du Clustering.

Pour une bonne classification « Clustering »le processus doit inclure trois phases essentielles : la première dans laquelle on prépare les données à utiliser, après on choisit l'algorithme de Clustering et enfin on exploite les clusters trouvés dans des analyses et des prises de décisions. Le schéma ci-dessous résume le processus du Clustering :

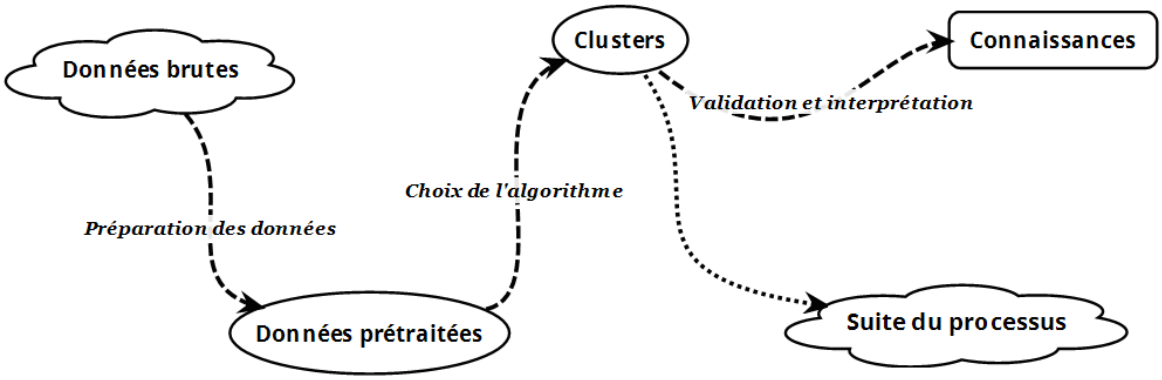


Figure IV.2 : Les phases du processus de Clustering. (Réalisée par nos soins)

IV.2 .1 Préparation des données :

Dans cette première partie, on assimile les données (objets) à des variables (attributs) quantitatives, continues ou discrètes, ou à des variables qualitatives ou structurées.

L'objectif est de choisir à partir de ces variables celles pouvant être lisibles normalisées avec le même type de données et d'en exclure celles erronées et répétitives.

A travers des mesures ou indices discriminants, les variables sont utilisées ou éliminées dans la classification supervisée ou non. Le classement peut se faire à travers des méthodes de filtrage pour écarter les variables non utiles ou à travers des classifieurs pour bien définir les classes de données à utiliser.

L’importance d’une variable à utiliser n’est pas toujours facile à déterminer sans les règles prédéfinies aux préalables associés aux classes, d’où la complexité du choix du nombre de classes et des algorithmes de Clustering.

Comme critères utilisés dans les algorithmes de Clustering on note l’indice de similarité, dissimilarité ou encore la distance ou mesure de proximité.

Le choix de la mesure de distance entre variables (objets) dépend des données étudiées et des objectifs et il est déterminant pour le processus de classification.

Chaque domaine d’application nécessite ces propres indices de mesure de proximité : par exemple, les données textuelles n’ont pas le même indice de mesure que les données spatiales qui utilisent la distance euclidienne dans un espace multidimensionnel.

Notons que :

- La distance Euclidienne [85]: le type de distance le plus couramment utilisé. Il s'agit d'une distance géométrique dans un espace multidimensionnel.

$$distance(x, y) = \sqrt{\sum_i (x_i, y_i)^2}$$ (Equation IV.6)

- La distance Euclidienne au carré permet de "surpondérer" les objets atypiques (éloignés), en élevant la distance euclidienne au carré.

$$distance(x, y) = \sum_i (x_i, y_i)^2$$ (Equation IV.7)

- La distance du City-block (Manhattan) (2006) : cette distance est simplement la somme des différences entre les dimensions.

$$distance(x, y) = \sum_i |x_i, y_i|$$ (Equation IV.8)

IV.2.2 Choix des algorithmes du Clustering.

Pour bien choisir l’algorithme de Clustering, il est essentiel de connaitre, tout d’abord le type de données à utiliser. Si elles sont mesurables ou non, qualitatives ou quantitatives, leurs tailles, les critères et les règles qui déterminent le nombre des clusters à considérer, leurs natures, leurs formes …Aussi doit-on savoir la façon de visualisation de ces données à deux dimension ou plus, sous forme de d’endogames, de partitions, de graphes …En effet la taille de données peut être déterminante pour le choix du processus de Clustering, surtout quand les données sont des images. En plus la performance des machines diminue avec la complexité des algorithmes de Clustering .Le temps de traitement augmente avec le nombre d’objets et une complexité non linéaire. Les algorithmes utilisés sont souvent de complexité linéaire comme l’algorithme des k-moyennes…

De même la nature des données influe souvent sur le choix d’algorithme de Clustering, notons que, notre objectif est de construire une collection d’objets similaires au sein d’un même groupe ou dissimilaires quand ils appartiennent à des groupes différents, d’où la nécessité d’une matrice de similarité pour mesurer la distance ou la qualité.

Différents types permettent de standardiser la nature des données, on site les types intervalles, binaires, nominales, catégories, ordinales, ratio …

Par ailleurs on peut définir d’autres types de données comme les Centroïdes à travers l’algorithme des k-moyennes …

La visualisation de données (après traitement et classification) peut se présenter sous plusieurs aspects, selon l'algorithme de Clustering, les clusters peuvent être de formes significatives « typiques » dont les objets entourent des Centroïdes ou sous forme d'outlier n'obeillissant à aucune règle. La taille et la forme des clusters dépendent de la densité des données prises en compte, des mesures de similarité entre les objets voisins, du choix d'algorithmes de classification et des conditions et hypothèses définies au préalable dans le model de classification.

Les connaissances à extraire à partir de données interviennent dans le choix de méthodes de groupement à utiliser et de clusters à trouver, soit sous forme de partitions, de dendrogrammes ou de graphe…

IV.2.3 Exploitation des clusters

L'exploitation des clusters peut se faire dans la phase d'apprentissage et de correction ou bien pour avoir des connaissances utilisables dans la prise de décision finale. Selon la forme, la densité, la taille des clusters, les classes semblables seraient localisées de façon claire sinon on serait obligé de changer la technique de Clustering et d'aller chercher une autre.

Toutefois, il n'est pas obligatoire de décrire les clusters s'ils vont servir à corriger les erreurs d'apprentissages à travers la mesure statistique de la qualité des informations, mais dans le cas contraire c'est-à-dire quand le Clustering a pour rôle principal l'exploration de classes, ceux-ci doivent être décris par des matrices de similarité ou bien par des méthodes de description de classes de conception utilisant des variables décrivant les objets de données .

IV.3 Exploitation de la similarité de données.

L'analyse de données exploite fréquemment la notion de ressemblance dite de similarité de données, cette mesure existe dans la majorité des algorithmes de Clustering : de reconnaissance de formes (RF), d'apprentissage symbolique (AS), d'analyse de données (AD) et des sciences cognitives(SC).

Comme exemple d'algorithmes utilisant la similarité on trouve :

- L'Analyse en Composantes Principales (ACP).
- Le k-moyennes (k-means en anglais) : la méthode des nuées dynamiques.
- La classification ascendante hiérarchique (CAH).
- L'algorithme EM (Algorithme espérance-maximisation).

- **Définition formelle de similarité[86] :**

Une « mesure » de similarité est une application symétrique s de $X \times X$ dans $\mathbb{R}^+$ telle que $s(x_i,\ x_i)$ est maximale et $s(x_i,\ x_j)$est d'autant plus élevée que les descriptions des objets x_i et x_j sont similaires.et le contraire est vrai.

- **Propriété de Minimalité**

Une mesure de dissimilarité $d : X \times X \to \mathbb{R}^+$ vérifie la propriété de minimalité si et seulement si : $\forall x_i \in X, d(x_i, x_i) = 0$ (Equation IV.9)

- **Propriété de Symétrie**

Une mesure de dissimilarité $d : X \times X \rightarrow \mathbb{R}^+$ est symétrique si et seulement si :

$$\forall x_i, x_j \in X, d(x_i, x_j) = d(x_j, x_i) \quad \text{(Equation IV.10)}$$

- **Propriété d'Identité**

Une mesure de dissimilarité $d : X \times X \rightarrow \mathbb{R}^+$ vérifie la propriété d'identité si et seulement si :

$$\forall x_i, x_j \in X, d(x_i, x_i) = 0 \Longrightarrow x_i = x_j$$

- **Propriété d'Inégalité triangulaire**

Une mesure de dissimilarité $d : X \times X \rightarrow \mathbb{R}^+$ vérifie l'inégalité triangulaire si et seulement si :

$$\forall x_i, x_j, x_k \in X, d(x_i, x_j) \leq d(x_i, x_k) + d(x_k, x_j)$$

- **Propriété d'Inégalité ultramétrique.**

Une mesure de dissimilarité $d : X \times X \rightarrow \mathbb{R}^+$ vérifie l'inégalité ultramétrique si et seulement si :

$$\forall x_i, x_j, x_k \in X, d(x_i, x_j) \leq max\{d(x_i, x_k), d(x_k, x_j)\}$$

La correspondance de distance entre deux objets et la similarité s'exprime par l'équation suivante :

$$\forall x_i, x_j \in X, s(x_i, x_j) = d_{max} - d(x_i, x_j) \quad \text{(Equation IV.11)}$$

, avec d_{max} la distance maximale entre $x_i et\ x_j$ et $d(x_i, x_j)$ la dissimilarité.

IV.3.1 Variables numériques et la similarité.

Souvent pour une mesure de similarité on a besoin de définir ce que c'est une distance.

En généralise : La distance de Minkowski, ou p-distance, généralise la distance **euclidienne** : c'est la racine $p^{ème}$ de la somme des valeurs absolues des écarts à la puissance *p.*

Cette distance de **Minkowski** est exprimée par la formule suivante :

$$d(x_i, x_j) = (\sum_{k=1}^{p} |v_k(x_i) - v_k(x_j)|^l)^{1/l} \quad \text{(Equation IV.12)}$$

- avec $v_k(x_i)$ la valeur de l'objet x_i sur la variable v_k
- pour *l* =1 correspond à la distance de **Manhattan**
- pour *l* =2 correspond à la distance **Euclidienne**
- pour *l* =∞ correspond à la distance de **Tchebychev**

Quand on a des données de type texte on utilise la distance du cosinus ci-dessous :

$$d(x_i, x_j) = \frac{x_i . x_j}{\|x_i\| \|x_j\|} \text{ (Equation IV.13)}$$

Avec «.» le produit scalaire et la norme $\|x_i\| = \sqrt{\sum_k v_k(x_i)^2}$ (Equation IV.14)

Une autre distance : distance de Mahalanobis [87] introduite en 1961 par Prasanta Chandra Mahalanobis [88] .Elle diffère de la distance euclidienne, elle prend en compte la variance et la corrélation de la série de données. Elle traite les composants de vecteurs séparément.

Sa définition est : $d(x_i, x_j) = \sqrt{(x_i - x_j)^T S^{-1} (x_i - x_j)}$ (Equation IV.15)

, avec S : la matrice de variance/covariance. $(x_i - x_j)^T$: la matrice transposée .

Cette distance néglige les données réparties.

IV.3.2 Similarité et variables symboliques [89]

Pour des données symboliques, l'évaluation à travers des formules de calcul n'est pas valide, donc on est sensé passer par décrire la similarité en passant par la technique suivante : associer à chaque valeur d'objet une donnée binaire ensuite on dénombre les propriétés communes à celle différentes pour des couples d'objets .D'où la définition des indices de Jaccard [90] et de Rand suivants :

Indices de Jaccard [91] pour les deux ensembles A et B :

$$J(A,B) = \frac{cardinal(A \cap B)}{cardinal(A \cup B)} \qquad \text{(Equation IV.16)}$$

Indices de Rand pour l'ensemble Al de paires (x_i, x_j) qui vérifie le cas l avec :

$a_l = |A_l|$ est :

$$Rand = \frac{a_1 + a_4}{a_1 + a_2 + a_3 + a_4} \qquad \text{(Equation IV.17)}$$

Dans notre situation pour le couple(x_i, x_j) l'indice de Rand noté R et celui de Jaccard J [92] :

$$J(x_i, x_j) = \frac{d_{++}}{d_{++} + d_{+-} + d_{-+}} \qquad \text{(Equation IV.18)}$$

$$R(x_i, x_j) = \frac{d_{++} + d_{--}}{d_{++} + d_{+-} + d_{-+} + d_{--}} \qquad \text{(Equation IV.19)}$$

Avec d_{++}le nombre de propriétés partagées, d_{+-} respecté par $x_i$seul,d_{-+} respecté par $x_j$seul et d_{--} non respecté par les deux objets.

D'autres indices peuvent être exploités comme ceux de Fowlkes [93] et Mallows qui utilisent des variables symboliques binaires « espace de description binaire à deux valeurs » pour évaluer la similarité des objets. Dans le cas où les variables symboliques possèdent plus que deux valeurs, on devrait discrétiser les valeurs en intervalles découpés …

A noter que dans le cas d'un langage textuel, là où les variables symboliques possèdent plus de deux valeurs, le formalisme de Martin et Moal exprime la similarité entre deux objets par le nombre de propriétés qui satisfait un langage « L » constitué par un ensemble de termes {t_1,…, t_n}, la similarité étant exprimée par la formule suivante :

$$similarité_L(x_i, x_j) = \frac{1}{|L|} \sum_{t \in L} \delta_t\,(x_i, x_j) \qquad \text{(Equation IV.20)}$$

Avec $\delta_t(x_i, x_j)$ =1 si les deux variables $x_i et\ x_j$ satisfont la propriété t.et 0 si un seulement.

En conclusion, dans cette partie, nous avons donné des définitions de la similarité selon la nature et le type d'objets à traiter. En général si les objets sont numériques mesurables, la distance euclidienne est la plus adapté à exprimer la mesure de la similarité, dans le cas contraire la formule de Martin et Moal peut être exploitées pour des variables symboliques à deux valeurs ou plus.

IV.4 Types de Clustering

Les techniques de Clustering peuvent être de différents types, on distingue trois types du processus de Clustering :

a. Le Clustering flou «fuzzy-clustering [94] ».
b. Le Clustering dur « hard/crisp-clustering [95] ».
c. Le Clustering avec recouvrement «soft-Clustering [96]».

Le Clustering dite « flou [97] » visualise les données dans une disposition simple .les objets peuvent être partagées par les clusters selon leurs propriétés, et on parle des fonctions d'appartenance ou d'affection liés à l'objet $\boldsymbol{x_i}$ .Ces fonctions d'affection notées $\{\boldsymbol{u_j}(\boldsymbol{x_i})\}_{\boldsymbol{j=1,\dots,t}}$ peuvent être relatives (probabilistes) ou absolues (possibilistes).

La règle suivante doit être satisfaite pour une fonction d'affection probabiliste :

$$\forall\, x_i \in X, \sum_{j=1}^{t} \boldsymbol{u_j}(\boldsymbol{x_i}) = \mathbf{1} \text{ (Equation IV.21)}$$

Le processus de Clustering flou est appliqué dans le traitement de données textes et d'images, le problème qui se pose c'est le nombre important d'informations de clusters trouvés et la difficulté de l'extraction de la connaissance à partir de ces clusters, puisque des propriétés des objets peuvent être partagé la prise de décision peut être incertaine.

Pour le Clustering « Dur », les critères sur les objets sont forts, ils les classent selon les contraintes dans une et une seule classe .Alors il y aura une sorte de disposition hiérarchique stricte et sans ambiguïté. Le point fort de ce type de Clustering, c'est le faite d'avoir une structure simple compréhensible et si l'on veut préalablement mettre des règles de classification dans ce cas, c'est le type de classement le plus adéquat.

Le troisième type de Clustering : le Clustering avec recouvrement qui regroupe les précédents, il rallie les avantages des approches dures et floues. Dans ce cas un objet peut être lié avec une contrainte dure à une ou plusieurs classes, d'où la souplesse de l'organisation et la simplicité de représentation des données.

IV.5 Les méthodes du Clustering.

Les méthodes de Clustering sont multiples .le partitionnement de données et la hiérarchisation poussent à les utiliser sous forme paramétrique ou non.

Aussi, leur utilisation se trouve influencée par des algorithmes de nature probabilistes lors du partitionnement des données.

D'autres méthodes d'algorithmes sont choisies en s'appuyant sur le résultat du Clustering qu'on veut avoir. Ici les contraintes appliquées aux traitements de données et le choix du type de processus de Clustering (Dur, flou, ou avec recouvrement) nous amène à utiliser soit des graphes ou des fonctions probabilistes.

Nous abordons dans ce qui suit les différents processus de Clustering, les algorithmes et les représentations graphiques et visuelles utilisés, nous citons comme exemple le Clustering par partitionnement, hiérarchique, par mélange de densités de probabilités, par grille, conceptuel et par densités …

Les schémas ci-dessous donnent une vision globale sur les différentes techniques de Clustering :

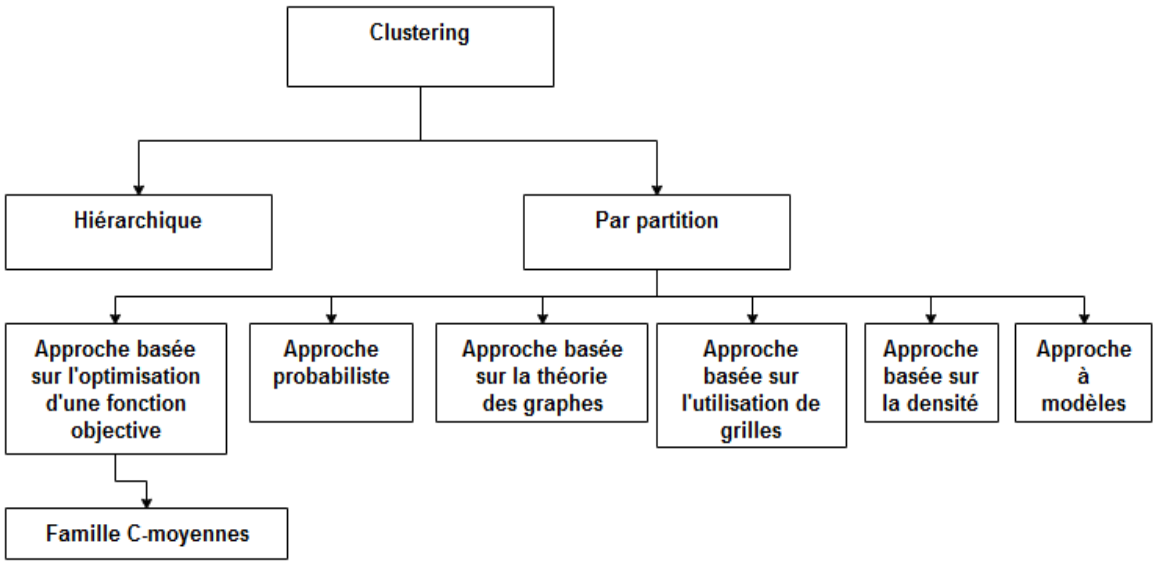


Figure IV.3 : différentes approches en Clustering (réalisée par nos soins):

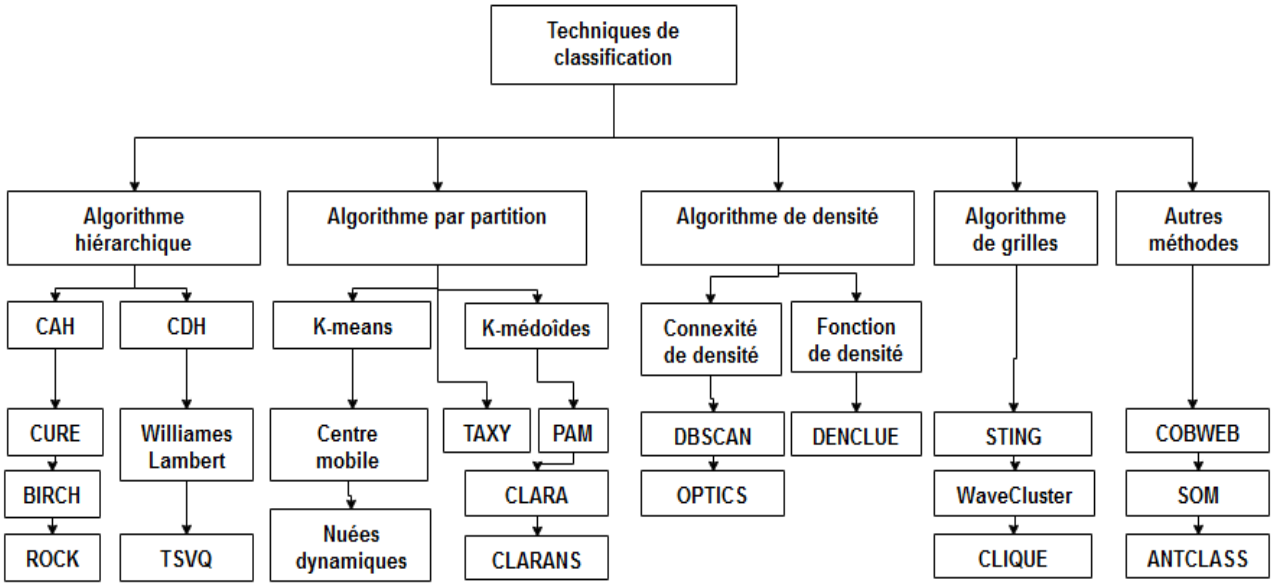


Figure IV.4 : Techniques de classification(réalisée par nos soins)

IV.5.1 Le Clustering hiérarchique [98-99]

La hiérarchisation des clusters nécessite l'utilisation des arbres appelés aussi dendrogrammes .pour être appelé ainsi, il doit respecter les conditions suivantes :

- Le cluster X compose la racine de l'arbre et contient l'ensemble d'objets.
- Dans chaque nœud se trouve un cluster $C_i \subset X$.
- Les $\{x_1\}, \ldots, \{x_n\}$ constituent les feuilles de l'arbre.
- Les objets présents dans le nœud contiennent tous les objets contenus dans ces fils.
- Au fur et à mesure de la construction de l'arbre on indexe les niveaux ou paliers de l'arbre.

La visualisation des objets dans le dendrogramme est claire surtout pour des niveaux et de nombre d'objets moins grands. Les partitions peuvent être identifiées facilement et sans interférence.

Dans le cas de traitement de cinq objets $x_1, \dots, x_5$ avec trois niveaux l =3 on représente le dendrogramme ci-dessous :

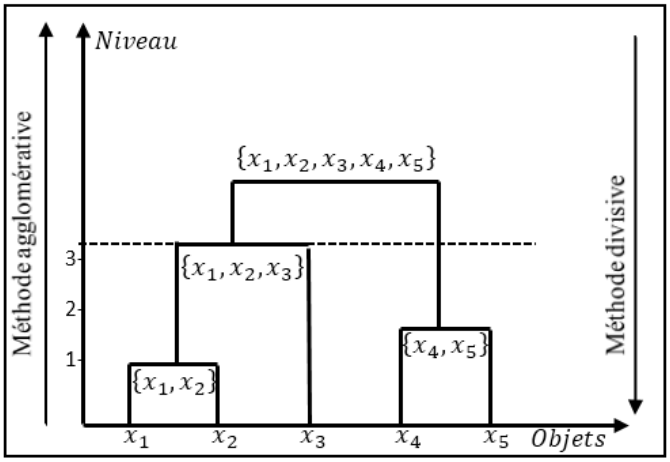


Figure IV.5 : Exemple de dendrogramme à 3 niveaux (réalisée par nos soins).

Le choix du niveau l =3 de l'exemple précèdent permet de définir les deux partitions $\{x_1, x_2, x_3\}$ et $\{x_4, x_5\}$.

Pour réaliser de tel arbre, on peut soit remonter des fils jusqu'à atteindre la racine : dans ce cas on parle de la méthode agglomérative, ou de la racine jusqu'aux fils pour la méthode divisible. Dans tous les cas on peut stopper le regroupement une fois on atteint le nombre de clusters fixé d'avance ou la mesure de qualité d'arrêt du processus.

Par la suite, nous traitons quelques algorithmes respectant la hiérarchie stricte ou pseudo-hiérarchie .Le problème qui se pose c'est qu'il n'est pas possible d'effectuer un ajustement ou correction une fois la décision de fusion ou de partage en clusters a été exécutée.

C'est-à-dire, si une fusion particulière ou une division en partition est effectuée, et l'on se rend contre plus tard qu'elle est mal choisie, la méthode ne permet pas sa rectification.

Les algorithmes qu'on va traiter par la suite sont :

- Algorithme (DIANA) : « DIvisive ANAlysis ».
- Algorithme (SAHN): « Sequential Agglomerative Hierarchical and Nonoverlapping».
- Algorithme (CAP) : « Classification Ascendante Pyramidale ».
- Algorithme (HFCM): « Hierarchical Fuzzy-k-Means».

➢ Algorithmes de la hiérarchie stricte :

Le premier exemple d'algorithme qu'on traite et celui de l'algorithme (DIANA [100]), il commence par la racine et divise jusqu'à attendre les singletons, tout d'abord on cherche l'objet le plus irrégulier et par la suite on lui associe les objets voisins de telle sorte à former les clusters. Les étapes de l'algorithme (DIANA) sont les suivantes :

Algorithme DIANA

En entrée : on déclare une matrice de similarité $\boldsymbol{S}$ de l'ensemble $\boldsymbol{X}$ des objets à traiter.

En sortie : on déclare une Hiérarchie $\boldsymbol{P}$ qu'on va trouver comme résultat.

Début

1. On initialise à 1 cluster la racine, on met $\boldsymbol{C = \{\{x_1, \dots, x_n\}\}\ et\ P = C}$
2. On choisit le cluster atypique $\boldsymbol{c \in C}$ de diamètre maximum.
3. On localise dans ce cluster un objet $\boldsymbol{x^*}$ ou celui qui a la plus faible

similarité moyenne avec les autres objets :

$$x^* = \arg min_{x_i \in c} \frac{1}{|c| - 1} \sum_{j \neq i} s(x_i, x_j)$$

avec x^* initialise à chaque fois un nouveau cluster c^*

4. On calcule pour chaque objet $x_i \notin c^*$:

$$s_i = \left[moyenne\ des\ s(x_i, x_j), x_j \in c \backslash c^*\right] - \left[moyenne\ des\ s(x_i, x_j), x_j \in c^*\right]$$

5. Si x_k est l'objet pour lequel s_k est minimal et si $s_k \leq 0$ alors on ajoute x_k à c^*
6. On répète la boucle de 3 et 4 jusqu'à $s_k > 0$
7. On remplace c par $c \backslash c^*$ et c^* dans C puis on ajoute $c \backslash c^*$ et c^* dans P.
8. On répète les étapes de 2 à 7 jusqu'à ce que chaque cluster contienne un singleton.
9. On retourne P qui correspond à la hiérarchie.

Fin.

Algorithme IV.1 : DIANA « DIvisive ANAlysis »

Il convient de constater que l'exécution de l'algorithme devienne rapide avec la diminution du nombre d'objets. La division en cluster peut se faire jusqu'à atteindre la condition d'une similarité minimale et d'un cluster contenant un seul objet. La complexité est de l'ordre de $O(n^2)$.

Le deuxième exemple d'algorithme qu'on traite est celui de l'algorithme agglomératif nommé : (SAHN [101-102]) : « Sequential Agglomerative Hierarchical and Nonoverlapping» :

Algorithme SAHN [103]

En entrée : on déclare une matrice de similarité S de l'ensemble X des objets à traiter.

En sortie : on déclare une Hiérarchie P qu'on va trouver comme résultat.

Début

1. On initialise à 1 cluster la racine, on met $C = \{\{x_1\}, \ldots, \{x_n\}\}\ et\ P = C$
2. On identifie les deux clusters $c_k et\ c_l$ de C les plus proches selon S :

$$(c_k, c_l) = arg\ max_{(c_i, c_j) \in C^2} \{sim(c_i, c_j)\}$$

3. On remplace $\{c_k\}$**et** $\{c_l\}$ par $\{c_k \cup c_l\}$ dans C et ajouter $\{c_k \cup c_l\}$ à P
4. On recalcule la matrice S en suite.
5. Si $C \neq \{\{x1\ , \ldots, x_n\}\}$ « la racine » on revient à l'étape 2.
6. On retourne P contenant les parties non vides de X représentant la hiérarchie.

Fin.

Algorithme IV.2 : SAHN « Sequential Agglomerative Hierarchical and Nonoverlapping»

Cet algorithme (SAHN)[104] effectue l'opération inverse : à partir de clusters composés de singletons on construit des clusters par regroupement des plus proches voisins en se basant sur la mesure de similarité .C'est une construction efficace des partitions jusqu'à la racine. Le temps dépend du nombre n d'objets à traiter et dans le pire des cas la complexité est de l'ordre de $O(n^2)$.

- Les algorithmes pseudo-hiérarchiques.

La représentation pseudo-hiérarchie permet de visualiser les clusters sous forme de pyramide. Chaque nœud de la pyramide peut avoir deux prédécesseurs. A un niveau de la pyramide on forme une pseudo-partition de X.

L'algorithme CAP [105] (Classification Ascendante Pyramidale) met en évidence cette représentation de pyramide .L'algorithme peut s'étendre à une représentation spatiale à travers l'algorithme (CAPS) pour rendre plus simple la visualisation des pseudo-partitions et pour éviter d'avoir des interférences au niveau de croisements et d'inversions dans les pyramides.

Les principales étapes de L'algorithme CAP (Classification Ascendante Pyramidale) sont :

Algorithme CAP

En entrée : on déclare une matrice de similarité $\boldsymbol{S}$ de l'ensemble $\boldsymbol{X}$ des objets à traiter, $\boldsymbol{\theta}$ **un ordre sur** $\boldsymbol{X}$.

En sortie : $\boldsymbol{P}$ une pyramide qu'on va trouver comme résultat.

Début

1. On initialise $\boldsymbol{P}$ à n singletons : $\boldsymbol{P} = \{\{\boldsymbol{x_1}\}, \dots, \{\boldsymbol{x_n}\}\}$.
2. On associe les deux groupes $\boldsymbol{C_i}, \boldsymbol{C_j} \boldsymbol{\epsilon P}$ les plus proches selon $\boldsymbol{S}$ et qui vérifient les conditions suivantes :
   a. $\boldsymbol{C_i}$ **et** $\boldsymbol{C_j}$ ont été associés une seule fois au plus
   b. $\boldsymbol{C_i} \cup \boldsymbol{C_j}$ est connexe selon l'ordre $\boldsymbol{\theta}$.
   c. Soit $\boldsymbol{C_k} \in \boldsymbol{P}$ tel que $\boldsymbol{C_i} \subset \boldsymbol{C_k}$ avec $\boldsymbol{C_i}$ contient l'une des extrémités de la partie connexe $\boldsymbol{C_k}$
3. On ajoute $\boldsymbol{C_i} \cup \boldsymbol{C_j}$ dans $\boldsymbol{P}$
4. On répète l'étape 2 jusqu'à ce qu'un groupe de $\boldsymbol{P}$ soit égale à $\boldsymbol{X}$
5. On retourne $\boldsymbol{P}$ la pyramide.

Fin.

Algorithme IV.3 : CAP (Classification Ascendante Pyramidale)

Pour mieux visualiser L'algorithme CAP[106] (Classification Ascendante Pyramidale) voir le schéma ci-dessous :

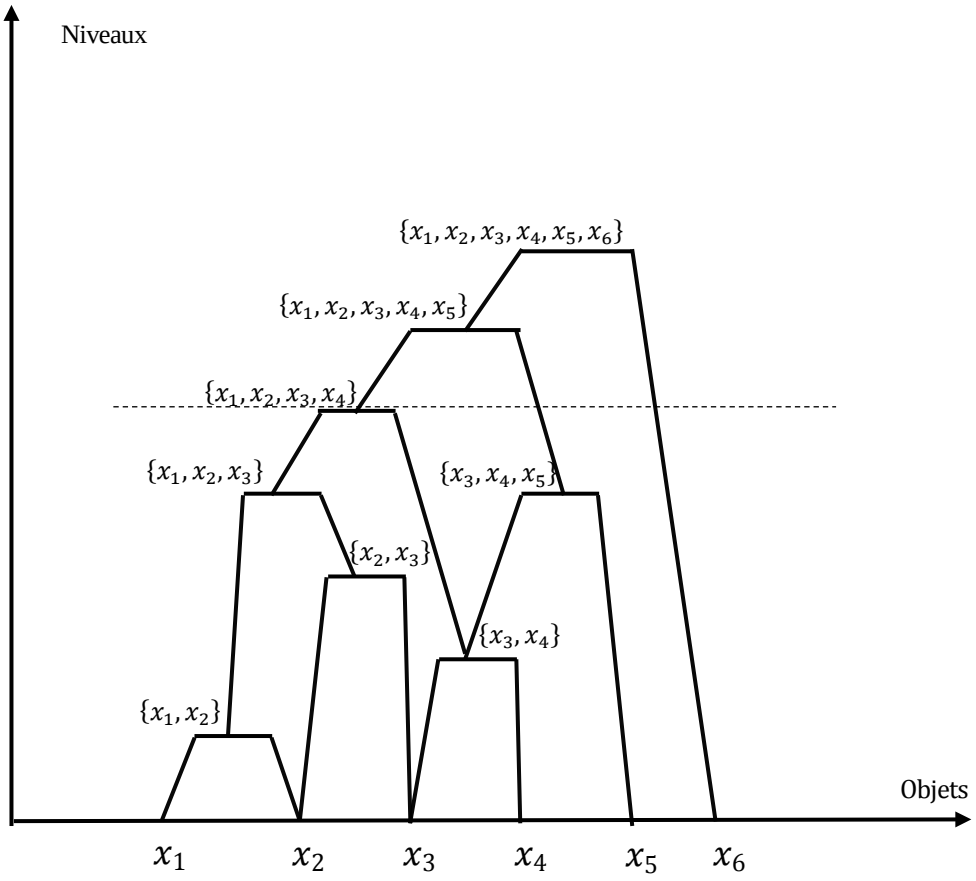


Figure IV.6 : Dendrogramme d'une pyramide (réalisée par nos soins)

Dans ce dendrogramme nous pouvons définir la pseudo-partition limitée par le trait en pointillé suivante : $C = \{\{x_1, x_2, x_3\}, \{x_3, x_4\}, \{x_3, x_4, x_5\}, \{x_6\}\}$

- Les algorithmes des hiérarchies floues

Les algorithmes flous ne sont pas souvent utilisés, comme il est signalé auparavant, les objets partages des propriétés communes, d'où la difficulté d'un classement stricte .On les utilise dans l'exploitation et le découpage d'images et de textes.

Comme algorithme des hiérarchies floues on utilise l'algorithme des k-moyennes [107] flou hiérarchique (FCM).Cet algorithme s'exécute de façon itérative sur chaque nœud, il commence par la racine qui regroupes tous les objets et décent selon le nombre des itérations k qu'on devrait fixer au départ. On arrête le traitement une fois la mesure de la qualité des clusters est valide. On trouve alors des clusters flous partageant des objets. D'où la difficulté d'interpréter le dendrogramme résultant.

L'algorithme des k-moyennes flou hiérarchique, noté HFCM (Hierarchical Fuzzy-k-Means) est définie par les étapes suivantes :

Algorithme des k-moyennes [108]

En entrée : on déclare l'ensemble $\boldsymbol{X}$ des objets à traiter, la mesure de qualité $\boldsymbol{\phi}$ et un seuil de qualité $\boldsymbol{\alpha}$.

En sortie : $\boldsymbol{P}$ une hiérarchie floue comme résultat.

Début

1. On initialise un cluster racine :$\boldsymbol{C} = \{\boldsymbol{X}\}$ et $\boldsymbol{P} = \boldsymbol{C}$
2. Pour chaque cluster $\boldsymbol{c_i} \in \boldsymbol{C}$ tel que $\boldsymbol{\phi}(\boldsymbol{c_i}) < \boldsymbol{\alpha}$:
3. On applique l'algorithme FCM en recherchant le nombre optimal $\boldsymbol{k_i}$ de clusters : $\boldsymbol{c_i} \longrightarrow \{\boldsymbol{c_{i,1}}, \dots, \boldsymbol{c_{i,k_i}}\}$
4. On supprime $\boldsymbol{c_i}$ dans $\boldsymbol{C}$ et on ajoute $\boldsymbol{c_{i,1}}, \dots, \boldsymbol{c_{i,k_i}}$ dans $\boldsymbol{C}$ et dans $\boldsymbol{P}$
5. On reprend l'étape 2 jusqu'à ce que tous les clusters $\boldsymbol{c_i}$ dans $\boldsymbol{C}$ soient tel que $\boldsymbol{\phi}(\boldsymbol{c_i}) > \boldsymbol{\alpha}$
6. On retourne $\boldsymbol{P}$ le résultat final.

Fin.

Algorithme IV.4 : Algorithme des k-moyennes

Cet algorithme correspond à une hiérarchie divisive, on commence de la racine et on descend jusqu'aux singletons. Le schéma ci-dessous illustre la difficulté de l'interprétation du dendrogramme.

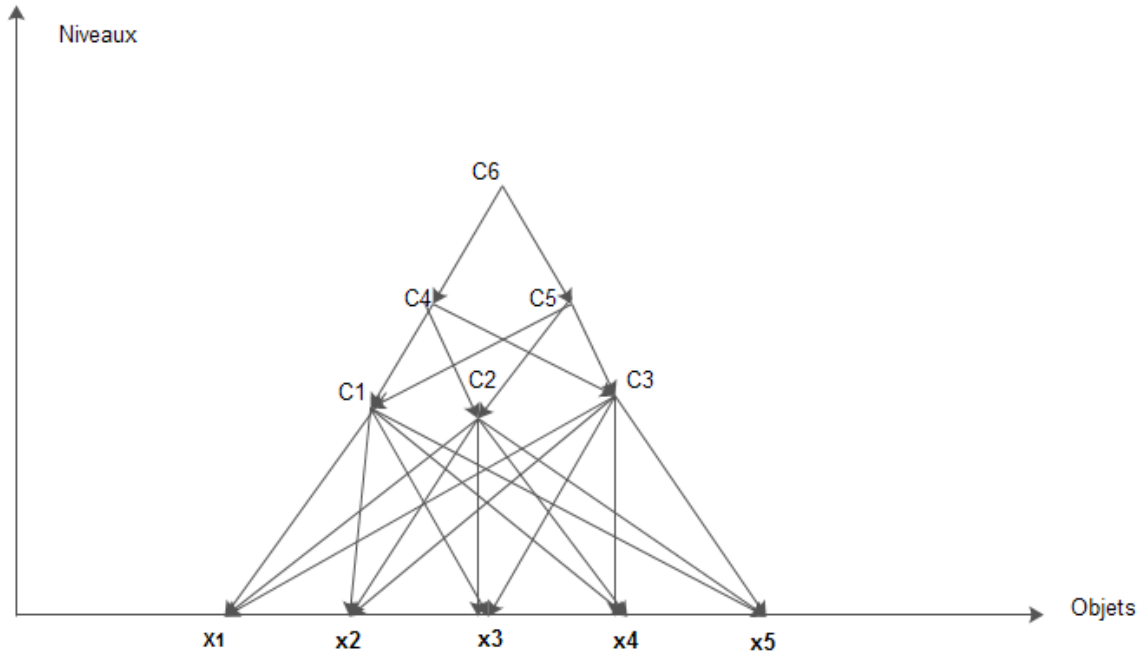


Figure IV.7 : Exemple de dendrogramme d'une hiérarchie floue (réalisée par nos soins).

Il semble nettement que la reproduction des dendrogrammes flous est pénible puisque chaque nœud de la hiérarchie doit accepter un vecteur de taille n. En définitive, ce dendrogramme participe légèrement à la conception du processus de Clustering.

IV.5.2Le Clustering par partitionnement

Inversement aux méthodes hiérarchiques vues précédemment, les algorithmes de partitionnement donnent, comme résultat, une partition de l'espace des objets à la place d'dendrogramme.

Le but est de chercher un schéma pour améliorer la mesure de qualité parmi un ensemble de schémas de Clustering. Ce schéma peut être transformé de manière interactive à partir d'un choix aléatoire, au début. Les objets analysés vont être associés autour d'un centre dit de gravité. L'algorithme des k-moyennes permet de retrouver certaines bonnes représentations de clusters et selon la valeur de k itérations fixée d'avance et même par le choix du type de partitionnement, stricte, pseudo-partition ou flou.

Dans le cas du partitionnement strict, on définit les étapes de l'algorithme de k-moyennes (k-means) comme suit :

Algorithme k-moyennes [109] :

En entrée : on déclare k le nombre de clusters et d'une mesure de similarité sur l'ensemble $\boldsymbol{X}$ des objets à traiter

En sortie : une partition qui va contenir les clusters $\boldsymbol{C} = \{\boldsymbol{c_1}, \dots, \boldsymbol{c_k}\}$

Début

1. On initialise dans $\boldsymbol{X}$ aléatoirement de k centres $\boldsymbol{x^*_{1,0}}, \dots, \boldsymbol{x^*_{k,0}}$
2. On construit une partition initiale $\boldsymbol{C_0} = \{\boldsymbol{c_1}, \dots, \boldsymbol{c_k}\}$ par allocation de chaque objet $\boldsymbol{x_i} \in \boldsymbol{X}$ au centre le plus proche :
$$\boldsymbol{C_l} = \{\boldsymbol{x_i} \in \boldsymbol{X} / \ \boldsymbol{d}(\boldsymbol{x_i}, \boldsymbol{x^*_{l,0}}) = \boldsymbol{min_{h=1,\dots,k}} \ \boldsymbol{d}(\boldsymbol{x_i}, \boldsymbol{x^*_{h,0}})\}$$
3. On calcule les Centroïdes des k classe obtenues $\boldsymbol{x^*_{1,}1}, \dots, \boldsymbol{x^*_{k,1}}$
4. On construit une nouvelle partition $\boldsymbol{C_t} = \{\boldsymbol{c_1}, \dots, \boldsymbol{c_k}\}$ par allocation de chaque objet $\boldsymbol{x_i} \in \boldsymbol{X}$ au centre le plus proche :
$$\boldsymbol{C_l} = \{\boldsymbol{x_i} \in \boldsymbol{X} / \ \boldsymbol{d}(\boldsymbol{x_i}, \boldsymbol{x^*_{l,t}}) = \boldsymbol{min_{h=1,\dots,k}} \ \boldsymbol{d}(\boldsymbol{x_i}, \boldsymbol{x^*_{h,t}})\}$$
5. On calcule les Centroïdes des k classes obtenues : $\boldsymbol{x^*_{1,t+1}}, \dots, \boldsymbol{x^*_{k,0=t+1}}$
6. On répète les étapes 4 et 5 tant que des changements se réalisent d'un schéma $\boldsymbol{C_t}$ à un schéma $\boldsymbol{C_{t+1}}$ ou jusqu' à un nombre $\boldsymbol{\tau}$ d'itérations
7. On retourne la partition finale $\boldsymbol{C}$

Fin

Algorithme IV.5 : K-moyennes (k-means) pour partitionnement strict

On fixe le nombre k des clusters recherchés. On initialise aléatoirement k objets représentant les centres des clusters initiaux. Après on affecte chaque objet au cluster dont le centre est le plus proche. Ensuite tant qu'il y a au moins un objet qui change de cluster, on doit mettre à jour les centres des clusters en fonction des objets qui leurs sont affectés, et réactualiser aussi les affectations des objets aux clusters en fonction de la mesure de distance de proximité des nouveaux centres et ainsi de suite.

Ce processus de Clustering possède les problèmes suivants :

- Le résultat final dépend beaucoup du choix du cluster initial.

- Il peut être très sensible aux outliers d'où une mauvaise interprétation de clusters.
- L'algorithme peut dans certaines situations générer des clusters vides ce qui rend le traitement long et incohérent.

Toutefois, la complexité de l'algorithme est linéaire, ce qui réduit le temps de traitement d'un nombre très grand d'objets. D'autre part, il s'arrête souvent sur un optimum local. Pour être atteint, l'optimum global nécessite, lui, les algorithmes génétiques comme autres techniques.

Cet algorithme n'est appliqué que sur des objets dont les attributs sont numériques pour pouvoir calculer les Centroïdes. Dans le cas d'attributs symboliques ou catégoriels, on utilise l'algorithme des k-médoïdes [110] développé par Kaufman et Rousseeuw en 1990.

L'algorithme des k-médoïdes utilise des médoïdes au lieu des Centroïdes, il se base sur une matrice de dissimilarité ce qui n'est pas le cas dans l'algorithme k-means. En outre, il permet de minimiser une somme de dissimilarité à la place de celle des distances Euclidiennes carrées. Les étapes définissant cet algorithme sont les suivantes :

Algorithme des k-médoïdes [111]

En entrée : on déclare k le nombre de clusters et l'ensemble $\boldsymbol{X}$ des objets à traiter :

$$\boldsymbol{X} = \{\boldsymbol{x_1}, \dots, \boldsymbol{x_n}\}$$

En sortie : un ensemble de clusters $\boldsymbol{C} = \{\boldsymbol{c_1}, \dots, \boldsymbol{c_k}\}$

Début

1. On choisit aléatoirement les k premiers médoïdes.
2. On associe chaque médoïde à un cluster.
3. On affecte chaque objet au cluster représenté par le médoïde le plus proche à cet objet.
4. On recalcule les positions des k médoïdes.
5. On répète l'étape 3 et 4 jusqu'à ce que l'ensemble des clusters se stabilise.

Fin.

Algorithme IV.6 : Des k-médoïdes

Le médoïde de chaque cluster est calculé en cherchant un objet $\boldsymbol{x_i}$ à l'intérieur du cluster qui minimise la somme suivante $\sum_{j \in C_i} d(x_i, x_j)$ avec C_i est le cluster contenant l'objet x_i et $d(x_i, x_j)$ la distance entre les deux objets.

Comme point fort : cet algorithme s'adapte à n'importe quel type de données, et il n'est pas sensible aux objets isolés comme les outliers. Comme point faible, c'est qu'il est couteux puisque la complexité est de l'ordre de $O(k * (n-k)^2$.

Dans le cas d'une Pseudo-partitions et partitions floues, on utilise des variables floues dans un processus de Clustering flou, c'est l'algorithme des k-moyennes flous (Fuzzy k-means) dont les étapes sont définies de la façon suivante :

Algorithme des k-moyennes flous

En entrée : on déclare k le nombre de clusters et une mesure de dissimilarité *d* sur l'ensemble $\boldsymbol{X}$ des objets à traiter, on note *T* le nombre maximum d'itération et un poids *m* >1, ainsi qu'un seuil $\boldsymbol{\varepsilon > 0}$.

En sortie : on déclare une partition floue $\boldsymbol{C = \{c_1, \dots, c_k\}}$ définie par la fonction d'appartenance $\{\boldsymbol{u_h}\}_{\boldsymbol{h=1\dots k}}$

Début

1. On choisit t =0, on choisit aléatoirement une partition initiale $\{\boldsymbol{u_h}\}_{\boldsymbol{h=1\dots k}}$
2. On calcule les centres de gravités $\boldsymbol{x^*_{1,t}, \dots, x^*_{k,t}}$ de chacune des k classes à l'instant t

$$\boldsymbol{v_j(x^*_{h,t}) = \frac{1}{\sum_{x_i \in X}[u_{h,t}(x_i)]^m} \sum_{x_i \in X} [u_{h,t}(x_i)]^m . v_j(x_i)}$$

3. On calcule les nouvelles valeurs d'appartenance $\{\boldsymbol{u_{h,t+1}(x_i)}\}_{\boldsymbol{h=1,\dots,k}}$ de chaque objet $\boldsymbol{x_i}$ à chaque centre de classe $\boldsymbol{x^*_{h,t}}$

$$\boldsymbol{u_{h,t+1}(x_i) = \frac{[d(x_i, x^*_{h,t})]^{2/(1-m)}}{\sum_{h=1}^{k}[d(x_i, x^*_{h,t})]^{2/(1-m)}}}$$

4. On calcule les centres de gravité de chaque classe $\boldsymbol{x^*_{1,t+1}, \dots, x^*_{k,t+1}}$ de chacune des k classes à l'instant *t+1*
5. On calcule le déplacement global $\boldsymbol{E_t = \sum_{h=1}^{k} d(x^*_{h,t+1}, v_{h,t})}$
6. Si $\boldsymbol{E_t \leq \varepsilon}$ alors on retourne la partition floue définie par($\{\boldsymbol{x^*_{h,t+1}, u_{h,t+1}}\}_{\boldsymbol{h=1,\dots,k}}$), sinon $\boldsymbol{t = t + 1}$ et on retourne à l'étape 3

Fin.

Algorithme IV.7 : Les k-moyennes flous

Un critère de variabilité $\boldsymbol{J_m}$ intra classe permet d'optimiser les itérations de cet algorithme, il est définit ainsi :

$$J_m = \sum_{i=1}^{n} \sum_{h=1}^{k} [u_h(x_i)]^m . d(x_i, x^*_h) \qquad \text{(Equation IV.22)}$$

Ou m est une grandeur nommée « fuzzifier » permettant d'accroître les dissimilitudes entre les objets distants et les objets centraux d'une même classe,$m \in \ ]1, \infty[$

Si m s'approche de 1, l'ensemble $\{0,1\}$ définie les valeurs des fonctions d'appartenance.

Pour le cas de données symbolique ou catégoriales on définit l'algorithme des k-médoïdes flou suivant :

Algorithme des k-médoïdes flous ;

En entrée : on déclare k le nombre de clusters et une mesure de dissimilarité *d* sur l'ensemble $\boldsymbol{X}$ des objets à traiter, on note *T* le nombre maximum d'itération et un poids *m* >1, ainsi qu'un seuil $\boldsymbol{\varepsilon > 0}$.

En sortie : on déclare une partition floue $\boldsymbol{C = \{c_1, \dots, c_k\}}$ définie par les médoïdes des classes $\boldsymbol{x^*_1, \dots, x^*_k}$

Début

1. On choisit t =0, on choisit aléatoirement k médoïdes $\boldsymbol{x^*_{1,t}, \dots, x^*_{k,t}}$ dans $\boldsymbol{X}$

2. On calcule les valeurs d'appartenance $\{\boldsymbol{u_{h,}(x_i)}\}_{\boldsymbol{h=1,\dots,k}}$ de chaque objet $\boldsymbol{x_i}$à ***chaque*** **médoïde** $\boldsymbol{x^*_{h,t}}$

$$\boldsymbol{u_{h,t}(x_i) = \frac{\left[d(x_i, x^*_{h,t})\right]^{-1/(m-1)}}{\sum_{l=1}^{k}\left[d(x_i, x^*_{l,t})\right]^{-1/(m-1)}}}$$

3. On recalcule les médoïdes $\boldsymbol{x^*_{1,t+1}, \dots, x^*_{k,t+1}}$

$$\boldsymbol{x^*_{h,t+1} = arg\ min_{x_i \in X} \sum_{l=1}^{n} u_{h,t}(x_l)^m . d(x_i, x_l)}$$

4. Si $\boldsymbol{\forall h, x^*_{h,t+1} = x^*_{h,t}\ ou\ si\ t \geq T}$ alors on retourne la partition floue définie par $(\boldsymbol{\{x^*_{h,t+1}, u_{h,t+1}\}_{h=1,\dots,k}})$, sinon $\boldsymbol{t = t + 1}$ et on retourne à l'étape 2

Fin.

Algorithme IV.8 : Des k-médoïdes flous

Cet algorithme est couteux vue le nombre d'itérations et le nombre d'objets à traiter, la complexité est de l'ordre de $O(n^2)$.

IV.5.3 Le Clustering par mélange de densités de probabilités

Dans cette partie, on suppose les clusters comme des groupes cohérents de grande densité. L'algorithme qu'on traite considère deux paramètres : le premier met en valeur un $\boldsymbol{\varepsilon}$ représentant le « radius », distance maximale entre les objets de l'ensemble X voisins de l'objetx_i. Le deuxième paramètre $\boldsymbol{M}$: c'est le nombre minimum d'objets du voisinage pour traduisant la densité de celui-ci.

L'algorithme EM [112] (**Expectation et maximisation) de Dempster de 1977 [113]** est fondé sur l'approche de l'estimation gaussienne avec le maximum de vraisemblance.

C'est un processus itératif qui se déroule comme suit :

- ✓ Sélection d'un objet non encore appartenant à un cluster.
- ✓ Si son voisinage respecte la règle de densité définie par $(\boldsymbol{\varepsilon, M})$, alors les objets correspondants vont être inclus au cluster courant.
- ✓ De la même manière, on répète pour chaque objet du voisinage.

Avant de lister les étapes de l'algorithme EM, on définit les variables suivants :

- ✓ $\mathcal{L}$ Le maximum de vraisemblance lié à l'estimation gaussienne.
- ✓ $\{\mu_h\}_{h=1,\dots,k}$: les valeurs moyennes des k gaussiennes.
- ✓ $\{\sigma_h^2\}_{h=1,\dots,k}$: les variances des k gaussiennes.
- ✓ τ_h : le paramètre du mélange qui ne dépend pas de la distribution gaussienne choisie.

E : une fonction qui correspond au logarithme népérien négatif de la vraisemblance : $E = -\ln(\mathcal{L}) = -\sum_{i=1}^{n} \ln p(x_i) = -\sum_{i=1}^{n} ln \sum_{h=1}^{k} \tau_h p(x_i|h)$ (Equation IV.23)

L'algorithme est le suivant :

Algorithme EM (Expectation Maximisation)[114] ;

En entrée : on déclare k le nombre de clusters ainsi qu'un seuil de tolérance $\boldsymbol{\varepsilon}$, et un l'ensemble des observations $\boldsymbol{X} = \{\boldsymbol{x_1}, \dots, \boldsymbol{x_n}\}$ des objets à traiter.

En sortie : on déclare l'ensemble $\{\boldsymbol{\mu_h}, \boldsymbol{\sigma_h^2}, \boldsymbol{\tau_h}\}_{\boldsymbol{h=1,\dots,k}}$ de k vecteurs.

Début

1. **On initialise k vecteurs de paramètres** $\{\boldsymbol{\mu_h^0}, \boldsymbol{\sigma_h^{20}}, \boldsymbol{\tau_h^0}\}_{\boldsymbol{h=1,\dots,k}}$
2. **On calcule à chaque étape t le vecteur** $\{\boldsymbol{\mu_h^{t+1}}, \boldsymbol{\sigma_h^{2t+1}}, \boldsymbol{\tau_h^{t+1}}\}_{\boldsymbol{h=1,\dots,k}}$ **à partir des estimations de l'étape précédente** $\{\boldsymbol{\mu_h^t}, \boldsymbol{\sigma_h^{2t}}, \boldsymbol{\tau_h^t}\}_{\boldsymbol{h=1,\dots,k}}$

$$\boldsymbol{v_j(\mu_h^{t+1}) = \frac{\sum_{i=1}^{n} p^t\,(h|x_i)\,.\,v_j(x_i)}{\sum_{i=1}^{n} p^t\,(h|x_i)}}$$

$$\boldsymbol{\sigma_h^{2t+1} = \frac{\sum_{i=1}^{n} p^t\,(h|x_i)\|x_i - \mu_h^t\|^2}{\sum_{i=1}^{n} p^t\,(h|x_i)}}$$

$$\boldsymbol{\tau_h^{t+1} = \frac{1}{n}\sum_{i=1}^{n} p^t\,(h|x_i)}$$

3. **On calcule la variation de la fonction d'erreur : le logarithme népérien de la vraisemblance négatif :** $\boldsymbol{\Delta^{t+1} = -\sum_{i=1}^{n} \ln(\frac{p^{t+1}(x_i)}{p^t(x_i)})}$

avec $\boldsymbol{p^{t+1}(x_i) = \frac{p^{t+1}(x_i|h).\tau_h^t}{p^t(x_i)}}$ **la probabilité trouvée par l'égalité de Bayes**

4. **Si** $\boldsymbol{\Delta > \varepsilon}$ **alors on retourne à l'étape 2, sinon on retourne les vecteurs** $\{\boldsymbol{\mu_h^{t+1}}, \boldsymbol{\sigma_h^{2t+1}}, \boldsymbol{\tau_h^{t+1}}\}_{\boldsymbol{h=1,\dots,k}}$

Fin.

Algorithme IV.9 : EM (Expectation Maximisation)

Cet algorithme présente des inconvénients comme le nombre croissant d'itérations qui rend le processus lent, de même le choix de k et l'initialisation peut intervenir dans le résultat obtenu. En plus le calcul de probabilité et le nombre de variables augmente le coût de calcul et fausse l'estimation des initialisations de paramètres initiaux.

IV.5.4 Le Clustering par grilles

L'espace de données est divisé en cellules (grouper des zones jugées denses sur la base d'un seuil fixé à priori, les zones déterminées peu denses permettent d'établir des limites.). Donc ce type d'algorithme est conçu pour des données spatiales. Une cellule peut être représentée par un cube, un terroir ou un hyper rectangle. Elle est un produit cartésien de sous intervalles d'attributs de données.

Le principe de base de ce type de Clustering, c'est le découpage de l'espace de « d » dimension en unités ou cellules trop petites pour former des partitions composées de granularité cubiques. Ici, on divise l'espace en une sorte de maillage de cellules denses et connectés : cellules forment une grille et elles sont voisines en termes de distance.

Trois exemples d'algorithmes utilisent le partitionnement par grille :

- ✓ L'algorithme **STING [115]** « Statistical Information Grid Approach to Spatial Data Mining » construit une hiérarchie de grille : un cluster est composé de cellules denses et connectés, pour chaque cellule du niveau courant, STING calcul l'intervalle de probabilité pour lequel des sous- cellules soient denses, et écarte les autres ; de même il répète le même traitement jusqu'à atteindre le plus bas niveau.

- ✓ L'algorithme de **WaveCluster** « Wavelet-Based Clustering» : il se base sur une transformation des signaux nommés « ondelettes ». Celui-ci suppose que l'espace des attributs est à d dimensions, et le découpe de façon à ce que chaque dimension i soit divisée en $m_i$intervalles, de telle sorte à obtenir une grille de $\Pi_i m_i$ cellules ou chaque objet est affecté à une cellule. La répartition des objets sur les cellules constitue un signal d-dimensionnel auquel est appliquée la technique des ondelettes pour former les clusters.

- ✓ **CLIQUE [116]** (Clustering In QUest [Agrawal 1998]) est un procédé basé sur la densité, il recherche automatiquement des sous-espaces de plus grande dimensionnalité où existent des classes de forte densité.

CLIQUE divise l'ensemble de l'espace des données en unités, si rectangulaires disjointes, et recherche les unités denses basées sur l'idée : si une région k-dimensionnelle est dense, la sous-région de dimension (k-1) la contenant devrait être aussi dense.

Quand les unités denses sont découvertes, plusieurs ensembles d'unités denses sont connectées pour constituer des classes.

Soit un espace à p dimensions, CLIQUE commence par découper chaque dimension de l'espace des données en un nombre fixe d'intervalles de même largeur, après identifie l'ensemble des cellules denses dans l'ensemble des sous-espaces de l'espace des données, ensuite détermine des classes comme un ensemble maximal de cellules denses contiguës.

Autrement dit l'algorithme de CLIQUE : comprend un Clustering des données en les projetant dans chaque dimension, on identifie les clusters denses dans les projections de données, ces clusters doivent correspondre aux régions denses de l'espace dans les dimensions. La méthode suppose que toute la base de données soit accessible pour les projections.

Dans ce qui suit on liste les différents algorithmes. On commence par l'algorithme STING.

L'algorithme STING (STatistical INformation Grid) exploite une grille multi résolution où les données, dans chaque cellule sont représentées par le nombre d'objets et par leurs moyennes.

Toute cellule est découpée de manière récursive en 4 sous-cellules. On commence par initialiser la cellule ancêtre représentant l'espace entier, on finit la récursivité si le seul est inférieur au nombre de point dans une cellule.

Chaque cellule contient les informations essentielles pour la recherche d'un point :

- ✓ La moyenne de toutes les valeurs dans la cellule,

- ✓ La variance de toutes les valeurs d'attribut dans la cellule,
- ✓ La valeur minimale et celle maximale des attributs dans la cellule,
- ✓ Le genre de distribution de la cellule : normale, uniforme, exponentielle…

Lorsqu'on remonte dans la hiérarchie, on calcule ces données statistiques en se basant sur celles du niveau inférieur.

STING est souple car il nécessite une seule passe sur les données et à chaque niveau de notre hiérarchie, on insère facilement un objet nouveau dans les cellules adéquates.

La complexité de cet algorithme dépend de « k » qui est le nombre de cellule, elle peut être de l'ordre de $O(k)$, ce nombre k est inférieur au nombre de données puisque l'algorithme écarte les données absurdes.

Pour commencer l'algorithme on fixe une base de données et on devrait répondre à une requête de type sélection basée sur les paramètres d'une cellule « point ».

Algorithme Sting [117] ;

Entrée : une base de données contenant des paramètres d'objets « cellules », et une requête de type SQL.

Sortie : ensemble de cellules représentant une couche ou grille.

Début

1. Déterminer une couche avec laquelle on commence.
2. Pour chaque cellule de cette couche, nous calculons l'intervalle de confiance (ou l'estimation de la marge) de la probabilité que cette cellule ait la réponse à la requête de la base de données posée au début.
3. Selon la valeur de l'intervalle calculé ci-dessus, nous marquons la cellule comme appropriée ou non appropriée.
4. Si cette couche est la couche inférieure, passer à l'étape 6 ; autrement, passer à l'étape 5.
5. on descend la structure de hiérarchie par un niveau. On passe à l'étape 2 pour ces cellules qui forment les cellules appropriées de la couche de niveau plus élevé.
6. Si les spécifications de la requête sont vérifiées, passer à l'étape 8 ; sinon, passer à l'étape 7.
7. Rechercher la chute de ces données dans les cellules appropriées et faire une transformation plus ultérieure. Renvoyer le résultat qui répond à l'exigence de la requête. Passer à l'étape 9.
8. Trouver les régions des cellules appropriées. Renvoyer ces régions qui répondent à l'exigence de la requête. Passer à l'étape 9.
9. arrêter.

Fin.

Algorithme IV.10 : STING

Le deuxième algorithme de WaveCluster, on découpe l'espace d'attributs en grilles comme suit :

Algorithme Wavelet-Based Clustering [118] ;
En entrée : on a un ensemble de données X multidimensionnelles.
En sortie : on doit trouver un ensemble de clusters.
Début
1. On découpe l'espace des attributs en cellules.
2. On affecter chaque objet a une cellule.
3. On applique la transformation par ondelettes.
4. On recherche les composants connectes (clusters)
5. On affecte les objets aux clusters.
Fin.

Algorithme IV.11 : Wavelet-Based Clustering

Un tel algorithme présente la particularité d'être efficace dans le cas des bases de données très grandes. La complexité de produire des clusters pour ce processus est de l'ordre de $O(n)$. Les résultats ne sont pas affectés par des outliers et la méthode n'est pas sensible au nombre d'objets d'entrée à traiter. WaveCluster est bien capable de trouver des clusters de formes arbitrairement complexes, telles que les clusters concaves ou autres. Dans ce processus, la connaissance préalable du nombre de clusters n'est pas exigée. Cependant, une évaluation du nombre prévu de clusters, aide énormément à mieux choisir la résolution appropriée des grilles.

Algorithme CLIQUE [119] ;
En entrée : une base de données D, un ensemble de points P, un seuil de la densité $\boldsymbol{\tau}$.
En sortie : ensemble de points.
Début
1. Discrétiser la base de données D sur l'ensemble P.
2. Déterminer les combinaisons denses de cellules de la grille qui possède le seuil minimum de densité τ en utilisant n'importe quel Algorithme d'exploration de motifs fréquence.
3. Créer le graphique dans lequel des combinaisons denses de grille sont connectées si elles sont adjacentes ;
4. Déterminer les composants connectés du graphique ;
5. Retourner le Couple de (point, sous-espace) pour chaque composant connecté

Fin.

Algorithme IV.12 : CLIQUE

Le point fort de cet algorithme c'est qu'il permet de trouver automatiquement des sous-espaces de la plus grande dimension, tels que des clusters de haute densité existant dans ces sous-espaces.

Par ailleurs, il présente les avantages suivants :

- il est insensible à l'ordre des enregistrements en entrée ;
- il n'impose pas une certaine distribution de données canoniques ;
- il s'exécute indépendamment de la taille de l'entrée ;
- il évolue excellemment avec le nombre de dimensions des données ;
- sa complexité est de l'ordre de $O(n)$.

IV.5.5 Le Clustering par densités

Le Clustering à base de densité utilise la notion de voisinage pour déterminer un cluster de noyau x_i .

$N_\varepsilon(x_i)$ *est le voisinage de* x_i : c'est l'ensemble de points de X dont la distance avec x_i est inférieure ou égale au rayon ε :

$$N_\varepsilon(x_i) = \{x_j \epsilon X |\ d(x_i, x_j) \leq \varepsilon\} \quad \text{(Equation IV.24)}$$

Pour ce type d'algorithme, on a besoin de définir deux paramètres : ε le rayon minimum autour du noyau et M le nombre de point minimum pour le voisinage $N_\varepsilon(x_i)$.

L'algorithme DBSCAN [120] (Density-Based Spatial Clustering of Applications with Noise) est l'un des plus connu utilisant ces deux paramètres pour l'identification des clusters basés sur la notion de voisinage auteur d'un noyau.

L'algorithme est très facile à comprendre et ne nécessite pas qu'on lui fournit le nombre de clusters à trouver. Il est capable de gérer des données absurdes et de les éliminer du processus de partitionnement. Cet algorithme peut être décrit comme suit :

Algorithme DBSCAN(Density-Based Spatial Clustering of Applications with Noise)

En entrée : les deux paramètres : $\boldsymbol{\varepsilon}$ le rayon minimum autour du noyau et $\boldsymbol{M}$ le nombre de point minimum pour le voisinage fixés d'avance et l'ensemble de points $\boldsymbol{X}$.

En sortie : une partition $\boldsymbol{C} = \{\boldsymbol{C_1}, \dots, \boldsymbol{C_k}\}$ de $\boldsymbol{X}$ en k d-clusters.

Début

- Initialiser le cluster $\boldsymbol{C_{id}} = \boldsymbol{\phi}$ avec *id=1*
- Pour i =1 à n faire
  a. Si $\boldsymbol{x_i}$ n'est pas un noyau ou si $\boldsymbol{x_i} \boldsymbol{\epsilon} \bigcup_{\boldsymbol{j=1,\dots,id}} \boldsymbol{C_j}$ alors retourner à l'étape 2
  b. Construire Cluster $(\boldsymbol{x_i}, \boldsymbol{X}, \boldsymbol{C_{id}}, \boldsymbol{\varepsilon}, \boldsymbol{M})$
  c. $\boldsymbol{id} = \boldsymbol{id} + \boldsymbol{1}$ et $\boldsymbol{C_{id}} = \boldsymbol{\phi}$

  Fin pour
- Retourner l'ensemble des d-clusters : $\boldsymbol{C_1}, \dots, \boldsymbol{C_{id-1}}$

Fin.

Algorithme IV.13 : DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

Bien que cet algorithme est facile à la compréhension, il reste d'exécution informatique lente, surtout lorsque le nombre de points est grand. Sa complexité est

quadratique, de l'ordre de (n^2) , mais peut être réduite à $O(nlog(n))$ en simplifiant l'implémentation de l'algorithme.

Un autre algorithme permet aussi le Clustering par densité, c'est celui de DENCLUE [121] (DENsity-based CLUstEring), il a été proposé par Hinneburg et keim entre 1998 et 2000, basé sur des fonctions mathématiques. Bien qu'il acquière une complexité élevée avec le nombre de paramètres d'entrée, il présente comme intérêts :

- très efficace lors du traitement de données aberrantes présentant des bruits ;
- capable de décrire mathématiquement des clusters choisis arbitrairement appartenant à des ensembles de données de grandes dimensions ;
- rapide par rapport à DBSCAN et donc plus fort.

Cet algorithme est basé sur l'estimation de la densité du noyau à travers différentes fonctions, c'est presque le même principe que l'algorithme DBSCAN, sauf qu'ici, on a la possibilité d'illustrer la structure hiérarchique interne dans la distribution de données par réglage de la largeur σ de fenêtre de la fonction d'influence du noyau.

Soit $D = \{x_1, \dots, x_n\}$ l'ensemble de données de n objets dans l'espace Ω .on décrit l'algorithme DENCLUE par les notions suivantes :

- L'estimation du noyau par la fonction de densité globale :

  $\forall\, x \in \Omega$, la fonction de densité de probabilité est donnée par :

$$f^D(x) = \frac{1}{n}\sum_{i=1}^{n} K\left(\frac{x-x_i}{\sigma}\right) \quad \text{(Equation IV.25)}$$

Avec $K(x)$ la fonction d'influence du noyau qui est une fonction de densité symétrique avec un pic à l'origine, elle peut être une fonction Gaussienne, fonction d'onde carrée …

σ : la largeur de fenêtre de la fonction du noyau.

- Attracteur de densité et attraction de densité .Soit x^* un point maximal local de la fonction de densité globale, pour un point $x \in \Omega$ s'il existe un ensemble de points $x_0, \dots, x_k$, tel que $x_0 = x$ et $x_k = x^*$ et x_i $(0 < i < k)$ de sorte que cela se situe dans le sens de gradient de x_{i-1}, alors *x* est attiré en densité par x^* et x^* est un attracteur de densité de *x.*

Si la fonction du noyau $K(x)$ est continue et différentiable à chaque point, la méthode d'escalade basée sur le gradient peut être utilisée pour trouver la densité des attracteurs.

- Le Clustering basé sur le centre. Avec x^* un attracteur de densité donné, s'il existe un sous-ensemble $C \subseteq D$ tel que x est attiré en densité par x^* et $f^D(x^*) \geq \xi$ avec ξ est un seuil de bruit préréglé, alors C est le cluster avec x^* son centre.

- Le Clustering avec une forme arbitraire : soit X un ensemble composé de densité Attracteurs. S'il existe un sous-ensemble $C \subseteq D$ qui vérifie :

$\forall\, x \in C$, il existe un attracteur de densité $x^* \in X$ pour que x soit attiré en densité par x^* et $f^D(x^*) \geq \xi$;

$\forall x_i^*, x_j^* \in X\ (i \neq j)$, il existe un chemin $P \subset \Omega$ de x_i^* à x_j^* qui satisfait la condition suivante :

$y \in X, f^D(y\ ) \geq \xi$. C s'appelle le cluster de la forme arbitraire déterminée par X .

Nécessairement, il faut prévoir deux paramètres pour exécuter cet algorithme qui sont σ : la largeur de fenêtre de la fonction du noyau et ξ le seuil de bruit préréglé, le choix de ces deux paramètres influence les attracteurs et le nombre de clusters trouvés.

Les Étapes de base de l'Algorithmes DENCLUE [122] sont les suivantes :

- Déterminer les attracteurs de densité.
- Associer des objets de données avec des attracteurs de densité utilisant l'escalade.
- Si possible, fusionner les clusters initiaux en s'appuyant davantage sur une approche de Clustering hiérarchique.

Toutefois, bien qu'il tienne compte des données aberrantes et des outliers, sa complexité peut être de l'ordre $O(nlog(n))$, ce qui est acceptable.

IV.5.6 Le Clustering conceptuel

C'est un modèle de regroupement non supervisé qui vise à trouver des classifications appropriées pour un ensemble d'exemples. Une telle technique non supervisée, nécessite l'utilisation d'une fonction d'évaluation pour découvrir des classes avec de bonnes descriptions conceptuelles, ce qui la qualifie, ainsi, de technique d'apprentissage par observation et non par utilisation d'exemples.

Les méthodes classiques de regroupement organisent des objets dans des classes fondées uniquement sur une base de similitude numérique. La mesure du degré de similitude est généralement définie comme une mesure de proximité dans un espace multidimensionnel. Les objets sont regroupés dans des classes de haute similarité intra classe, mais de faible similarité interclasse.

L'existence de propriétés non pertinentes peut fausser cette mesure. Les méthodes numériques traditionnelles visent à éliminer les attributs non pertinents des classes, telles que l'analyse des facteurs ou la mise à l'échelle multidimensionnelle, qui sont inadéquates pour les attributs nominatifs (catégoriques). En outre, les mesures numériques ne tiennent pas compte de toute information contextuelle. Les propriétés globales des classes d'objets ne sont pas prises en compte, ce qui rend souvent difficile l'interprétation et la description des clusters.

L'exemple ci-dessous illustre l'incapacité des méthodes numériques à prendre en compte les propriétés globales, puisqu'elles se basent sur le regroupement des voisins les plus proches pour former un cluster à partir des points A et B, elles ne vont pas voir comme un observateur humain qui verrait instantanément que les points représentent deux diamants, regroupant les deux points.

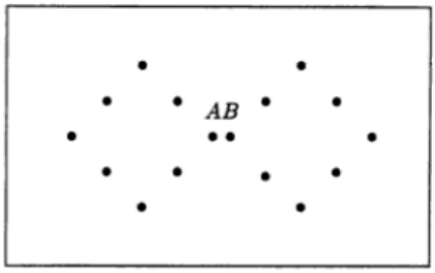


Figure IV.8 : Cohérence globale du regroupement conceptuel (réalisée par nos soins).

Michalski et Stepp ont qualifié ce type de vision de cohésion conceptuelle « cohesiveness ».

Pour ce dernier, cette idée est à la base du regroupement conceptuel. Pour celui-ci, la «similitude» entre deux points A et B, appelé la cohésion conceptuelle de A et B, dépend non seulement des points et des points environnants E, mais aussi d'un ensemble de concepts C q disponibles pour décrire A et B ensemble :

Cohésion conceptuelle (A, B) =f (A, B, C, E) (Equation IV.26)

Toutefois Hansen (1997) a déclaré que le choix d'une méthode de Clustering différente, comme l'utilisation de Centroïdes plutôt que les plus proches voisins « PPV » pourrait produire un regroupement correct.

En conséquence, cette distinction entre le regroupement numérique et conceptuel n'est pas entièrement valable, et des travaux supplémentaires doivent être faits pour déterminer les points forts de chaque méthode.

Néanmoins, les méthodes de Clustering numériques typiques, sans contrainte, dépendent uniquement des propriétés des objets individuels, et non de tous les concepts externes qui peuvent être utiles pour servir de base au Clustering. Les méthodes conceptuelles de regroupement permettent un tel accès aux concepts globaux et externes.

Michalski [123] et Stepp (1983) déterminent le regroupement conceptuel comme «un processus de construction d'un réseau concept caractérisant un ensemble d'objets, avec des nœuds marqués par des concepts décrivant des classes d'objets et des liens marqués par les relations entre les classes».

La prédiction des caractéristiques des groupes est l'une des motivations essentielles en regroupement, si la classe d'un objet est connue, alors on connaît ses propriétés, car les objets héritent des propriétés des super classes. Cette hypothèse est la base de l'algorithme COBWEB (1987)[124].

Comme l'être humain observe le monde, il établit des concepts en arrangeant ses observations sous forme de caractéristiques partagées parmi les choses qu'il observe.

De l'exemple des chiens observés, leur concept est formé au fil du temps, il est enrichi par des observations d'apparence et de comportement différents de chiens particuliers .A mesure que le nombre de chiens observé augmente, le concept grandit, la généralisation des caractéristiques est devenue utile, un certain moment des prédictions exactes étaient faites sur un chien en particulier pour la première fois. Ce classement se fait malgré des informations non pertinentes ou approximatives.

A l'instar des humains, COBWEB forme des concepts en regroupant des objets avec des attributs semblables. Il le fait bien que les informations non adéquates et incomplètes dans les observations reçues, et de manière non supervisée. De plus, COBWEB est incrémental, il

construit des clusters en analysant les exemples à la fois.

COBWEB présente les clusters sous forme d'une distribution de probabilité sur l'espace des valeurs d'attributs, générant alors un arbre de classification hiérarchique, dans lequel les nœuds intermédiaires définissent des sous-concepts.

L'algorithme COBWEB de base est de conception simple, la classification et l'apprentissage sont reliés, chaque objet étant trié par une hiérarchie conceptuelle et en même temps celle-ci est rectifie dans son passage. Le processus initialise sa hiérarchie sur un premier nœud, la première instance fournit ses attributs comme base du concept. Lors de la deuxième instance, COBWEB utilise le concept pour mesurer ses valeurs et crée alors deux enfants, un se base sur la première instance et l'autre sur la deuxième instance.

COBWEB sauvegarde tous les enfants en chaque nœud et classe les objets selon leurs catégories. Chacun d'eux compose un regroupement alternatif de nouvelles instances avec un parent commun.

COBWEB utilise une fonction d'évaluation permettant de choisir le meilleur Cluster, prévoit de créer de nouveaux types contenants de nouvelles instances et compare celui-ci au meilleur utilisant uniquement les catégories existantes.

Si le Cluster basé sur des classes existantes a le meilleur score de la compétition, COBWEB change la probabilité de la catégorie sélectionnée et les probabilités conditionnelles pour ses valeurs d'attributs. Voir l'algorithme ci-dessous :

Algorithme COBWEB [125] ;

Entrée : Le nœud courant N de la hiérarchie conceptuelle, et une instance I non classifiée.

Résultants : une hiérarchie conceptuelle qui classe l'instance I. Appel de niveau supérieur des méthodes Cobweb(…),…

Variables : C, P, Q et R sont des nœuds dans la hiérarchie.

U, V, W et X sont des scores de Clusters.

Début

Cobweb (N, I) ;

Si N est un nœud terminal alors

créez-nouveaux_terminaux (N, I)

Incorporer (N, I).

Sinon

    Incorporate (N, I).

Pour chaque enfant C du nœud N,

Calculez le score pour placer I dans C.

Soit P le nœud avec le score le plus élevé W.

Soit R le nœud avec le deuxième meilleur score.

Soit X le score pour placer I dans un nouveau nœud Q.

Soit Y le score pour fusionner P et R en un seul nœud.

Soit Z le score pour diviser P dans ses enfants.

Si W est le meilleur score alors

Cobweb (P, I) (place I dans la catégorie P).

Sinon

Si X est le meilleur score alors
Initialisez les probabilités de Q en utilisant les valeurs de I
(Place I par lui-même dans la nouvelle catégorie Q).
Sinon
Si Y est le meilleur score alors
O=Fusionner (P, R, N)
Cobweb (O, I)
Sinon
Si Z est le meilleur score alors
Diviser (P, N)
Cobweb(N,I)
Fin Si…
Fin.

Les Sous-programmes de COBWEB :
Variables : N, O, P et R sont des nœuds dans la hiérarchie.
I une instance non classifiée
A est un attribut nominal.
V est une valeur d'un attribut.
Procédure Incorporer (N, I) ;
Mettre à jour la probabilité de la catégorie N.
Pour chaque attribut A de l'instance I,
Pour chaque valeur V de A,
Mettre à jour la probabilité de V de donnée de catégorie N ;
Fin pour ;
Fin procédure.

Procédure créez_nouveaux_terminaux (N, I)
Créer un nouvel enfant M du nœud N.
Initialiser les probabilités de M à celles de N.
Créer un nouvel enfant O du nœud N.
Initialiser les probabilités de O à l'aide des valeurs de I
Fin procédure.
Procédure Fusionner (P, R, N)
Faire de O un nouvel enfant de N.
Définir les probabilités de O pour qu'elle soit la moyenne de P et R ;
Supprimez P et R en tant qu'enfants du nœud N ;
Ajoutez P et R comme enfants du nœud O ;
Retourner 0 ;
Fin procédure.
Procédure Diviser (P, N)
Retirer l'enfant P du nœud N.
Ajouter les enfants de P pour être enfants de N ;
Fin procédure.

Algorithme IV.14 : COBWEB

Ainsi, les scores de prévisibilité pour les valeurs se produisant dans l'instance augmenteront, alors que ceux pour les valeurs qui ne se produiront vont diminuer. Les scores de prédictions changent aussi, mais comme le système ne les stocke pas, il ne les met pas à jour explicitement. En outre, COBWEB continue de trier l'instance dans la hiérarchie, en considérant récursivement les enfants de la catégorie sélectionnée. Le nœud N3 dans la figure ci-dessous montre le résultat de l'incorporation d'une nouvelle instance dans un nœud existant. À un stade antérieur. Il se créait un nœud terminal basé sur une instance unique. Toutefois, l'action d'hébergement d'une nouvelle instance a laissé ses probabilités de couleur uniformément réparties et lui a donné deux enfants.

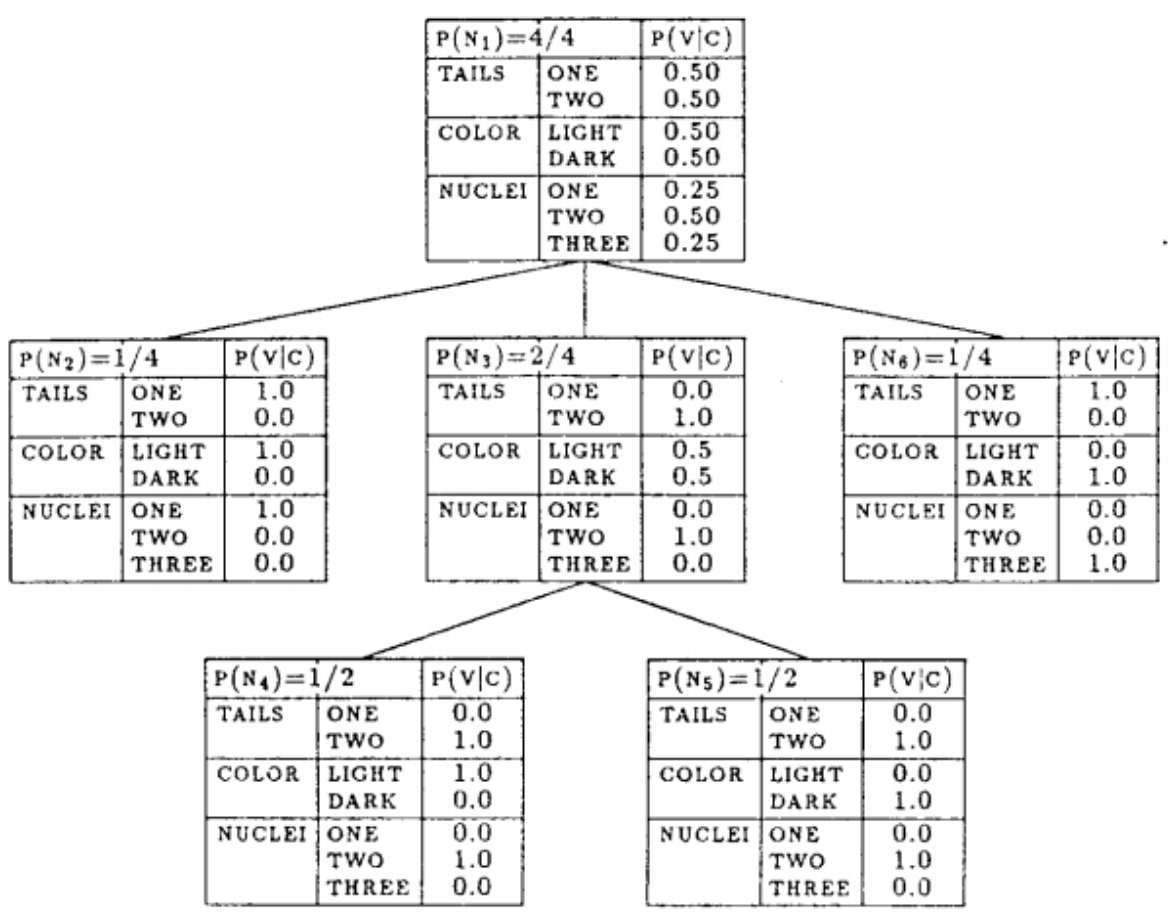


Figure IV.9 : Un exemple de hiérarchie COBWEB [126] avec des nœuds numérotés par ordre de création

Si le Clustering de classe singleton apparaît en tant que vainqueur, COBWEB crée cette nouvelle catégorie et la met en tant que fils du nœud père courant. Le système conçoit les valeurs des attributs de ce nouveau concept sur ceux trouvés dans l'instance, chaque score de prévisibilité leurs est donné. Dans ce cas, la classification s'arrête à cette étape. Puisque le nouveau concept est un nœud terminal.

Le nœud N_6 de la figure ci-dessus a été créé de cette façon, puisque l'instance qu'elle résume est suffisamment différente du nœud N_2 et N_3.

Bien qu'en principe, la méthode ci-dessus fournisse tout ce qui est essentiel pour créer des hiérarchies de concepts probabilistes, elle peut être sensible à l'ordre de la présentation d'instance, en créant différentes hiérarchies à partir de différents ordres des mêmes données. En particulier, si les cas initiaux ne sont pas significatifs de l'ensemble de la population, on peut avoir des hiérarchies avec une mauvaise capacité prédictive. Par exemple, si les premières instances sont toutes des membres conservateurs, l'algorithme créerait des sous-catégories de

celles-ci au niveau supérieur. Lorsqu'il a enfin recensé des instances de députés libéraux, il créerait une catégorie pour eux du niveau supérieur. Cependant, il aurait encore toutes les instances conservatrices à ce même niveau, alors qu'on préférerait qu'elles soient regroupées dans une catégorie distincte.

COBWEB comprend deux opérateurs supplémentaires pour l'aider à se remettre de ces hiérarchies non optimales. À chaque niveau du processus de classification, le système envisage de fusionner les deux nœuds qui classent le mieux la nouvelle instance. Si le Clustering résultant est meilleur (selon la fonction d'évaluation) que l'original, il combine les deux nœuds. Dans une catégorie unique, tout en conservant les nœuds originaux comme enfants. Cela transforme un cluster de N nœuds en un ayant des nœuds N-1, comme dans la transition illustrée par la figure ci-dessous :

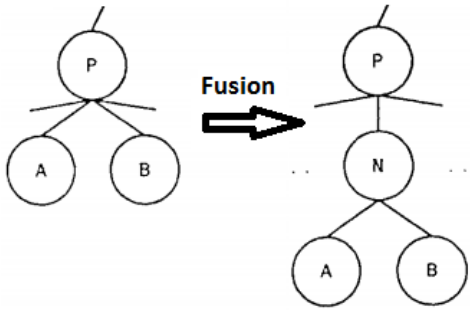


Figure IV.10 : Fusion de catégories en COBWEB (réalisée par nos soins).

Le système intègre également l'opération inverse des nœuds de fractionnement. À chaque niveau, si COBWEB décide de classer une instance en tant que membre d'une catégorie existante, il envisage également de supprimer cette catégorie et d'élever ses enfants. Si cette action conduit à un Clustering amélioré, le système modifie la structure de sa hiérarchie en conséquence. Ainsi, si l'un des N nœuds d'un niveau donné possède M enfants, le fractionnement de ce nœud donnerait N + M-1 nœuds à ce niveau, tel que représenté par la transition à la figure suivante :

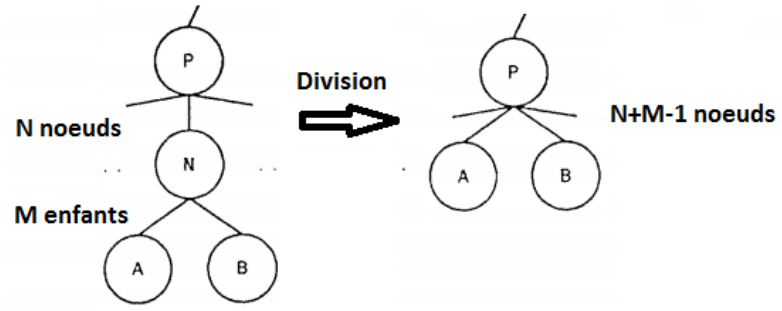


Figure IV.11 : Division de catégories en COBWEB (réalisée par nos soins).

La fonction d'évaluation de COBWEB est une mesure de l'utilité de catégorie (CU), elle favorise les regroupements qui maximisent le potentiel d'inférence d'informations. Pour ce faire, on tente de maximiser la similitude intra-classe et les différences entre classes, et on fournit également un arrangement de principe entre la prévisibilité et la prédiction. La mesure de base suppose que les descriptions de concepts sont de nature probabiliste. Nous n'avons pas l'espace nécessaire pour regagner cette métrique, mais nous pouvons considérer certaines de ses caractéristiques.

Pour tout-ensemble d'instances, toute paire attribut-valeur, $A_i = V_{ij}$ et toute classeC_k, on peut calculer la probabilité conditionnelle de la valeur donnée V_{ij} à l'appartenance à la classe $P(A_i = V_{ij}|C_k)$ ou sa prévisibilité.

On peut également calculer $P(C_k|A_i = V_{ij})$: la probabilité conditionnelle d'appartenance à la classe donnée à cette valeur ou sa prédiction. On peut combiner ces mesures des attributs et valeurs individuelles en une mesure globale de la qualité de cluster.

Plus exactement :

$$\sum_k \sum_i \sum_j P(A_i = V_{ij}) P(C_k|A_i = V_{ij}) P(A_i = V_{ij}|C_k) \quad \text{(I) (Equation IV.27)}$$

représente un compromis entre la prévisibilité $P(A_i = V_{ij})$ et la prédiction $P(C_k|A_i = V_{ij})$ qui a été résumée pour les attributs (i) et les valeurs (j) dans toutes les classes (k).

La probabilité $P(A_i = V_{ij}|C_k)$ pondère les valeurs individuelles, de sorte que les valeurs fréquentes jouent un rôle plus important que ceux se produisant moins fréquemment.

En utilisant la règle de Bayes, nous avons :

$$P(A_i = V_{ij}) P(C_k|A_i = V_{ij}) = P(C_k) P(A_i = V_{ij}|C_k) \text{ (Equation IV.28)}$$

L'expression (I) devient : $\sum_k P(C_k) \sum_i \sum_j P(A_i = V_{ij}|C_k)^2$ (II)

Gluck et Carter (2002) ont montré que la sous-expression $\sum_i \sum_j P(A_i = V_{ij}|C_k)^2$ est le nombre attendu de valeurs d'attribut que l'on peut deviner correctement pour un membre arbitraire de la classe C_k. Cette attente considère une stratégie de concordance de probabilité, dans laquelle on suppose une valeur d'attribut avec une probabilité égale à sa probabilité de se produire. Ainsi, ils présument que l'on admette une valeur avec la probabilité $P(A_i = V_{ij}|C_k)$ et que cette estimation est correcte avec la même probabilité.

Gluck et Carter s'appuient sur l'expression (II) dans leur dérivation. Ils définissent l'utilité de catégorie comme l'augmentation du nombre attendu de valeurs d'attribut qui peuvent être correctement devinées, compte tenu d'un ensemble de n catégories, sur le nombre attendu de suppositions correctes sans une telle connaissance. Ce dernier terme est simplement $(\sum_i \sum_j P(A_i = V_{ij})^2)$, il faut donc soustraire ceci de l'expression (II).

L'expression complète pour l'utilité de catégorie (CU) est donc :

$$CU(\{C_1, \ldots, C_t\}) = \frac{1}{t} \sum_{k=1}^{t} P(C_k) \left[\sum_i \sum_j P(A_i = V_{ij}|C_k)^2 - \sum_i \sum_j P(A_i = V_{ij})^2 \right]$$

(Equation IV.29)

Sachant que t est le nombre de catégories. Cette division permet de comparer différents groupes de dimensions, se produit chaque fois que l'on envisage de fusionner, de diviser ou de créer une nouvelle catégorie. Etant donné l'utilité de la catégorie dans le nombre attendu de suppositions correctes concernant les valeurs d'attribut, la capacité prédictive s'impose comme mesure naturelle du comportement. Fisher a testé COBWEB sur des domaines naturels et artificiels, en mesurant ses performances en lui demandant de prédire les valeurs d'attributs manquants sur les instances de test.

Cette vision est similaire à celle de Quinlan (1986)[127] pour juger les systèmes d'apprentissage supervisés, à la différence que l'on obtienne une moyenne de nombreux attributs au lieu de prédire un seul correspondant au nom de la classe.

COBWEB peut être alors vu comme un algorithme d'une recherche d'escalade dans un espace de hiérarchies conceptuelles. Dans ce cas, il existe quatre étapes principales à suivre :

- Tout d'abord on classe l'objet dans une classe existante ;
- Après on crée une nouvelle classe ;
- Ensuite on fusionne deux classes en une seule classe ;
- Et en dernier, on passe au fractionnement en divisant une classe en plusieurs ;

A chaque étape du processus de classification, le système utilise la fonction d'évaluation (CU) « l'utilité de catégorie » pour savoir quel objet à créer ou à utiliser ensuite.

L'emploi d'une fonction d'évaluation bien déterminée est d'une grande importance par rapport aux travaux antérieurs pour former les concepts, ajouter à ça la reformulation de Fisher de la prévisibilité et de la prédiction en termes de probabilités conditionnelles.

L'utilisation explicite de la fusion et du fractionnement s'avère très utile et souhaitable, car à partir d'échantillons non représentatifs, COBWEB avance sans perdre d'efficacité malgré les limites de la mémoire de l'ordinateur ou il s'exécute.

Cependant, le travail de Fisher reste limité, car COBWEB ne traite que des données nominales et non numériques.

Finalement, COBWEB garde toutes les instances manipulées dans son concept hiérarchique comme des nœuds terminaux et néglige les données floues ou numériques.

IV.5.7 Autres techniques de Clustering

Il est très intéressant d'analyser les données qui sont les mieux représentées sous forme de graphique. Les exemples incluent la WWW, les réseaux sociaux, les réseaux biologiques, les réseaux de communication, les réseaux de transport, les réseaux énergétiques et bien d'autres. Ces graphiques sont fréquemment multi-relationnels et dynamiques. À l'ère des grandes données, l'importance de pouvoir les exploiter efficacement et d'en tirer des leçons augmente au fur et à mesure que des données structurées et semi-structurées deviennent disponibles.

Comme nous l'avons vu au chapitre 2, les graphes sont très utiles dans l'extraction de connaissances et la recherche de motifs fréquents afin de classifier les données et de trouver les règles d'association entre les données.

Cette approche est fondée sur le formalisme théorique des graphes, constitués de sommets et d'arêtes.

De façon générale, un graphe représente la structure et les liaisons d'un ensemble complexe dit système (S), en exprimant les relations entre ses éléments tel que les réseaux de communication, les réseaux routiers, l'interaction entre les diverses variétés animales, les circuits électriques, la programmation et le plus intéressant les applications aux sciences de l'Internet.

La méthode des k-NN [128] (plus proches voisins) est liée souvent aux graphes de connaissances. A chaque nœud on affecte une instance pour former à la fin un graphe complet.

L'approche CNN (Concepts of Nearest Neighbors) présente la similarité symbolique entre deux objets comme un concept de voisins dont la similarité définit ce que les deux objets ont en commun .Par la suite on identifie le cluster qui regroupe les instances se trouvant entre les deux

objets. La grandeur des clusters dépend de la distance ou de la similarité numérique.

On peut alors exploiter de prés des clusters, sans transformation ou extraction de fonctionnalités, car chaque plus proche voisin est clairement représenté en se basant sur sa similarité.

Plusieurs graphes peuvent être utilisé pour représenter les instances de données aux sommets, nous citons quelque uns comme :

- Le graphe du plus proche voisin [129] NNG (Nearest Neighbor Graph) (2011)
- Le graphe de Gabriel GG [130] (Gabriel Graph) (2012)
- L'arbre minimum de recouvrement [131] MST (X) (Minimum Spanning Tree)
- La triangulation de Delaunay [132] DT (X) (Delaunay Triangulation)
- Le graphe de voisinage relatif RNG [133] (Relative Neighborhood Graph)

Pour le processus de Clustering utilisant les graphes, deux méthodes recherche de clusters existent :

➢ Soit on fixe k le nombre de clusters à priori et on choisit comme fonction d'évaluation le nombre d'arêtes minimum entre clusters basé sur la distance de similarité entre objets des sommets. Dans ce cas le processus de Clustering choisit une partition du graphe en k sommets cohérents en taille de telle sorte que les arêtes inter-clusters soient en nombre minimal.

➢ Soit on utilise le diamètre maximum des clusters et on passe au partitionnement des graphes en cherchant le nombre minimum de clusters qui possèdent le diamètre maximum fixé.

➢ Une autre technique encourageante qui a récemment apparu dans un certain nombre de domaines est d'utiliser le Clustering dite spectral [134] (Thomas G. Dietterich, Suzanna Becker, Zoubin Ghahramani) (2001).

Ici, on utilise les vecteurs propres supérieurs d'une matrice dérivée de la distance entre les points. De tels algorithmes ont montré leurs preuves dans de nombreuses applications, y compris le processus de vision par ordinateur. Mais malgré leurs succès empiriques, différents auteurs sont encore en désaccord sur le choix des vecteurs propres à utiliser et comment en dériver des clusters [135] (Yair Weiss, Michael I Jordan) (1999).

Notons aussi que les réseaux de neurones artificiels, sont très connus pour leur utilisation en classification. Ce sont des réseaux de processeurs élémentaires, fortement connectés entre eux, fonctionnant en parallèle. Chaque processeur (neurone artificiel) calcule une sortie unique à partir des informations qu'il reçoit.

Les réseaux de neurones utilisent la technique d'apprentissage pour améliorer le modèle, dans cette phase le style et le comportement du modèle se transforme progressivement jusqu'à atteindre un état stationnaire et acceptable.

En pratique, on associe des algorithmes d'apprentissage au modèle étudié et on modifie les variables, comme les poids de connexion, pendant l'apprentissage.

Pour ces algorithmes d'apprentissage, on identifie deux grandes classes selon la nature de l'apprentissage : supervisé ou non supervisé. Cette différenciation se base sur la forme des exemples d'apprentissage. Lorsque celui-ci est supervisé : les exemples sont sous forme de couple (Entrée, Sortie associée) alors que pour celui non supervisé on a que des valeurs d'entrées.

Dans un réseau de neurones chaque neurone effectue un traitement élémentaire et ce sont le nombre de neurones et leurs connectivités qui vont faire la force du réseau. Dans les réseaux de neurones dits "en couche", le réseau est constitué de trois sous-ensembles de neurones à savoir les neurones d'entrée, les neurones cachés (entièrement connectés aux neurones, d'entrée ou, si le réseau possède plus d'une couche cachée, à la couche précédente) et les neurones de sortie connectés à la dernière couche de neurones cachés.

L'exemple de la figure ci-dessous schématise un réseau de neurone, ce réseau effectue N_o fonctions algébriques des variables d'entrées du réseau ; chacune des sorties est une fonction, réalisée par le neurone de sortie correspondant, des fonctions non linéaires réalisées par les neurones cachés.

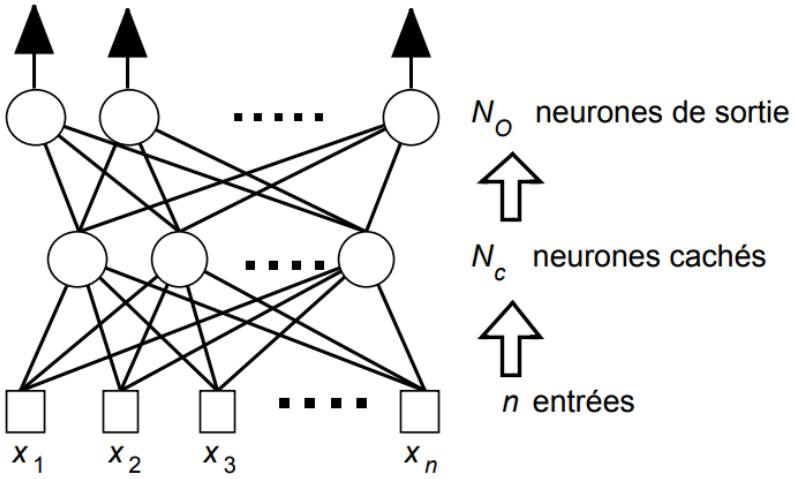


Figure IV.12 : Un réseau[4] de neurones à n entrées, une couche de Nc neurones cachés, et No neurones de sortie.

Pour ce type de réseau, appelé réseau de neurones non bouclé, le temps ne joue aucun rôle fonctionnel. Si les entrées sont fixes, les sorties le sont également. Le temps nécessaire pour le calcul de la fonction réalisée par chaque neurone est minime, et on peut considérer ce calcul comme instantané étant donné sa grande rapidité. Pour cette raison, les réseaux non bouclés sont souvent appelés « réseaux statiques », par opposition aux réseaux bouclés ou « dynamiques ».

IV.6 Critères d'évaluation du Clustering

L'évaluation des résultats d'une méthode de Clustering reste un problème ouvert. Mais ici, la principale difficulté réside dans le fait que l'évaluation des résultats de la classification est subjective par nature. En conséquence, il existe plusieurs manières pertinentes pour la classification des objets de données.

En pratique, quatre techniques principales sont utilisées pour mesurer la qualité d'une technique de Clustering. Cependant chacune de ces techniques a des limites :

- La première est d'utiliser des jeux de données artificiels lorsqu'on connaît le regroupement souhaité. Toutefois les résultats ne sont évalués que sur la distribution générée

[4] Http : //www.eyrolles.com/Chapitres/9782212110197/chap01.pdf, page consultée le 12/12/2017 à 16h30.

correspondante, et les résultats sur des données artificielles ne peuvent pas être transposés à des données réelles.

- La 2ème technique est d'utiliser des jeux de données étiquetés et vérifier si l'algorithme de Clustering recouvre ces données dans les classes. Pourtant les classes d'un problème supervisé ne sont pas nécessairement celles qui doivent être trouvés par un algorithme de Clustering parce que d'autres groupements peuvent aussi être utiles.

- La 3ème technique consiste à faire appel à un expert pour comprendre la signification du regroupement dans un champ particulier. Cependant, s'il est possible pour un expert de dire si un regroupement de Clustering donné a un sens, il est beaucoup plus difficile de quantifier son intérêt, ou de dire si un résultat donné est meilleur qu'un autre. En outre, l'applicabilité de la méthode ne peut être étendue à d'autres types de données.

- Dans la 4ème technique, on utilise un critère interne, comme l'inertie intra-cluster ou entre clusters. Cependant, ces critères, prédéfinis, sont également subjectifs par nature car ils utilisent une notion préétablie de ce que c'est qu'un bon Clustering. Par exemple, la séparation inter-clusters n'est pas toujours le meilleur critère à utiliser, les groupes qui se chevauchent peuvent parfois être plus appropriés.

En résumé on peut différentier entre un critère d'évaluation interne, externe ou relatif pour mettre en valeur la qualité d'une technique de Clustering.

Un autre critère appelé indice Γ nommé « statistique de Huberts »(2005) permet de donner une valeur mesurable au critère d'évaluation des techniques de Clustering. En voici la définition :

Soit $X = \{x_1, \dots, x_n\}$ l'ensemble d'objets à traiter

$M = \frac{n(n-1)}{2}$ Le nombre de paires possibles dans X : $\{(x_i, x_j)\}_{i \neq j}$

Avec $U(i,j)\ et\ V(i,j)$correspondent aux distances entres les clusters contenant les objets $x_i\ et\ x_j$ respectivement dans le schéma obtenu et dans la classification issue des connaissances externes (classification préfinie).

$$\Gamma = \frac{1}{M}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} U(i,j).V(i,j) \quad \text{(Equation IV.30)}$$

IV.6.1 Critères externes

Il s'agit de comparer un schéma avec une classification prédéfinie. L'évaluation porte donc sur la concordance entre le schéma obtenu et une connaissance externe sur les données (schéma attendu).Ici on a donc les deux distances $U(i,j)\ et\ V(i,j)$ sont identiques et l'indice Γ devient :

$$\Gamma = \frac{1}{M}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} U(i,j)^2 \quad \text{(Equation IV.31)}$$

Ajoutons trois autres indices utiles dans l'évaluation, à savoir ceux de Rand [136], de Jaccard ou encore de Fowlkes et Mallows, lesquels mesurent le taux de liaisons ou non-liaisons correctes dans le schéma à évaluer :

$$Rand(C, P)=\frac{a+d}{a+b+c+d} \qquad \text{(Equation IV.32)}$$

$$Jaccard(C,P) = \frac{a}{a+b+c} \qquad \text{(Equation IV.33)}$$

$$FM(C,P) = \sqrt{\frac{a}{a+b}\frac{a}{a+c}} \qquad \text{(Equations IV.34)}$$

Avec :

- C schéma de Clustering et P une classification préétablie.
- a : le nombre de paires (x_i, x_j) telles que x_i et x_j se retrouvent dans une même classe dans C et dans P (liaison correcte),
- b : le nombre de paires (x_i, x_j) telles que x_i et x_j se retrouvent dans une même classe dans C mais non dans P (liaison incorrecte),
- c le nombre de paires (x_i, x_j) telles que x_i et x_j se retrouvent dans une même classe dans P mais non dans C (non-liaison incorrecte),
- d le nombre de paires (x_i, x_j) telles que x_i et x_j ne se retrouvent dans une même classe ni dans C ni dans P (non-liaison correcte).

Souvent, il n'y a pas de classification connue pour servir de référence. Il est donc indispensable de procéder au jugement d'un schéma de façon indépendante, en utilisant que les informations fournies par l'algorithme. Dans ce cas on devrait s'intéresser des critères d'évaluation interne ou relatifs dans l'évaluation.

IV.6.2 Critères internes

Il existe peu de mesures d'évaluation interne. Dans le cas d'un partitionnement, c'est l'indice Γ (interne) qui est généralement utilisé.

Dans la littérature, plus de 30 indices différents peuvent être trouvé. Dans ce cadre, plusieurs études et comparaisons ont été effectuées. Parmi elles, celles entreprises par Arbelaitz(2013)[137] qui ont examiné ces indices, et ont aboutis à des performances différentes pour ceux-ci.

Plusieurs autres indices de qualité, utilisant la distance entre les individus ont été développés. Ceux de Dunn, de validation de Davies-Bouldin [138], de Silhouette [139] et Calinski Harabasz [140] en constituent des exemples types dont l'évaluation a été jugée satisfaisante dans une large gamme de situations.

Par ailleurs, lorsqu'il s'agit d'évaluer le Clustering hiérarchique, on utilise le coefficient de corrélation cophénétique (CPCC).

IV.6.3 Critères relatifs

Pour ces critères, on choisit les paramètres optimaux connaissant une méthode de Clustering et un ensemble de données, dont le nombre k de clusters à construire. La technique consiste alors à faire changer un seul paramètre, à chaque fois, et à observer la qualité des schémas obtenus.

Le paramétrage optimal est identifié par un changement significatif d'un meilleur schéma, dans la représentation graphique de l'indice utilisé.

Plusieurs types d'indices existent, utilisés comme critères internes et relatifs, en fonction des données que l'on souhaite utiliser (matrice de (dis) similarité), et du type de schéma produit par la méthode (hiérarchie ou partitionnement).

IV.6.4 Critères d'évaluation pour le Clustering flou

Pour le Clustering flou, on définit l'indice de séparation $S(C)$, qui est une variante floue de l'indice de Davies-Boulin :

$$S(C) = \frac{\sum_{h=1}^{t}\sum_{i=1}^{n} u_h^2(\quad) d(x_i,x_h^*)^2}{n.min_{h\neq l} d(x_h^*,x_l^*)^2} \qquad \text{(Equation IV.35)}$$

Avec $(\{u_h(x_i)\}_{i=1\ldots n,h=1\ldots t})$ valeurs d'appartenance des objets aux clusters à évaluer.

Le numérateur de $S(C)$ représente la somme des inerties intra-clusters (floue) tandis que le dénominateur évalue la séparation inter-clusters. Une bonne évaluation correspond à un indice de séparation $S(C)$minimal.

IV.7 Conclusion

Comme nous l'avons vu dans un premier temps, la notion d'informations, de données, et de connaissance sont indissociables. Les techniques d'exploitation et de manipulation sont nombreuses, passant par la construction de modèles mathématiques de la théorie d'information à la théorie des probabilités .D'où l'apparition de plusieurs processus de classification et de Clustering traités dans ce chapitre. Mais, avant, nous avons mis en valeur l'information dans différents domaines, chez les entreprises, dans le commerce et pour la sécurité …

Nous avons analysé les techniques d'extraction de connaissance à partir de motifs fréquents et des règles d'association, de même, nous avons invoqué les relations et les concepts formels ainsi que les méthodes de recherche sélectives et la programmation logique inductive sans oublier les graphes comme support d'information.

Nous avons également exposé la problématique globale du Clustering, passant par la préparation de données au choix d'algorithmes jusqu'à l'exploitation de clusters ; de même, nous avons défini diffèrent paramètres de mesure de similarité et les variables symboliques et numériques ainsi les indices d'évaluation comme celui de Jaccard et de Randet ceux en se basant sur les résultats d'investigation du deuxième chapitre …

Après nous avons surtout distingué les différents types de Clustering envisageables suivant les critères et contraintes existantes, et observé que la construction des hiérarchies ou partitions strictes est commune à la plupart des algorithmes. Les algorithmes de regroupement flou offrent un résultat plus riche du point de vue connaissance.

Dans le chapitre suivant, nous proposons une nouvelle approche visant à extraire la connaissance à partir d'un texte, eu égard sur l'importance d'utilisation de données numériques textes dans les réseaux sociaux ….

Chapitre V. Étude d'un cas pratique sur le processus d'extraction de connaissances à partir de données textes.

V.1 Introduction :

De nos jours, une grande importance est donnée aux publications d'informations numériques via le net, d'où la nécessité d'un système d'extraction de connaissance à partir de texte(ECT).Le principe de base consiste à identifier les besoins d'une personne et par la suite faire une recherche dans un texte pouvant ne pas être structuré, et par la suite récupérer les termes et les informations en relation avec le sujet de recherche et de les structurer sous forme de classes d'informations utiles .

Actuellement, étant donné le rôle, combien important, des nouvelles technologies de l'information et de la communication (NTIC) entrainant un nombre considérable d'informations disponibles sous formes numérique (revues électroniques, livres publiés sur internet, informations non structurées diffusées dans des réseaux sociaux tels que Facebook, Twiter…), il s'avère nécessaire d'en extraire celles pertinentes et fiables.

D'où la nécessité de préconiser un système crédible et performant qui traite l'ensemble de données textuelles et d'en déduire une connaissance structurée et utile.

Le principe du système de recherche d'informations textuelle consiste à organiser l'information en entités et en classes de mots avec les associations entre ces classes et les interactions mutuelles de ses objets.

Toute nouvelle information trouvée va donc enrichir une base de données structurée représentant des tables de chaque type de données.

V.2 Le Texte Mining (TM)[141]

Est l'ensemble de processus de recherche de modèles lié à l'intelligence artificielle permettant de trouver les règles d'association à partir d'un texte non structuré. Plusieurs méthodes se basent sur le tri, le regroupement (à partir de requêtes SQL[5]) de mots et le comptage du nombre de répétition pour identifier leurs importances.

Un processus d'extraction de texte (TM) passe par les quatre étapes suivantes :

1. Préparation des données pour le traitement en transformant celles brutes d'un format à un autre pour pouvoir les traiter de façon adéquate ;
2. recherche des motifs fréquents dans le texte extrait et l'extraction des règles d'association ;
3. présentation des données sous forme visuel par des graphiques, ou des schémas, ici, un outil informatique comme les logiciels de visualisation de données en 2 D ou 3 D seraient nécessaire pour mieux reconnaître l'information pertinente et utile ;
4. nettoyage et optimisation des informations trouvées pour réduire leur taille.

[5] Structured Query Language.

Les techniques de traitement de l'information dépendent de là ou l'information devrait être exploitée et des algorithmes et méthodes utilisés. Tout d'abord, les données récupérées peuvent être triées et organisées au fur et à mesure de leurs acquisitions, après elles sont exposées aux algorithmes formels d'analyses de données. Il est à noter que ces opérations peuvent se faire en parallèle pour plus d'efficacité.

V.3 Extraction de l'information

L'opération d'extraction de l'information est la première étape la plus importante dans le prétraitement du texte.

On extrait l'information [142] pour trouver un texte structuré en langage naturel. L'intervention de DARPA « Message Understanding Conferences » (2004) (MUC) s'est concentré sur l'évaluation de la performance des processus qui se basent sur des tests de documents à « l'aveugle ». Les données à extraire se présentent sous forme d'un modèle qui spécifie une liste d'emplacements de sous-chaînes extraites du document.

En Général, les informations à extraire sont définies par un modèle représentant une liste d'issues à remplir, bien qu'il soit, parfois, représenté par des annotations dans le fichier. Les remplisseurs de vide peuvent être soit un groupe de valeurs précisées ou de chaînes extraites instantanément du document.

L'exemple suivant visualise les contraintes liées au prétraitement du texte :

Annonce : Besoin, de Développeur Web

Emplacement : Rabat. Maroc

Cette personne est responsable de la conception et de la mise en œuvre des composants d'interface Web du serveur ABC et des tâches générales de développement de l'arrière-plan.

Un candidat retenu doit posséder une expérience qui comprend :

- **Un ou plusieurs des éléments suivants : Solaris, Linux, Windows / NT**
- **Programmation en C / C ++, Java**
- **Accès à la base de données et intégration : Oracle, ODBC**
- **CGI et scripting : un ou plusieurs de JavaScript,**
- **Perl, PHP, ASP**
- **L'exposition à ce qui suit est un plus : JDBC, FrontPage et / ou Fusion de cuivre.**

Une expérience de 2+ ans (ou équivalent) est nécessaire.

Modèle rempli

• Réf : \ Développeur Web "

• Lieu : Rabat. Maroc

\ Langages : \ C / C ++ ", \ Java", \ JavaScript ", \ Perl", \ PHP ", \ ASP"

• Plates-formes : \ Solaris ", \ Linux", \ Windows / NT "

• Applications : \ Oracle ", \ ODBC", \ JDBC ", \ FrontPage", \ Cold Fusion "

• zones : \ Data base ", \ CGI", \ Scripting "

• degré requis : expert

• Années d'expérience : \ 2+ ans "

Figure V.1 : Exemple de texte et de modèle rempli pour une offre d'emploi.

D'une figure illustre le cas d'un modèle de document simplifié récupéré à partir d'une opération d'extraction d'informations d'un réseau social de publication d'offre d'emplois. Cet exemple de texte contient seulement des secteurs qui sont chargés par des chaînes extraites directement du document. Plusieurs secteurs peuvent avoir des remplisseurs multiples pour l'usage de l'annonce d'activités en tant que langages (de programmation), environnement de développement, applications et domaines.

Il a été prouvé que l'extraction de la connaissance à partir du texte, est encore une méthode adéquate pour automatiser la mise à jour des pages Web statiques et dynamiques, le traitement dans des guides d'hôtelleries, des pages d'accueil de cours, des annonces de colloques, des offres d'emploi, des annonces de location et des articles de journal sur l'activité d'entreprises.

Des techniques de l'intelligence artificielle pour apprentissage automatique sont d'usage continu pour extraire des informations à partir de fichiers et documents textes .afin de générer aisément des bases de données de renseignements rendant ainsi le texte en ligne plus abordable.

A titre d'exemple, les informations extraites des postes de travail sur le Web peuvent être utilisées pour construire une base de données consultable pour l'inventaire du matériel informatique afin de définir clairement ses besoins.

V.4 Aperçu sur le Texte Mining (Fouille de texte) :

L'exploration d'informations brutes suppose que l'information à extraire est déjà dans une base de données locale. Malheureusement, dans plusieurs applications, l'information numérique n'est disponible que sous forme de fichiers en langue naturelle gratuits plutôt que de bases de données bien organisées. Puisque l'objectif visé de l'extraction d'informations est de surpasser la difficulté de changer un ensemble de fichiers textes en une base de données plus structurée, dont la construction par un par un module Texte Mining va être soumise au processus KDD « Knowledge Discovery in Databases » « Intelligence artificielle » pour une utilisation plus poussée de la connaissance comme illustré dans la Figure suivante. L'extraction d'information peut jouer un rôle évident dans le texte mining.

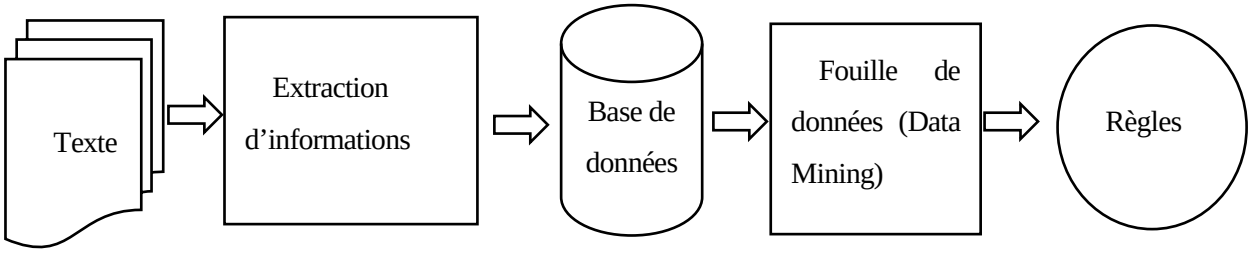


Figure V. 2: Exemple de processus du Texte Mining (réalisée par nos soins).

Certes la construction d'un système TM soit une opération difficile, il y a eu des améliorations nouvelles significatives dans l'emploi de procédés d'apprentissage automatique pour aider à automatiser la réalisation de système TM.

En manipulant manuellement un petit nombre de documents avec les données à extraire, le système TM peut s'avérer utile dans le cas d'un texte long pour la construction d'une base de données. Néanmoins, la précision dans le traitement des systèmes TM actuels est limitée et, par conséquent, une base de données déduite automatiquement comportera sûrement un nombre important de fautes. Cependant, la question qui peut nous interpeler est de savoir si les connaissances acquises à partir de cette base de données sont clairement moins crédibles que dans le cas d'une base de données plus propre.

Ce chapitre présente des exemples montrant que les règles appliquées à une base de données extraite automatiquement sont imprécises au même titre que celles appliquées à une base de données construite manuellement.

V.5 Processus d'extraction de l'information

Plusieurs concepteurs repartissent le processus d'extraction de l'information en étapes de granularité distincte, puis les regroupent en leurs associant les composants des systèmes d'extraction d'information afin d'atteindre à l'objectif fixé. Pourtant, une analyse des différentes approches permet d'identifier six étapes pour le fonctionnement du processus d'extraction d'information comme suit :

- ✓ le traitement préliminaire ;
- ✓ La découverte des noms propres ;
- ✓ L'analyse syntaxique ;
- ✓ l'extraction des événements et des relations ;
- ✓ Résolution de l'anaphore (L'anaphore ,substantif féminin, est une figure de style qui consiste à commencer des vers, des phrases ou des ensembles de phrases ou de vers par le même mot ou le même syntagme) ;
- ✓ Production des résultats d'exploitation.

V.5.1 Traitement préliminaire

Dans cette étape on divise le document texte en plusieurs portions constituant des phases, des segments, des zones vides …

Cette opération peut être effectuée par différents composants liés au langage de programmation comme les stringtokenizer de java, les splitters, les segmenters …, notons qu'une tokenisation permet de diviser le texte en plusieurs Tokenas délimités par un caractère ou espace prédéfinis d'avance.

Cette technique présente l'avantage d'être efficace dans le cas des textes écrits dans divers langages sauf pour ceux asiatiques (le chinois, japonais etc.).

La phase qui suit le traitement est l'analyse lexicale et morphologique du texte.

Il consiste à repérer les mots et les phrases représentants des exceptions , d'ambiguïtés et vient ensuite l'analyse de ces contraintes en utilisant à l'aide des dictionnaires spécialisés de plusieurs langues, ces derniers peuvent regrouper les noms de pays, de villes ou de termes scientifiques …

Un exemple simple consiste à repérer les mots d'un éditeur de texte, comme Word, lequel précise les fautes commises let propose par la même occasion des corrections. Et du coût une mise à jour de son dictionnaire se fait en validant le choix effectué.

V.5.2 Découverte des noms propres.

C'est l'une des tâches les plus importantes dans le processus d'extraction des informations, elle permet de repérer l'ensemble de classes et d'entités représentants les noms propres comme les personnes, les sociétés, les monuments, les pays … Ces informations peuvent être facilement identifiées grâce à leur aspect textuel et aussi la disponibilité des outils de control des langages de programmation « expressions régulières de java » …

V.5.3 Analyse syntaxique

Une analyse syntaxique des phrases s'effectue dans les documents. Après identification des entités et des classes de base dans ce qui précède, les phrases sont analysées pour identifier le groupe de noms de certaines de ces entités et les groupes de verbes. Dans cette étape d'analyse, le travail se prépare pour la prochaine phase d'extraction des événements et des relations dans lesquels ils interagissent. Les groupes de noms et de verbes sont utilisés comme sections pour commencer à travailler à l'étape de correspondance de motif. L'identification de ces groupes est faite à l'aide d'un ensemble d'expressions régulières construites spécialement.

Toutefois, l'analyse complète n'est pas une tâche aisée ; elle nécessite en fait des calculs onéreux qui, à leurs tours, ralentissent tout le processus d'extraction de l'information. Comme il s'agit d'un problème difficile, l'analyse complète est susceptible d'introduire des erreurs. En

revanche, parfois, l'analyse syntaxique totale peut ne pas être nécessaire du tout. De ce fait, de plus en plus de groupes de recherche sur l'extraction d'informations ont tendance à utiliser ce qu'on appelle l'analyse partielle ou superficielle au lieu de l'intégralité. En utilisant uniquement des informations locales, l'analyse peu profonde crée des fragments syntaxiques partiels qui ne se chevauchent qu'avec un niveau de confiance supérieur. Au début du processus d'évaluation, tous les participants de la MUC ont utilisé l'analyse complète. Et le groupe qui est venu avec la nouvelle idée d'analyse peu profonde était Lehnert et al.[143] au cours de MUC[6] (1991), à la suite de l'application de l'analyse syntaxique partielle, ils ont montré une meilleure performance que celle des sites auxquels qui ont essayé de créer des structures syntaxiques complètes [144].

V.5.4 Extraction d'événements et de relations

Ce processus est effectué par la création et l'application de règles d'extraction qui spécifient des motifs différents. Le texte est ajusté à ces motifs et si une correspondance est trouvée, l'élément du texte est étiqueté et extrait après. Le formalisme d'écriture de ces règles d'extraction diffère d'un système d'extraction d'informations à un autre [145].

V.5.5 Résolution de l'anaphore

Toute classe donnée dans un texte peut être révoqué à plusieurs reprises et à chaque fois qu'il pourrait être renvoyé différemment. Afin de reconnaître toutes les manières utilisées pour donner un nom à cette entité, on effectue tout au long une résolution documentée de référence. Il existe plusieurs types de corrélation, mais les plus courants sont la nomenclature pronominale et les noms propres, quand un nom est remplacé par un pronom dans le premier cas et par un autre nom ou un syntagme nominal dans le second [146].

V.5.6 Production des résultats d’exploitation.

Cette phase contient la modification des structures qui ont été ressorties au cours des opérations précédentes dans les modèles de sortie selon les formats spécifiés par un processus. Celui-ci peut inclure des opérations de normalisation différentes pour les dates, l'heure, les monnaies, etc. Par exemple, une méthode d'arrondissement pour les pourcentages peut être exécutée et le numéro d'aire 75.96 sera converti en entier 76.

Toutes les opérations ne doivent pas obligatoirement être achevées dans un seul extrait d'information. Par conséquent, un processus d'extraction d'informations particulier peut ne pas avoir tous les composants possibles. Selon Appelt et Israël (1993)[147], il existe plusieurs facteurs qui affectent le choix des composants des systèmes, tels que les langues .Comme on l'a invoqué précédemment pour le traitement de textes en chinois et en japonais, les langues ne comportent pas de mots clairs ou de limites de phrases ou de textes, qui est bien le cas de l’allemand ou les mots possèdent une structure morphologique difficile, et où le recourt à des modules de traitement est nécessaire.

---

[6] Message Understanding Conference

Par ailleurs, dans les transcriptions de discours informels, pour le type de texte et ses propriétés, des erreurs d'orthographe peuvent apparaitre avec une délimitation de phrases implicites. Si l'information doit être extraite de ces textes, ces remarques devront être prises en considération lors de la conception d'un système en ajoutant les modules correspondants.

Enfin, pour le Processus d'extraction et surtout la reconnaissance des noms, les modules d'analyse et de résolution d'anaphore peuvent ne pas être nécessaires du tout.

V.6 Évaluation de l'extraction de l'information

En tenant compte de l'entrée du texte ou d'un bloc de textes, la sortie attendue d'un système d'extraction d'information peut être définie de façon précise. Pour faciliter l'évaluation des différents systèmes et approches de l'extraction d'information, des paramètres de précision et de rappel ont été adoptés par la communauté de recherche internationale (the IR research community) à cet égard. Pour répondre aux besoins de l'utilisateur, elle a procédé à la mesure de l'efficacité du système, c'est-à-dire la mesure dans laquelle le système produit le rendement maximum (rappel) et seulement la sortie appropriée (précision). Ainsi, le rappel et la précision peuvent être considérés comme la mesure de l'exhaustivité et de l'exactitude, respectivement. Pour les définir de façon formelle, elle a attribué les paramètres #key (nombre total des slots qui doivent être remplis en fonction d'un corpus de référence annoté, représentant un degré de fiabilité ou un étalon), #correct (le nombre de slots correctement remplis en réponse à la demande du système) et #incorrect (le nombre de slots incorrectement remplis en réponse à la demande du système). Une fente est dite remplie correctement si elle ne s'aligne pas avec une fente dans l'étalon (fente parasite) ou si une valeur non valide lui a été attribuée. Pour évaluer le degré de fiabilité du système, le rappel et la précision sont définis comme suit :

$$\boldsymbol{Précision} = \frac{\#correct}{\#correct+\#incorrect} \quad \text{(Equation V.1)}$$

$$\boldsymbol{rappel} = \frac{\#correct}{\#key} \quad \text{(Equation V.2)}$$

D'autre part, afin d'obtenir une image plus fine de la performance des systèmes d'extraction d'information, la précision et le rappel sont souvent mesurés séparément pour chaque type d'emplacement.

La mesure *F* est utilisée comme moyenne harmonique pondérée de précision et de rappel. Elle est définie comme suit :

$$\boldsymbol{F} = \frac{(\beta^2+1)*Précision*rappel}{(\beta^2*Précision)+rappel} \quad \text{(Equation V.3)}$$

Dans la définition ci-dessus, β est une valeur non négative, utilisée pour ajuster leur pondération relative (β^2 = 1,0 donne une pondération égale au rappel et à la précision, et des valeurs de β données inférieures à 1 augmentent la précision).

D'autres paramètres sont également utilisés dans la littérature, par exemple le taux d'erreur de fente, SER [148-149], qui est défini comme suit :

$$\boldsymbol{SER} = \frac{\#\boldsymbol{incorrect}+\#\boldsymbol{missing}}{\#\boldsymbol{key}} \quad \text{(Equation V.4)}$$

Où #missing indique le nombre d'emplacements dans la référence qui ne s'alignent avec aucun emplacement dans la réponse du système. Il reflète, en fait, le rapport entre le nombre total de slot erronés et le nombre total de slots dans la référence. Selon les besoins particuliers, certains types d'erreurs (par exemple des fentes parasites) peuvent être pondérés afin de les juger leur importances relatives par rapport aux d'autres.

V.7 Conclusion

L'automatisation de l'extraction de l'information semble être la meilleure technique pour procéder à celle du texte. Le procédé général correspondant est exploité dans le domaine des réseaux sociaux et sur le web. Les facteurs tels que la précision, Rappel, F-mesure et le taux d'erreur de la fente sont utilisés pour mieux améliorer l'évaluation dans l'extraction d'information.

Chapitre VI. La recherche d'informations numériques en évaluant quatre modèles.

Résumé : Alors que l'information devient de plus en plus abondante et accessible sur le Web, les chercheurs n'ont pas à fouiller dans les livres et les bibliothèques. Les pages web sont riches en informations textuelles, les moteurs de recherche web mettent à disposition des internautes différents fichiers correspondant aux mots clés recherchés. Ce grand nombre de données numériques rend difficile le tri manuel, il est donc nécessaire d'automatiser la collecte d'informations utiles en utilisant des techniques basées sur l'intelligence artificielle.

À l'ère numérique d'aujourd'hui, une grande importance est accordée aux techniques de recherche d'informations via Internet. Dès lors, il apparaît essentiel de prévoir un système crédible et performant traitant de l'ensemble de l'information textuelle, afin d'en déduire des connaissances structurées et utiles. Ce travail se concentre sur quatre modèles utilisés dans le domaine de la recherche documentaire, et met en évidence leurs limites d'utilisation, en vue de développer de nouvelles techniques pouvant combler les lacunes détectées. À la fin, les paramètres d'évaluation seront discutés pour améliorer l'intervention humaine dans la prise de décision.

VI.1 Introduction

Les informations textuelles répondant à une demande de recherche sont classées en fonction de leurs scores de pertinence. Cependant, les données obtenues ne sont généralement pas structurées ; la recherche n'utilise pas de requêtes SQL sur les bases de données. Les pages Web diffèrent des documents texte ; ils contiennent des hyperliens et des textes d'ancrage. Les hyperliens sont très utiles pour la recherche et jouent un rôle important dans le classement des algorithmes. De même, les textes d'ancrage associés aux hyperliens sont nécessaires, car un texte d'ancrage est souvent une description plus précise de la page vers laquelle pointe l'hyperlien. En HTML, le contenu d'une page web est structuré en blocs de champs (titre, métadonnées, corps, etc.).

Certains champs sont très importants par rapport à d'autres, ils sont utiles pour l'indexation et le référencement de pages web par les robots des moteurs de recherche, cependant le spam ou la redirection vers des pages web publicitaires reste un problème, et peut nuire à la qualité des résultats d'une requête de recherche sur la toile.

De plus, les informations textuelles écrites dans des langues différentes, et dans la même base de données posent un problème majeur de classification.

A travers quatre modèles nous mettrons en évidence l'importance de l'intervention humaine pour une bonne classification des données textuelles.

VI.2 Le processus proposé pour rechercher des informations à partir de documents

L'information n'est utile que si elle apporte des réponses à la question de l'utilisateur. Pour ce faire, un système de recherche d'informations[150] doit d'abord récupérer tous les fichiers et documents répondant aux demandes des clients, puis les indexer et les organiser avant de les stocker et de les publier. La figure ci-dessous représente l'architecture générale du processus de recherche d'informations [151] :

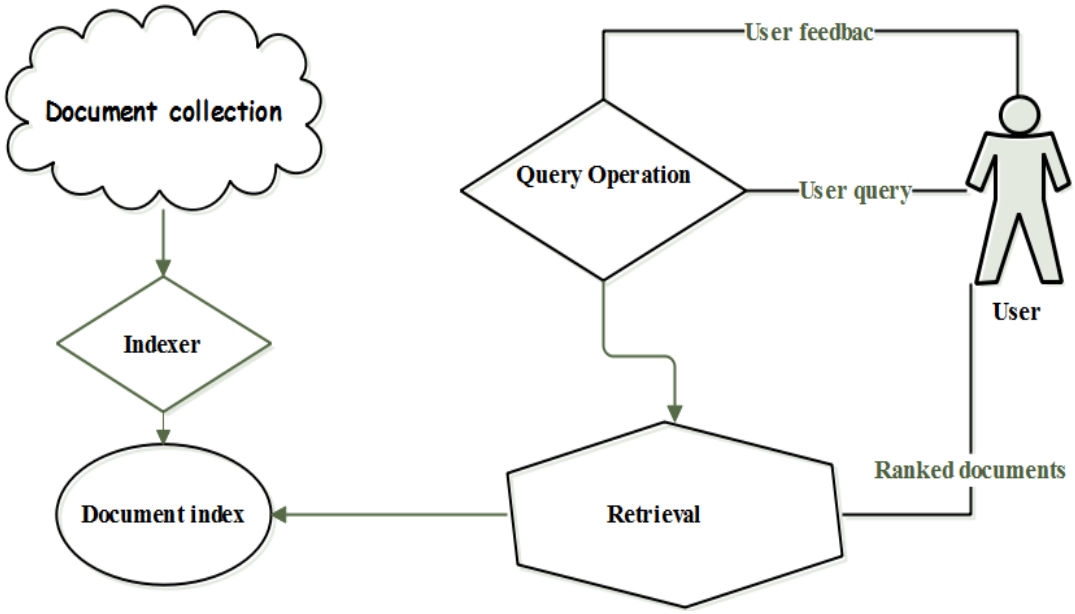


Figure VI.1. Une architecture générale du processus de recherche d'informations

Un utilisateur envoie une demande au système de récupération ; ce dernier consulte au préalable les documents indexés, il les classe selon le score de pertinence répondant aux besoins du client puis renvoie un ensemble de résultats.

Pour bien comprendre le principe de recherche d'informations, l'utilisateur exprime ses besoins sous différentes formes, soit une requête composée de mots-clés indépendants ou liés par des opérateurs logiques tels que ET, OU, NON..., certaines applications de recherche d'information les utilisent pour afficher les index des bases de données contenant des pages Web ou des fichiers collectés. Les moteurs de recherche ont leur propre syntaxe pour traiter de telles requêtes. Dans la plupart des cas, l'utilisateur commence sa recherche en écrivant des phrases simples, et selon la langue et le style d'écriture, cependant, l'ordre des mots du texte saisi influence la qualité des résultats obtenus et en conséquence, le document récupéré peut ne pas contenir tous les termes de la requête. Dans ce cas, un autre type de requêtes appelées requêtes de proximité est requis. Le système calcule la proximité (distance) des termes composants, et classe les pages et documents fondés, en tenant compte de ce facteur de proximité et de l'ordre des termes ; l'utilisateur peut définir cette commande à l'avance. D'autres expressions recherchées peuvent être mises entre deux guillemets forçant ainsi le moteur de recherche à trouver le document complet et à renvoyer les résultats ou les liens URL de pages similaires.

Poser des questions sur le langage naturel est une autre façon de répondre aux besoins de l'utilisateur. Cependant, cette technique est difficile à mettre en œuvre en pratique, le principe est de préparer un ensemble de questions avec leurs réponses, puis de les structurer selon un modèle de système d'information. Après avoir préparé la requête de la question et l'avoir mise en forme, en tenant compte des prépositions du langage naturel utilisé, l'utilisateur lance alors la requête et obtient ses résultats.

Dans les sections suivantes, quatre modèles différents définissent les facteurs de pertinence[152] et de proximité exploités dans les requêtes, afin d'extraire des documents et des pages Web répondant aux besoins des utilisateurs.

VI.3 Méthode de recherche de différents modèles de recherche d'information

Quatre modèles de recherche d'information peuvent être distingués : le modèle booléen, le modèle spatial vectoriel, le modèle probabiliste et le modèle connexionniste et linguistique.

Les documents « D » ou les pages Web collectées peuvent être considérés comme un ensemble de mots ou de termes, leurs positions sont ignorées dans les phrases. Chaque terme « ti » est lié à un poids « wij », tous ces termes distincts représentent une collection V :

- $V = \{t_1, t_2, t_3, \ldots., t_N\}$ avec N = la taille du document.
- Pour un document dj∈D, on associe un poids wij au terme ti
- Si ti existe dans dj alors wij = 1 sinon wij = 0
- Chaque document dj représente un vecteur des termes : $d_j = (w_{1j}, w_{2j}, w_{3j}, \ldots., w_{Nj})$

On peut donc représenter l'ensemble des documents « D » par une matrice « table » de vecteurs di « ou d'objets » dont les attributs sont wij.

VI.3.1 Le modèle booléen pour la recherche d'informations

Ce modèle a été développé par Edward Alan Fox[7] en 2006. Une requête de recherche est représentée par une expression de termes liés par des opérateurs logiques Et, Ou et Non.

Pour mieux expliquer le principe de base, nous supposons trois documents d_1 , d_2 et d_3) de la collection D, et les termes de la requête t1 et t2. Chaque document dj est représenté par un vecteur des termes liés aux poids wij avec i∈ {1,2} et j∈ {1, 2, 3, 4} où :

- $d_1 = (w_{11}, w_{21})$
- $d_2 = (w_{12}, w_{22})$
- $d_3 = (w_{13}, w_{23})$
- $d_4 = (w_{14}, w_{24})$

Le tableau ci-dessous représente le modèle booléen de recherche d'informations :

[7] Garcia E.: The Extended Boolean Model. Published at www.minerazzi.com (2016).

Tableau VI.1. Modèle booléen d'extraction d'informations

	Termes de la requête		Requête de similarité	
	t1	*t2*	*t1 OR t2*	*t1 AND t2*
d1	1	1	1	1
d2	1	0	1	0
d3	0	1	1	0
d4	0	0	0	0

Lors de la recherche des termes t1 et t2 dans les documents di, et selon l'expression booléenne utilisée dans la requête, on choisit les documents qui ont le plus grand nombre de 1 dans le tableau. La mise en œuvre de ce modèle en pratique est aisée, mais les résultats ne sont pas satisfaisants, car la fréquence et la proximité des termes dans un document ne sont pas prises en compte. Les requêtes composées par des expressions logiques complexes des termes, sont utilisées par les moteurs de recherche pour extraire certains documents pertinents et en exclure d'autres. Ainsi, l'apparition du modèle spatial vectoriel prend en compte la fréquence de répétition des termes dans la recherche d'informations.

VI.3.2 Le modèle spatial vectoriel pour la recherche d'informations :

Un texte est un regroupement de termes, il est assimilé à un vecteur dans un espace vectoriel[153], on mesure le degré d'importance de chaque terme dans le document par un nombre réel, et une fréquence représentant sa répétition dans le document. Un document répondant à une requête est celui dont le vecteur est proche de celui d'une requête, la mesure de pertinence est donc le calcul du cosinus des deux vecteurs.

Pour un espace vectoriel V : $V = \{t_1, t_2, t_3, \ldots, t_n\}$ représentant n termes ti, on considère un document D tel :

D =(a_1 , a_2 , a_3 ,…, a_n), avec ai représente le poids de ti dans D.

Soit la requête Q : Q = (b_1, b_2, b_3, ..., b_n), avec bi le poids de ti dans Q.

On définit le score ou le rappel « R », également noté comme similitude, par une fonction telle que :

$$R(D, Q) = Sim(D, Q)$$

La figure ci-dessous schématise le modèle spatial vectoriel par une matrice représentative.

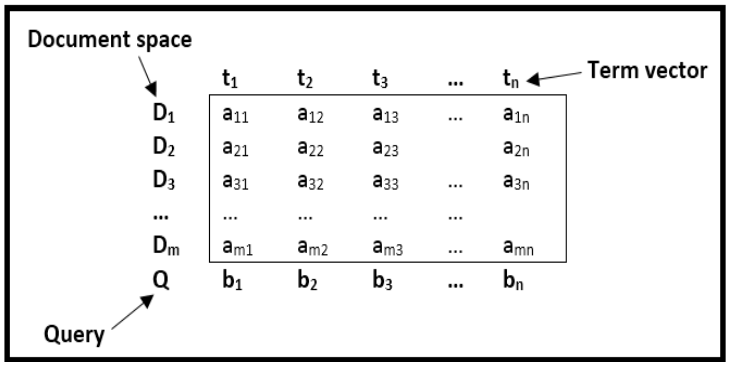


Figure VI.2. Représentation matricielle des éléments du modèle spatial vectoriel

Certains facteurs de similarité ou de distance peuvent être utilisés, comme le produit scalaire de deux vecteurs ou le cosinus, ainsi que l'indice de Jaccard ou l'indice de Sørensen-Dice 1945, également appelé coefficient de Dice :

- Le produit scalaire du document D et de la requête Q est défini comme :

$$Sim(D,Q) = D.Q = \sum_i (a_i * b_i) \text{(Equation VI.1)}$$

- De même, nous définissons Cosinus

$$Sim(D,Q) = \frac{\sum_i (a_i * b_i)}{\sqrt{\sum_i a_i^2 * \sum_i b_i^2}} \quad \text{(Equation VI.2)}$$

- Pour l'indice de Sørensen-Dice, on a

$$Sim(D,Q) = \frac{2*\sum_i (a_i * b_i)}{\sum_i a_i^2 + \sum_i b_i^2} \text{Sim(D, Q)} = \frac{2*\sum_i (a_i * b_i)}{\sum_i a_i^2 + \sum_i b_i^2} \text{Sim(D, Q)} = \frac{2*\sum_i (a_i * b_i)}{\sum_i a_i^2 + \sum_i b_i^2}$$

(Equation VI.3)

- Et pour l'index Jaccard on a

$$Sim(D,Q) = \frac{\sum_i (a_i * b_i)}{\sum_i a_i^2 + \sum_i b_i^2 - \sum_i (a_i * b_i)} \text{(Equation VI.4)}$$

Dans la matrice de la Fig.2, on compare la similitude (le cosinus du clou entre deux vecteurs) entre deux documents pour les classer selon leurs termes ti, ou bien calcule le score de correspondance des termes entre les documents Di et la requête Q.

Une autre façon d'évaluer le degré de pertinence est de calculer un score de pertinence pour chaque document Di relatif à la requête Q. La méthode Okapi proposée en 1976 par Robertson

[154] et ses variantes sont des techniques de pondération largement utilisées dans la recherche d'informations.

Le calcul du score de pertinence Okapi donne plus de précision que le cosinus pour les requêtes courtes.

Dans ce contexte, nous définissons ce score comme suit :

$$okapi(Dj,Q) = \sum_{t_i \in Q,D_j} \ln \frac{N-df_i+0.5}{df_i+0.5} * \frac{(k_1+1)f_{ij}}{k_1(1-b+b\frac{dl_j}{avdl})+f_{ij}} * \frac{(k_2+1)f_{iq}}{k_2+f_{iq}} \quad \text{(Equation VI. 5)}$$

Avec:

- t_i un terme.
- f_{ij} est le nombre de fréquences brutes du terme ti dans le document Dj
- f_{iq} est le nombre de fréquences du terme ti dans la requête Q
- N est le nombre total de documents dans D.
- df_i est le nombre de documents contenant ti
- dl_j est le langage d'octets du document Dj
- $avdl$ est la longueur moyenne des documents dans la collection D

Avec les paramètres : k1 varie entre 1 et 2, k2 entre 1 et 1000 et b souvent égal à 0,75.

Dans ce modèle vectoriel, les mots considérés comme fréquents et vides comme « de, et, ... », selon la langue du dictionnaire utilisé, doivent être supprimés des termes pour avoir des résultats pertinents qui se rapprochent de la requête.

Un autre inconvénient est qu'il peut ignorer les termes peu fréquents dans le document, qui ont un poids considérable dans la recherche comme "terrorisme...", cependant le modèle permet une bonne indexation des documents qui répondent en tout ou en partie à la demande.

Ce modèle est relativement simple à mettre en œuvre, sa complexité est linéaire comme le modèle booléen. Mais à cause de la taille des documents sur le net et de la taille de la requête en termes recherchés, cela devient coûteux pour les moteurs de recherche. Une solution possible est de traiter les documents à l'avance et de les indexer selon les termes dans les répertoires. Des robots ou des processus autonomes effectuent ces opérations automatiquement.

VI.3.3 Le modèle probabiliste de recherche d'informations pertinentes :

Dans ce modèle, un document D peut être utile pour répondre à une requête Q de l'utilisateur, la pertinence de D par rapport à une requête est indépendante des autres documents. La probabilité de pertinence est définie comme un facteur de classement dans la recherche pour information.

Robertson (1977) dans son article[155] a présenté le principe du classement probabiliste (PRP), dans lequel le score de pertinence d'un document D dépend d'une requête Q telle que :

$$RSV(D,Q) = \frac{P(R|D)}{P(NR|D)} \qquad \text{(Equation VI.6)}$$

Avec R : l'ensemble des documents pertinents pour la requête Q et NR représentent celui qui n'est pas pertinent.

P (R | D) : probabilité que D soit dans R, et P (NR | D) : probabilité que D soit dans NR.

Pour estimer le calcul de P (R | D, on utilise le théorème de Bayes ; il donne la probabilité de pertinence avant et après observation du document D.

Comme défini au début, chaque document est décrit par la présence ou l'absence de termes d'index, il est représenté par un vecteur binaire$(x_1, x_2, x_3, \ldots, x_N)$, avec :

$$x_i = \begin{cases} 0 \text{ si le terme d'index est manquant} \\ 1 \text{ si le terme d'index est présent} \end{cases} \quad \text{(Equation VI.7)}$$

Ainsi pour ce document, on peut associer deux situations possibles W1 et W2 telles que :

- w_1 = le document est pertinent
- w2 = le document n'est pas pertinent.

On essaie de calculer la probabilité de x en l'observant au hasard :

$$P(x) = P(x/w_1)P(w_1) + P(x/w_2)P(w_2) \quad \text{(Equation VI.8)}$$

Avec:

- P (x / w_1) et P (x / w_2) représentent la probabilité de pertinence de x la probabilité de non-pertinence de x, respectivement.
- P (w_1) et P (w_2) représentent la probabilité de documents pertinents et de documents non pertinents, respectivement.

La valeur P(x) permet de mesurer quantitativement la densité de x par rapport à des documents pertinents ou non.

Ainsi, pour une bonne prise de décision, nous utilisons la règle suivante :

Si P (x / w1) > P (x / w2)) alors x est pertinent sinon x n'est pas pertinent et si .

En tenant compte de cette règle, les documents pertinents x sont extraits aléatoirement et triés par ordre décroissant de P (x).

La récupération des documents est liée au seuil de probabilité minimum imposé par l'utilisateur et au coût des opérations de dépôt de l'application utilisée. Il est souvent indispensable de connaître la taille pour mieux gérer le chemin parcouru lors de recherches basées sur des heuristiques. Par conséquent, l'utilisation d'arbres, également appelés dendrogrammes, pour la classification des données ou l'introduction d'algorithmes de clustering en mélangeant des densités de probabilité tels que l'algorithme EM (Expectation and Maximization) de Dempster [156] devient important.

De plus, il existe un algorithme qui utilise le partitionnement de l'espace de données de la grille, comme l'algorithme STING [157] « Statistical Information Grid Approach to Spatial Data Mining », qui construit une hiérarchie de grille : un cluster est composé d'éléments denses et connectés nommées cellules. Pour chaque cellule du niveau courant, STING calcule l'intervalle

de probabilité pour lequel les sous-cellules sont denses, et distance les autres ; de même, il répète le même traitement jusqu'à ce qu'il atteigne le niveau le plus bas.

A l'exécution de cet algorithme, nous fixons une base de données et nous devons répondre à une requête sélection basée sur les paramètres d'une cellule "point".

STING algorithm;

Input: a database containing "cell" object parameters, and a SQL type query.

Output: A set of cells representing a layer or grid.

Begin

1. Determine a layer with which one begins.
2. For each cell in this layer, we calculate the confidence interval (or margin estimate) of the probability that this cell will have the answer to the database query asked at the beginning.
3. Depending on the value of the interval calculated above, we mark the cell as appropriate or inappropriate.
4. If this layer is the bottom layer, go to step 6; otherwise, go to step 5.
5. Descend the hierarchy structure by one level. Step 2 is followed for those cells that form the appropriate cells of the higher level layer.
6. If the specifications of the query are verified, go to step 8; otherwise, go to step
7. Look for the fall of these data in the appropriate cells and make a further transformation. Return the result that meets the requirement of the query. Go to step 9.
8. Find the appropriate cell regions. Return those regions that meet the requirement of the query. Go to step 9.
9. Stop.

End.

Algorithme VI.1 STING

Cet algorithme est flexible ; il nécessite un seul passage sur les données et à chaque niveau de notre hiérarchie, l'utilisateur peut facilement insérer un nouvel objet dans les cellules appropriées.

La complexité de cet algorithme dépend du nombre de cellules k, elle peut être de l'ordre de O (k), le nombre k est inférieur au nombre de données puisque l'algorithme rejette les données absurdes.

En résumé, le point fort du modèle probabiliste est qu'il présente une base théorique solide, le classement optimal des résultats de recherche est justifié en théorie, mais en pratique, il est difficile de calculer avec précision la probabilité, la fréquence des termes est ignoré. Par conséquent, les résultats ne peuvent être convaincants et nécessitent une intervention humaine.

VI.3.4 Le modèle de langage statistique pour la recherche d'informations :

Ce type de modèle[158] s'appuie sur des probabilités et des statistiques pour modéliser la capture, la position et la probabilité de fréquence des mots dans une langue.

Pour un corpus, on essaie de classer les documents en tenant compte de critères comme l'auteur, la date de parution, les thèmes etc...

La probabilité de la requête de recherche dépend du modèle de langue de chaque document, du traitement de la langue parlée et de la fréquence des mots ainsi que des règles logiques de la langue[159]. Les statistiques permettent la conception, la configuration et l'évaluation du modèle de recherche d'informations dans le corpus. La pertinence des résultats de la requête est liée au traitement statistique lors de l'indexation, de la correction orthographique et grammaticale, ainsi qu'à la reconnaissance automatique courte de l'écriture, de la parole et de la traduction des données. Ceux-ci poussent les moteurs de recherche à utiliser plus de volume de texte dans des répertoires alimentés par des processus robotiques et des applications robustes pour les langues d'apprentissage automatique et de reconnaissance vocale.

VI.4 Résultats et discussion des évaluations des modèles de recherche

Compte tenu du nombre de modèles de recherche d'informations et des contraintes liées aux types de données existants sur le web, il est nécessaire d'évaluer les performances, la qualité et la fiabilité des résultats récupérés par le modèle. Le paramètre « recall »[160] est défini par :

$$recall = \frac{number\ of\ relevant\ document\ returned}{number\ of\ relevant\ database\ documents} \quad \text{(Equation VI.9)}$$

Ce paramètre est un pourcentage indiquant la capacité du modèle à récupérer les documents pertinents qui satisfont la demande de l'utilisateur. Cette recherche est effectuée dans une base de données déjà disponible.

Un autre paramètre appelé "précision", mesure la capacité du système de recherche à éliminer les documents non pertinents à la requête, il est défini comme suit :

$$precision = \frac{number\ of\ relevant\ documents\ returned}{number\ of\ document\ returned} \quad \text{(Equation VI.10)}$$

Les bons résultats récupérés correspondent à une précision égale à 1, ceux-ci sont loin d'être vérifiés en pratique, donc un rappel « recall » égal à 1 signifie que le système est efficace pour classer les documents pertinents de la base de données.

Deux autres paramètres « Silence » et « Bruit ou Noise » mesurent le cas contraire :

Silence = 1-recall (Equation VI.11)

Noise = 1-precision (Equation VI.12)

Le paramètre Silence indique le pourcentage de documents pertinents non renvoyés et le paramètre Bruit(Noise) indique le pourcentage d'éléments non pertinents renvoyés par le système de recherche.

Pour un moteur de recherche Web, nous ne regardons souvent que les 20 premières pages, puis nous pouvons calculer les détails des 5, 10, 15, 20, 25 et 30 pages renvoyées. Par contre, le

paramètre « recall » n'est pas très intéressant puisqu'on ne connaît pas le nombre de documents pertinents dans la base de recherche du moteur de recherche. Un autre problème rencontré lors de l'évaluation est que la pertinence d'un document est souvent déterminée par l'utilisateur, ce qui rend l'évaluation non spécifique.

L'évaluation expérimentale sur les données ci-dessous est représentée par la courbe donnant la précision et le rappel des documents (Figure VI. 2).

Tableau VI.2. Valeurs de rappel et de précision pour une recherche sur le Web

Document	score	Relevant	Precision	Recall
d1	9.92	Relevant	1.00	0.2
d2	9.77	Non relevant	0.50	0.2
d3	9.76	Relevant	0.67	0.40
d4	9.59	Relevant	0.75	0.60
d5	8.72	Non relevant	0.60	0.60
d6	6.85	Relevant	0.67	0.80
d7	6.51	Relevant	0.57	0.80
d8	4.32	Non relevant	0.63	1
d9	4.16	Non relevant	0.56	1
d10	3.47	Non relevant	0.50	1
d11	2.69	Non relevant	0.45	1
d12	2.04	Non relevant	0.42	1
d13	1.84	Non relevant	0.38	1
d14	1.67	Non relevant	0.36	1
d15	0.07	Non relevant	0.33	1

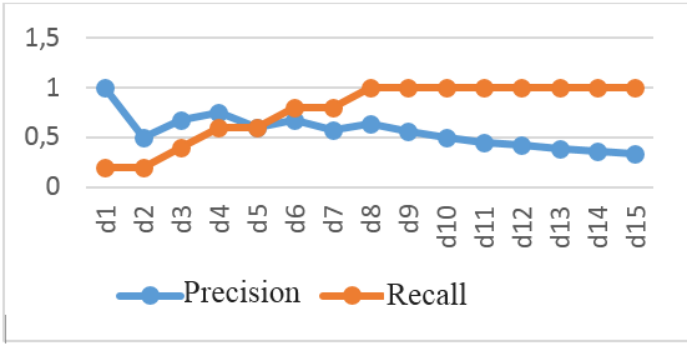


Figure VI. 3. Représentation de précision et rappel « recall » de documents.

Notons qu'à partir de d5, la précision diminue par rapport au rappel, et elle est meilleure pour les premiers documents renvoyés qui ont de petites valeurs du paramètre de rappel, cette précision est mauvaise pour les grandes valeurs de rappel. Ainsi, l'idéal est de choisir les trois premiers documents récupérés et de refaire une analyse humaine sur la pertinence de chacun.

a. Notre système et analyse comparative des modèles :

A l'aide d'une application, réalisée avec le langage C-Sharp, dans une plateforme Windows, nous avons essayé d'implémenter les quatre modèles et de faire des tests sur des bases de données depuis le Serveur «https://archive.ics.uci.edu/ml/datasets/Banque+Marketing ».

L'application est un système d'extraction de connaissances à partir de données ; il permet de traiter, classer et visualiser des données. L'exemple ci-dessous montre la visualisation globale des deux classes de demandeurs de crédit selon leurs états martiaux "homme célibataire, marié..." selon l'ensemble des autres attributs :

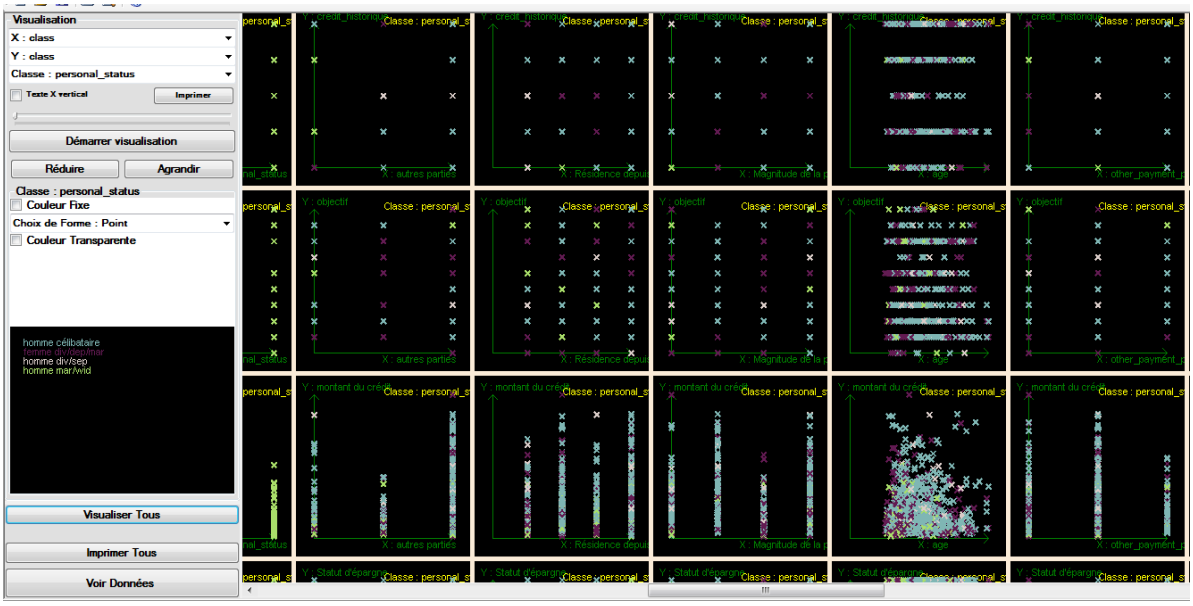


Figure VI.4 Visualisation globale de deux classes de demandeurs de crédit (célibataires et mariés)

Dans cette figure les clusters sont representés par des points colorés, dans cette situation ,ilest difficile de les identifier facilement puisqu'ils se chauvochent .

Une évaluation comparative connue sous le nom de benchmarking a été développée pour identifier les forces et les faiblesses de chaque modèle, voir tableau VI.3

Tableau VI.3. Comparaison des quatre modèles de recherche d'informations.

Modèle	Forces	Faiblesses
MODÈLE BOOLÉEN	• Simple à mettre en pratique • Complexité linéaire	• Résultats insatisfaisants • Ignorer la fréquence et la proximité des termes
MODÈLE ESPACE VECTORIEL	• Simple à mettre en pratique • Mesure la fréquence des termes et la proximité.	• Coût élevé, traitement lent • Nécessite une indexation préalable des documents

MODÈLE PROBABILISTE	• Bases théoriques solides • Existe dans plusieurs algorithmes de Clustering	• Difficulté à calculer la probabilité en pratique • Fréquence des termes ignorés • Nécessite une intervention humaine
MODÈLE DE LANGAGE STATISTIQUE	• Utilise des statistiques pour l'indexation	• Dépendance des langues. • Trop de paramètres à prendre en compte.

À l'aide de Systèmes d'intelligence d'affaires « Business Intelligence Systems »[161], ces différents modèles de recherche d'informations sont exploités dans la recherche de compétences en marketing numérique destinées à l'industrie, au commerce, à la santé et bien d'autres.

Plutôt que de se baser sur l'intuition et l'expérience humaine, les forces de chaque modèle sont utilisées pour trouver des paramètres de référence fiables et précis. Le benchmark exploite les données catégorisées à travers ces modèles pour permettre aux entreprises d'évaluer et de développer leurs activités. La récupération de données manuelle n'est pas facile. Tout en fournissant leurs services aux utilisateurs, les applications Web comme Google Analytics sont là pour recouper et classer les données pertinentes sur le comportement et les besoins des utilisateurs, ensuite ces applications utilisent les algorithmes d'apprentissage continu et automatique pour extraire les connaissances et renforcer les résultats des modèles.

VI.5 CONCLUSION

Ce document décrit le processus général de recherche d'informations ; il examine les principaux modèles de recherche d'informations numériques qui sont le modèle booléen, celui de l'espace vectoriel, le langage probabiliste et statistique.

En introduisant la notion de score pour la classification des documents, les contraintes et les limites de chaque modèle ont été identifiées. Pour l'évaluation du degré de pertinence, le calcul du score de pertinence « Okapi » a été introduit ; il prend en considération le nombre de fréquences des termes de la demande dans les documents retournés. Comme le modèle booléen, le vectoriel est simple à réaliser mais la taille de la requête en terme recherché et le nombre de données à traiter, diminuent les performances du système.

La solution est de préparer d'avance les bases de données en automatisant les techniques d'indexation et de clustering. Par la suite, le modèle probabiliste exploite la probabilité comme unité de mesure de pertinence sans oublier l'utilisation du théorème de Bayes pour le calcul.

L'algorithme STING peut être très intéressant dans la classification, puisqu'il divise l'espace de données en une grille et une couche puis calcule la probabilité que les cellules répondent à la requête de l'utilisateur. Ainsi, il présente une structure hiérarchique des documents pertinents pour améliorer les performances du système de recherche. De plus, le modèle de langage statistique comporte plusieurs facteurs qui le rendent difficile à mettre en œuvre dans la pratique. Dans cette perspective, certains moteurs de recherche ont déjà commencé à exploiter l'intelligence artificielle pour la reconnaissance de la voix et de l'image.

Enfin, les paramètres d'évaluation d'un système de recherche d'informations ont été discutés, mais jusqu'à présent, il est apparu que l'intervention humaine est nécessaire pour la prise de décision.

Chapitre VII. La Classification basée sur la densité avec l'algorithme DENCLUE

Résumé :

La classification de l'information est un domaine de recherche vague et difficile à explorer, d'où l'émergence de techniques de regroupement, souvent appelées Clustering. Il faut faire la différence entre une classification non supervisée et une classification supervisée. Les méthodes de clustering sont nombreuses. Le partitionnement et la hiérarchisation des données poussent à les utiliser sous forme paramétrique ou non. Aussi, leur utilisation est influencée par des algorithmes de nature probabiliste lors du partitionnement des données. Le choix d'une méthode dépend du résultat du Clustering que l'on veut avoir. Ce travail porte sur la classification à l'aide des algorithmes DBSCAN (Density-Based Spatial Clustering of Applications with Noise) et DENCLUE (DENsity-based CLUstEring) à travers une application réalisée en csharp. Grâce à l'utilisation de trois bases de données qui sont la base de données IRIS, Breast Cancer Wisconsin (Diagnostic) Data Set et Bank Marketing Data Set, nous montrons expérimentalement que le choix des paramètres de données initiaux est important pour accélérer le traitement et peut minimiser le nombre d'itérations pour réduire le temps d'exécution de l'application.

VII.1 INTRODUCTION

La volonté de classer pour réduire et mieux maîtriser s'est progressivement développée vers l'ambition d'automatiser la classification pour concevoir et, pourquoi pas, prédire l'avenir. Cette vision humaine a poussé la communauté scientifique du domaine, depuis des années, à s'améliorer et à trouver de nouvelles solutions permettant de libérer des connaissances utiles à travers des applications puissantes et autonomes qui concernent la classification automatique, la simulation et la visualisation à deux. ou trois dimensions d'événements qui peuvent se produire dans le futur

Dans l'analyse des données statistiques, un individu est associé à une classe parmi plusieurs classes prédéfinies. Mais dans le cas d'une classification non supervisée, les classes ne sont pas connues à l'avance, on regroupe alors en individus ou objets ayant des propriétés communes à partir d'un grand nombre de données, d'où la complexité du regroupement et de l'identification du nombre de classes.

Une telle classification est apparue dans les analyses de données archéologiques (Classer les objets selon l'âge), et de données médicales (Classer les patients selon l'âge, le poids, les symptômes, etc…)

Par la suite, d'autres usages de la classification[162] dans le traitement de données textuelles, de reconnaissance d'images d'extraction de connaissances, qui ont poussé les chercheurs à se concentrer sur des recherches avancées d'algorithmes et de techniques de clustering au fur et à mesure de leur progression. et avec le développement des outils

informatiques, d'où l'émergence d'une science visant les processus d'extraction de connaissances à partir de données [163].

Dans certaines situations, l'apparence et le format des clusters peuvent être très utiles pour l'analyse et l'extraction de connaissances lors de la visualisation des méga données et du choix des clusters. On peut trouver des clusters noyés dans d'autres ou isolés, selon la représentation des données. Dans ce contexte, la contribution que nous apportons est la réalisation d'une application informatique d'exploration de données qui analyse les données de la phase de prétraitement à la classification en utilisant le clustering basé sur la densité.

Plus récemment, Platoš [164] a présenté une analyse de clustering basée sur la densité. L'idée était d'identifier des régions denses et à grain fin dans les données, leur regroupement produit des grappes de forme arbitraire. Il s'agit d'un algorithme de classification hiérarchique appelé Density-Based Spatial Clustering of Applications with Noise (DBSCAN) basé sur la densité. Les groupes sont formés par des régions de grille denses de connectivité adjacente, car ils partagent un côté et un coin communs. Les régions connectées peuvent être trouvées en traversant d'abord ou en profondeur à l'aide d'un modèle basé sur un graphique. L'algorithme donne de bonnes représentations avec les points de bruit. Le contour des amas est plus lisse, tandis que les régions rectangulaires sont remplacées par une zone sphérique identifiée par le rayon

Nous abordons dans notre méthode de recherche deux algorithmes, le DBSCAN qui permet le Clustering[165] par densité, et le DENCLUE (DENsity-based CLUstEring)[166], il a été proposé[167] entre 1998 et 2000, basé sur des fonctions mathématiques. Bien qu'il acquière une grande complexité avec le nombre de paramètres d'entrée, il montre des résultats acceptables. Compte tenu de l'importance des usages de la classification en botanique, en médecine et en marketing bancaire, nous avons testé dans la section 3 trois bases de données correspondantes.

VII.2 MÉTHODE DE RECHERCHE : Clustering basé sur la densité

Le clustering basé sur la densité[168-169] utilise la notion de voisinage[170-171] pour déterminer un noyau de cluster.

$N_\varepsilon(x_i)$ est son voisinage, c'est l'ensemble des points de X dont la distance à x_i est inférieure ou égale au rayon ε

$$N_\varepsilon(x_i) = \{x_j \epsilon X |\ d(x_i, x_j) \leq \varepsilon\} \qquad \text{(Equation VII.1)}$$

Pour ce type d'algorithme, nous devons définir deux paramètres : le rayon minimum autour du noyau et M le nombre minimum de points pour le voisinage $N_\varepsilon(x_i)$.

L'algorithme DBSCAN (Density-Based Spatial Clustering of Applications with Noise)[172] est l'un des plus connus utilisant ces deux paramètres pour l'identification de clusters basée sur la notion de voisinage autour d'un noyau.

L'algorithme est très simple à comprendre et ne nécessite pas que l'on lui fournisse le nombre de clusters à trouver. Il est capable de gérer des données absurdes et de les éliminer du processus de partitionnement. Cet algorithme peut être décrit comme suit (Algorithme VII.1) :

Algorithm DBSCAN(Density-Based Spatial Clustering of Applications with Noise)

Input : two parameters: $\boldsymbol{\varepsilon}$ the minimum radius around the nucleus and $\boldsymbol{M}$ the minimum number of points for the neighborhood fixed in advance and the set of points $\boldsymbol{X}$.

Output : Partition $\boldsymbol{C} = \{\boldsymbol{C_1}, \dots, \boldsymbol{C_k}\}$ of $\boldsymbol{X}$ with k d-clusters.

Begin

- Initialize cluster $\boldsymbol{C_{id}} = \boldsymbol{\phi}$ with *id=1*
- For i =1 to n do
    a. If $\boldsymbol{x_i}$ is not a core or if $\boldsymbol{x_i} \boldsymbol{\epsilon} \bigcup_{\boldsymbol{j=1,\dots,id}} \boldsymbol{C_j}$ then go back to step 2
    b. Build Cluster $(\boldsymbol{x_i}, \boldsymbol{X}, \boldsymbol{C_{id}}, \boldsymbol{\varepsilon}, \boldsymbol{M})$
    c. $\boldsymbol{id} = \boldsymbol{id} + \boldsymbol{1}$ and $\boldsymbol{C_{id}} = \boldsymbol{\phi}$

  End for
- Return all the d-clusters: $\boldsymbol{C_1}, \dots, \boldsymbol{C_{id-1}}$

End.

Algorithm VII.1 : DBSCAN

Bien que cet algorithme soit facile à comprendre, son exécution informatique reste lente, surtout lorsque le nombre de points est important. Sa complexité est quadratique, de l'ordre de (n2)[173-174], mais peut être réduite à O (nlog (n)) en simplifiant la mise en œuvre de l'algorithme.

Un autre algorithme permet également le clustering de densité, c'est l'algorithme DENCLUE (DENsity-based CLUstEring)[175]. Bien qu'il acquière une grande complexité avec le nombre de paramètres d'entrée, il présente les avantages suivants :

- Très efficace face aux données aberrantes présentant du bruit ;
- Capable de décrire mathématiquement des clusters choisis arbitrairement appartenant à de grands ensembles de données ;
- Rapide par rapport au DBSCAN [176] et donc plus puissant.

Cet algorithme est basé sur l'estimation de la densité du noyau à travers différentes fonctions, c'est quasiment le même principe que l'algorithme DBSCAN, sauf qu'ici on a la possibilité d'illustrer la structure hiérarchique interne dans les données de distribution en ajustant la largeur de fenêtre σ de la fonction d'influence du noyau.

Notons l'ensemble de données de n objets dans l'espace. Nous décrivons l'algorithme DENCLUE par les notions suivantes :

Notons $D = \{x_1, \dots, x_n\}$ l'ensemble de données de n objets dans l'espace Ω. Nous décrivons l'algorithme DENCLUE par les notions suivantes :

- L'estimation du noyau par la fonction de densité globale :

$\forall\, x \in \Omega$, la fonction de densité de probabilité[177] est donnée par :

$$f^D(x) = \frac{1}{n}\sum_{i=1}^{n} K\left(\frac{x - x_i}{\sigma}\right) \quad \text{(Equation VII.2)}$$

Avec K(x)la fonction d'influence du noyau qui est une fonction de densité symétrique avec un pic à l'origine, cela peut être une fonction gaussienne, une fonction d'onde carrée...

σ : la largeur de la fenêtre de la fonction noyau.

- Attracteur de densité et attraction de densité, désignons x^* un point local maximum de la fonction de densité globale, pour un point $x \in \Omega$. s'il existe un ensemble de points $x_0, \ldots, x_k$, tel que $x_0 = x$ et $x_k = x^*$ et $x_i \ (0 < i < k)$ de sorte qu'il se trouve dans la direction du gradient de x_{i-1} , alors x est attiré en densité par x^* et x^* est un attracteur de densité de x.

Si la fonction du noyau $K(x)$ est continue et différentiable à chaque point, alors la méthode d'escalade basée sur le gradient peut être utilisée pour trouver la densité des attracteurs.

- Clustering centré. Avec x^* un attracteur de densité donné, s'il existe un sous-ensemble $C \subseteq D$ tel que x est attiré en densité par x^* et $f^D(x^*) \geq \xi$ avec ξ un seuil de bruit prédéfini, alors C est le cluster avec x^* son centre.
- Clustering avec une forme arbitraire : soit X un ensemble constitué de la densité d'Attracteurs. S'il existe un sous-ensemble $C \subseteq D$ qui vérifie :

$\forall x \in C$, il existe un attracteur de densité $x^* \in X$ tel que x est attiré en densité par x^* et $f^D(x^*) \geq \xi$;

$\forall x_i^*, x_j^* \in X \ (i \neq j)$, il existe un chemin $P \subset \Omega$ de x_i^* à x_j^* qui satisfait la condition suivante :

$y \in X, f^D(y) \geq \xi$. C est appelé le cluster de la forme arbitraire déterminée par X.

Nécessairement, deux paramètres doivent être fournis pour exécuter cet algorithme, qui sont σ : la largeur de fenêtre de la fonction noyau et ξ le seuil de bruit prédéfini, le choix de ces deux paramètres influence les attracteurs et le nombre de clusters trouvés.

Les étapes de base de l'algorithme DENCLUE[178] sont les suivantes :

- Déterminer les attracteurs de densité.
- Associez des objets de données à des attracteurs de densité à l'aide de l'escalade.
- Si possible, fusionnez les clusters initiaux en s'appuyant davantage sur une approche de clustering hiérarchique.

Cependant, bien qu'il prenne en compte des données incomplètes et des valeurs aberrantes[179], sa complexité peut être de l'ordre de $O(nlog(n))$, ce qui est acceptable.

Nous avons implémenté dans notre application les étapes définies dans l'algorithme DENCLUE ci-dessous , il reçoit en paramètres la base de données D et le seuil de bruit ξ

:

```
Algorithm DENCLUE(Dataset: D, Threshold: ξ )
Begin
Determine the density attractor of each data point in a dataset with gradient ascent rule;
Create clusters of data points that converge to the same density attractors;
Discard clusters whose density attractors have density less than  (noise and outliers);
Merge clusters whose density attractors are connected with a path of density at least ;
return clusters ;
End.
```

Algorithme VII.2 : DENCLUE[8]

[8] Dua, Dheeru and Graff, Casey"{UCI} Machine Learning Repository",*University of California, Irvine, School of Information and Computer Sciences*, 2019.

Cependant, avant d'utiliser des données brutes, on est passé par une étape de prétraitement pour éliminer celles qui contiennent des erreurs de type, celles ci pourraient présenter des valeurs aberrantes et retarder l'étape de classification et par la suite brouiller la visualisation des clusters.

VII.3 RÉSULTATS ET DISCUSSION

L'évaluation des résultats d'une méthode de clustering reste un problème ouvert. Mais ici, la principale difficulté réside dans le fait que l'évaluation des résultats du classement est de nature subjective. En conséquence, il existe plusieurs manières pertinentes de classer les objets de données.

En pratique, et pour vérifier la fiabilité de cette technique de classification basée sur la densité, une application a donc été faite en C-Sharp. Elle regroupe plusieurs fonctionnalités en commençant par le prétraitement, puis le traitement et la classification des données.

L'application permet à l'utilisateur de décider du choix des données initiales et du seuil de bruit.

Afin de tester et d'évaluer, nous utilisons une base de données du site web du serveur américain en libre accès nommé « the UC Irvine Machine Learning Repository » (UCI)[180].

Ce serveur héberge un référentiel d'apprentissage automatique qui est une collection de bases de données, de théories de domaine et de générateurs de données utilisés par la communauté d'apprentissage automatique pour l'analyse empirique des algorithmes.

VII.3.1 Tests sur la base de données IRIS

Dans un premier temps nous avons testé l'application sur une base de données dont nous connaissons les clusters à rechercher.

Dans ce cadre, nous utilisons la base de données iris ; elle contient 3 classes de 150 enregistrements sous les cinq attributs suivants : longueur et largeur des sépales et des pétales ainsi que l'espèce.

Le choix de cette base est l'exemple le plus connu dans le domaine de Machine Learning. Le système classe les fleurs d'iris en trois espèces (setosa, versicolor et virginica) en fonction des mesures de la longueur et de la largeur des sépales et des pétales. Nous appliquons d'abord une phase de prétraitement et de traitement aux données afin d'éliminer celles contenant des erreurs, puis nous les classons à l'aide de DBSCAN et DENCLUE.

La figure 1 ci-dessous représente toutes les données après nettoyage :

Détails Attributs

N°	Nom	Type	Eléments Déscritisés
1	sepal length	decimal	{5.1}{4.9}{4.7}{4.6}{5.0}{5.4}{4.4}{4.8}{4.3}{5.8}{5.7
2	sepal width	decimal	{3.5}{3.0}{3.2}{3.1}{3.6}{3.9}{3.4}{2.9}{3.7}{4.0}{4.4
3	petal length	decimal	{1.4}{1.3}{1.5}{1.7}{1.6}{1.1}{1.2}{1.0}{1.9}{4.7}{4.5
4	petal width	decimal	{0.2}{0.4}{0.3}{0.1}{0.5}{0.6}{1.4}{1.5}{1.3}{1.6}{1.0
5	class	string	{Iris-setosa }{Iris-versicolor }{Iris-virginica }

Détails Données

Nombre d'attributs : 5
nombre d'instances accéptées :150 nbre en % : 100 %
nombre d'instances ignorées :0
==========Liste d'attributs et types====
sepal length : decimal
sepal width : decimal
petal length : decimal
petal width : decimal
class : string

Figure VII.1. Visualisation des données après nettoyage.

Le temps de traitement et de classification est très court compte tenu du nombre et de la taille de cette base de données, il est de l'ordre de 150 millisecondes.

Pour avoir une vision claire de la nature des données, la Figure ci-dessous détermine le nombre d'occurrences pour chaque attribut :

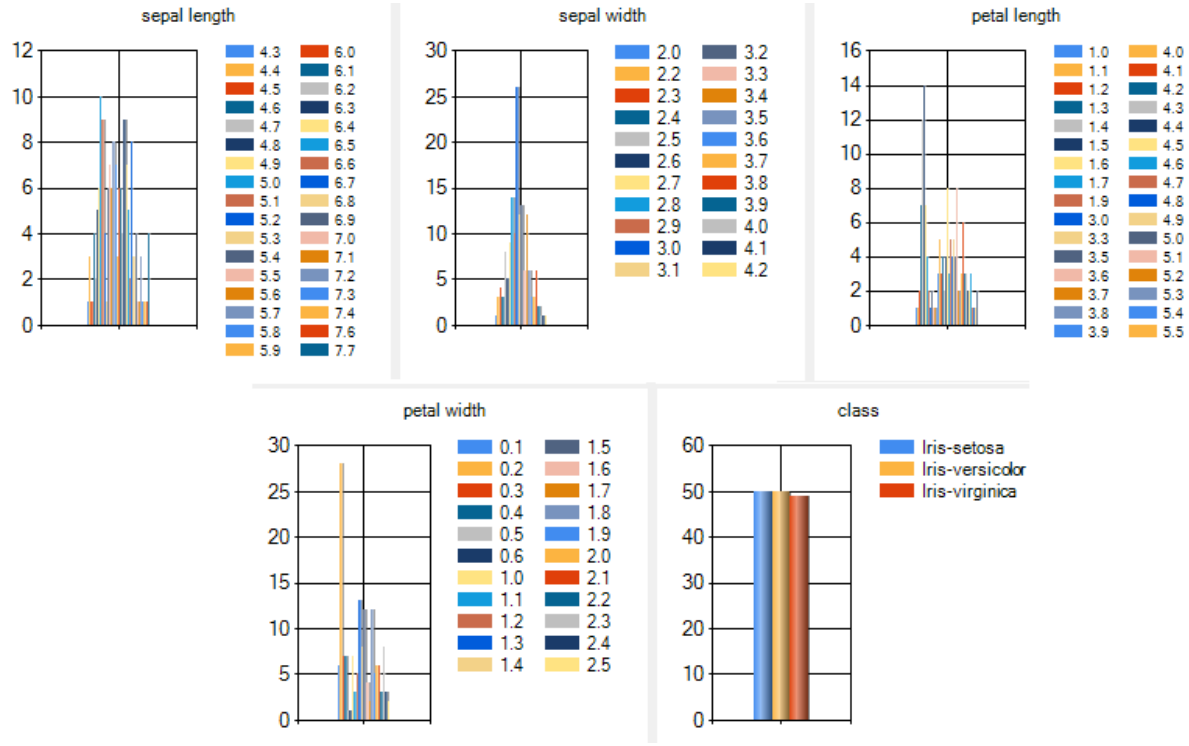


Figure VII.2. visualisation de l'occurrence des données après traitement.

Enfin, pour les deux algorithmes, le choix des attributs initiaux pour la classification est déterminant pour l'identification des clusters. Dans la figure VII.3 ci-dessous, le croisement de la longueur des pétales avec la largeur des sépales donne de bons résultats.

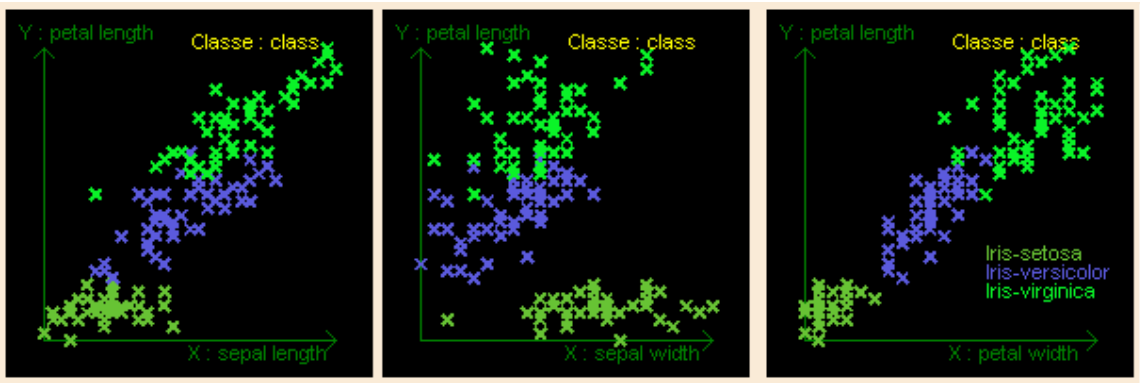


Figure VII.3. visualisation des clusters des Iris.

Les trois clusters sont visibles par des regroupements de points de couleurs différentes.

VII.3.2 Tests sur la base de données Wisconsin (diagnostic du cancer du sein)

Un autre test est appliqué à la base de données sur les maladies du cancer du sein de l'hôpital du Wisconsin.

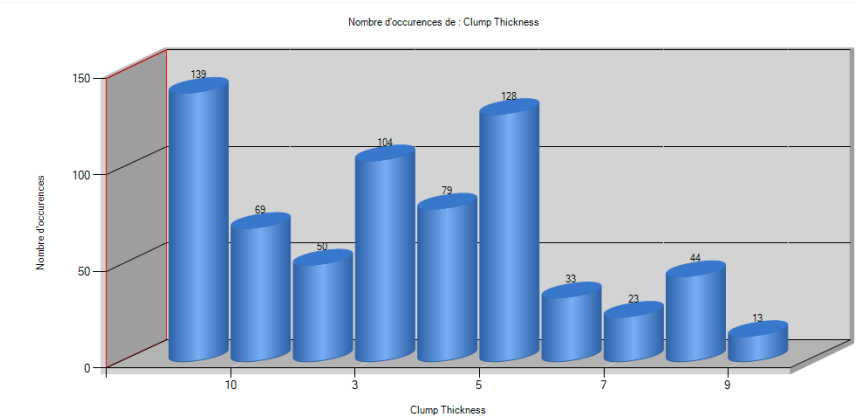
Comme le montre la table ci-dessous, cette base de données contient 683 lignes et 11 colonnes.

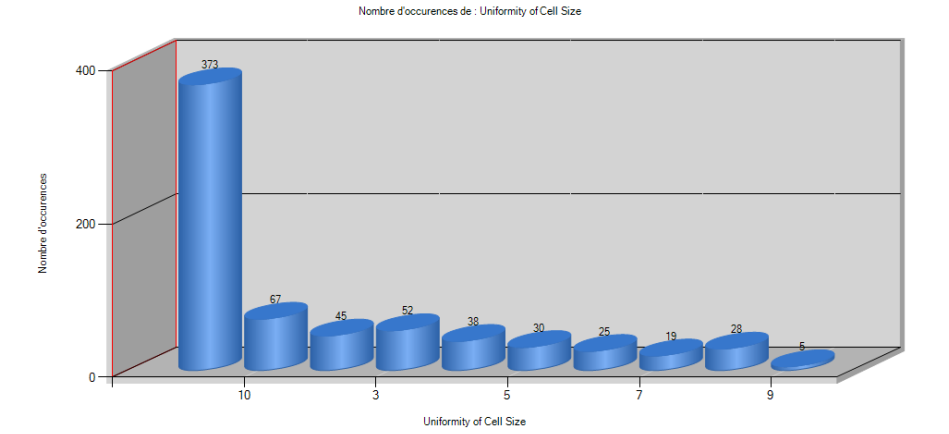
Tableau VII.1. Structure de la base de données.

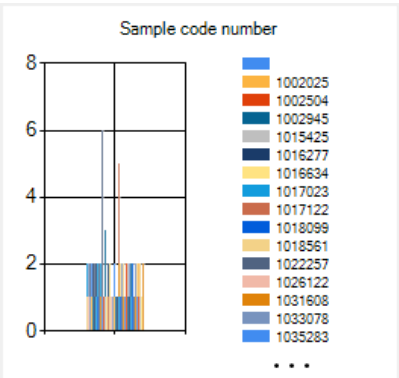
Attribute	Domain
Sample code number	id number
Clump Thickness	1-10
Uniformity of Cell Size	1-10
Uniformity of Cell Shape	1-10
Marginal Adhesion	1-10
Single Epithelial Cell Size	1-10
Bare Nuclei	1-10
Bland Chromatin	1-10
Normal Nucleoli	1-10
Mitoses	1-10
Class	(2 for benign, 4 for malignant)

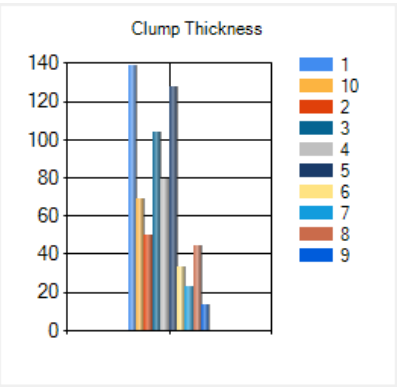
L'intérêt de ce choix est d'identifier les paramètres déterminant le type de cancer du sein afin de mieux cibler la posologie adoptée pour les patients.

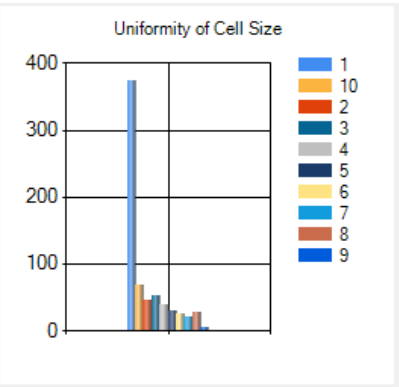
Après nettoyage des données, nous avons présenté leurs occurrences pour l'ensemble d'attributs dans la figure ci-dessous :

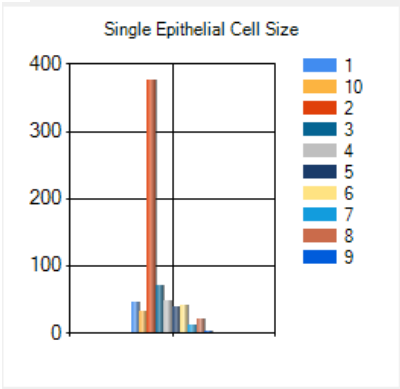

Nombre d'occurences de : Clump Thickness
Nombre d'occurences
150
100
50
0
139
69
50
104
79
128
33
23
44
13
10
3
5
7
9
Clump Thickness

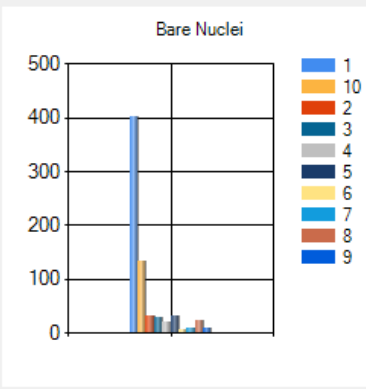

Nombre d'occurences de : Uniformity of Cell Size
Nombre d'occurences
400
200
0
373
67
45
52
38
30
25
19
28
5
10
3
5
7
9
Uniformity of Cell Size

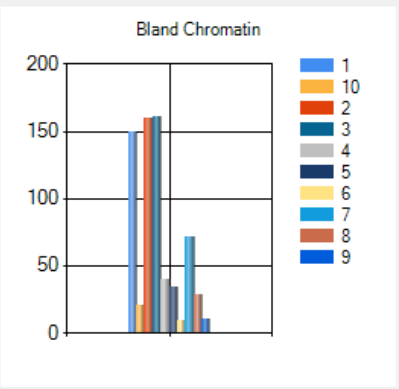

Sample code number
8
6
4
2
0
1002025
1002504
1002945
1015425
1016277
1016634
1017023
1017122
1018099
1018561
1022257
1026122
1031608
1033078
1035283


Clump Thickness
140
120
100
80
60
40
20
0
1
10
2
3
4
5
6
7
8
9


Uniformity of Cell Size
400
300
200
100
0
1
10
2
3
4
5
6
7
8
9


Single Epithelial Cell Size
400
300
200
100
0
1
10
2
3
4
5
6
7
8
9


Bare Nuclei
500
400
300
200
100
0
1
10
2
3
4
5
6
7
8
9


Bland Chromatin
200
150
100
50
0
1
10
2
3
4
5
6
7
8
9

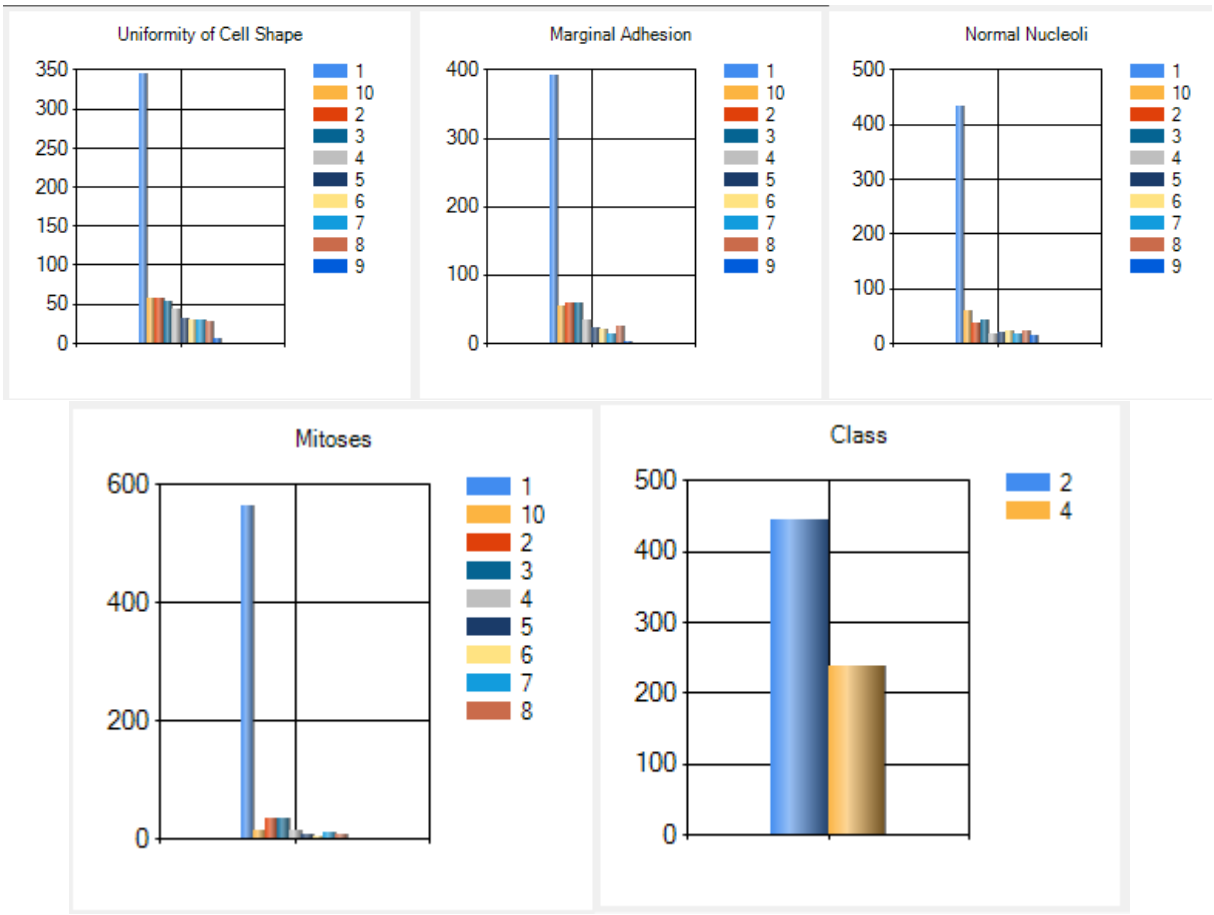


Figure VII.4. Visualisation de l'occurrence des données après traitement

L'exécution des deux processus de classification est assez rapide, des clusters sont visibles pour certains graphes, ce qui permet d'identifier les attributs pertinents, les formes des clusters observés ne sont pas régulières, il y a aussi quelques valeurs aberrantes, mais les objets avec cancer malin sont les plus commun et plus dominant.

Juste en dessous, une visualisation globale de tous les paramètres du patient est donnée par notre application de classification. Une représentation agrandie de deux graphiques est observée, les patients souffrant du type concert malin sont visualisés par des points rouges et ceux de type bénin par des points blancs comme le montre la figure suivante :

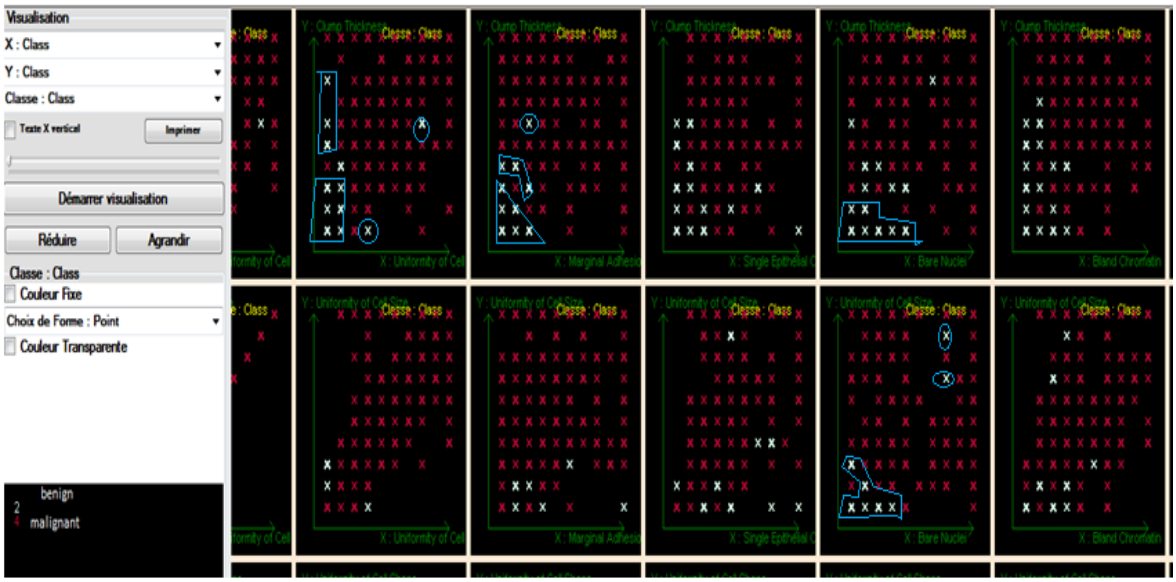


Figure VII.5. Visualisation des clusters de cancer (bénigne, maligne)

D'après la figure ci-dessus, la forme des clusters peut être utile pour la validation des paramètres initiaux à utiliser dans le processus de classification, on retrouve comme attributs significatifs : l'épaisseur des touffes(Clump Thickness), l'uniformité de la forme et de la taille des cellules, les mitoses…

Cependant, ces éléments ne peuvent pas être analysés à l'oeil humain, compte tenu de la grande taille et de la complexité des informations qu'ils contiennent, d'où la nécessité d'enrichir la base de données pour mieux visualiser les cas isolés.

VII.3.3 Tests sur l'ensemble de données marketing bancaire

Cette base de données contient environ 45 000 lignes et 17 colonnes, il est donc important de choisir les données et le nombre d'attributs à saisir dans le système.

Comme attributs de cette base de données nous citons : l'âge, l'emploi, l'état civil… et la cible souhaitée (le client a-t-il souscrit un dépôt à terme ? « oui », « non »).

Compte tenu de la taille et du nombre de données de cette base de données, nous n'avons présenté que les trois graphiques suivants (Figure VII.6) :

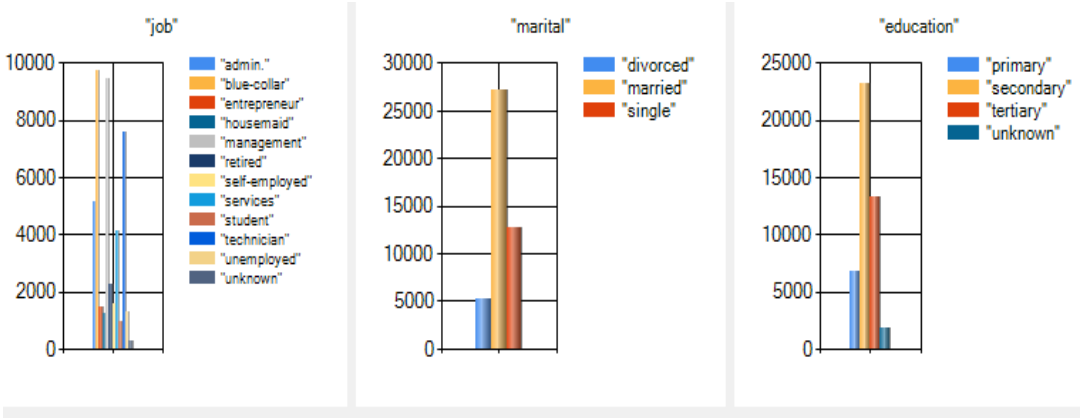


Figure VII.6. visualisation des occurrences (type de travail, état martial, niveau d'étude) des individus.

On peut voir certains clusters sous différentes formes et dans certains cas, ils sont trop proches les uns des autres, d'où la difficulté de les entourer :

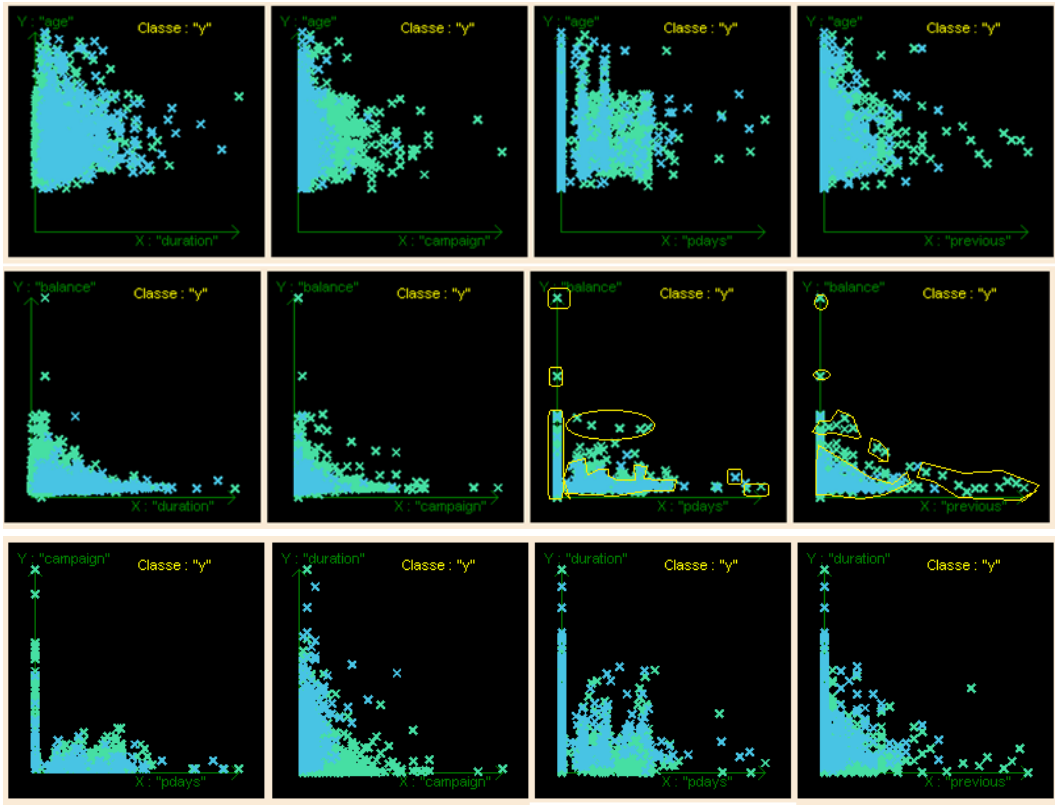


Figure VII.7. visualisation de l'ensemble de données marketing bancaire

Après avoir fixé les paramètres initiaux pour mieux calculer les attracteurs de densité, nous construisons des points de même densité qui convergent. Le système en cours d'exécution produit un regroupement de points dans des formes non régulières. Les clusters peuvent être vus à l'œil nu, mais il existe également des valeurs aberrantes qui obscurcissent leur visibilité. De plus, le choix du seuil de bruit et de la taille des données ralentit le traitement et produit des exceptions liées aux performances limites du matériel informatique.

VII.3.4 Discussion et évaluation :

L'évaluation de la méthode[181] peut se faire sur la base des clusters trouvés et de la qualité des connaissances[182-184] extraites pour une bonne prise de décision. Cependant le choix de la densité des attracteurs et du seuil de bruit[185] sont des facteurs déterminants pour une bonne mesure d'évaluation. L'escalade lors de la recherche d'attracteurs de densité et la taille des données à traiter interviennent dans l'évaluation de la méthode. Dans l'exemple de la base de données (Bank Marketing Data Set), le temps d'attente a été augmenté de 10 minutes jusqu'à ce que le système plante, en fonction du nombre d'attributs et de lignes initialement choisis.

Tableau VII.2. Comparative analysis of the two density based algorithms

Algorithm	Complexity	Cluster format	Parameters	Noise resistant	Cluster quality	RunTime
DBSCAN	O(n2)	Arbitrary	No input parameter	Well	10%	200 ms, infinitely diverges for large data
DENC-LUE	O (nlog(n))	Arbit rary	Depending on the size of database	Better	20%	100 ms better than DBSCAN and increases for data> 10,000

La table 2 montre que l'algorithme DENCLUE est meilleur pour les grandes bases de données, il permet de choisir des paramètres d'entrée pour réduire le temps d'exécution. De plus, lors du choix des paramètres initiaux pour accélérer le processus, il est possible d'omettre l'identification de certains des valeurs aberrantes d'intérêt dans la prise de décision.

Pour évaluer la qualité du clustering, nous considérons à quel point les clusters sont compacts et à quel point ils sont séparés, cela dépend de ce que nous voulons comme résultat du clustering. La mesure de la largeur moyenne de la silhouette [186] des clusters nous a permis de présenter la qualité des clusters dans le tableau ci-dessus.

Une autre façon d'évaluer est de faire appel à un expert [187] pour comprendre le sens du regroupement dans un domaine particulier. Cependant, s'il est possible pour un expert de dire si un regroupement donné a du sens, il est beaucoup plus difficile de quantifier son intérêt, ou de dire si tel résultat est meilleur qu'un autre. De plus, l'applicabilité de la méthode ne peut pas être étendue à d'autres types de données.

VII.4 CONCLUSION

Comme nous l'avons vu dans un premier temps, les notions d'information, de données et de connaissances sont indissociables. Il existe de nombreuses techniques d'exploitation et de manipulation, notamment la construction de modèles mathématiques de l'information et la théorie des probabilités. D'où l'apparition de plusieurs processus de classification et de clustering. Nous avons mis en évidence la classification en utilisant la notion de voisinage pour le calcul de la densité de points voisins. Ensuite, nous nous sommes concentrés sur l'étude des algorithmes DBSCAN [188-189] et DENCLUE ; nous avons également exposé la problématique globale de ce type de Clustering, de la préparation des données au choix des paramètres. Ensuite, nous avons implémenté l'algorithme DENCLUE dans une application C-Sharp. Nous avons fait des tests sur trois bases de données pour vérifier la validité de

l'algorithme, les résultats peuvent être acceptés et utilisés sous certaines conditions. Ceci était motivé par la réduction, autant que possible, du temps d'exécution des calculs et une meilleure exploitation des données.

Nous avions donc réussi la réalisation d'une application regroupant les différents processus d'extraction de connaissances à partir de données, passant du prétraitement à la recherche de données incohérentes et incomplètes, vers le traitement et la prise de décision sur le choix des valeurs initiales pour l'apprentissage des processus, après avoir enregistré les données préparées sans affecter les données initiales. Parmi les perspectives à court terme, il convient d'envisager de développer l'algorithme DENCLUE dans une nouvelle version permettant ainsi une meilleure classification des big data et un temps d'exécution réduit, sans oublier les outliers qui peuvent être déterminants dans certaines situations.

Chapitre VIII. L'approche proposée : Mise en œuvre d'une application informatique «Système ECD d'extraction de connaissances à partir de données et expérimentations »

VIII.1 Introduction

Dans les chapitres précédents nous avons présenté différentes approches de traitement de données basé sur la notion de similarité et de distance entre objets et attributs.

Nous nous sommes confrontés à divers problèmes liés aux valeurs de données incomplètes dans la fouille de données et de l'apprentissage automatique dans la base d'apprentissage.

Nous avons rencontré des objets ayant des valeurs manquantes et ou imprécises pour certains attributs. Cela est survenu pendant la phase d'acquisition des données du processus de l'ECD. Le manque de données est dû à leur non enregistrement, ou à leur acquisition jugée onéreuse.

La collecte de données nécessaire à la prise de décision est une opération difficile, en particulier dans le domaine de la santé, où la fiabilité est souvent contestée.

Dans ce chapitre, Nous présentons une application, réalisée avec le langage C-Sharp, dans une plateforme Windows. C'est un système d'extraction de connaissance à partir de données.

Dans un premier temps Nous appliquons les différentes phases liées au prétraitement vu précédemment sur des données textes brutes, avant de passer au traitement et à la visualisation de clusters ….

Nous utilisons pour le teste et l'évaluation, des bases de données issues des sites d'un serveur américain d'accès libre nommé « the UC Irvine Machine Learning Repository » (UCI) (https://archive.ics.uci.edu/ml/index.php et www.openml.org).

Ce serveur heberge l'« UCI », un référentiel d'apprentissage machine qui est un ensemble de bases de données, de théories de domaines et de générateurs de données utilisés par la communauté d'apprentissage automatique pour l'analyse empirique des algorithmes. L'archive a été créée en tant que fichiers ftp en 1987 par David Aha [147] et d'autres étudiants diplômés de UC Irvine. Depuis lors, il a été largement utilisé par les étudiants, les éducateurs et les chercheurs du monde entier comme principale source d'ensembles de données d'apprentissage automatique. Pour indiquer l'importance de l'archive, celui-ci est cité dans plus de 1000 articles publié à l'échelle internationale.

VIII.2 Choix des bases de données :

S'il y a un champ où la difficulté d'imprécision et d'ambiguïté de données est une caractéristique essentielle, c'est bien le domaine médical. La raison en est que la manière de raisonnement du médecin dans sa démarche repose, sur son savoir, et sur l'expérience liée à la résolution de cas rencontrés dans la pratique.

L'autre environnement d'incertitude est le domaine économique là où on a besoin de traiter les demandes de crédits liés aux êtres humains.

Etant donné que le facteur humain est incontrôlable et incertain, et devant le nombre important d'informations diverses et incomplètes des demandeurs de crédits, le responsable de la banque aurait beaucoup de risques à en valider une demande.

De ce fait, nous sommes en présence de trois bases de données suivantes IRIS **(**base de référence**)**, **DIABETE** et **CREDIT DE BANQUE**.

Iris est la plus connue et la plus utilisée pour les tests et la validation de la classification.

VIII.3 Méthodes mises en œuvre :

L'application utilise des méthodes de filtrage et d'analyse de données selon le type de données, leurs disponibilités, la similarité et la dissimilarité d'objets et de leurs attributs.

En premier lieu, nous procédons à l'affichage de la page d'accueil ci-dessous :

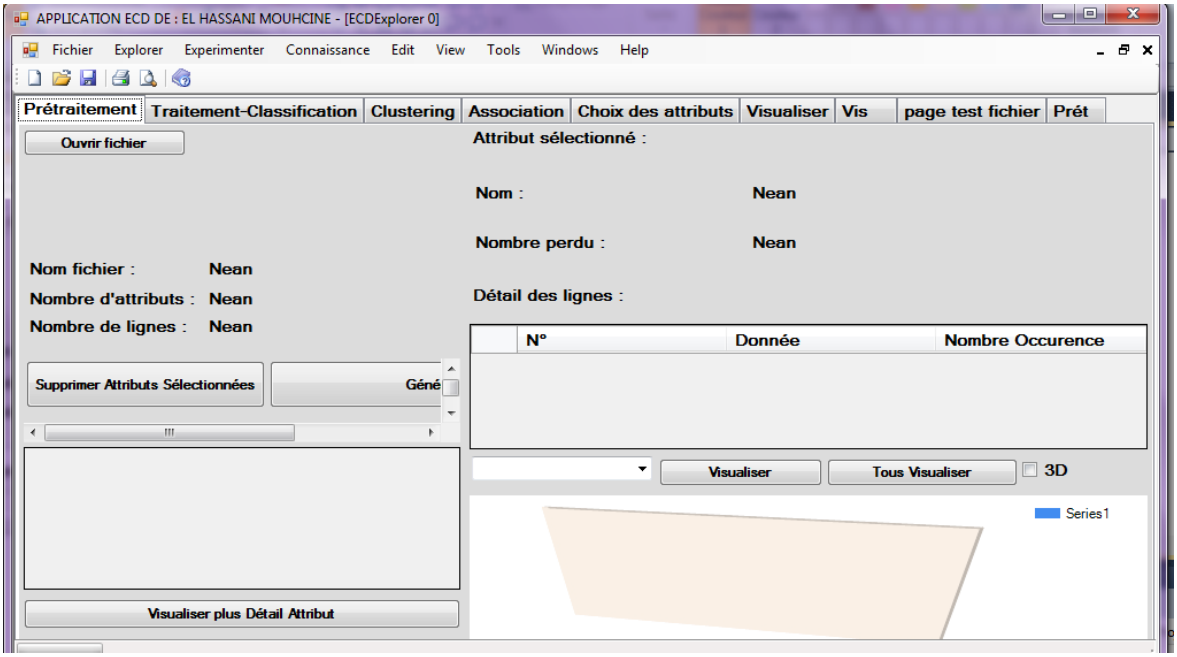


Figure. VIII.1 : Menu général de l'Application (ECD)

Cette fenêtre présente plusieurs fonctionnalités allant du chargement de la base de données, au prétraitement, au traitement et à la classification de clusters et visualisation.

VIII.3.1 Chargement et prétraitement de données

Avant de commencer le traitement, notons que les données utilisées se trouvent dans un fichier texte ou dans une base de données, elles sont tout d'abord récupérées dans un texte formaté par un délimiteur « ; ». La première ligne du texte regroupe les noms de colonnes séparées par le délimiteur. Après les valeurs d'attributs sont stockées chacune dans une ligne.

Le choix du fichier texte se fait dans le but de réaliser un stockage meilleur et le transport de grandes quantités de données.

Le chargement de données se fait dans un objet Data-grid comme l'indique la figue suivante :

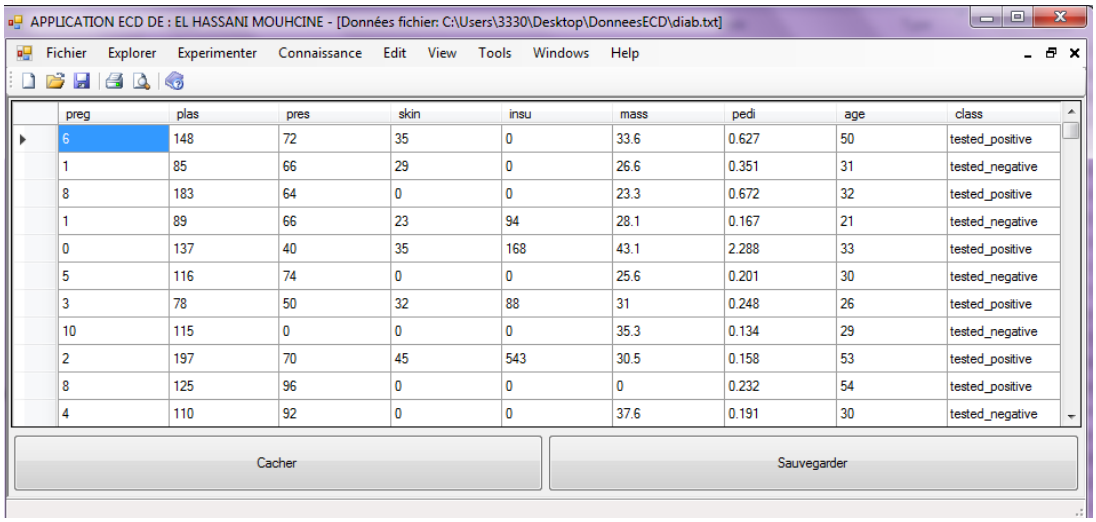


Figure VIII.2 : Chargement de données test dans Data-grid

Au cours du chargement, nous appliquons un premier filtrage et nettoyage des données manquantes et incohérentes. Le résultat se présente sous forme de coloration de cellules ne possédant pas d'informations suivi d'un message d'information pouvant être exploité pour la correction.

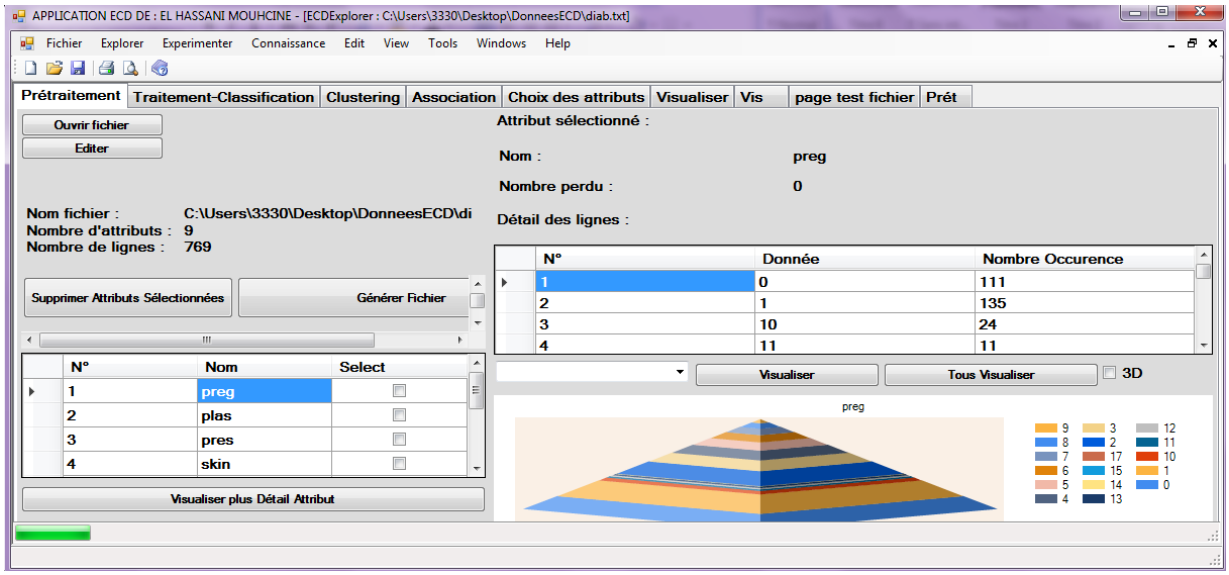
A la fin du traitement, les données utiles et erronées vont être enregistrées séparément pour l'exploitation.

VIII.3.2 Visualisation graphique de l'étape de prétraitement

Dans cette phase, on affiche les données récupérées en faisant une analyse statistique du nombre d'attributs à utiliser et des classes correspondantes ainsi que celle de la similarité des données d'attributs.

Une telle analyse aboutit à un affichage initial de données (nombre d'attributs) ; les occurrences et même une visualisation initiale sous forme pyramidale ou graphique selon la figure ci-dessous :

« Voir aussi l'Annexe 1 et l'Annexe 12 pour les bases de données 'IRIS et Crédit de banque' »



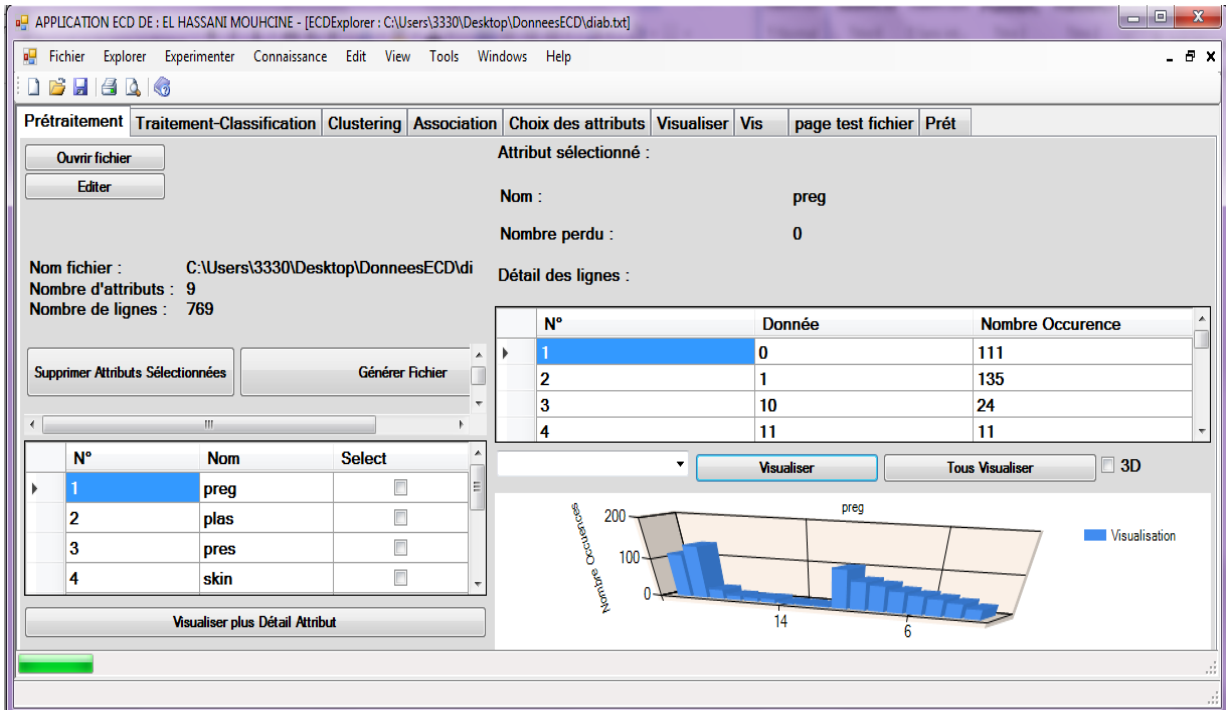


Figure VIII.3 : Chargement de données test pour le prétraitement.

L'application offre aussi la possibilité de voir en détail le comportement d'informations de chaque attribut de classe ou d'avoir une visualisation globale de toutes les données de la classe de prise de décision (voir figure ci-dessous) :

« Voir aussi l'Annexe 2 et l'Annexe 13 pour les bases de données 'IRIS et Crédit de banque' »

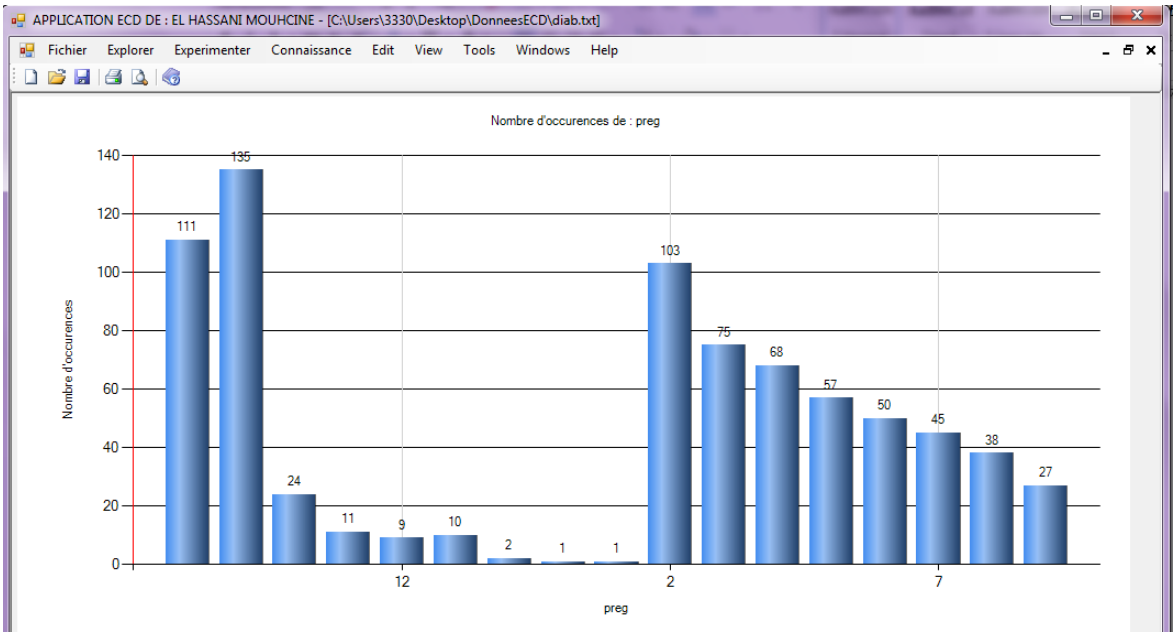


Figure VIII.4 : Exemple de représentation graphique d'occurrences du paramètre « preg » propres aux patients diabétiques

Dans ce qui suit, nous présentons les données de la base diabète de façon condensée pour pouvoir détecter la fréquence de répétition de données selon les attributs de classes.

« Voir aussi l'Annexe 5 pour la base de données 'IRIS' et l'Annexe 13 propre à la base 'Crédit de banque '»

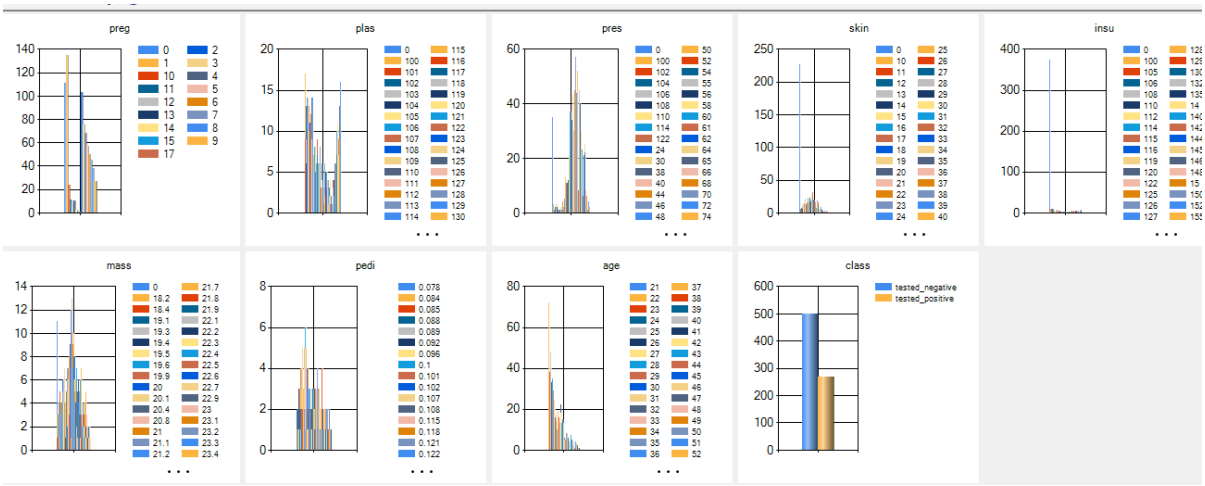


Figure VIII.5 : Exemple de représentation graphique globale d'occurrences de tous les attributs propres aux patients diabétiques

VIII.3.3 Traitement et classification

Au cours de cette étape, les données vont être récupérées et chargées pour l'exploitation par l'expert .La fenêtre de la figure ci-dessous montre les données et leurs natures .Durant cette phase, soit nous laissons le système décider à la place de l'être humain sur la nature et le type de données, soit nous modifions manuellement ce choix automatique :

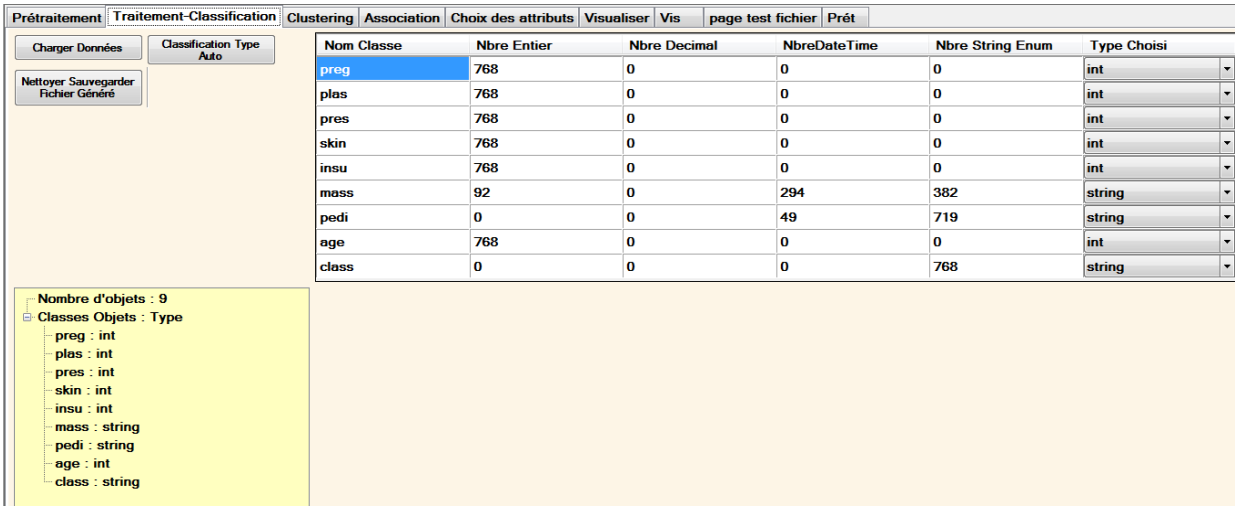


Figure VIII.6 : Phase de préparation et traitement des données pour l'exploitation

Dans cette fenêtre, nous prévoyons un nettoyage de données non précises es à l'aide d'un bouton, créé à cet effet, et nous passons à l'affichage et à la sauvegarde .Voir la *figure ci-dessous :*

« Voir aussi l'Annexe 6 pour la base de données 'IRIS' et l'Annexe 14 propre à la base 'Crédit de banque '»

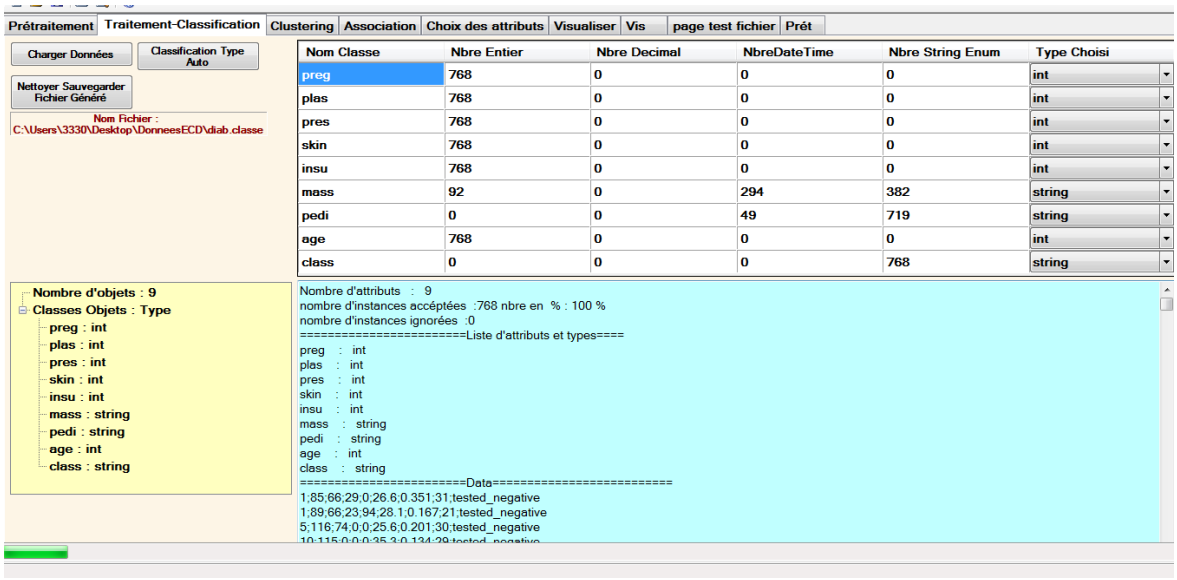
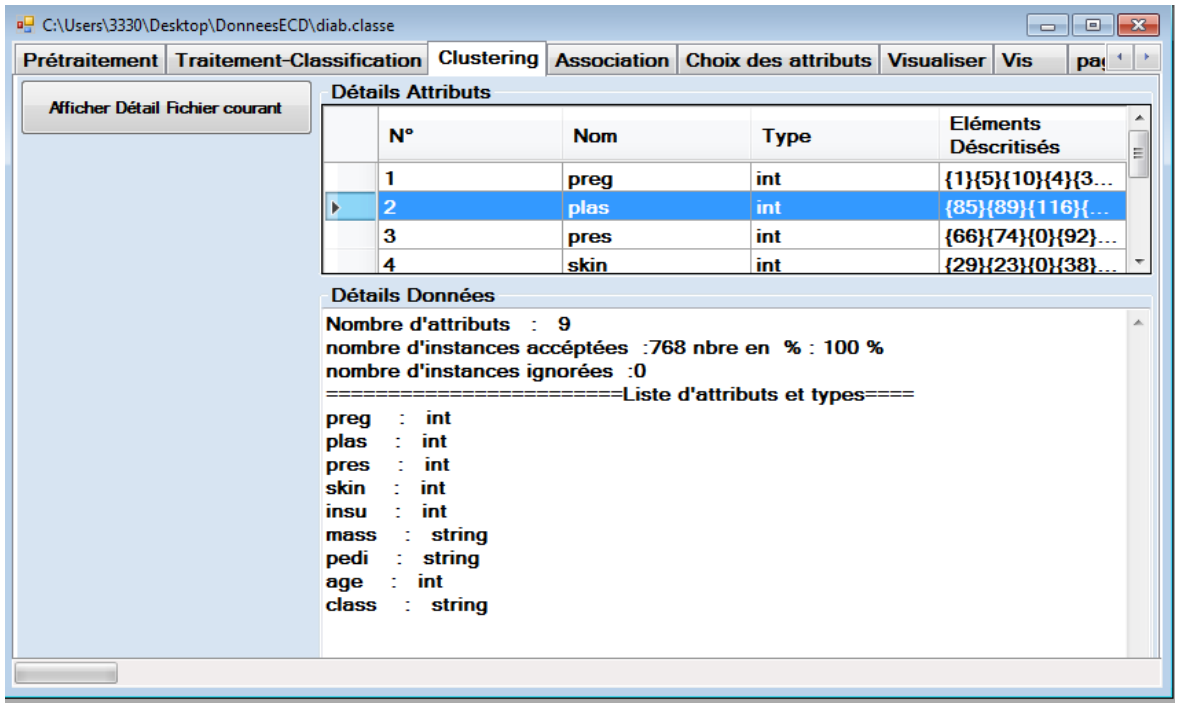


Figure VIII.7 : Phase d'affichage et sauvegarde de données pour l'exploitation

VIII.3.4 Clustering et visualisation

Dans cette étape, on cherche à organiser les données dans des clusters en passant par la discrétisation de données et la recherche de motifs fréquents si possibles tout en précisant le nombre d'instances acceptées et celles ignorées, ainsi que les éléments discrétisés et le détail de données et types .

« Voir aussi l'Annexe 14 propre au Crédit de banque' »



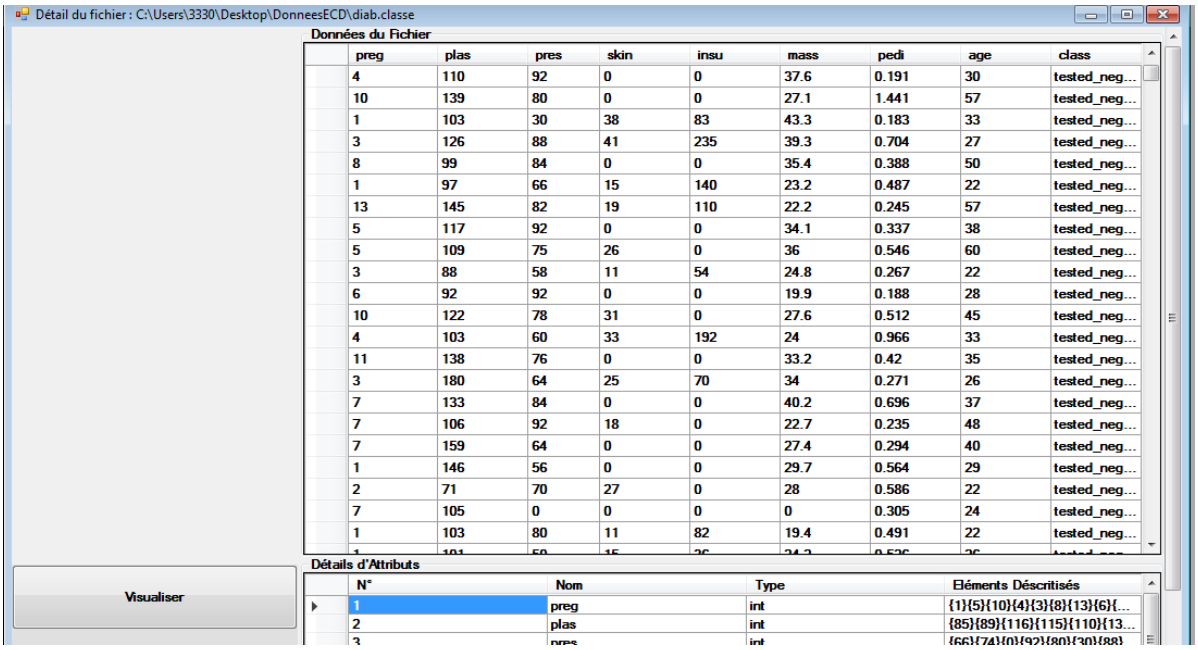


Figure VIII.8 : Phase de classification et Clustering données pour la visualisation

VIII.3.5 Exploitation du résultat pour l'extraction de connaissances

Dans cette dernière étape, l'application nous permet de mieux visualiser les données en utilisant les techniques de coloration et de représentation par des motifs points, carrés, cercles ou de façon aléatoire.

De même on peut effectuer des zooms pour mieux voir les données. Ainsi, on a prévu des objets listes se chargeant automatiquement pour croiser les données selon nos désirs et de les visualiser de différentes manières …la figure ci-dessous donne une visualisation de données de la base diabète manipulée au paravent :

« Voir aussi l'Annexe 9 pour la base de données 'IRIS' et l'Annexe 15 propre à la base 'Crédit de banque '»

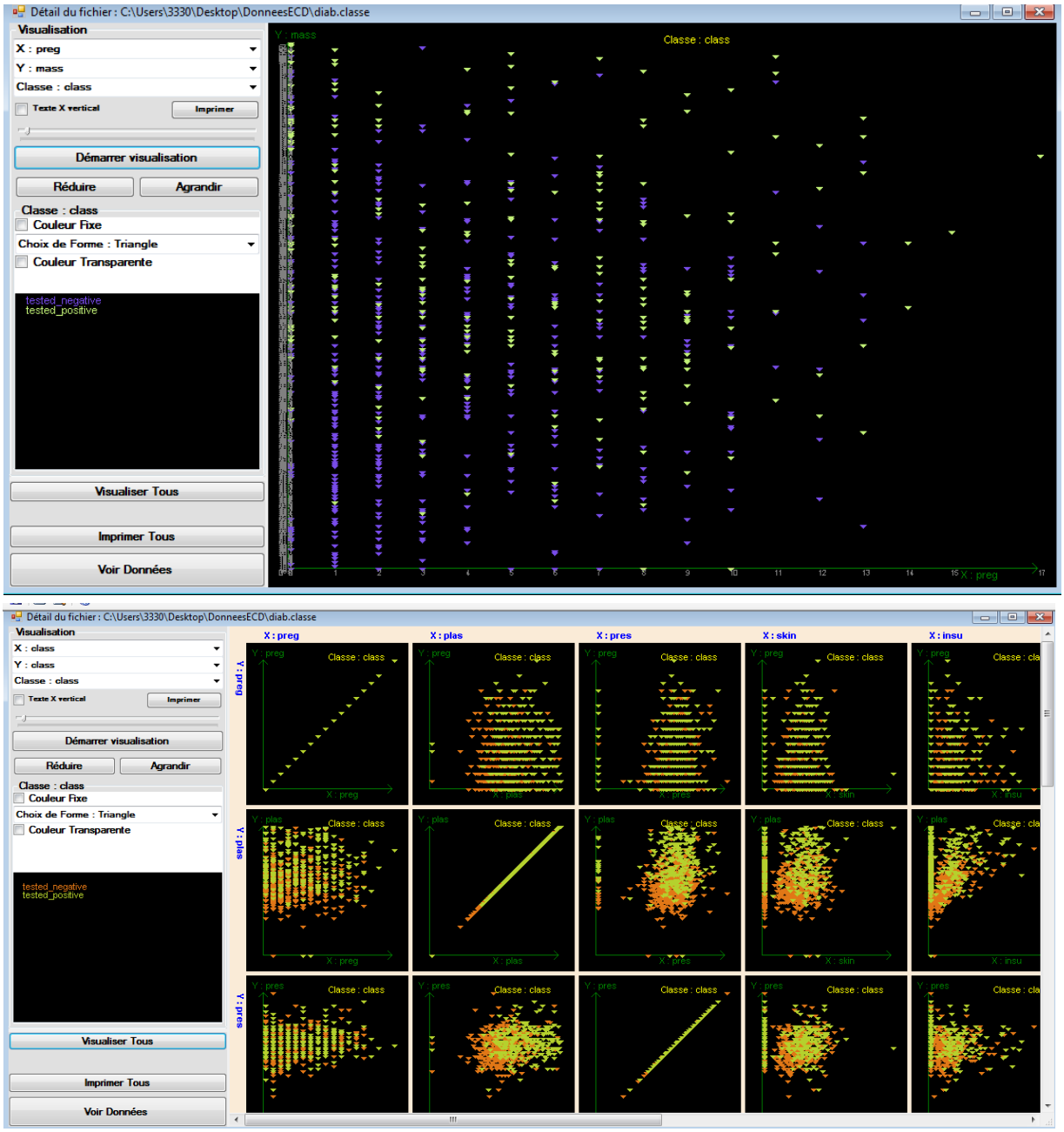


Figure VIII.9 : Phase de visualisation de clusters après croisement de données.

On peut à première vue identifier les clusters puisque la visualisation se base sur la couleur et la géométrie des points .Cela peut être très utile pour des masses de données élevées. Précisons aussi que ce système n'exige pas d'intervention humaine lors du traitement et, de ce fait, ne peut être sollicité que lorsque ce qu'il aura accomplis la tâche qui lui est confiée.

« Voir aussi l'Annexe 10 pour la base de données IRIS et l'Annexe 16 propre à la base 'Crédit de banque ' »

Aussi , il peut être exécuté rien que sur une plateforme Windows et les résultats issues peuvent être récupérés sous forme d'images de grande résolution, ce qui permet alors aux experts de procéder à une analyse visuelle rapide , telle que le permet l'affichage total des résultats ,qui est réalisé pour une meilleurs comparaison et une bonne prise de décision. « Annexe 17 »

VIII.4 Conclusion

Le prototype que nous avons présenté a été développé en C-Sharp, et est de ce fait "portable "sous
diverses plates-formes Windows à condition d'y introduire le Framework nécessaires.

Comme toutes applications, il reste à passer par les différentes phases de test et de correction pour permettre une utilisation aisée et efficace. De même il faut penser à implémenter d'autres fonctionnalités sans oublier les contraintes liées aux performances de la machine et au choix initial des paramètres à injecter, et pouvant rendre efficace et rapide la collecte de résultats fiables.

Conclusion générale

Dans ce mémoire, nous avons défini au début les notions d'informations de données et de connaissances, par la suite nous avons mis le point sur l'importance des données et de connaissances certaines dans le commerce, la sécurité et la prise de décision.

De nombreuses approches ont étés utilisées pour gérer la connaissance et l'indexer. Nous avons de même remarqué une prolifération de méthodes permettant d'extraire la connaissance et nous nous sommes trouvés face à un problème d'embarrât de choix et pour opter pour telle ou telle procédure nous avons procédé à des calculs optimaux afin de tomber initialement sur les paramètres adéquats pour le traitement. Cela a été motivé par la réduction, le mieux possible, du temps d'exécution des calculs et une meilleure exploitation des données.

Après, nous nous somme focalisés sur l'étude des relations entre données et sur la programmation logique inductive, et à cet égard, l'utilisation des graphes étaient une solution acceptable. Ensuite, nous nous sommes intéressés, à l'analyse des algorithmes de Clustering et à l'étude comparative des contraintes liées .En effet, il est souvent plus aisé pour un décideur de donner des exemples de valeurs déjà affectés aux catégories que de produire directement et explicitement les paramètres fondamentaux, dont il ne peut connaitre la véritable signification. Cela, impose donc, la mise en œuvre de techniques d'évaluation et de mesure de la qualité d'une méthode à l'aide des indices d'évaluation se basant sur la mesure de similarité, de distance etc…

Par ailleurs ,avant de finir par la réalisation d'une application informatique utilisant différents algorithmes de Clustering, on a fait l'étude d'un cas pratique d'extraction de connaissance à partir du texte, une telle technique est très exploitée de nos jours dans la recherche d'information à travers les réseaux sociaux et par les robots de moteurs de recherche comme Google pour permettre l'indexation et le classement et par la suite l'accès rapide et efficace à la connaissance .

Nous nous étions, donc, parvenu à la réalisation d'une application regroupant les différents processus d'extraction de connaissances à partir de données, passant par le prétraitement à la recherche de données incohérentes et incomplètes, vers le traitement et la prise de décision sur le choix des valeurs initiales pour le processus d'apprentissage, après la sauvegarde des données préparées sans affectées celles initiales.

Finalement, la classification basée sur les notions de similarités, de distances et des algorithmes a été effectuée pour aboutir à une visualisation claire et significative des clusters. Cela a permis, en outre, une vision générale ou détaillée avec des options de couleurs et de formes pouvant simplifier visualisation de données croisées et même un stockage dans des fichiers photos de grande résolution , ce qui constitue un point de plus pour l'extraction de la connaissance .

Parmi les perspectives à court terme, il convient de projeter à développer l'application pour permettre à l'utilisateur de se déplacer dans un espace de données et de se mettre là où il le faut tout en étant soumis aux contraintes imposées par le système de prise de décision.

Bibliographies :

[1] Leonhard Euler (1736), "Solution problematis ad geometriam situs pertinentis", Commentarii Academiae Scientiarum Imperialis Petropolitanae Vol 8, pages 128–140.

[2] Jaccard P. (1901), "Distribution de la flore alpine dans le bassin de Dranses et dans quelques régions voisines", Bulletin de la Société Vaudoise des Sciences naturelles, pages 37, 241-272.

[3] Dénes König (1936), "Theorie der Endlichen und Unendlichen Graphen", Teubner, Leipzig, page 45.

[4] Claude E. Shannon and Warren Weaver, July, October 1948, "A Mathematical Theory of Communication", Reprinted with corrections from The Bell System Technical Journal, Vol. 27, pages. 379–423, 623–656.

[5] Cover TM & Hart PE (1967), "nearest neighbor pattern classification". IEEE Trans. Inform. Theory IT-13:21-7, 1967. Dept. Electrical Engineering, Stanford Univ., Stanford, and Stanford Res. Inst., Menlo Park, CA, page 20

[6] Donald. Schön (1983), "Reflection-in-action and reflection-on-action", page 60

[7] Ryszard S.Michalski (1983) , "A theory and methodology of inductive learning", 1983 Elsevier pages 111-161

[8] E. B. Fowlkes and C. L. Mallows (1983), "A Method for Comparing Two Hierarchical Clustering", Journal of the American Statistical Association, Vol. 78, No. 383 (Sep., 1983), pages 553-569.

[9] Leo Breiman, Jerome Friedman, Charles J. Stone, R.A. Olshen (1984), "Classification and Regression Trees (Wadsworth Statistics/Probability) ", J.R. QUINLAN, Induction of Decision Trees, 1986 Kluwer Academic Publishers, Boston, Machine Learning 1, ISBN-13: 978-0412048418, pages 81-106

[10] Douglas Fisher(1987), "Machine Learning & Knowledge Acquisition", Improving Inference Through Conceptual Clustering, Department of Information and Computer Science University of California Irvine, California 92717,AAAI-87 Proceedings. Copyright ©1987, pages 461-465

[11] John H. Gennari, Pat Langley and Doug Fisher (1989), "Models of Incremental Concept Formation", Irvine Computational Intelligence Project, Department of Information and Computer Science, University of California, Irvine, CA 92717, U.S.A, pages 30-31.

[12] John H. Gennari, Pat Langley and Doug Fisher (1989), "Models of Incremental Concept Formation", Irvine Computational Intelligence Project, Department of Information and Computer Science, University of California, Irvine, CA 92717, U.S.A, pages 30-31.

[13] R BERTRAND, E. D1DAY (1990), "Une généralisation des arbres hiérarchiques : les représentations pyramidales", Revue de statistique appliquée, tome 38, n° 3 (1990), pages 53-78.

[14] Lehnert, W. Cardie, C. Fisher D. Riloff, E, & Williams,R. (1991), "Description of the CIRCUS ,System as Used for MUC-3", a. University of Massachusetts: Third Message Understanding Conference (MUC-3). San Diego, CA, Morgan Kaufrnann, pages. 223-233.

[15] Rakesh Agrawal et Ramakrishnan Srikant (1994), "Fast Algorithms for Mining Association Rules", IBM Almaden Research Center,650 Harry Road, San Jose, CA 95120, pages 488-492

[16] Nada Lavrac, Sašo Džeroski (April 1996), "Review of “Inductive Logic Programming: Techniques and Applications“Machine Learning", Volume 23, pages 103-108.

[17] Nada Matta, Jean Louis Ermine, Gérard Aubertin, Jean-Yves Trivin (1996), "How to capitalize knowledge with the MASK method" , pages 8-13

[18] Usama M. Fayyad (1996), "Advances in Knowledge Discovery and Data Mining", Jet Propulsion Laboratory, California Institute of Technology, pages 1-3

[19] Hendrik.Blockeel, Luc.DeRaedt (1997), "Top-down Induction of Logical Decision Trees", Katholieke Universiteit Leuven Department of Computer Science Celestijnenlaan 200A, pages 2-20

[20] Rakesh Agrawal Johannes Gehrke, Dimitrios Gunopulos, Prabhakar Raghavan (1997), "Automatic Subspace Clustering of High Dimensional Data for Data Mining", Applications IBM Almaden Research Center650 Harry Road, San Jose, CA 95120, pages 3-5.

[21] Ralph Grishman, (1997), "Information Extraction: Techniques and Challenges", Computer Science Department, New York University New York NY10003 U.S.A, pages. 7-9.

[22] Saurav Kumar Singh (1997), "Statistical Information Grid", Department of Computer Science & Engineering Dual degree 4th year, pages 17-18.

[23] Wei Wang, Jiong Yang, Richard Muntz, STING (1997), " A Statistical Information Grid Approach to Spatial Data Mining", University of California, Los Angeles CA 90095, U.S.A, pages 190-197.

[24] Brooking (1998), Brooking A. Corporate Memory: "Strategies for Knowledge Management", pages 4-5.

[25] Jerzy W. Jaromczyk and Godfried T. Toussaint(1998), "Relative Neighborhood Graphs and Their Relatives", pages 4-9

[26] Bernhard Ganter and Rudolf Wille (1999), "Conceptual Scaling", Springer pages 139-167.

[27] Yair Weiss, Michael I Jordan (1999), "Segmentation using eigenvectors", CA 94720 - 1776, pages 2-8.

[28] ISMO KÄRKKÄINEN and PASI FRÄNTI(2000), "MINIMIZATION OF THE VALUE OF DAVIES-BOULDIN INDEX", Department of Computer Science, University of Joensuu Box 111, FIN-80101 Joensuu, FINLAND,pages 2-7.

[29] Thomas G. Dietterich, Suzanna Becker, Zoubin Ghahramani (2001), Edition Cambridge, Massachusette , "Advances in Neural Information Processing Systems": Proceedings of the 2001 conference, London, England, pages 849-853

[30] Kramer S., Lavrac N., Flach P (2001), "Propositionalization approaches to relational data mining", in relational data mining, Dzeroski S., Springer Edition, pages 262- 291.

[31] Sandra Berasaluce (2002) thèse doctorat, "Fouille de données et acquisition de connaissances à partir de bases de données de réactions chimiques", Pages 53-66.

[32] Michail Vlachos Jessica Lin Eamonn Keogh Dimitrios Gunopulos (2002), "A Wavelet-Based Anytime Algorithm for K-Means Clustering of Time Series", Computer Science & Engineering Department University of California - Riverside, CA 92521, pages 2-4.

[33] Jean-Pierre Nakache (2004), Josiane Confais, "Approche pragmatique de la classification : arbres hiérarchiques ... ", Edition TRCHNIP, ISBN 271080848X, page 203.

[34] CHIH-TANG CHANG1, JIM Z. C. LAI2 AND MU-DER JENG (2011), "A Fuzzy K-means Clustering Algorithm Using Cluster Center Displacement", JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 27, 995-1009, pages 995-997.

[35] Stephen MUGGLETON (2012), "Inductive Logic Programming", The Turing Institute, 36 North Hanover St, Glasgow G1 2AD, United Kingdom, pages 295-345.

[36] Bernardo M. Abrego (2013), Ruy Fabila-Monroy Silvia Fernandez-Merchant, David Flores-Penaloza, Ferran Hurtado, Vera Sacristan, Maria Saumell, "On Crossings in Geometric Proximity Graphs", pages 3-5

[37] Nils Kriege, Petra Mutzel, Till Schäfer (2013), "SAHN Clustering in Arbitrary Metric Spaces Using Heuristic Nearest Neighbor Search Algorithm Engineering", Report TR13-1-003 August 2013 ISSN 1864-4503, pages 2-13

[38] Claire Chabanet, Fabrice Dessaint (2015), "Outliers ou données extrêmes, comment les détecter ?que faire de ces observations", pages 1-7

[39] Ricco RAKOTOMALALA(2016), "Algorithme des k-médoïdes, classification automatique", Université Lumière Lyon 2, pages 9-11.

[40] Jan Platoš (2017), "Data Analysis 2, Density-based Clustering", Department of Computer Science Faculty of Electrical Engineering and Computer Science VŠB - Technical University of Ostrava, pages 14-16.

[41] Antoine Lefébure (1979), "Le monde diplomatique", pages 14-15.

[42] Jean-Louis Ermine (2010), "Using cartography to sustain inter-generation knowledge transfer", the M3C methodology 2010, pages 2-6

[43] Claude Elwood Shannon (Juliet 1948): "A Mathematical Theory of Communication", Bell System Technical Journal, vol. 27, no 3, pages 379-423

[44] Brooking (1998), Brooking, "A. Corporate Memory: Strategies for Knowledge Management", pages 4-5.

[45] Penalva J.-M, & Montmain J (2002), "Travail collectif et intelligence collective : les référentiels de connaissances", In Proceedings of IPMU'2002, Annecy, France.

[46] Ikujiro Nonaka, "The Knowledge Creating Company", Harvard Business Business School Publishing Corporation 1991-2007 ,ISBN 987-1-4221-7974-1

[47] Nada Matta, Jean Louis Ermine, Gérard Aubertin, Jean-Yves Trivin (1996), "How to capitalize knowledge with the MASK method ", pages 8-13

[48] Jean-François Ballay (1997), "Capitaliser et transmettre les savoir-faire de l'entreprise", Paris, Eyrolles, (Direction des études et recherches d'Électricité de France).

[49] Jean-Yves Prax, "Le guide du Knowledge Management. Concepts et pratiques du management de la connaissance Paris", Dunod, 2002.

[50] Donald. Schön (1983), "Reflection-in-action and reflection-on-action", page 60

[51] Sandra Berasaluce (2002), "thèse : Fouille de données et acquisition de connaissances à partir de bases de données de réactions chimiques", pages 84-85

[52] Ryszard S.Michalski : "A theory and methodology of inductive learning" ,1983 Elsevier pages 111-161

[53] Usama M. Fayyad(1996), "Advances in Knowledge Discovery and Data Mining", Jet Propulsion Laboratory, California Institute of Technology

[54] Rakesh Agrawal(1993), "Fast Discovery of Association Rules", pages 308-327

[55] Frédéric Pennerath (2009), "Méthodes d'extraction de connaissances à partir de données modélisables par des graphes. Application à des problèmes de synthèse organique", Thèse de doctorat, Université Henri Poincaré - Nancy I, page 24.

[56] MAMI Mohammed Nassim(2013) (thèse) : "Extraction des connaissances dans un environnement distribué : synthèse", pages 31-33

[57] Rakesh Agrawal, Tomasz Imielinski, Arun Swami (1993) , " Mining Associations between Sets of Items in large Databases", ACM SIGMOD'93, pages 207-216.

[58] Thierry Lecroq, "Extraction de règles d'association", (2016), Université de Rouen, France, page 10-11

[59] Rakesh Agrawal et Ramakrishnan Srikant (1994), "Fast Algorithms for Mining Association Rules", IBM Almaden Research Center,650 Harry Road, San Jose, CA 95120, pages 488-492

[60] Guillaume CALAS (2009), "Études des principaux algorithmes de data mining", EPITA 14-16 rue Voltaire, 94270 Le Kremlin-Bicêtre, France, pages 6-7.

[61] Franz Baader, Bernhard Ganter, Baris¸ Sertkaya,and Ulrike Sattler(2006), "Completing Description Logic Knowledge Bases using Formal Concept Analysis", TU Dresden, Germany and The University of Manchester, UK, pages 2-6.

[62] Willy Ugarte Rojas(2014), "Thèse de doctorat : Extraction de motifs sous contraintes souples", Université de Caen Basse-Normandie ´ Ecole doctorale SIMEM, pages 19-22.

[63] Albrecht Zimmermann, Katholieke Universiteit Leuven (2007), "Constraint-Based Pattern Set Mining", pages 237-247.

[64] Bernhard Ganter and Rudolf Wille (1999), "Conceptual Scaling", Springerm pages 139-167.

[65] Xavier Dolques, Florence Le Ber, Marianne Huchard, Clémentine Nebut (2012) : "Analyse Relationnelle de Concepts pour l'exploration de données relationnelles", pages 3-12.

[66] Stephen MUGGLETON, "Inductive Logic Programming", The Turing Institute,36 North Hanover St, Glasgow G1 2AD,United Kingdom, pages 295-345.

[67] Nada Lavrač, Sašo Džeroski (April 1996), "Review of Inductive Logic Programming: Techniques and Applications Machine Learning ", Volume 23,pages 103-108.

[68] Jean–Christophe Routier (1994), "Terminaison, Satisfiabilité, Puissance de calcul d'une clause de Horn Binaire", Thèse de Doctorat en Informatique, LIFL – UA CNRS 369, Université des Sciences et Technologies de Lille, Pages 41-58.

[69] Kramer S., Lavrac N., Flach P (2001), "Propositionalization approaches to relational data mining", in relational data mining, Dzeroski S., Springer Edition, pages 262- 291.

[70] Hendrik.Blockeel, Luc.DeRaedt (1997), "Top-down Induction of Logical Decision Trees", Katholieke Universiteit Leuven Department of Computer Science Celestijnenlaan 200A, pages 2-20

[71] Nadjim Chelghoum, Karine Zeitouni, Thierry Laugier, Annie Fiandrino, Lionel Loubersac (2007), "Fouille de données spatiales Approche basée sur la programmation logique inductive", Institut Français de Recherche pour l'Exploitation de la Mer (IFREMER), Laboratoire Environnement- Ressources du Languedoc Roussillon (LER/LR), pages 4-12

[72] J. R. Quinlan and R. M. Cameron-Jones (2006), FOIL: "A Midterm Report, Basser Department of Computer Science", University of Sydney, Sydney Australia, pages 2-15

[73] Nadjim Chelghoum, Karine Zeitouni, Thierry Laugier, Annie Fiandrino, Lionel Loubersac (2007), "Fouille de données spatiales : Approche basée sur la programmation logique inductive", pages 503-540

[74] Leonhard Euler (1736), "Solution problematis ad geometriam situs pertinentis", Commentarii Academiae Scientiarum Imperialis Petropolitanae Vol 8, pages 128–140.

[75] Dénes König (1936), "Theorie der Endlichen und Unendlichen Graphen", Teubner, Leipzig, page 45.

[76] Alain Bretto, Alain Faisant, François Hennecart (2012), "Éléments de théorie des graphes", Springer, pages 1-34.

[77] Jean-Charles Régin, Arnaud Malapert (2015), "Théorie des Graphes", pages 9-15.

[78] Eric MATON (2007), "Représentation graphique et pensé managerielle cas de la Harvard Business", Review de 1922 à 1999, Thèse DOCTEUR DE L'ECOLE POLYTECHNIQUE, pages 39-42

[79] Cover TM & Hart PE (1967), "nearest neighbor pattern classification". IEEE Trans. Inform. Theory IT-13:21-7, 1967. Dept. Electrical Engineering, Stanford Univ., Stanford, and Stanford Res. Inst., Menlo Park, CA

[80] Leo Breiman, Jerome Friedman, Charles J. Stone, R.A. Olshen (1984), "Classification and Regression Trees (Wadsworth Statistics/Probability) ", J.R. QUINLAN, Induction of Decision Trees, 1986 Kluwer Academic Publishers, Boston, Machine Learning 1, pages 81-106

[81] Guillaume Cleuziou (2006) Thèse doctorat, "Une méthode de classification non-supervisée pour l'apprentissage de règles et la recherche d'information", pages 7-10

[82] Romain Quéré (2014), "Quelques propositions pour la comparaison de partitions non strictes", thèse doctorat, pages 5-7

[83] Guillaume Cleuziou (2004), "Une méthode de classification non-supervisée pour l'apprentissage de règles et la recherche d'information", thèse doctorat, page 10.

[84] Claire Chabanet, Fabrice Dessaint (2015), "Outliers ou données extrêmes, comment les détecter ?que faire de ces observations", pages 1-7

[85] Sylvain Foisy, Ph. D.Bio-informaticien conseil (2006), Diploide BioIT : "Introduction à la transcriptomique, Clustering et classification supervisée", pages 8-10

[86] Bruno Pinaud (2013), Regroupement (Clustering), pages 1-30.

[87] Serge Vladimir Emteu Tchagou (2016) thèse doctorat, "Réduction à la volée du volume des traces d'exécution pour l'analyse d'applications multimédia de systèmes embarqués", pages 49-55.

[88] Jean-Louis Prieur (2014), "Thèse d'habilitation à diriger des recherches Contribution en Signal, Image et Instrumentation pour l'Astronomie", page 75.

[89] Julie Dubois-Chevalier (2012), Chimio-thèques ; "vers une approche rationnelle de la sélection de sous-chimio-thèques", thèse doctorat, pages 32-34.

[90] Florin Gorunescu (2011), "Intelligent Systems Reference Library", Volume 12, Editors-in-Chief, Data Mining, 2011,ISBN 978-3-642-19720-8, page 279.

[91] Marion Chevrier(2014), "Evaluation des récents changements de la diversité floristique en France", pages 13-19.

[92] Jaccard P. (1901), "Distribution de la flore alpine dans le bassin de Dranses et dans quelques régions voisines", Bulletin de la Société Vaudoise des Sciences naturelles, pages 37, 241-272.

[93] E. B. Fowlkes and C. L. Mallows (1983), "A Method for Comparing Two Hierarchical Clustering", Journal of the American Statistical Association, Vol. 78, No. 383 (Sep., 1983), pages 553-569.

[94] NCSS Statistical Software (2017), NCSS.com, "Fuzzy Clustering", Chapter 448

[95] Veronica S. Moertini (2002), "Introduction To Five Data Clusterin Galgorithms", Integral, Vol. 7, No. 2, pages 2-10

[96] Nidhi Grover(2014), "A study of various Fuzzy Clustering Algorithms", Assistant Professor, Institute of Information Technology & Management, Delhi, India, International Journal of Engineering Research ISSN:2319-6890)(online), 2347-5013(print), Volume No.3, Issue No.3, pages : 177-181 .

[97] Aleksander Przybylo (2005), "optimisation par algorithmes de Clustering de la construction automatique de bases de connaissances floues", thèse doctorat, pages 30-42.

[98] D. Chessel, J. Thioulouse & A.B. Dufour (2004), "introduction à la classification hiérarchique", pages 22-48

[99] Stéphane Vialle (2017), "Big Data : Informatique pour les données et calculs massifs : Algorithmes de Clustering", pages 3-7.

[100] Dhafer Malouche (2013), "Méthodes de classifications", ESSAI-U2S-ENITANR-Do Well Be, Saint Nectaire, page 13.

[101] Maurice Roux (2015), A comparative study of divisive hierarchical clustering algorithms, pages2-12

[102] Nils Kriege, Petra Mutzel, Till Schäfer (2013), "SAHN Clustering in Arbitrary Metric Spaces Using Heuristic Nearest Neighbor Search Algorithm", Engineering Report TR13-1-003 August 2013 ISSN 1864-4503, pages 2-13

[103] William H. E. Day, Herbert Edelsbrunner (1984), "Efficient Algorithms for Agglomerative Hierarchical Clustering Methods", Journal of Classification 1:7-24 (1984), pages 9-18

[104] Mark J. Embrechts1, Jonathan D. Linton2, and Christopher J. Gatti1 (2012), "Hybrid Hierarchical Clustering: Cluster Assessment via Cluster Validation Indices", ESANN 2012 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2012, pages 317-318

[105] R BERTRAND, E. D1DAY (1990), "Une généralisation des arbres hiérarchiques : les représentations pyramidales", Revue de statistique appliquée, tome 38, n° 3 (1990), pages 53-78.

[106] Laure Vescovo (2007), "Outils et méthodes pour la classification pyramidale de données biologiques", thèse de doctorat, pages 52-94.

[107] Vincent Lemaire and Oumaima Alaoui, IsmailiAn, Antoine Cornuejols (2012), "Initialization Scheme for Supervized K-means", pages 2-3.

[108] CHIH-TANG CHANG1, JIM Z. C. LAI2 AND MU-DER JENG(2011), "A Fuzzy K-means Clustering Algorithm Using Cluster Center Displacement", JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 27, 995-1009, pages 995-997.

[109] Sueli A. Mingoti, Joab O. Lima (2005), "Comparing SOM neural network with Fuzzy c-means K-means and traditional hierarchical clustering algorithms", European Journal of Operational Research 174 (2006), pages 1742–1759.

[110] Ricco RAKOTOMALALA(2016), "Algorithme des k-médoïdes, classification automatique", Université Lumière Lyon 2, pages 9-11.

[111] Hae-Sang Park, Chi-Hyuck Jun (2009), "A simple and fast algorithm for K-médoïdes clustering, Expert Systems with Applications 36", pages 3336–3341.

[112] Frank Dellaert (2002), College of Computing, Georgia Institute of Technology, Technical Report number GIT-GVU-02-20, February 2002, pages 1-7.

[113] A. P. Dempster, N. M. Laird, D. B. Rubin, Journal of the Royal Statistical Society. Series B (Methodological), Vol. 39, No. 1. (1977), pages 1-38.

[114] Yihua Chen and Maya R. Gupta (2010), Department of Electrical Engineering, University of Washington Seattle UWEETR-2010-0002, WA 98195, pages 13-17.

[115] Wei Wang, Jiong Yang, Richard Muntz, STING (1997): "A Statistical Information Grid Approach to Spatial Data Mining", University of California, Los Angeles CA 90095, U.S.A, pages 190-197.

[116] Rakesh Agrawal Johannes Gehrke, Dimitrios Gunopulos, Prabhakar Raghavan (1997), "Automatic Subspace Clustering of High Dimensional Data for Data Mining" Applications IBM Almaden Research Center650 Harry Road, San Jose, CA 95120, pages 3-5.

[117] Saurav Kumar Singh (1997), "Statistical Information Grid", Department of Computer Science & Engineering Dual degree 4th year, pages 17-18.

[118] Michail Vlachos Jessica Lin Eamonn Keogh Dimitrios Gunopulos (2002), "A Wavelet-Based Anytime Algorithm for K-Means Clustering of Time Series", Computer Science & Engineering Department University of California - Riverside, CA 92521, pages 2-4.

[119] Suman, Pinki Rani (2017), "A Survey on STING and CLIQUE Grid Based Clustering Methods", Department of Computer Science, Kurukshetra University, Kurukshetra, India, International Journal of Advanced Research in Computer Science, Volume 8, No. 5, May-June 2017, ISSN No. 0976-5697, pages 1510-1512

[120] Jan Platoš (2017),Data Analysis 2, "Density-based Clustering", Department of Computer Science Faculty of Electrical Engineering and Computer Science VŠB - Technical University of Ostrava, pages 14-16.

[121] Alexander Hinneburg1 and Hans-Henning Gabriel2 (2004), "DENCLUE 2.0: Fast Clustering based on Kernel Density Estimation", Institute of Computer Science Martin-Luther-University Halle-Wittenberg, Germany, pages 3-11.

[122] Deyi Li and Yi Du, "ARTIFICIAL INTELLIGENCE, WITH UNCERTAINTY", Tsinghua University Beijing (2007), China, Chapman & Hall/CRC Taylor & Francis Group 6000 Broken Sound Parkway NW, Suite 300, Boca Raton, FL 33487-2742, ISBN 13: 978-1-58488-998-4 (Hardcover), pages 229-236

[123] S. Michalski (1983), J. Carbonell and T. Mitchell (Editors), Chapter in the book, "Machine Learning: An Artificial Intelligence Approach", R., TIOGA Publishing Co., Palo Alto. California, pages 3-20.

[124] Douglas Fisher(1987), "Machine Learning & Knowledge Acquisition", Improving Inference Through Conceptual Clustering, Department of Information and Computer Science University of California Irvine, California 92717,AAAI-87 Proceedings. Copyright ©1987, pages 461-465

[125] Guillaume Cleuziou (2006), "Une méthode de classification non-supervisée pour l'apprentissage de règles et la recherche d'information", thèse doctorat, pages 42-44.

[126] John H. Gennari, Pat Langley and Doug Fisher (1989), "Models of Incremental Concept Formation", Irvine Computational Intelligence Project, Department of Information and Computer Science, University of California, Irvine, CA 92717, U.S.A, pages 30-31.

[127] Quinlan, J.R (1986), "Induction of decision trees", Mach. Learning 1, pages 81-106.

[128] Sébastien Ferré (2017), "Concepts de plus proches voisins dans des graphes de connaissances", IRISA/Université de Rennes 1 Campus de Beaulieu, 35042 Rennes cedex,ferre@irisa.fr, pages 2-10.

[129] Wei Dong, Moses Charikar, Kai Li(2011), "Efficient K-Nearest Neighbor Graph Construction for Generic Similarity Measures", Department of Computer Science, Princeton University 35 Olden Street, Princeton, NJ 08540, USA, pages 2-3.

[130] Abhijeet Khopkar, Sathish Govindarajan (2012), "On computing optimal Locally Gabriel Graphs", pages 1-5

[131] Sylvie Hamel (2011), "Arbre couvrant minimal", IFT2125, A 2011, Université de Montréal, pages 6-9.

[132] Bernardo M. Abrego (2013), Ruy Fabila-Monroy Silvia Fernandez-Merchant, David Flores-Penaloza, Ferran Hurtado, Vera Sacristan, Maria Saumell, "On Crossings in Geometric Proximity Graphs ", pages 3-5

[133] Jerzy W. Jaromczyk and Godfried T. Toussaint(1998), "Relative Neighborhood Graphs and Their Relatives ", pages 4-9

[134] Thomas G. Dietterich, Suzanna Becker, Zoubin Ghahramani (2001), Edition Cambridge, Massachusette , "Advances in Neural Information Processing Systems ", Proceedings of the 2001 conference, London, England, pages 849-853

[135] Yair Weiss, Michael I Jordan (1999), "Segmentation using eigenvectors", CA 94720 - 1776, pages 2-8.

[136] Jean-Pierre Nakache (2004), Josiane Confais, "Approche pragmatique de la classification : arbres hiérarchiques ... ", Edition TRCHNIP, page 203.

[137] Olatz Arbelaitz (2013), Ibai Gurrutxaga, Javier Muguerza, Jesus M. Pérez, Inîgo Perona, "An extensive comparative study of cluster validity indices", ELSEVIER Pattern Recognition 46 (2013), pages 243–256.

[138] ISMO KÄRKKÄINEN and PASI FRÄNTI(2000), "MINIMIZATION OF THE VALUE OF DAVIES-BOULDIN INDEX", Department of Computer Science, University of Joensuu Box 111, FIN-80101 Joensuu, FINLAND,pages 2-7.

[139] Slobodan Petrovic NISlab (2004), "A Comparison Between the Silhouette Index and the Davies-Bouldin Index in Labelling IDS Clusters", Department of Computer Science and Media Technology, Gjøvik University College, P.O. box 191, 2802 Gjøvik, Norway, pages 4-12.

[140] Caglar Cengizler and M. Kerem Un (2017), "Evaluation of Calinski-Harabasz Criterion as Fitness Measure for Genetic Algorithm Based Segmentation of Cervical Cell Nuclei", British Journal of Mathematics & Computer Science Article no.BJMCS.33729, pages 6-11.

[141] Un Yong Nahm and Raymond J. Mooney (2002), "Text Mining with Information Extraction ", Department of Computer Sciences, University of Texas, page 2

[142] Un Yong Nahm (2004), Doctoral Dissertation: "Text mining with information extraction", The University of Texas at Austin ©2004 ISBN: 0-496-01283-5, page 218.

[143] Lehnert, W. Cardie, C. Fisher D. Riloff, E, & Williams,R. (1991), "Description of the CIRCUS ,System as Used for MUC-3", a. University of Massachusetts: Third Message Understanding Conference (MUC-3). San Diego, CA, Morgan Kaufrnann, pages. 223-233.

[144] Ralph Grishman, (1997), "Information Extraction: Techniques and Challenges", Computer Science Department, New York University New York NY10003 U.S.A, pages. 7-9.

[145] Alexandre Saidi (2005), "Textual Information Extraction Using Structure Induction", LIRIS-CNRS (UMR 5205) École Centrale de Lyon, Mathematics and Computer Science Department, pages 4-9

[146] Ronen Feldman (2006): "THE TEXT MINING HANDBOOK Advanced Approaches in Analyzing Unstructured Data", livre ISBN-13 978-0-511-33507-5, pages 65-80.

[147] Appelt, Douglas E., Jerry R. Hobbs, John Bear, David Israel, and Mabry Tyson, (1993)."FASTUS: "A Finite State Processor for Information Extraction from Real World Text ", Proceedings. IJCAI-93, Chambery, France.

[148] Thierry Poibeau, Horacio Saggion, Jakub Piskorski, Roman Yangarber (2013), "Multi-source, Multilingual Information Extraction and Summarization", pages 27-28

[149] Gokhan Tur, Renato De Mori (2011)."Spoken Language Understanding: Systems for Extracting Semantic Information from speetch". John Wiley & Sons, Ltd 2011, ISBN: 978-0-470-68824-3, pages 93-118.

[150] RIJSBERGEN B. , "Information retrieval, Department of Computing Science", University of Glasgow, published by the press syndicate of the university of Cambridge (2004) , pages. 3-6.

[151] Liu B. Web Data Mining, "Exploring Hyperlinks, Contents, and Usage Data, Data-Centric Systems and Applications". Springer-Verlag Berlin Heidelberg. DOI 10.1007/978-3-642-19460-3 (2011), pages 213-214.

[152] Ali M., Zainal A., Christiano S., Suryo I. , "Indonesian News Classification Using Naïve Bayes and Two-Phase Feature Selection Model", Indonesian Journal of Electrical Engineering and Computer Science (IJEECS) (2016), pages 401-408.

[153] Manwar B., "A vector space model for automatic indexing", Indian Journal of Computer Science and Engineering (IJCSE) (2012), pages 223-228.

[154] Stephen E, Robertson, Karen S., "Relevance weighting of search terms", Journal of the American Society for Information Science, (1976), 27(3), pages 129–146.

[155] Stephen E., "The Probability Ranking Principle in IR", University College London , Journal of Documentation, (1977), 33(4), pages 294-303

[156] Frank Dellaert (2002), College of Computing, Georgia Institute of Technology, Technical Report number GIT-GVU-02-20, February 2002, pages 1-7.

[157] Saurav Kumar S., "Statistical Information Grid", Department of Computer Science & Engineering Dual degree 4th year (1997), pages 17-18.

[158] Alain P, Boëffard O, " Evaluation des Modèles de Langage n-gramme et n=m-multigramme", Dourdan Journal TALN, (2005), pages 1- 4.

[159] Iftakher Md. et al., "An investigative design of optimum stochastic language model for bangla autocomplete", Indonesian Journal of Electrical Engineering and Computer Science (IJEECS (2019),13(2), pages) 671-676.

[160] Nakache D., Metais E., " Evaluation : nouvelle approche avec juges", Conference : Actes du XXIIIème Congrès INFORSID. Grenoble, (2005), pages 2-4

[161] Julyeta P., Irene R., " Vertical Information System: A Case Study of Civil Servant Teachers Data in Manado City", Indonesian Journal of Electrical Engineering and Computer Science (2017), 6(1), pages 42-49.

[162] Trikha, Priyanka, and Singh Vijendra. 2013., "Fast Density Based Clustering Algorithm. ", International Journal of Machine Learning and Computing 3 (1):10–12. https://doi.org/10.7763/IJMLC.2013.V3.262.

[163] Jean-Yves Prax, "Le guide du Knowledge Management. Concepts et pratiques du management de la connaissance Paris", Dunod, 2002.

[164] Jan Platoš, "Data Analysis 2,Density-based Clustering", Department of Computer Science Faculty of Electrical Engineering and Computer Science VŠB, Technical University of Ostrava, (2017), pages 14-16.

[165] Hag Barman, et al., "Clustering Techniques for Software Engineering", Indonesian Journal of Electrical Engineering and Computer Science, Vol. 4, No. 2, November 2016, pages 465-472.

[166] A. Hinneburg and H. Henning Gabriel, "DENCLUE 2.0: Fast Clustering based on Kernel Density Estimation", Institute of Computer Science Martin-Luther-University Halle-Wittenberg, Germany , pp 3-11, 2004.

[167] Alexander H, Daniel A. Keim,"An efficient approach to clustering in large multimedia databases with noise ", Proceedings of the Fourth International Conference on Knowledge Discovery and Data MiningAugust,pp. 58–65, 1998.

[168] J. Lu and Q. Zhu, "An effective algorithm based on density clustering framework, " IEEE Access, doi:10.1109/ACCESS.2017.2688477, vol. 5, 2017, pages 4991-5000

[169] Chen, Xiaoming and Liu, Wanquan and Huining, Qiu and Lai, Jianhuang, "APSCAN: A parameter free algorithm for clustering. Pattern Recognition", Letters. 32 (7): pp. 973-986, 2011

[170] Sébastien Ferré (2017), "Concepts de plus proches voisins dans des graphes de connaissances", IRISA/Université de Rennes 1 Campus de Beaulieu, 35042 Rennes cedex,ferre@irisa.fr, pages 2-10.

[171] D. Yu, G. Liu, M. Guo, X. Liu, and S. Yao, "Density peaks clustering based on weighted local density sequence and nearest neighbor assignment", IEEE Access, vol. 7, pp. 34301-34317, 2019, doi:10.1109/ACCESS.2019.2904254.

[172] Dua, Dheeru and Graff, Casey, "{UCI} Machine Learning Repository", University of California, Irvine, School of Information and Computer Sciences, 2019.

[173] R. Xu and D. Wunsch, "Clustering analysis" in Clustering, Ed. New Jersey: WileyIEEE Press ,ch. 1, sec. 1, pp.1-3, 2008

[174] Pooja B.,Priyanka A., "Comparative Study of Density based Clustering Algorithms", International Journal of Computer Applications (0975 – 8887), Vol. 27, No.11, August 2011.

[175] Sander J, "Density-based clustering", In Encyclopedia of Machine Learning, Springer, pp. 270–273, 2011.

[176] J. Hou and M. Pelillo, "A new density kernel in density peak based clustering", 23rd International Conferenceon attern Recognition (ICPR), Cancun, , doi:10.1109/ICPR.2016.7899678, pp. 468-473, Dec. 2016

[177] Deyi Li and Yi Du, "ARTIFICIAL INTELLIGENCE, WITH UNCERTAINTY", Tsinghua University Beijing (2007), China, Chapman & Hall/CRC Taylor & Francis Group 6000 Broken Sound Parkway NW, Suite 300, Boca Raton, FL 33487-2742,International Standard Book Number-13: 978-1-58488-998-4 (Hardcover), pages 229-236

[178] Trikha, Priyanka, and Singh Vijendra. 2013, "Fast Density Based Clustering Algorithm. ", International Journal of Machine Learning and Computing 3 (1):10–12. https://doi.org/10.7763/IJMLC.2013.V3.262, 2013

[179] Jan Platoš, "Data Analysis:Density Based Clustering, Cluster Validation", Department of Computer Science,Faculty of Electrical Engineering and Computer Science, VŠB ,Technical University of Ostrava,October 6, 2020.

[180] Glory H. Shah, C. K. Bhensdadia, Amit P. Ganatra, "An Empirical Evaluation of Density-Based Clustering Techniques", International Journal of Soft Computing and Engineering (IJSCE), ISSN: 2231-2307, Volume-2, Issue-1and March 2012.

[181] Khalid W. Al-Ani, Fairuz Bin Abdullah, Salman Yossuf, "Unequal clustering in wireless sensor network: a review", Indonesian Journal of Electrical Engineering and Computer Science, Vol. 22, No. 1, pp. 419~426,April 2021.

[182] P. H. Ahmad, and Shilpa Dang, "Performance evaluation of clustering algorithm using different datasets", International Journal of Advance Research in Computer Science and Management Studies, vol. 3, no. 1, January 2015 pp. 167-173, 2015.

[183] Sanskruti Patel, Atul Patel, "Performance Analysis and Evaluation of Clustering Algorithms", International Journal of Innovative Technology and Exploring Engineering(IJITEE)ISSN: 2278-3075,Vol. 8I, ssue-6S2, April 2019.

[184] Hans-Peter Kriegel, Peer Kröger, Jörg Sander, Arthur Zimek (2011), "Density-based Clustering", WIREs Data Mining and Knowledge Discovery 1 (3): 231–240. doi:10.1002/widm.30.

[185] Niphaphorn, Wiwat S.,"Optimal Choice of Parameters for DENCLUE-based and Ant Colony Clustering", International Conference on Modeling, Simulation and Control,2011,IPCSIT vol.10 IACSIT Press, Singapore.

[186] Fatima Batool,Christian Hennig, "Clustering with the Average Silhouette Width", Journal of Computational Statistics & Data Analysis • February 2021 DOI: 10.1016/j.csda.2021.107190

[187] Hans-Peter Kriegel, Peer Kröger, Jörg Sander, Arthur Zimek (2011) , "Density-based Clustering", WIREs Data Mining and Knowledge Discovery 1 (3): 231–240. doi:10.1002/widm.30.

[188] Shapol M. Mohammed, Karwan Jacksi, Subhi R. M. Zeebaree,"A state-of-the-art survey on semantic similarity for document clustering using GloVe and density-based algorithms", Indonesian Journal of Electrical Engineering and Computer Science, Vol. 22, No. 1, April 2021, pages 552-562.

[189] Narendra Sharma, Aman Bajpai, Mr. Ratnesh Litoriya "Comparison the various clustering algorithms of weka tools", International Journal of Emerging Technology and Advanced Engineering, ISSN 2250-2459, Volume 2, Issue 5 and May 2012

Publications scientifiques :

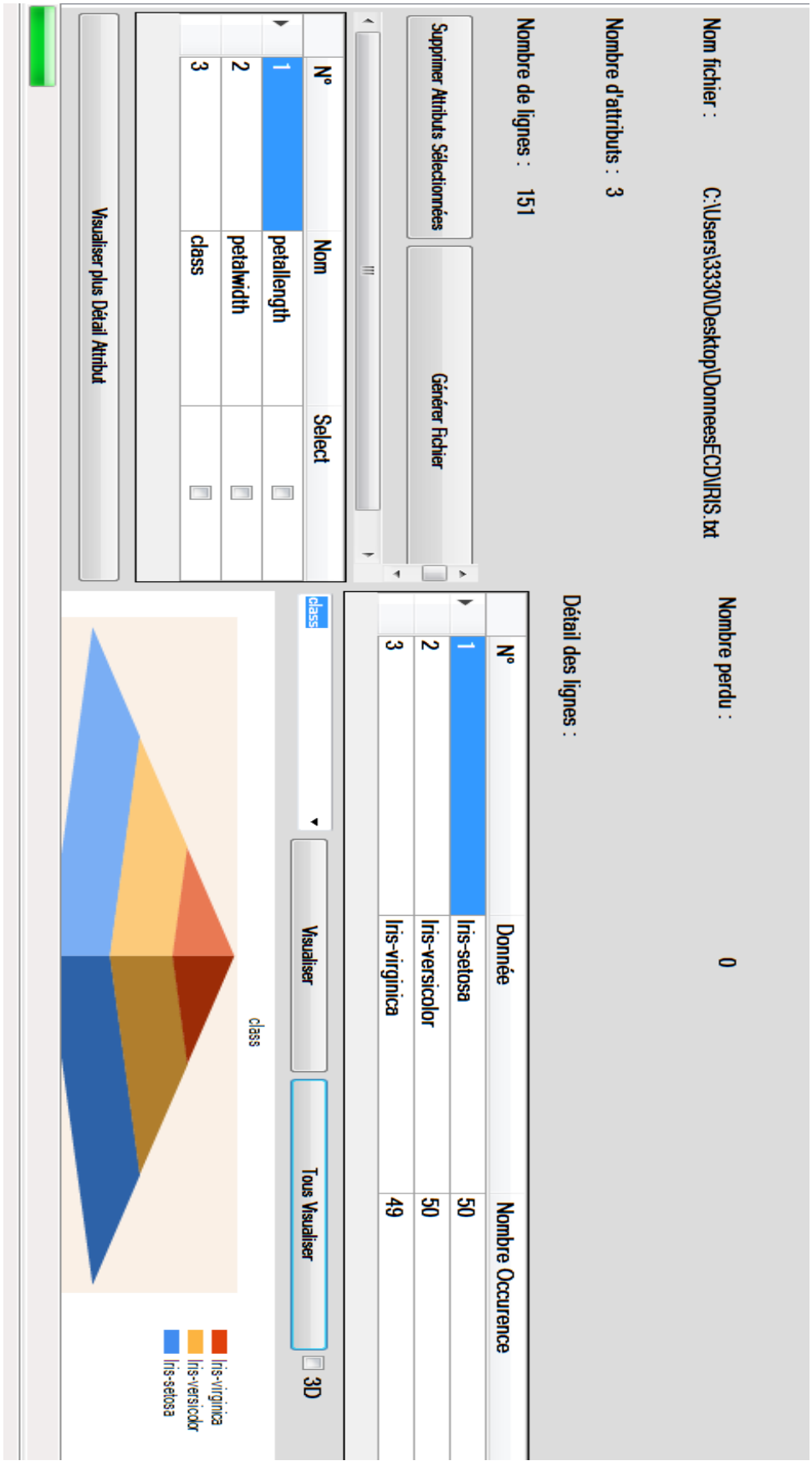
- **Octobre 2021**: Article « Density-based classification with the DENCLUE algorithm » au journal http://ijeecs.iaescore.com/index.php/IJEECS/article/view/25892.
- **Mai 2021** : Chapitre "The Search for Digital Information by Evaluating Four Models", Publisher : Springer International Publishing, Electronic ISBN : 978-3-030-76508-8
- **Août 2020** : Chapitre"Search for Information in Text Files" https://www.igi-global.com/chapter/search-for-information-in-text-files/237640 , DOI: 10.4018/978-1-7998-1021-6.ch004.
- **Août 2018** : Article "Process of Retrieval of Knowledge Starting From Data Texts" "https://www.iosrjournals.org/iosr-jce/papers/Vol20-issue4/Version-3/E2004033337.pdf"

Annexes : Les imprimés écrans de l’application ECD

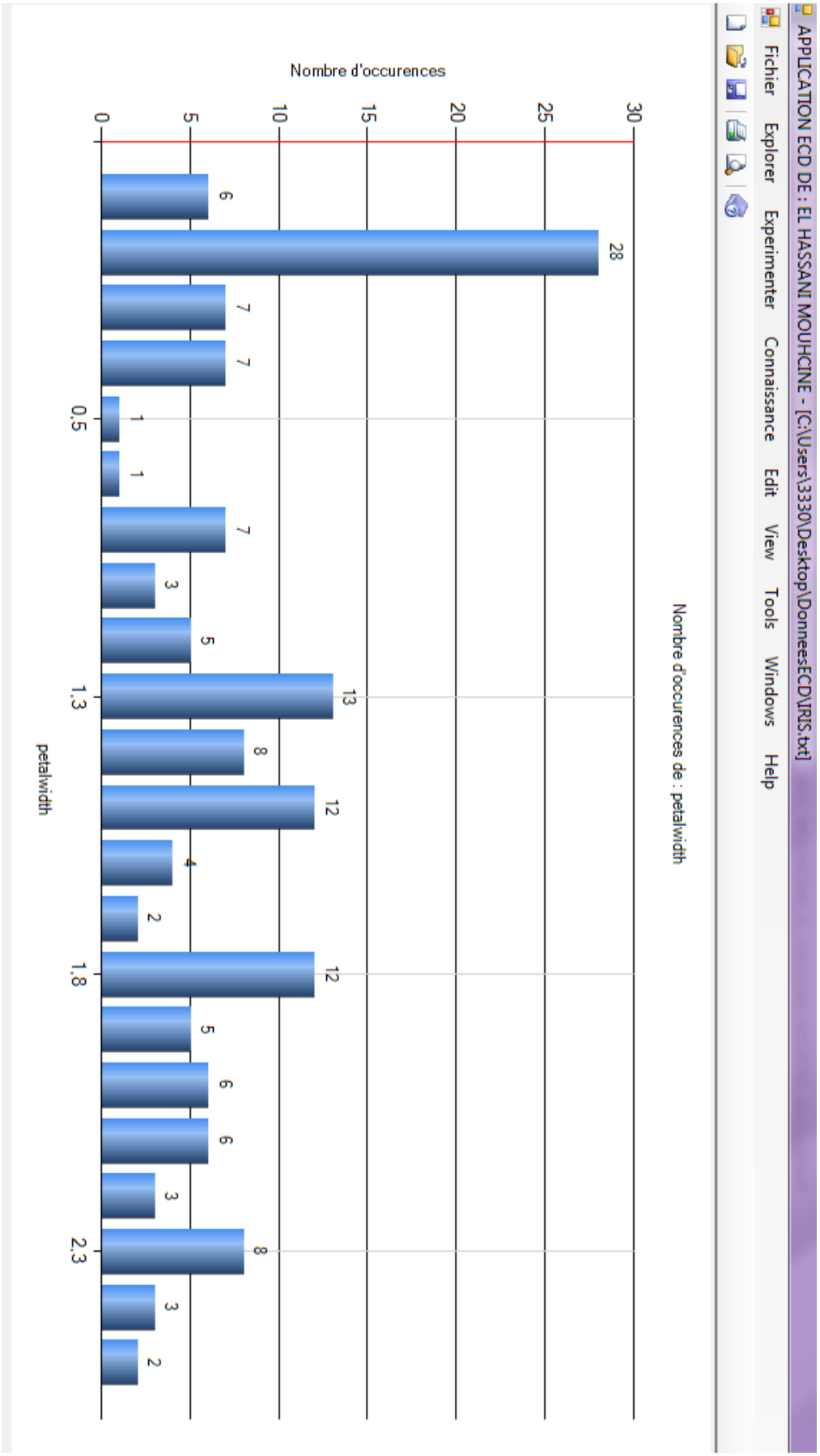
A. Etapes de traitement et de visualisation des données de la base de données « IRIS d'Anderson »

« De l'annexe 1 à l'annexe 10 »

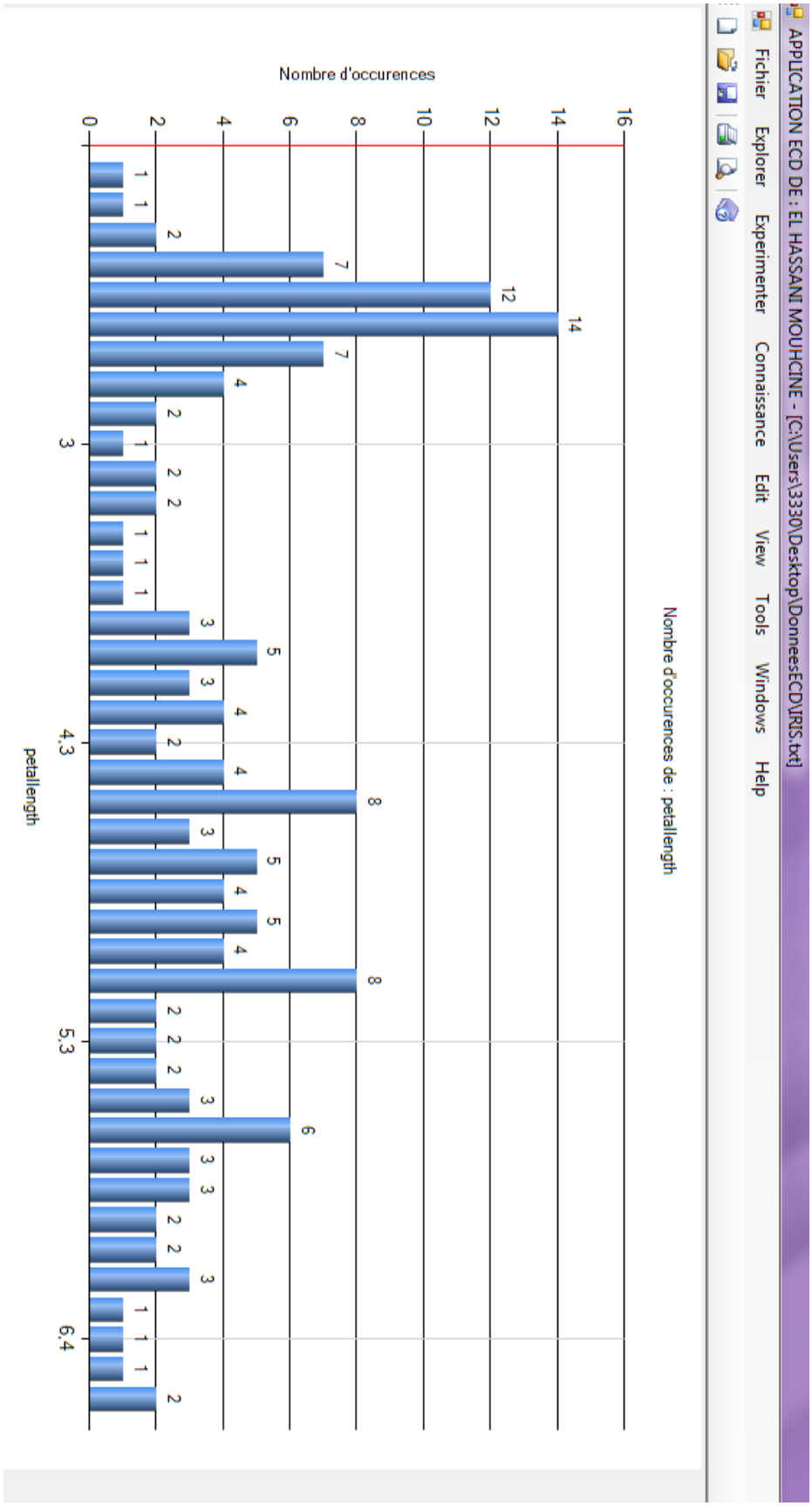
Images des IRIS



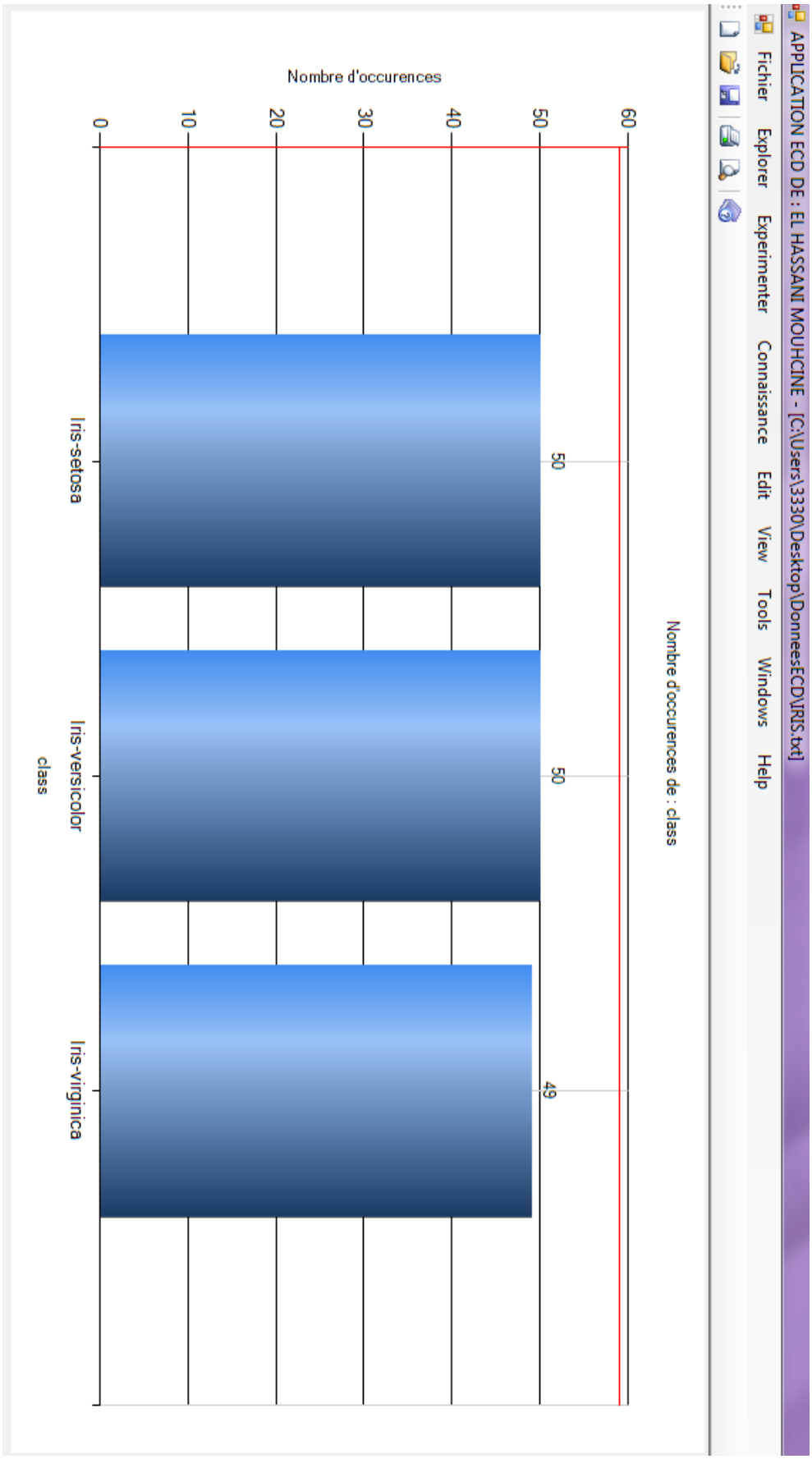
Annexe 1 : le menu général de chargement de la base de données des IRIS d'Anderson



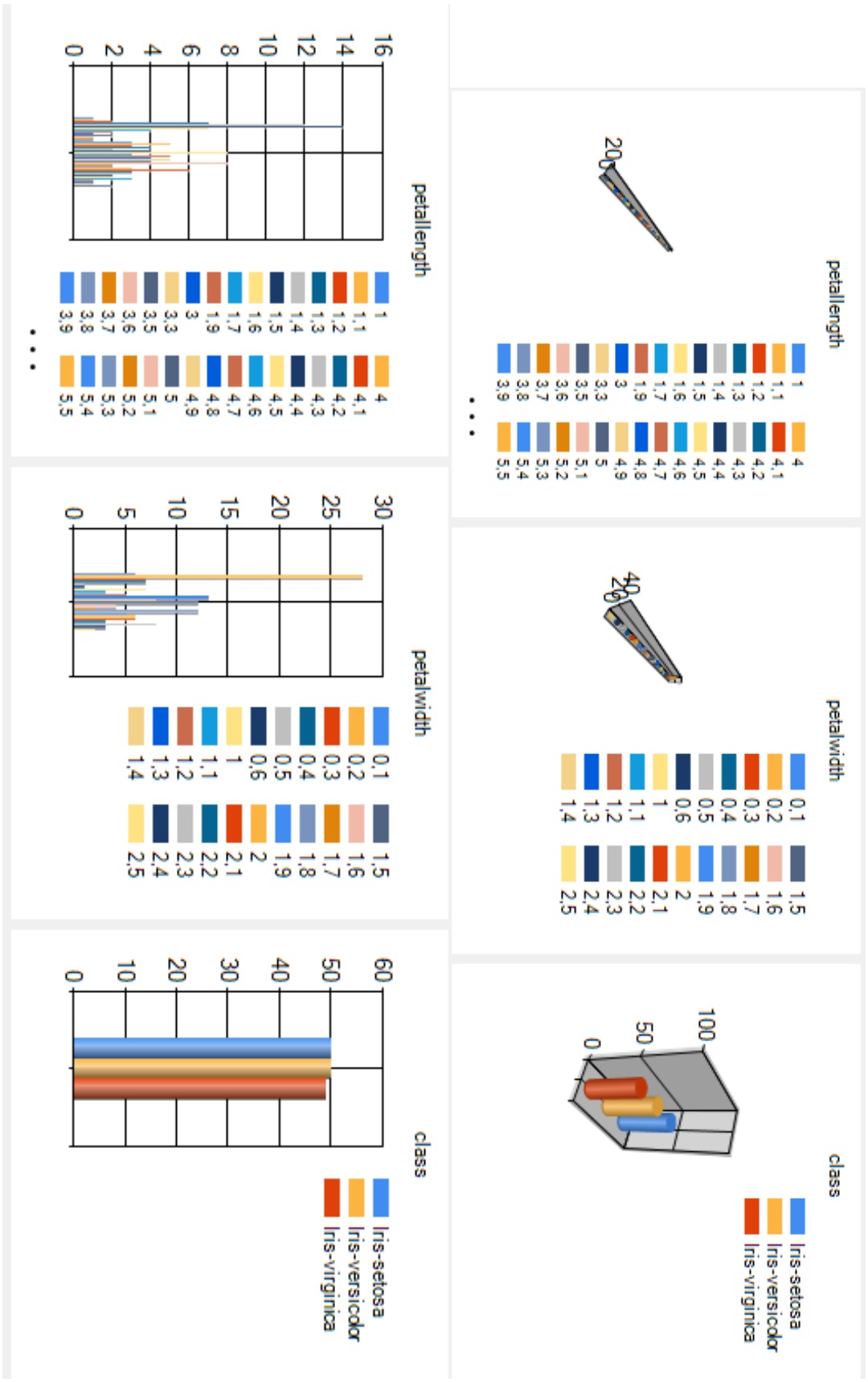
Annexe 2 : Histogramme en bâtons représentant le nombre d'occurrences de chaque valeur de largeur des pétales des IRIS.



Annexe 3 : Histogramme en bâtons représentant le nombre d'occurrences de chaque valeur de longueur des pétales des IRIS.

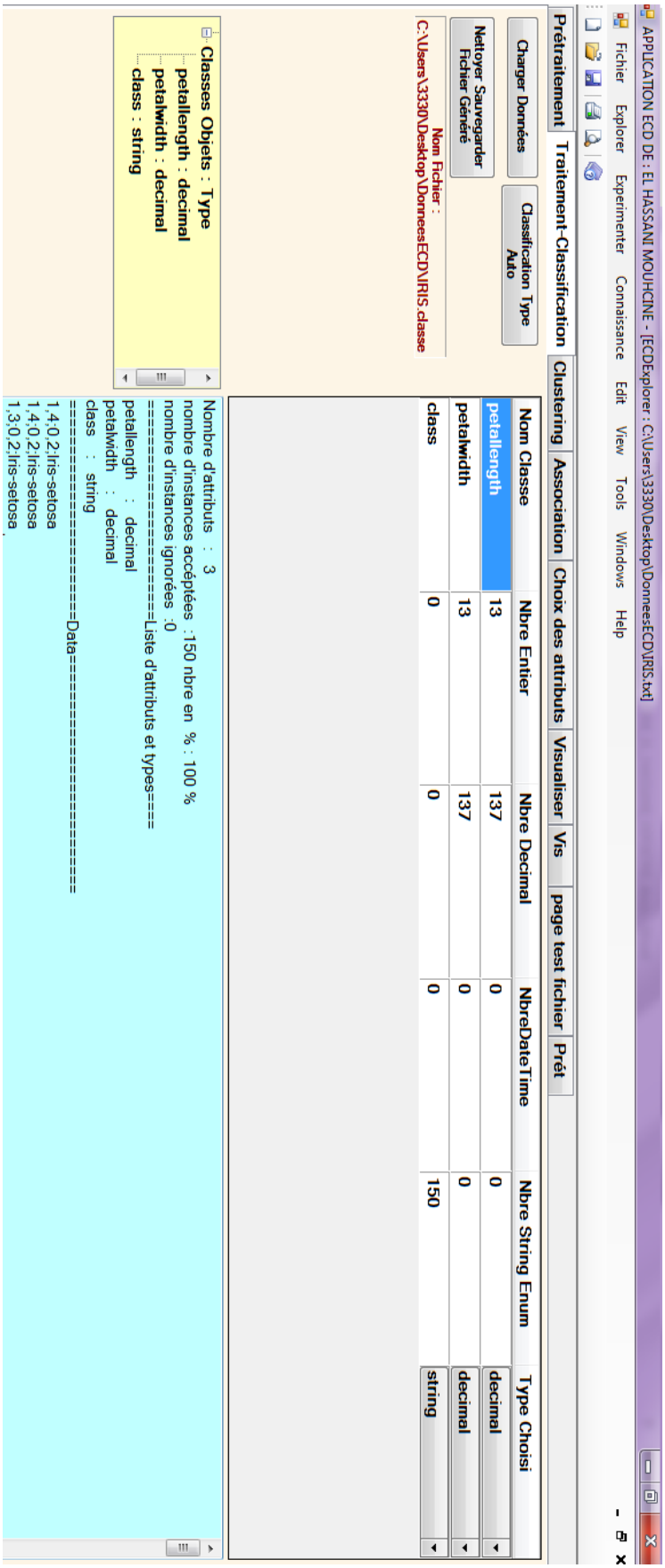


Annexe 4 : Histogramme en bâtons représentant le nombre d'éléments de chaque classe des IRIS (Setosa, Versicolor et virginica).



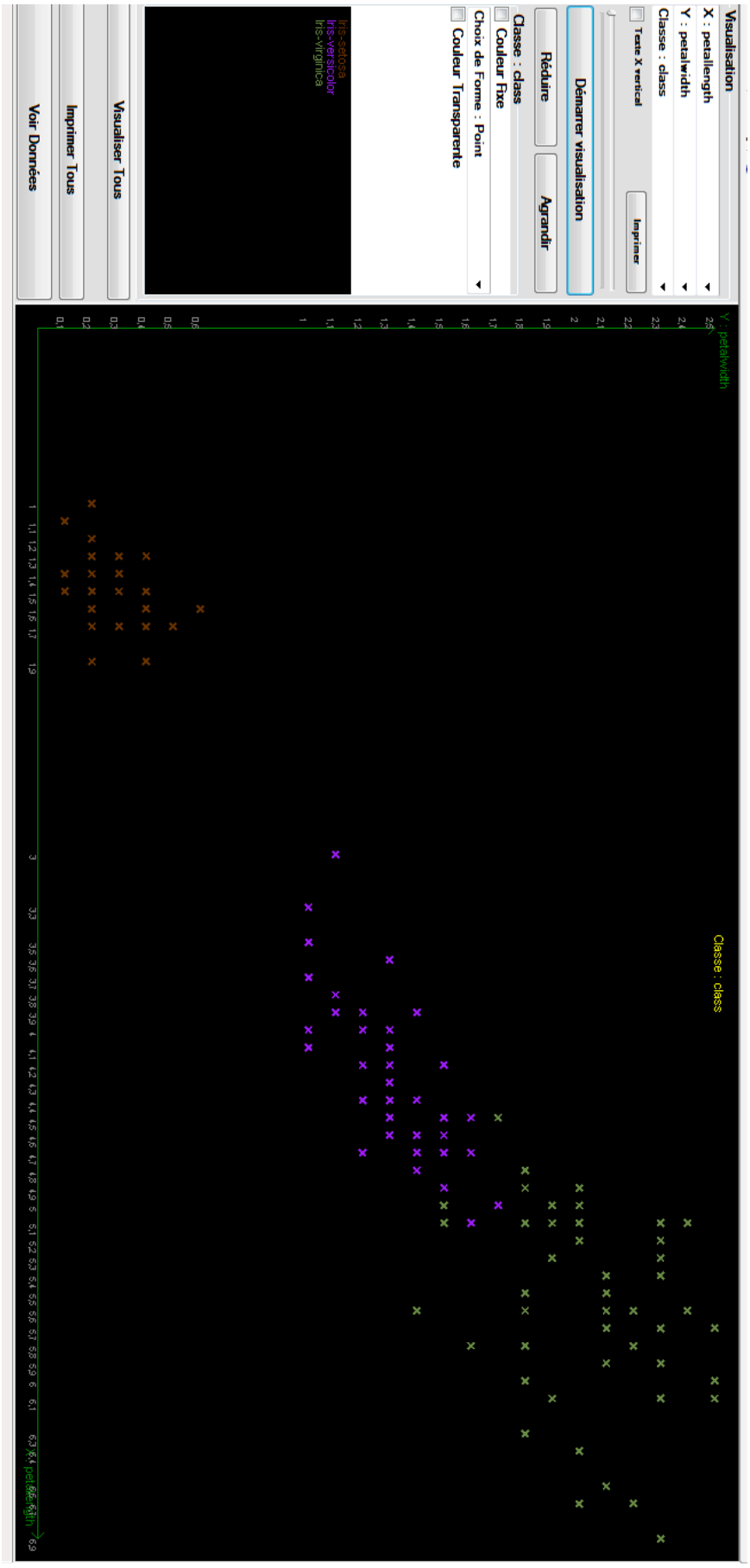
Annexe 5 : Histogrammes en bâtons en 2 et 3 dimensions représentants les nombres d'occurrences de chaque valeur de longueur et largeur des pétales des IRIS, ainsi que les éléments des classes : Setosa, Versicolor et virginica.

Annexe 6 : phase initiale de traitement et typage de données des IRIS.

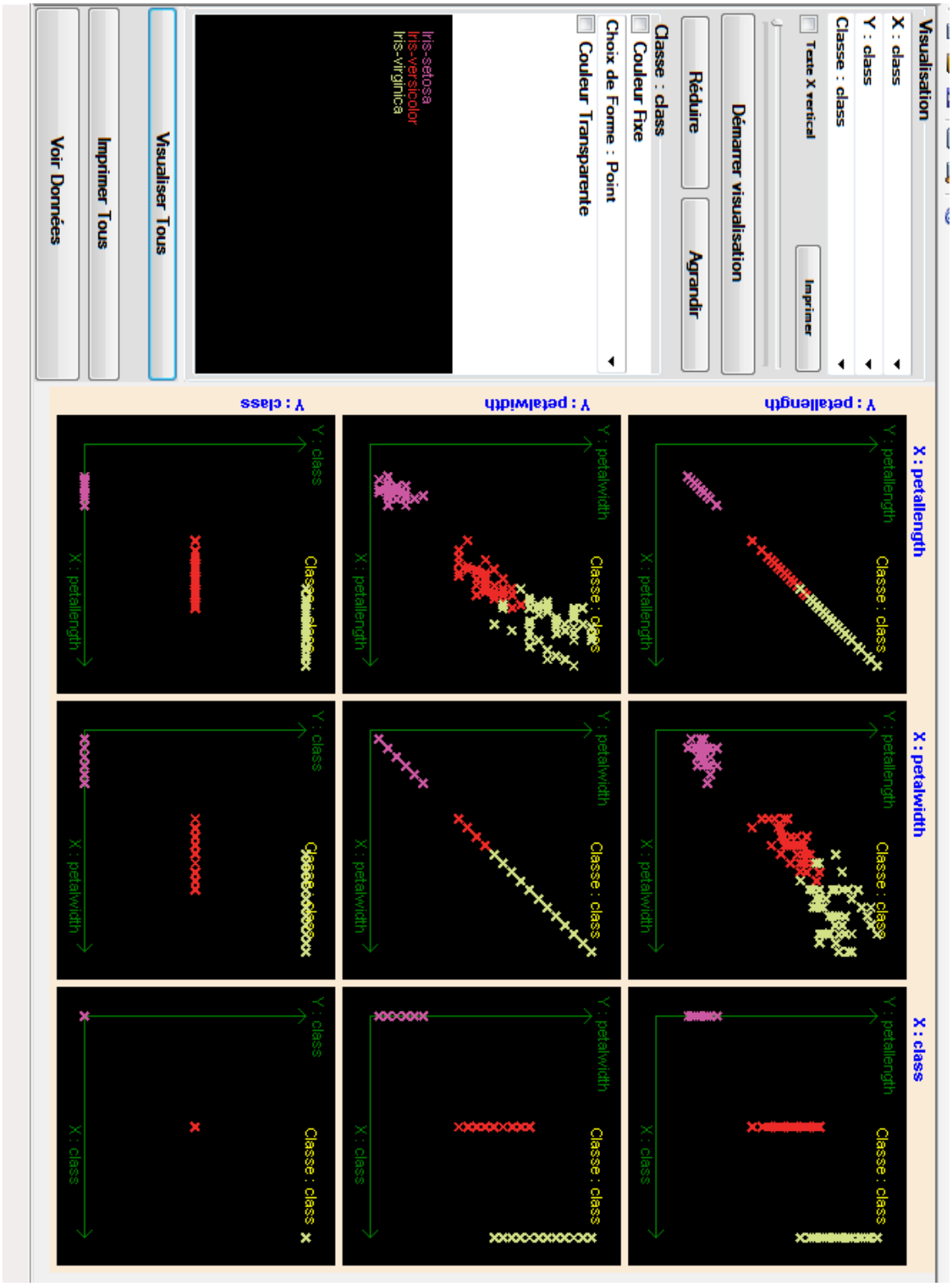


Annexe 7 : phase finale de traitement et typage de données des IRIS et génération du fichier de données final.

Annexe 8 : phase de visualisation finale de données d'analyse d'IRIS après leur discrétisation.



Annexe 9 : phase de la visualisation détaillée des classes d'IRIS



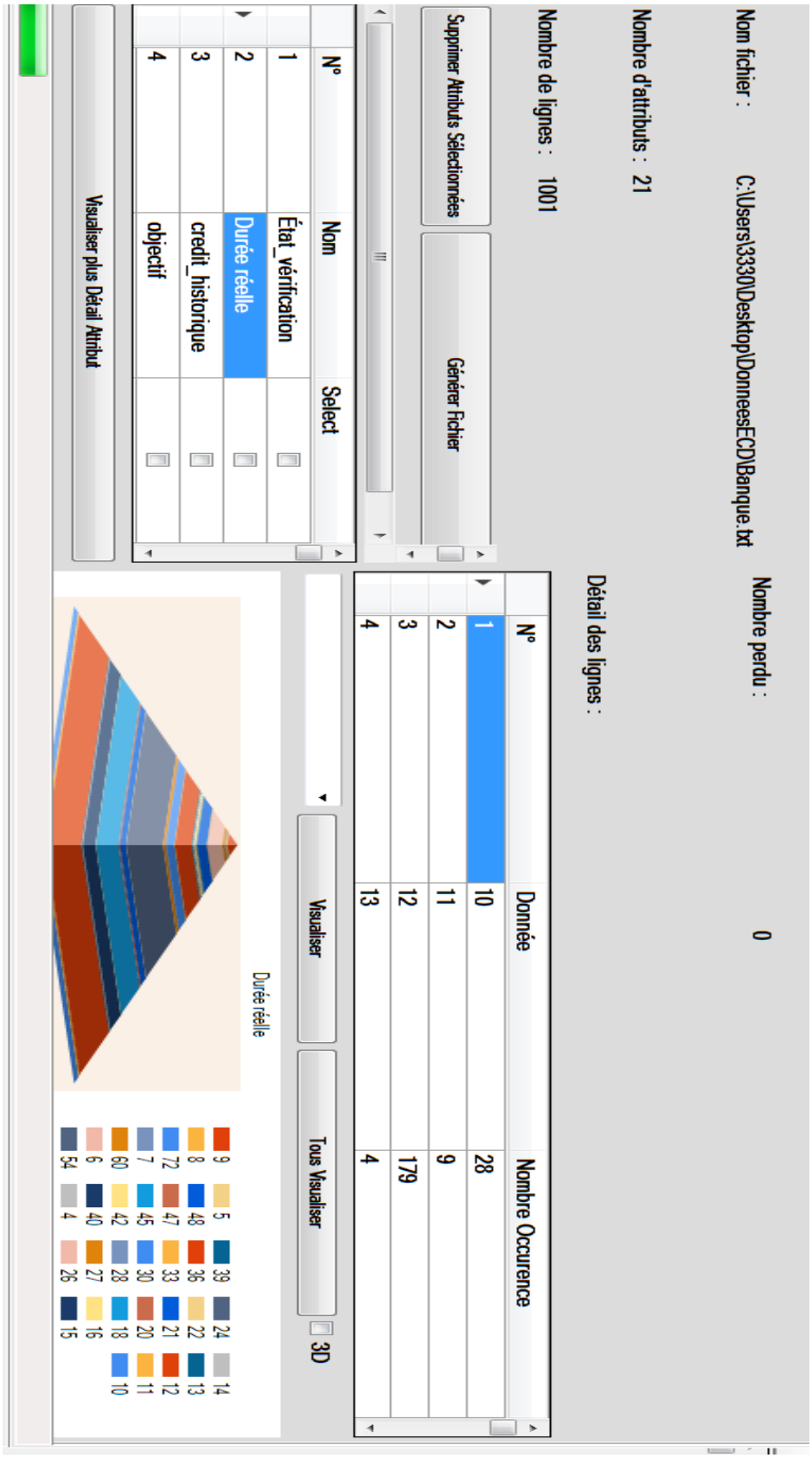
Annexe 10 : phase de la visualisation globale des classes d'IRIS

B. Etapes de traitement et de visualisation des données de la base de données :

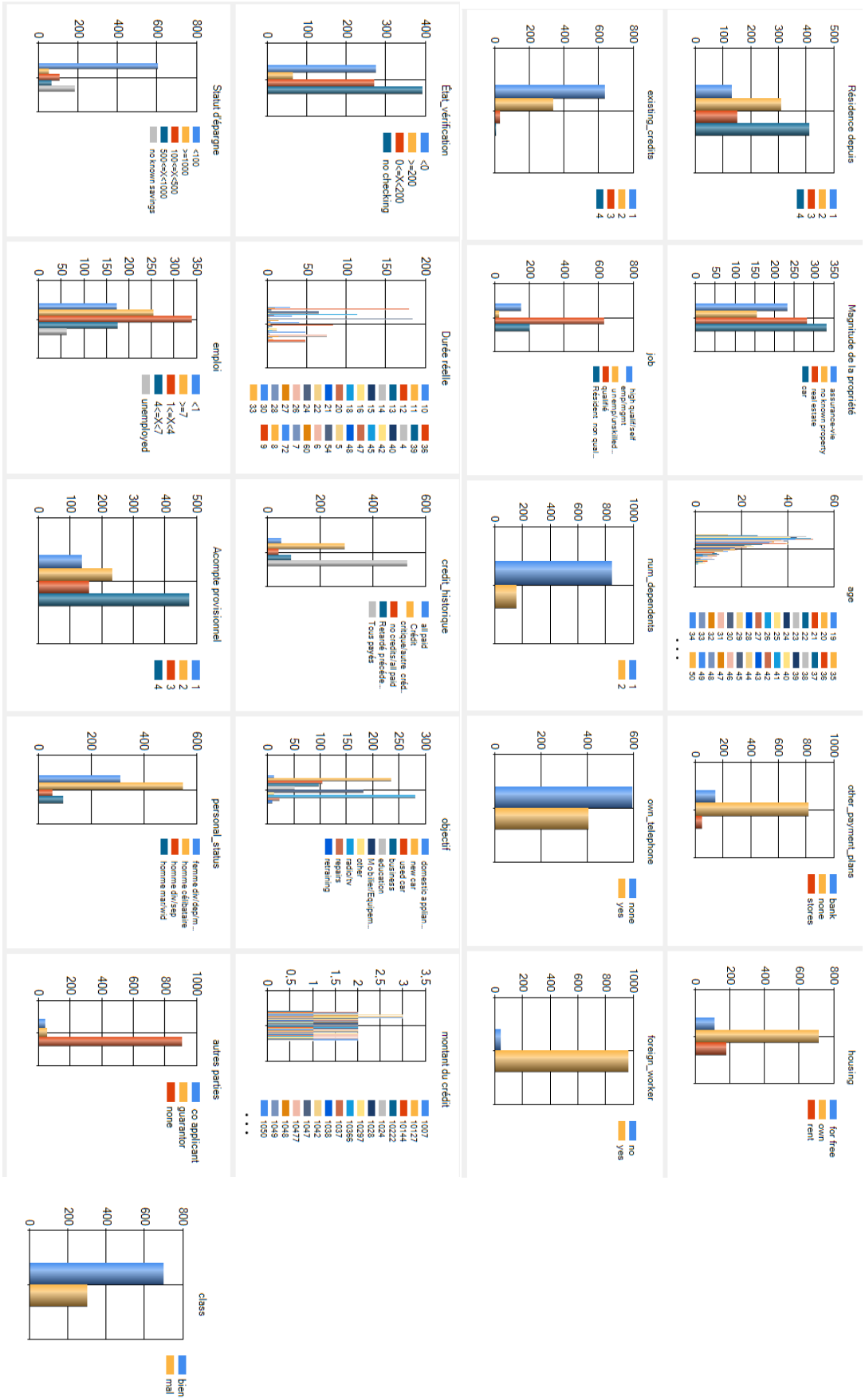
« Crédit banque »

« De l’annexe 11 à l’annexe 19 »

Annexe 11 : Chargement initial de la base de données « crédit banque »avant le traitement



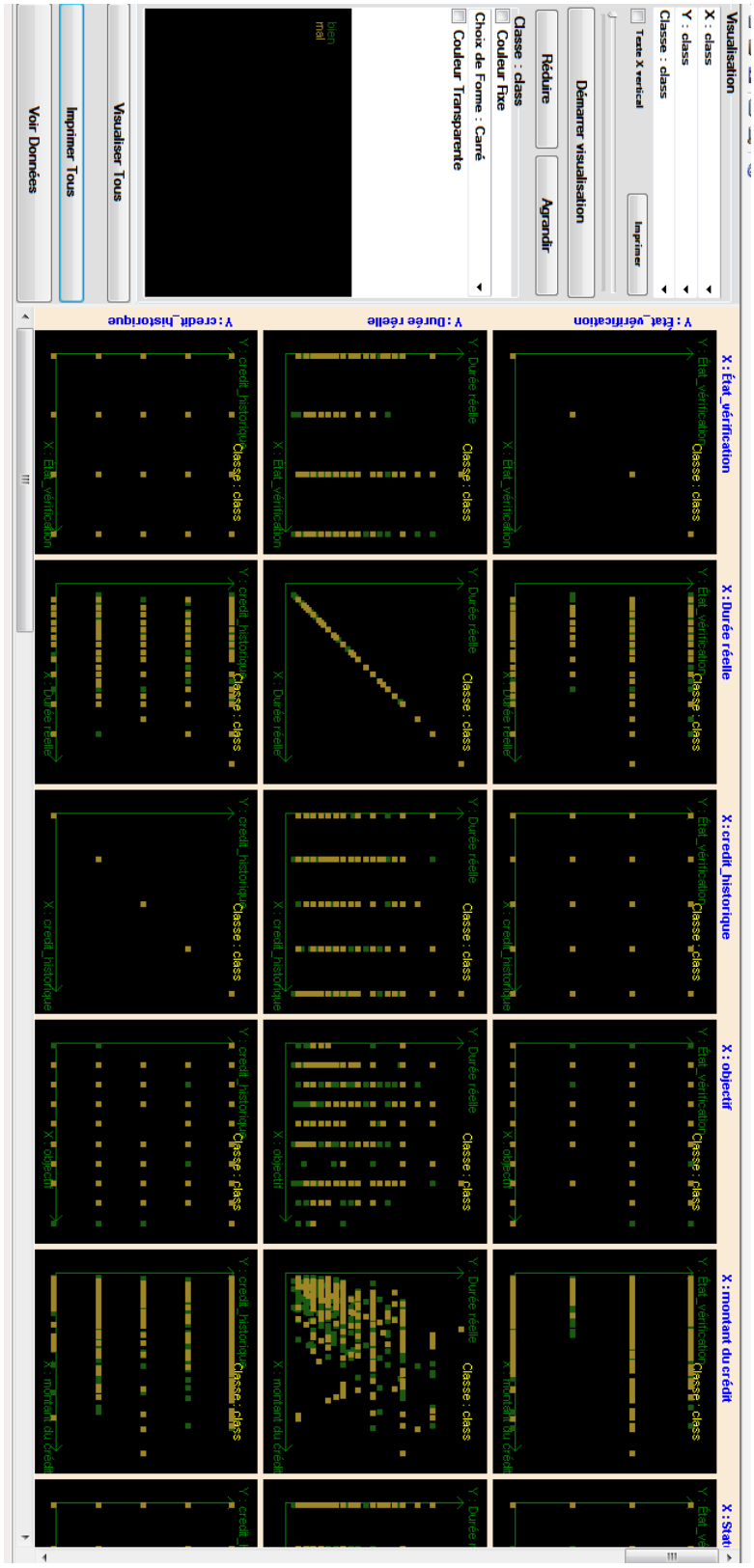
Annexe 12 : phase de prétraitement initial de la base de données « crédit banque »



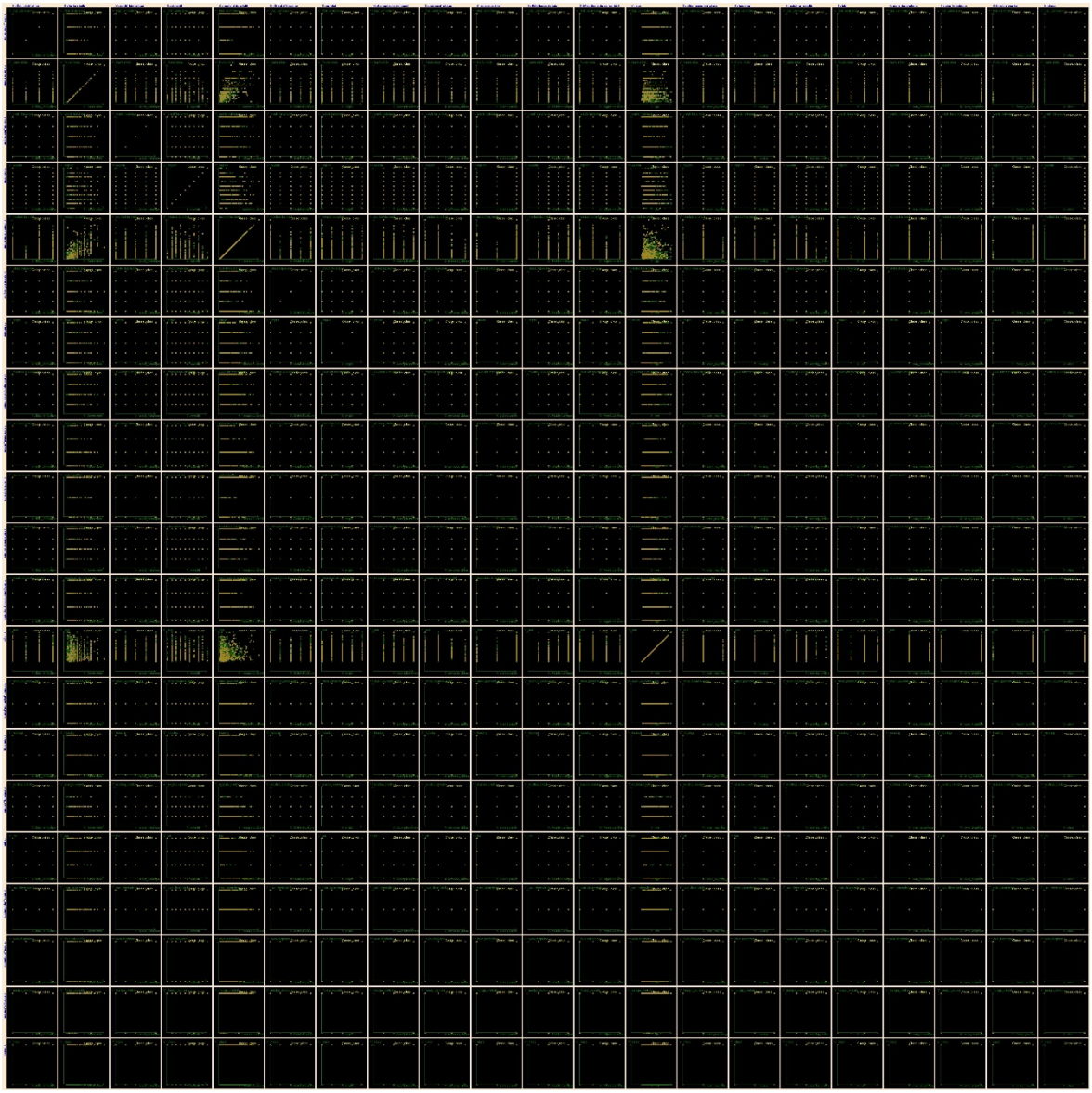
Annexe 13 : Visualisation globale des histogrammes des données de prétraitement initial de la base de données « Crédit banque »

Annexe 14 : Visualisation finale des données après traitement final dans deux tables.

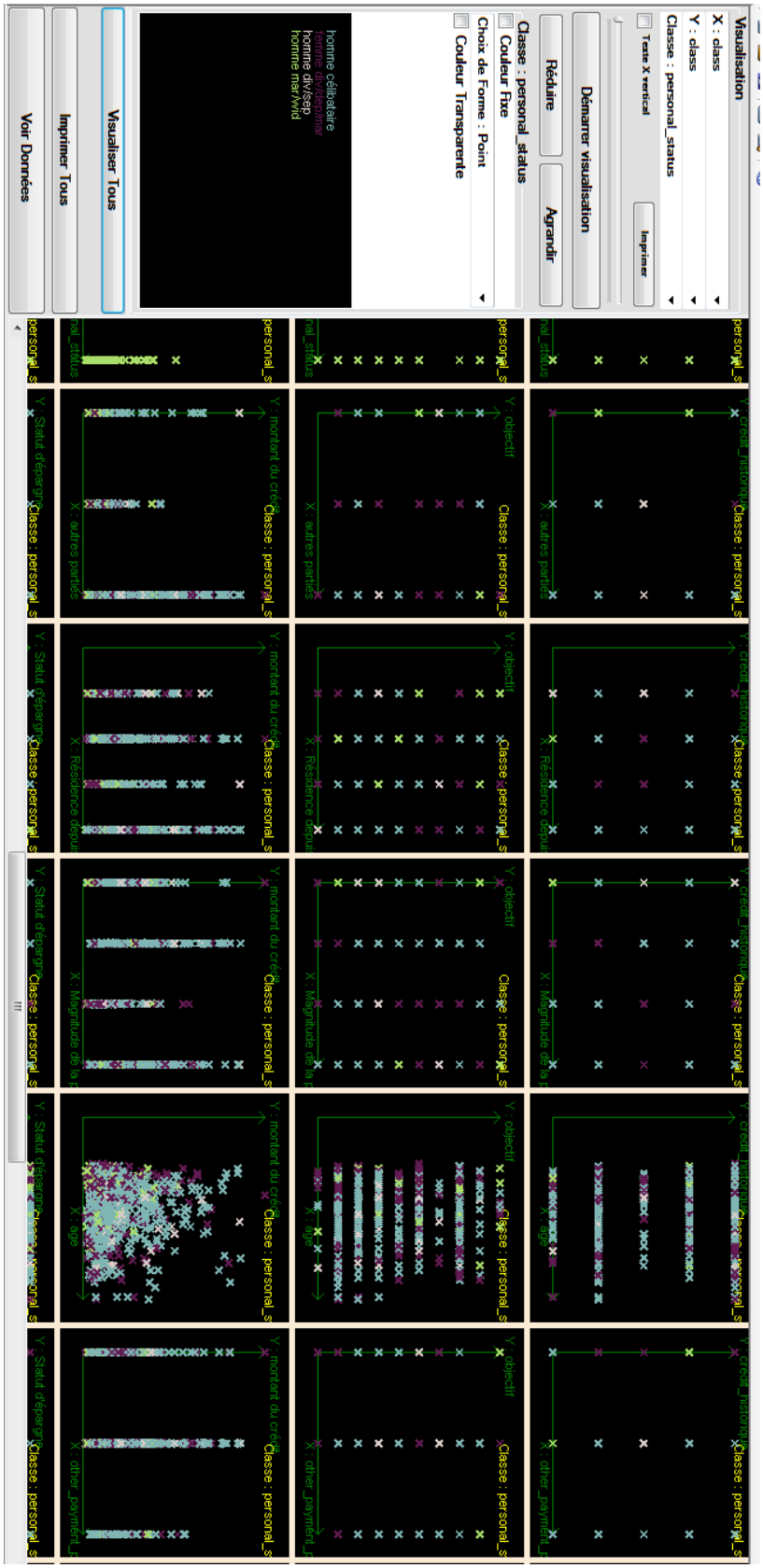
Annexe 15 : phase de la visualisation détaillée des classes de demandeurs de crédit en fonction de la durée réelle et de leurs âges.



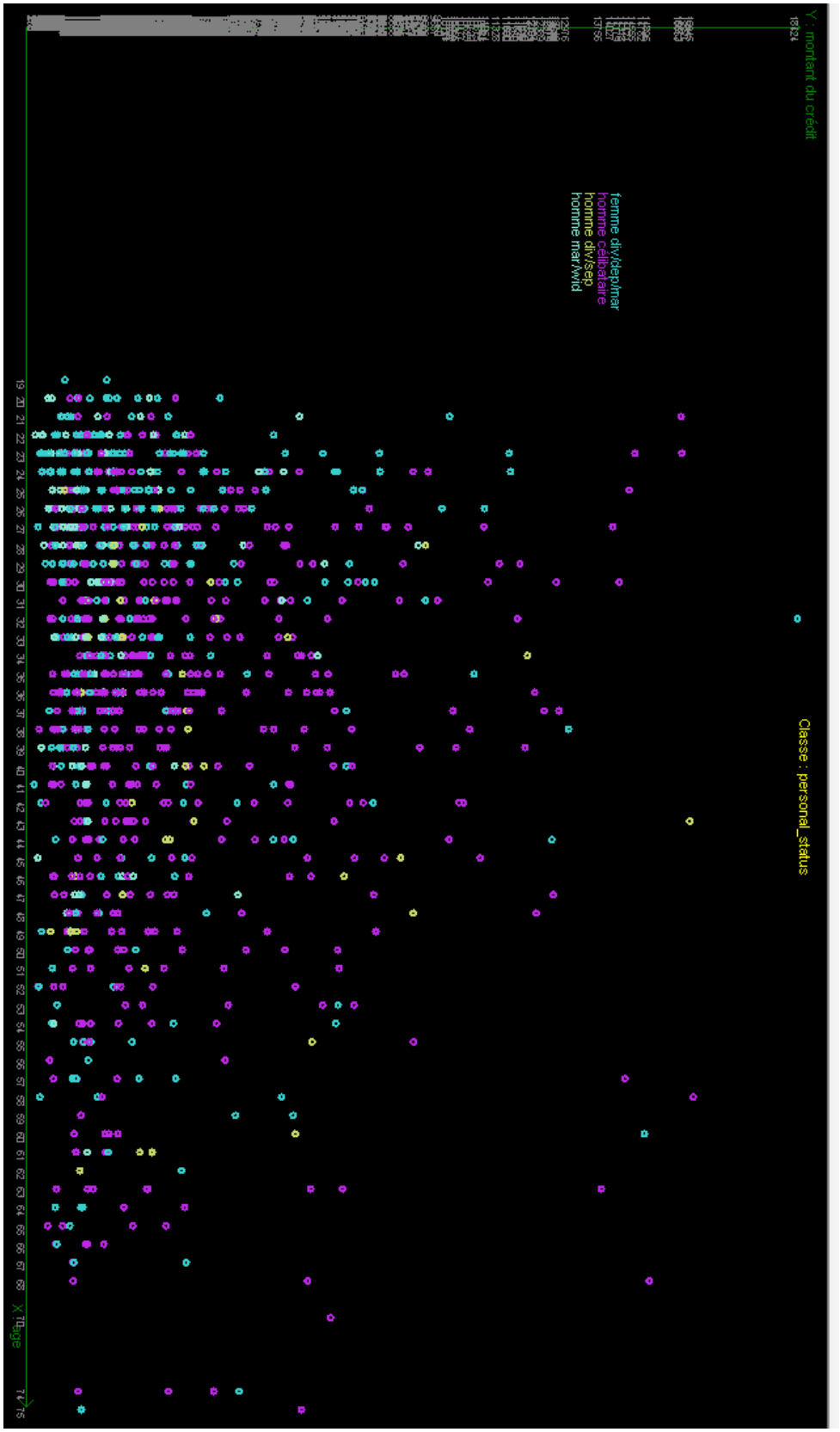
Annexe 16 : phase de la visualisation globale des deux classes de demandeurs de crédit « ceux qui règlent bien ou mal leurs crédits » en fonction de l'ensemble d'autres attributs



Annexe 17 : vue du fichier image créé par l'application ECD des deux classes de demandeurs de crédit « ceux qui règlent bien ou mal leurs crédits » en fonction de l'ensemble d'autres attributs



Annexe 18 : phase de la visualisation globale des deux classes de demandeurs de crédit selon leurs états martiaux «homme célibataire, marié … » en fonction de l'ensemble d'autres attributs



Annexe 19 : vue des statuts personnels en fonction de l'âge et du montant de crédit

Table des matières