TABLE DES MATIERES

NOTATIONS ET ABREVIATIONS

1. Minuscules latines

a	Nombre de niveaux de gris d'une image
b	Biais
b^*	Biais optimal
d	Mesure de dissimilarité
f	Classifieur ou fonction de décision
g	Valeur de l'intensité d'un pixel d'une image
h	VC-dimension
k	Nombre de clusters désirés
m	Dimension des exemples d'apprentissage
n	Taille des exemples d'apprentissage
r	Fonction de régression de risque minimal
t	Nombre des clusters
u_a	Degré d'appartenance de l'objet au cluster C_a
$\vec{w}$	Vecteur normal à l'hyperplan
$\vec{w}^*$	Vecteur normal à l'hyperplan optimal
w_i	Etiquette de la classe
x_i	Exemples d'apprentissage
x^*	Centroïde
y	Réponse fournie par le classifieur

2. Majuscules latines

C	Cluster, partition ou classe
D	Matrice de dissimilarité
F	Dimension supérieure à R^m
I	Image à traiter
Ke	Fonction noyau
L	Lagrangien
Lo	Fonction de perte

M	Nombre quelconque de classifieurs
M_i	Moyenne des points de S
N_{w_i}	Nombre d'éléments dans la classe i
P	Matrice de dépendance des niveaux de gris
P_g	Fréquence de l'intensité g dans une région de l'image
Q	Paramètre de régularisation
R	Région d'une image
S	Ensemble de toutes les partitions possibles des pixels en k ensembles
T	Tâche d'apprentissage
V	Ensemble de variables descriptives
X	Ensemble de prédiction
Y	Ensemble à prédire

3. Minuscules grecques

ε	Fonctionnelle
$\varepsilon_{i,j}$	Score
η	Probabilité $\epsilon[0; 1]$
θ	Angle de voisinage
λ^*	Solution du POQ pour le SVM linéaire
λ_i	Multiplicateurs de Lagrange
μ_i	Multiplicateurs de Lagrange
ξ_i	Variables souples
ψ	Ensemble de parties non vide

4. Majuscules grecques

Θ	Ordre
Π	Performances mesurées sur T
Φ	Fonction transférant les données de R^m à F
Ω	Base d'apprentissage

5. Abréviations

ANN	Artificial Neural Network
CRISP-DM	Cross Industry Standard Process for Data Mining
DM	Data Mining
DSC	Débit Sanguin Cérébral
DT	Decision Tree
ER	Entité-Relation
ERM	Empirical Risk Minimization
FN	False Negative
FP	False Positive
GLCM	Gray Level Cooccurrence Matrix
HSO	Hyperplan Séparateur Optimal
HTML	HyperText Markup Language
IP	Internet Protocol
KDD	Knowledge Discovery in Databases
KKT	Karush-Kuhn-Tucker
KNN	K-Nearest Neighbours
LBP	Local Binary Pattern
MA	Maladie d'Alzheimer
MATLAB	MATrix LABoratory
ML	Machine Learning
POQ	Problème d'Optimisation Quadratique
RAM	Random Access Memory
RBF	Radial Basis Function
ROC	Receiver Operator Characteristic

SGBDR	Systèmes de Gestion de Base de Données Relationnelles
SIFT	Scale Invariant Feature Transform
SPOT	Système Probatoire d'Observation de la Terre
SQL	Structured Query Language
SRM	Structural Risk Minimization
SVM	Séparateur à Vaste Marge
TEMP	Tomographie d’Emission Mono Photonique

INTRODUCTION GENERALE

Le Big Data est une expression que nous entendons de plus en plus dans le monde numérique. En effet, l'évolution des moyens d'acquisition, de partage et de stockage d'information engendre un volume extrêmement important de données. Big Data est donc le terme utilisé pour décrire ce phénomène. Mais en plus d'être volumineuses, les données sont aussi de plusieurs variétés, citons : les données texte, image, son et vidéo.

L'idée est alors apparue de ne pas juste abandonner ces données et les laisser « mourir » dans les dispositifs de stockage mais de les analyser pour en extraire de nouvelles informations. Ce processus conduisant un volume massif de données brutes jusqu'à la connaissance est appelé Data Mining. Cette technique permet l'extraction d'informations pertinentes menant à une décision.

Les entreprises traitant des données images n'échappent pas à cette explosion de données. Grâce à la diversité des moyens d'acquisition d'images, et la facilité de partage, on peut avoir jusqu'à des Téraoctets d'images dans une base. Pratiquer le Data Mining dans cette dernière s'avère nécessaire, en ayant recours à un analyste d'image.

Toutefois, l'énorme quantité d'images disponibles dans la base établit une limite à la capacité humaine pour analyser efficacement les données. Pour en finir avec les tâches manuelles, répétitives et l'analyse fondée sur l'intuition, de quelle manière peut-on alors faciliter l'extraction d'information dans les grandes bases de données images ?

La tendance consiste aujourd'hui à automatiser certains procédés pour faciliter l'analyse de données. Les objectifs sont la construction et l'optimisation de modèles prédictifs. C'est dans ce sens que s'oriente le sujet de ce mémoire intitulé : « Outil de classification d'images par méthode d'apprentissage automatique ».

Pour exposer en détail ce thème, cet ouvrage est scindé en quatre chapitres : après avoir présenté un état de l'art sur ce qu'on entend par Data Mining dans un premier chapitre, nous développerons les différentes techniques d'apprentissage automatique dans le chapitre suivant, ensuite nous allons modéliser un système de classification automatique d'image dans le troisième chapitre ; et dans le dernier chapitre, nous allons utiliser le système précédent pour concevoir un outil de classification d'images en prenant des images médicales comme application.

CHAPITRE 1
ETAT DE L'ART DU DATA MINING

1.1 Introduction

L'une des préoccupations des organismes et entreprises qui traitent des volumes massifs de données est le stockage même de ces données. Cependant, l'évolution des technologies grâce aux moyens matériels et logiciels, permet de résoudre le problème. En effet, la tendance actuelle consiste non seulement à stocker les données mais aussi à extraire les informations enfouies dans ces données. Cette extraction d'information est connue sous le nom de Data Mining (DM). Ce chapitre met un accent sur ses origines ; les données que l'on peut « miner » ; les techniques de DM existantes ; ainsi que ses domaines d'applications et enfin la méthodologie à adopter pour l'élaboration d'un projet de DM.

1.2 Origines du Data Mining

Le DM peut être vu comme la conséquence de l'évolution des systèmes de traitement de l'information et plus particulièrement des Systèmes de Bases de Données.

L'industrie des Systèmes de Bases de Données a connu une évolution dans le développement des fonctionnalités suivantes (voir figure 1.01) : la collecte de données et la création de bases de données, la gestion des données et l'analyse avancée des données.

Depuis les années 1960, les bases de données et les technologies de l'information se sont systématiquement transformées en partant des systèmes de traitement de fichiers primitifs vers des Systèmes de Bases de Données sophistiqués et puissants. Les progrès constants et étonnants de la technologie informatique a conduit à de grandes quantités d'ordinateurs puissants et abordables, de matériel de collecte de données et de supports de stockage. Les données peuvent être stockées dans différents types de bases de données et de dépôt d'informations. Une architecture de dépôt de données s'est alors manifestée : le Data Warehouse. C'est un entrepôt de données hétérogènes organisées sous un schéma unifié sur un seul site afin de faciliter la prise de décision.

La quantité croissante et rapide de données, collectées et stockées dans de grands et nombreux Data Warehouses, a largement dépassé notre capacité humaine de compréhension sans outils puissants. L'abondance des données, associée à la nécessité d'outils puissants d'analyse des données, a été décrite comme une situation riche en données mais pauvre en informations.

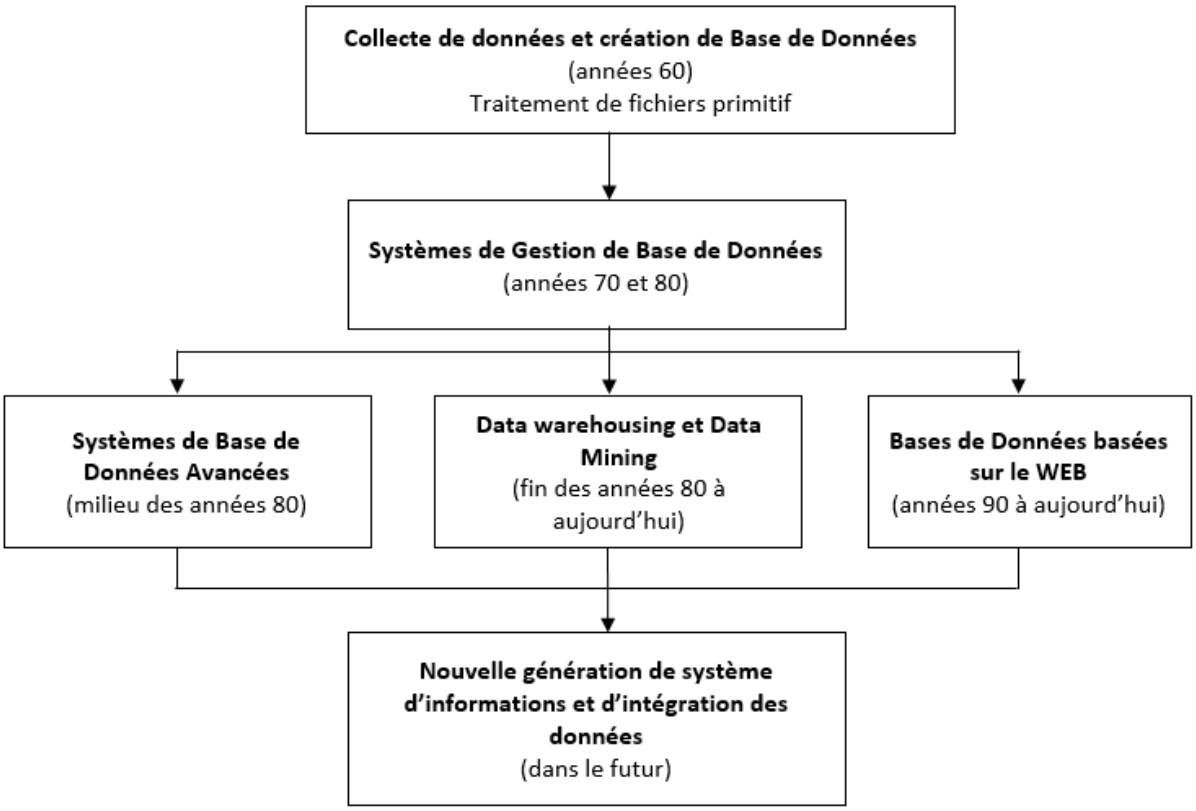


Figure 1.01 : *Evolution des systèmes de Bases de Données*

En conséquence, les données recueillies dans les grands dépôts de données deviennent des « tombes de données », des archives qui sont rarement visitées. Les décisions importantes sont souvent prises sur l'intuition d'un décideur, simplement parce que le décideur n'a pas les outils pour extraire les connaissances précieuses intégrées dans les vastes quantités de données. L'écart grandissant entre les données et les informations requiert un développement systématique d'outils de DM qui transformeront les tombeaux de données en « pépites d'or » du savoir. Le domaine du Data Mining a ainsi émergé dans les années 1990 à l'issu du premier workshop KDD (Knowledge Discovery in Databases) en 1989. [1][2]

1.3 Définitions

Le Data Mining, souvent traduit en français par « fouille de données », est une technique permettant l'extraction d'information d'intérêt (non triviale, implicite, inconnue à priori et potentiellement utile) à partir de données stockées dans de larges entrepôts de données, en utilisant des procédures automatiques ou semi-automatiques pour une prise de décision. Comme ce processus peut être très difficile, il est souvent comparé au minage de l'or dans les rivières : le gravier des alluvions

représente l'énorme quantité de données et les pépites d'or représentent les connaissances cachées que l'on veut trouver.

Le DM est un domaine qui se situe à l'intersection des statistiques, de l'apprentissage automatique et des bases de données (illustré par la figure 1.02). [3][4]

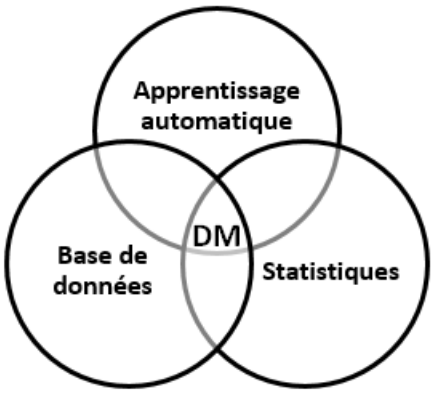


Figure 1.02 : *Domaine du Data Mining*

1.4 Terminologies sur le Data Mining

Pour une bonne compréhension sur ce qui suit, il est nécessaire de définir les termes suivants :

- Concept

Un concept désigne un problème à résoudre, un phénomène à prédire ou un objectif à atteindre à partir des données et des techniques de DM.

- Dataset

Un dataset fait référence aux données utilisées pour le DM représentées sous forme de table. Un exemple est mis en évidence dans le tableau 1.01.

Chaque ligne du dataset correspond à un événement tandis qu'une colonne désigne un attribut. Un événement est donc un vecteur composé de différents attributs. Ainsi, un dataset est une matrice dont les lignes sont les événements et les colonnes les attributs. Les attributs peuvent être sous forme nominale ou numérique.

- Classe

La classe désigne un attribut de la dataset dont la valeur est calculée ou conditionnée par la valeur des autres attributs. Elle représente l'attribut à prédire. Chaque événement peut avoir une classe qui lui est associée. Elle est souvent placée à la dernière colonne d'un dataset illustré en gris dans le tableau 1.01. [5]

Temps (*Nominal*)	**Température** (*Numérique*)	**Humidité** *(Numérique)*	**Vent** *(Nominal)*	**Jouer** *(Nominal)*
Ensoleillé	85	85	Faible	Non
Ensoleillé	80	90	Fort	Non
Couvert	83	86	Faible	Oui
Pluvieux	70	96	Faible	Oui
Pluvieux	68	80	Faible	Oui
Pluvieux	65	70	Fort	Non
Couvert	64	65	Fort	Oui
Ensoleillé	72	95	Faible	Non
Ensoleillé	69	70	Faible	Oui
Pluvieux	75	80	Fort	Oui
Ensoleillé	75	70	Fort	Oui
Couvert	72	90	Fort	Oui
Couvert	81	75	Faible	Oui
Pluvieux	71	91	Fort	Non

Tableau 1.01: *Exemple de dataset*

1.5 Type de données à explorer

La nature des données pouvant être utilisées pour le DM est variée. Cette section dresse une liste de ces types de données à explorer avec leur description.

1.5.1 Base de données relationnelle

Une base de données relationnelle est une base de données où l'information est organisée dans des tableaux à deux dimensions appelés relations ou tables. Selon ce modèle relationnel, une base de données consiste en une ou plusieurs relations. Les lignes de ces relations sont appelées des enregistrements tandis que les colonnes sont appelées attributs.

Chaque enregistrement dans une table représente un objet identifié par une clé unique et décrit par un ensemble de valeurs d'attribut. Un modèle de données sémantiques, tel qu'un modèle de données entité-relation (ER), est souvent construit pour des bases de données relationnelles. Un modèle de données ER représente la base de données comme un ensemble d'entités et leurs relations.

Les logiciels qui permettent de créer, utiliser et maintenir des bases de données relationnelles sont des Systèmes de Gestion de Base de Données Relationnelles ou SGBDR. [1]

1.5.2 Data Warehouse

Un Data Warehouse est un entrepôt de données collectées à partir de sources multiples, stockées sous un schéma unifié et qui résident habituellement sur un site unique. Les entrepôts de données

sont construits par un processus de nettoyage des données, d'intégration des données, de transformation des données, de chargement des données et de rafraîchissement périodique des données. Un exemple illustre ce concept sur la figure 1.03.

La structure physique réelle d'un Data Warehouse peut être une mémoire de données relationnelle ou un cube de données multidimensionnel. Un cube de données fournit une vue multidimensionnelle des données et permet l'accès rapide aux données déjà résumées. [1]

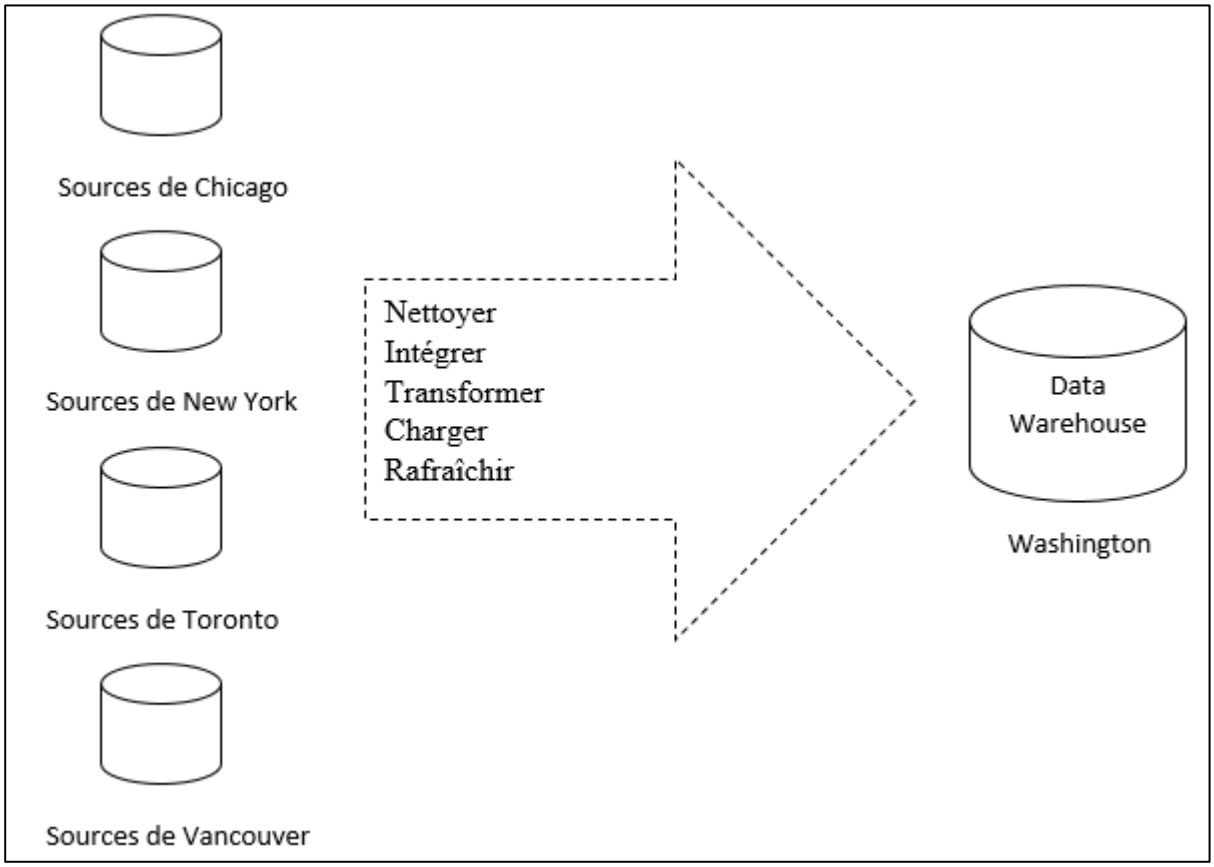


Figure 1.03 : *Data Warehouse d'une entreprise aux Etats-Unis*

1.5.3 Base de données objet-relationnelles

Les bases de données objet-relationnelles sont construites sur la base d'un modèle objet-relationnel. Ce modèle étend le modèle relationnel en fournissant un type de données riche pour gérer des objets complexes et l'orienté objet. Conceptuellement, le modèle objet-relationnel hérite des concepts essentiels des bases de données orientées objet, où, en termes généraux, chaque entité est considérée comme un objet. Les données et le code relatifs à un objet sont encapsulés dans une seule unité.

Chaque objet est associé à ce qui suit :

- un ensemble de variables qui décrivent les objets. ceux-ci correspondent aux attributs dans les modèles relationnels y compris le modèle ER ;
- un ensemble de messages que l'objet peut utiliser pour communiquer avec d'autres objets ou avec le reste du système de base de données ;
- un ensemble de méthodes, où chaque méthode détient le code pour implémenter un message. Lors de la réception d'un message, la méthode renvoie une valeur en réponse.

Pour le DM dans des systèmes objet-relationnels, des techniques doivent être développées pour gérer des structures d'objets complexes, des types de données complexes, des hiérarchies de classes et de sous-classes, l'héritage de propriétés, de méthodes et de procédures. [1]

1.5.4 Bases de données spatiales et spatio-temporelles

1.5.4.1 Base de données spatiale

Une base de données spatiale est une base de données optimisée pour stocker et requêter des données reliées à des objets référencés géographiquement y compris des points, des lignes et des polygones. Il existe une multitude d'exemples comprenant : des bases de données géographiques (carte), des bases de données dont la conception est assistée par ordinateur, ainsi que des bases de données d'imagerie médicales et d'images satellitaires.

Les données spatiales peuvent être représentées en format raster, consistant en des cartes binaires (bit maps) ou des cartes de pixels (pixel maps) à n dimensions. Par exemple, une image satellite 2D peut être représentée sous forme de données raster, chaque pixel enregistrant les précipitations dans une zone donnée.

Les relations entre un ensemble d'objets spatiaux peuvent être examinées par le DM afin de découvrir quels sous-ensembles d'objets sont spatialement auto-corrélés ou associés. De plus, la classification spatiale peut être effectuée pour construire des modèles de prédiction basés sur l'ensemble pertinent de caractéristiques des objets spatiaux. [1]

1.5.4.2 Base de données spatio-temporelle

Une base de données spatiale stockant des objets spatiaux qui changent avec le temps est appelée une base de données spatiotemporelle, à partir de laquelle des informations intéressantes peuvent être extraites. Par exemple, nous pourrions être en mesure de distinguer une attaque bioterroriste basée sur la propagation géographique d'une maladie avec le temps. [1]

1.5.5 Bases de données temporelle, séquentielle et série-chronologique

- Une base de données temporelle stocke généralement des données relationnelles qui incluent des attributs temporels. Ces attributs peuvent impliquer plusieurs horodatages (associations d'une heure et d'une date à un événement), chacun ayant une sémantique différente.
- Une base de données séquentielle stocke des séquences d'événements ordonnés, avec ou sans notion concrète de temps. Voici quelques exemples : les séquences d'achats des clients, les flux de clics Web et les séquences biologiques.
- Une base de données de série-chronologique stocke des séquences de valeurs ou d'événements obtenus sur des mesures répétées de temps (horaire, quotidien, hebdomadaire). L'exemple concret que l'on peut citer est l'observation de phénomènes naturels comme la température et le vent.

Les techniques de DM peuvent être utilisées pour trouver les caractéristiques de l'évolution des objets, ou la tendance à changer des objets dans la base de données. Ces informations peuvent être utiles dans la prise de décision et la planification de stratégie. [1]

1.5.6 Bases de données textuelles et multimédias

1.5.6.1 Base de données textuelles

Les bases de données textuelles sont des bases de données contenant des mots pour décrire les objets. Ces descriptions ne sont généralement pas de simples mots-clés mais des phrases ou des paragraphes assez longs, comme des rapports d'erreurs ou de bug, des messages d'avertissement, des notes ou d'autres documents. Les bases de données textuelles peuvent être fortement déstructurées (comme certaines pages Web sur le World Wide Web). Certaines peuvent être quelque peu structurées, c'est-à-dire semi-structurées telles que des messages électroniques et de nombreuses pages Web HTML (HyperText Markup Language) / XML (eXtensible Markup Language). Tandis que d'autres sont relativement bien structurées comme les bases de données de catalogues de bibliothèques.

En effectuant le DM sur les données textuelles (pratique appelée « text mining »), on peut découvrir des descriptions générales et concises des documents, des associations de mots clés ou de contenu. Pour ce faire, les méthodes standard de DM doivent être intégrées aux techniques de récupération d'information et à la construction ou l'utilisation d'hiérarchies spécifiques pour les données textuelles. [1]

1.5.6.2 Base de données multimédias

Les bases de données multimédias stockent des données image, audio et vidéo. Ces bases doivent prendre en charge les données à volume massif, car les données telles que la vidéo peuvent nécessiter des giga-octets de stockage. Des techniques spécialisées de stockage sont également requises, parce que les données vidéo et audio sont généralement récupérées en temps réel avec un débit constant et prédéterminé afin d'éviter les bruits d'image ou de son et les débordements de mémoire tampon du système.

Pour l'exploration de données multimédias, les techniques de stockage doivent être intégrées aux méthodes standard d'exploration de données. Les approches prometteuses incluent la construction de cubes de données multimédias, l'extraction de multiples fonctionnalités à partir de données multimédias et l'appariement de modèles basé sur la similarité. [1]

1.5.7 Le World Wide Web

Le « Web mining » est l'appellation donnée à l'application des techniques du DM aux données du Web (documents, structure des pages, des liens...). Il s'est développé à la fin des années 90 afin d'extraire des informations pertinentes sur l'activité des internautes sur le Web.

On peut classer les données utilisées dans le Web mining en quatre types :

- Données relatives au contenu : données contenues dans les pages Web (textes, graphiques),
- Données relatives à la structure : données décrivant l'organisation du contenu (structure de la page, structure inter-page),
- Données relatives à l'usage : données fournissant des informations sur l'usage telles que les adresses IP (Internet Protocol), la date et le temps des requêtes,
- Données relatives au profil de l'utilisateur : données fournissant des informations démographiques sur les utilisateurs du site Web.

Ces données sont généralement stockées dans un Data Warehouse, appelé Data Webhouse, dont l'objectif de construction est de collecter des données propres à la fréquentation des sites Web afin d'analyser les comportements de navigation. [6]

1.6 Applications du Data Mining

D'après les sections précédentes, les types de données que le DM peut exploiter sont nombreuses. Par conséquent, il existe de multiples domaines d'application du DM qui varient de la finance, au

marketing, à la médecine et à la télécommunication. Une liste non exhaustive de ces applications est représentée sur la figure 1.04 avec le secteur auquel elles sont rattachées. [3]

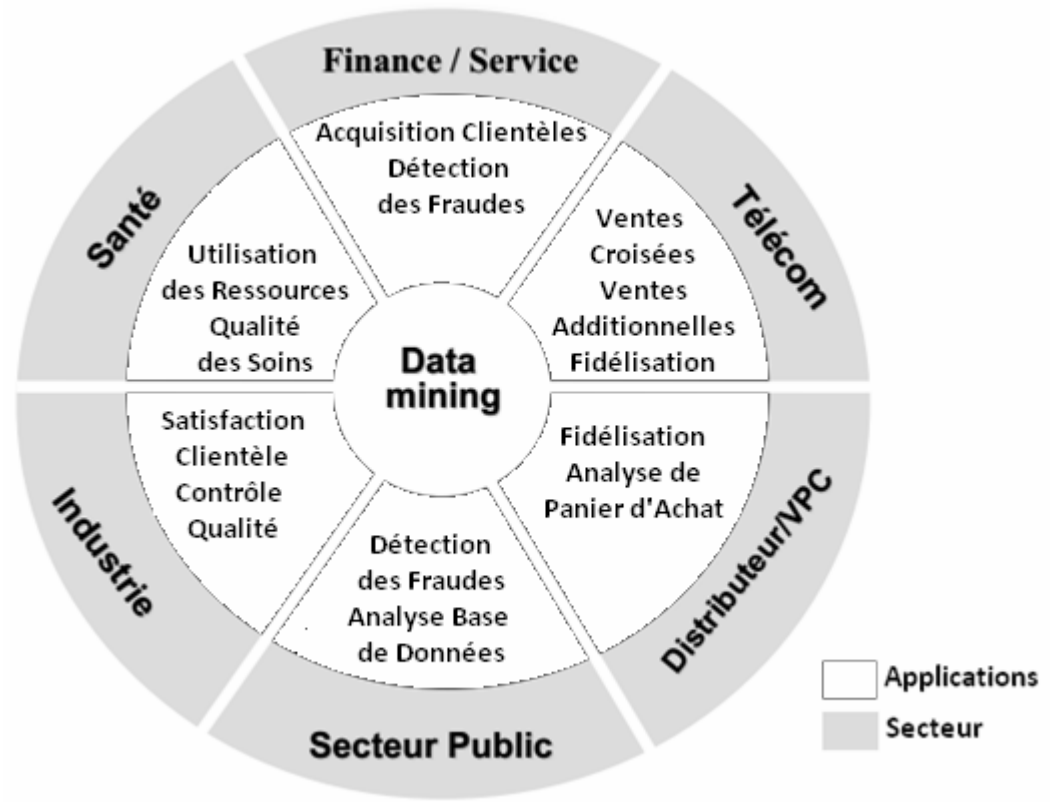


Figure 1.04 : *Applications du Data Mining*

1.7 Techniques de Data Mining

Les différentes techniques de DM peuvent être divisées en deux : l'analyse prédictive ou supervisée et l'analyse descriptive ou non supervisée. De plus amples informations à ce sujet seront détaillées dans le chapitre 2.

1.7.1 Analyse prédictive

La plupart des applications de DM ont pour but la prédiction. Elle consiste à calculer ou à prédire la classe d'un événement non contenu dans le dataset. L'objectif des techniques supervisées est d'apprendre, à l'aide d'un ensemble d'entraînement, des règles qui permettent de prédire (ou « deviner ») certaines caractéristiques de nouvelles observations. Le principe est que, dans tous les cas, on utilise des données connues pour construire un modèle. Ce modèle est ensuite utilisé dans le but de classer les nouvelles observations. [7][8]

1.7.2 Analyse descriptive

Dans le cas de la description, le but est de trouver de la connaissance à partir des données brutes, sans connaître la sortie. Elle consiste à fournir une description globale d'un dataset sans nécessairement construire un modèle permettant de déterminer la classe d'un nouvel événement. L'analyse descriptive permet de trouver les relations logiques ou mathématiques qui existent entre les attributs d'un dataset et aussi identifier les similarités entre les événements. [7][8]

1.8 Elaboration d'un projet de Data Mining

Une méthodologie standard a été définie, en 1996, par la Cross Industry Standard Process for Data Mining (CRISP-DM) pour l'élaboration d'un projet de DM. Selon ce modèle, un projet typique de DM devrait être effectué en six étapes indiquées sur la figure 1.05.

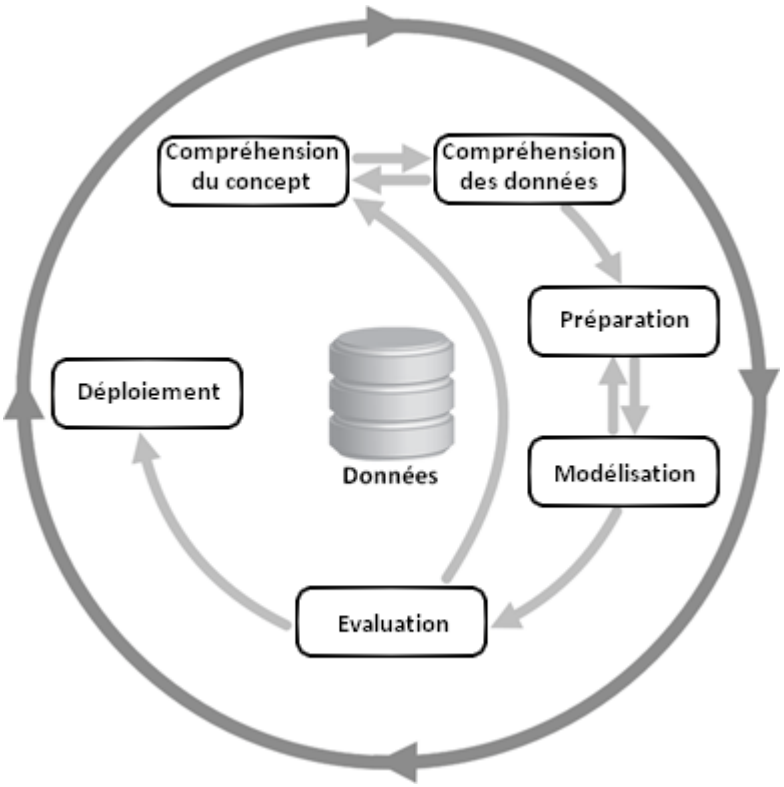


Figure 1.05 : *Différentes étapes du CRISP-DM*

Une dépendance significative entre les étapes est représentée par une flèche, tandis que la flèche circulaire externe représente la nature itérative du processus.
Ces différentes étapes seront brièvement présentées dans les sections suivantes. [9]

1.8.1 Compréhension du concept

Cette étape consiste à se familiariser avec le problème à résoudre. Cela comprend la définition des objectifs à atteindre et des critères de réussite pour aider à choisir les méthodes à utiliser.

1.8.1.1 Evaluer la situation

Cette tâche implique une recherche plus détaillée des ressources, des contraintes, des hypothèses et d'autres facteurs qui doivent être pris en considération pour déterminer l'objectif et le plan de projet de l'analyse des données.

- Inventaire des ressources : personnel impliqué, les données disponibles, les ressources informatiques (plateformes matérielles) et les logiciels (outils de DM, autres logiciels pertinents) ;
- Exigences : calendrier d'achèvement, compréhension et qualité des résultats, sécurité ;
- Hypothèses : concernant les données pouvant être vérifiées ou non au cours du DM ;
- Contraintes : disponibilité des ressources, taille des données à utiliser pour la modélisation ;
- Risques et éventualités : événements susceptibles de retarder ou d'échouer le projet. [9]

1.8.1.2 Déterminer les objectifs à atteindre

Les éléments ci-dessous sont à prendre en compte :

- Description des résultats escomptés du projet ;
- Définition des critères de réussite du projet en termes techniques ;
- Production d'un plan de projet. [9]

1.8.2 Compréhension des données de départ

Elle a pour but de se familiariser avec les données et d'évaluer la qualité des données collectées.
Les tâches incluent :

- la collection des données initiales : consiste à acquérir les données (ou accéder aux données) répertoriées dans les ressources du projet ;
- la description du format et de la quantité des données ;
- l'exploration des données par des requêtes ;
- la vérification : données complètes ou valeurs manquantes, données correctes ou présence d'erreurs. [9]

1.8.3 Préparation des données

L'étape de préparation des données couvre toutes les activités liées à la construction du dataset (données qui seront utilisées dans la phase de modélisation) à partir des données initiales. Elle constitue la phase la plus importante du processus. En effet, la qualité des résultats obtenus dépendra essentiellement des données à l'entrée, c'est pourquoi presque la moitié du temps alloué au traitement sera consacrée à cette phase.

- Le regroupement des données

Il s'agit de l'exploration d'autres sources de données pour compléter les données de départ. Ces données sont ajoutées dans l'entrepôt de données.

- Le nettoyage des données

Il consiste à traiter les événements de mauvaises qualités c'est-à-dire qui présentent des attributs manquants ou incohérents. En présence de tels événements il faut soit les remplacer, soit estimer la valeur de ces attributs manquants ou simplement les supprimer.

- L'intégration des données

Elle représente le transfert des données dans l'environnement d'analyse qui sera utilisé. Ces données doivent correspondre au format exigé par cet environnement d'analyse. [9]

1.8.4 Construction du modèle ou modélisation

Il s'agit du Data Mining proprement dit, en utilisant les statistiques, l'informatique et surtout l'apprentissage automatique.

Les tâches à exécuter sont les suivantes :

- sélectionner des techniques de modélisation ;
- séparer les données en deux : une partie (training set) pour construire le modèle et une autre (test set) pour évaluer la qualité et la validité du modèle ;
- construire le modèle. [9]

1.8.5 Evaluation du modèle

Cette étape permet de valider ou non les modèles obtenus précédemment et d'évaluer la fiabilité des connaissances extraites ainsi que les performances des algorithmes utilisés.

Ci-dessous la liste des tâches :

- Évaluation des résultats du DM en fonction des critères de succès prédéfinis ;
- Approbation du modèle répondant aux critères ;

- Détermination des prochaines étapes : prendre une décision. [9]

1.8.6 Déploiement

Il consiste à utiliser le modèle construit, pour la résolution du problème posé à l'étape 1.

- Résumer la stratégie adoptée pour le déploiement y compris les différentes étapes à suivre pour l'exécuter ;
- Planifier le monitoring et la maintenance ;
- Rédaction d'un rapport final et présentation à l'utilisateur. [9]

1.9 Conclusion

Le Data Mining est le résultat de l'évolution de la puissance de calcul et de la capacité à stocker les données et donc des systèmes de base de données. L'énorme quantité de données qui « reposait en paix » dans des matériels de stockage, est désormais accédée pour en extraire des informations utiles. Le Data Mining consiste alors à « torturer les données jusqu'à ce qu'elles avouent » afin de prendre une décision.

Le CRISP-DM a été développé pour servir de guide dans l'élaboration d'un projet de DM. Ce standard est composé de six étapes dont une est de représenter les données à explorer sous forme de table, le dataset. Une autre étape est la modélisation où sont utilisés les algorithmes d'apprentissage automatique. Le choix de ces algorithmes détermine la performance du système de DM. D'ailleurs, le chapitre suivant sera axé sur l'apprentissage automatique.

CHAPITRE 2
APPRENTISSAGE AUTOMATIQUE

2.1 Introduction

Depuis bientôt un demi-siècle, les chercheurs en Intelligence Artificielle travaillent à programmer des machines capables d'effectuer des tâches qui requièrent de l'intelligence. Cependant, programmer des machines aptes à s'adapter à différentes situations et éventuellement à évoluer en fonction de nouvelles contraintes est difficile. L'enjeu est de contourner cette difficulté en dotant la machine de capacités d'apprentissage lui permettant de tirer parti de son expérience. C'est pourquoi les recherches sur l'apprentissage par les machines se sont développées.

Dans ce chapitre, nous allons voir en détail l'apprentissage automatique notamment quelques généralités, les méthodes d'apprentissage supervisé et non supervisé ainsi que différents algorithmes utilisés par chacun d'eux. Nous allons spécialement mettre l'accent sur le Séparateur à Vaste Marge et l'algorithme des K-means.

2.2 Définitions

Le terme académique le plus courant est « apprentissage automatique ». Mais il est aussi connu sous l'appellation « apprentissage artificiel » et également « apprentissage machine », traduit de l'anglais « Machine Learning » (ML) qui semble plus approprié.

Le ML est l'étude des algorithmes qui permettent aux programmes de s'améliorer automatiquement par expérience. Ainsi, on dit qu'il y a apprentissage automatique lorsque les performances d'un programme informatique et les tâches réalisées à partir de ce programme, s'améliorent au fur et à mesure des expériences acquises.

Considérons T une tâche d'apprentissage, Ω un ensemble d'expériences, et Π les performances mesurées sur T. Il y a apprentissage automatique à partir de Ω si les performances Π de la tâche T s'améliorent en fonction de Ω. Cet ensemble Ω est également appelé ensemble d'apprentissage ou « training set ». L'apprentissage automatique consiste à améliorerΠ en appliquant des algorithmes informatiques sur Ω.

Le ML est un sous domaine de l'Intelligence Artificielle qui définit une façon d'acquérir les connaissances. Le ML est donc la partie Intelligence Artificielle du DM. Par conséquent, il existe un type d'apprentissage supervisé et un type non supervisé. [10] [11]

2.3 Apprentissage supervisé

Le principal objectif de l'apprentissage supervisé est la prédiction. Il s'agit de construire un modèle capable de prédire une variable de sortie à partir des variables d'entrée. Deux des problèmes centraux de l'apprentissage supervisé sont la classification et la régression. La différence entre les deux tâches est le fait que la variable de prédiction est nominale (ou qualitative) pour la classification et numérique pour la régression. [12]

2.3.1 Principe

L'apprentissage supervisé tire son principe de l'apprentissage statistique. La théorie de l'apprentissage statistique étudie les propriétés mathématiques des machines d'apprentissage. Ces propriétés représentent celles de la classe de fonctions ou modèles que peut implémenter la machine. L'apprentissage statistique utilise un nombre limité d'entrées (appelées exemples) d'un système avec les valeurs de leurs sorties pour apprendre une fonction qui décrit la relation fonctionnelle existante, mais non connue, entre les entrées et les sorties du système.

Soit une population Ω qui représente les données d'étude. Cette population est appelée base d'apprentissage ou ensemble d'entrainement. Elle est divisée en deux sous-ensembles : un sous ensemble X des variables de prédiction ou exemples d'apprentissage et un sous ensemble Y des variables à prédire ou variable cible ou label (étiquette). La population Ω peut être représentée par :

$$population\ \Omega = \begin{Bmatrix} Y\ ensemble\ des\ variables\ \text{à}\ prédire \\ X\ ensemble\ des\ variables\ de\ prédiction \end{Bmatrix} \quad (2.01)$$

L'apprentissage supervisé consiste à trouver une fonction f telle que :

$$\forall\ x_1, x_2, \dots, x \in X\ et\ \forall\ y \in\ Y\ on\ a\ y = f(X) \quad (2.02)$$

La fonction f permettra de calculer la valeur Y correspondant à une nouvelle série de variable de prédiction quelconque X. On dit aussi que f permet de modéliser Y. La fonction de classification f est appelée un classifieur.

Dans les sections qui suivent, nous allons nous intéresser à la classification proprement dite, à la régression ainsi qu'à quelques algorithmes utilisés pour l'apprentissage supervisé. [13] [7]

2.3.2 Classification

Effectuer une classification, c'est mettre en évidence, d'une part, des relations entre des objets et, d'autre part les relations entre ces objets et leurs paramètres. La classification a donc deux objectifs

à atteindre : trouver un modèle capable de représenter la répartition des données (catégorisation) et définir de manière formelle l'appartenance à l'une ou l'autre des classes de toute nouvelle donnée (généralisation). [14]

2.3.2.1 Formulation du problème de classification

On suppose premièrement que les exemples d'apprentissage sont générés selon une certaine probabilité inconnue (mais fixe) c'est-à-dire indépendants et identiquement distribués (*iid*). C'est une supposition standard dans la théorie d'apprentissage. Les exemples sont de dimension m ($x_i \in R^m$) et dans le cas d'apprentissage supervisé, accompagnés d'étiquettes caractérisant leurs types ou classes d'appartenance. Dans le cas d'une classification binaire cette étiquette est soit $+1$ ou -1 (quelques fois 0 ou 1). Ainsi, l'ensemble d'apprentissage est constitué par l'ensemble des exemples et leurs étiquettes correspondantes soit :

$$\Omega = \{(x_1, y_1), \dots, (x_n, y_n)\} \tag{2.03}$$

avec $x_i \in R^m$ et $y_i = \pm 1$.

Le problème est donc de trouver une fonction f qui assigne le label $+1$ (respectivement -1) aux éléments $\vec{x}$ tels que $f(\vec{x}) \geq 0$ (respectivement $f(\vec{x}) < 0$). La surface de séparation est donnée par l'équation $f(\vec{x}) = 0$ qui divise l'espace en deux parties, l'une positive, l'autre négative. [13] [15]

2.3.2.2 Minimisation du Risque Empirique

Une solution évidente au problème ci-dessus est de minimiser l'erreur d'apprentissage ou Risque Empirique, c'est-à-dire le taux de mauvaise classification sur la base d'apprentissage. On note $R_{emp}[f]$ le Risque Empirique, i.e. le taux d'erreurs effectuées par la fonction f sur la base d'apprentissage Ω tel que :

$$R_{emp}[f] = \frac{1}{n}\sum_{i=1}^{n} Lo\big(y_i, f(x_i)\big) \tag{2.04}$$

Avec Lo une fonction de perte (function loss) telle que :

$$Lo = \begin{cases} 1 & si \quad y_i = f(x_i) \\ 0 & sinon \end{cases} \tag{2.05}$$

qui mesure la différence entre la réponse y fournie par le superviseur et celle $f(x)$ fournie par la machine d'apprentissage. [14] [16]

2.3.2.3 Sur-apprentissage et risque total

Ramenons la formule du Risque Empirique à l'expression suivante :

$$R_{emp}[f] = \frac{1}{2n}\sum_{i=1}^{n}|y_i - f(\overrightarrow{x_i})| \tag{2.06}$$

En supposant que les données sont générées selon une distribution de probabilité inconnue $P(x, y)$, le risque total ou le taux de mauvaise classification sur R^m vaut :

$$R(f) = \int \frac{1}{2}|y - f(\vec{x})|\, dP(x, y) \tag{2.07}$$

Le principe de Minimisation du Risque Empirique (« Empirical Risk Minimization » - ERM) a néanmoins quelques inconvénients. Premièrement, cela ne permet pas de définir une unique solution. Deuxièmement, le taux d'erreur sur tout l'espace peut être beaucoup plus grand que le taux d'erreur d'apprentissage (minimiser l'erreur sur un sous-ensemble d'éléments, ici Ω, n'est pas équivalent à minimiser l'erreur sur tous les éléments c'est-à-dire sur R^m). Dans ce cas, la fonction n'a pas une bonne capacité de généralisation : on parle de sur-apprentissage. Ce phénomène de sur-apprentissage est obtenu lorsque la surface de séparation a une forme très complexe et est trop liée à la base d'apprentissage.

Pour garantir que f, prenne en charge même les exemples jamais vus, il faut contrôler sa capacité de généralisation, mesurée souvent sur un autre ensemble d'exemples appelé ensemble de test réservé uniquement pour tester la machine apprise. La fonction f recherchée doit donc minimiser les erreurs de classification sur les deux ensembles d'entrainement et de test. [15] [13]

2.3.2.4 Théorie de Vapnik

La théorie de l'apprentissage statistique, selon Vapnik en 1998, fournit une relation entre le Risque Empirique et le risque total, i.e. le taux de mauvaise classification sur R^m : soit $\eta \in [0,1]$, avec une probabilité de $1 - \eta$, on a :

$$R(f) \leq R_{emp}[f] + \sqrt{\frac{h[\log(2n/h) + 1] - \log \eta/4}{n}} \tag{2.08}$$

où h est un entier positif appelé "VC-dimension" (dimension de Vapnik-Chervonenkis). Pour un ensemble de fonctions $\{f\}$, il est défini comme le plus grand nombre de points qui peuvent être

séparés par l'ensemble $\{f\}$, quelles que soient les étiquettes des points. On peut donc noter que les deux risques dépendent de la fonction de décision choisie f, alors que le deuxième terme de l'inégalité précédente est monotone croissant par rapport à h et dépend donc de l'ensemble de fonctions choisi. [15]

2.3.3 Régression

Dans un problème de régression, y prend des valeurs continues et l'on cherche également à exprimer par une fonction la dépendance entre x et y. La fonction de perte qu'on considère principalement est l'écart quadratique défini par :

$$Lo\big(y, f(x)\big) = (y - f(x))^2 \tag{2.09}$$

Le risque ou l'erreur d'une fonction f est alors l'écart quadratique moyen défini par :

$$R(f) = \int_{X \times Y} (y - f(x))^2 \, dP(x, y) \tag{2.10}$$

Comme en classification, on peut exprimer simplement une fonction qui minimise l'erreur quadratique moyenne.

La fonction r définie par la formule suivante est la fonction de régression de risque minimal :

$$r = \int_{Y} y \, dP(y|x) \tag{2.11}$$

Elle calcule pour chaque élément x la moyenne des valeurs observées en x. [16]

2.3.4 Quelques algorithmes d'apprentissage supervisé

Les modèles fréquents utilisés en apprentissage supervisé sont nombreux. Toutefois, nous n'allons pas les détailler un à un car notre sujet se focalise essentiellement sur le Séparateur à Vaste Marge (SVM). Ils incluent :

- la méthode des K plus proches voisins (K-Nearest Neighbours – KNN) ;
- l'arbre de décision (Decision Tree – DT) ;
- la régression logistique ;
- le classifieur bayésien naïf ;
- les réseaux de neurones artificiels (Artificial Neural Network – ANN) ;
- et le SVM. [17]

2.3.5 Séparateur à Vaste Marge (SVM)

Les SVM sont de nouvelles méthodes d'apprentissage pour la classification binaire, motivées par les résultats de la théorie de l'apprentissage statistique. A la différence des autres algorithmes supervisés, les SVM sont fondés sur le principe de la Minimisation du Risque Structurel (« Structural Risk Minimization » - SRM) dont l'idée fondamentale est de minimiser la borne supérieure du risque total de l'équation 2.07. Ainsi, l'ensemble de fonctions $\{f\}$ est limité, ce qui implique une meilleure capacité de généralisation des SVM. Ce principe privilégie donc la capacité de généralisation par rapport à la classification sur la base d'apprentissage, ce qui permet d'éviter un sur-apprentissage. [13]

2.3.5.1 SVM à classe binaire

a. Cas des données linéairement séparables

L'idée des algorithmes de SVM est de partager l'espace en deux parties à l'aide d'un hyperplan qui maximise la distance minimale des observations à ce plan (i.e. la marge). Les observations qui sont situées les plus proches de l'hyperplan séparateur (sur la marge), sont appelées les « vecteurs supports ». Voici un schéma représentatif de ces concepts.

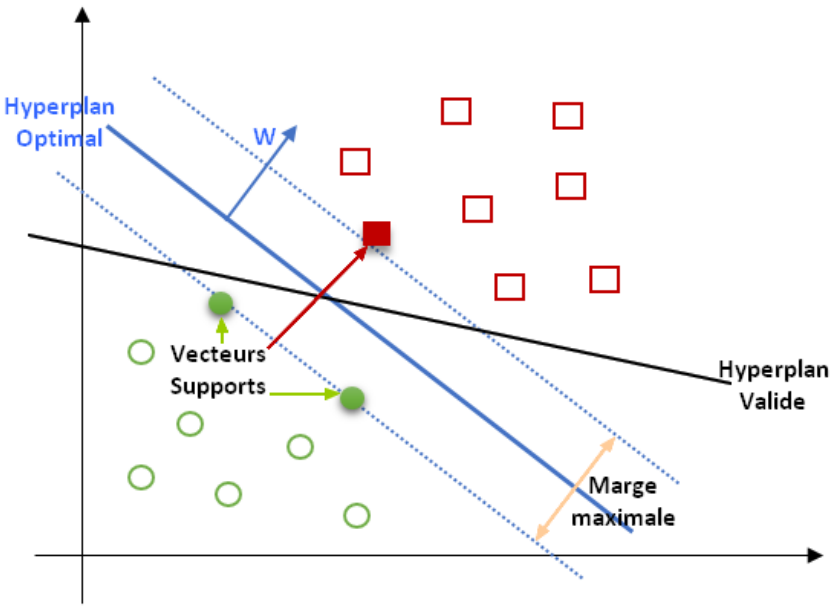


Figure 2.01 : *Concept du SVM*

Une base d'apprentissage est dite linéairement séparable s'il existe au moins une fonction linéaire f qui classifie correctement tous les objets de la base d'apprentissage.

Dans le cas linéairement séparable, l'hyperplan séparateur est défini par l'équation :

$$f(\vec{x}) = \vec{w}.\vec{x} + b = 0 \tag{2.12}$$

avec $\vec{w}$ un vecteur normal à l'hyperplan et $b \in R$ un scalaire appelé le biais.

Afin de classifier correctement la base d'apprentissage, nous devons avoir :

$$\begin{cases} \vec{w}.\vec{x_i} + b > 0 \quad si\ y_i = +1 \\ \vec{w}.\vec{x_i} + b < 0 \quad si\ y_i = -1 \end{cases} \tag{2.13}$$

Ce qui équivaut à :

$$\begin{cases} \vec{w}.\vec{x_i} + b \geq 1 \quad si\ y_i = +1 \\ \vec{w}.\vec{x_i} + b \leq -1\ si\ y_i = -1 \end{cases} \tag{2.14}$$

Ces contraintes impliquent une marge, qui est deux fois la plus petite distance entre un point de la base d'apprentissage et l'hyperplan séparateur, définie par la distance entre $\vec{w}.\vec{x} + b = -1$ et $\vec{w}.\vec{x} + b = 1$. Elle est égale à $\frac{2}{\|w\|}$.

L'hyperplan choisi doit donc faire partie de la famille d'hyperplans à marge maximale afin de minimiser la "VC-dimension". Cet hyperplan est appelé Hyperplan Séparateur Optimal (HSO).

Comme maximiser la marge est équivalent à minimiser l'inverse de la marge, les paramètres optimaux $\vec{w}^*$ et b^* sont obtenus après résolution du Problème d'Optimisation Quadratique (POQ) convexe sous contraintes linéaires :

$$\min_{(w,b)} \frac{\|\vec{w}\|}{2} \tag{2.15}$$

telles que les contraintes linéaires soient : $y_i(\vec{w}.\vec{x_i} + b) \geq 1, \forall i \in \langle 1, n \rangle$

Pour résoudre ce problème, on introduit les multiplicateurs de Lagrange $\{\lambda_i\}_{i \in \langle 1,n \rangle}$

On obtient alors le problème dual suivant :

$$L(\vec{w}, b, \vec{\lambda}) = \frac{\|\vec{w}\|^2}{2} - \sum_{i=1}^{n} \lambda_i.[y_i(\vec{w}.\vec{x_i} + b) - 1] \tag{2.16}$$

L'unique point d'inflexion du Lagrangien L, qui est un minimum par rapport à $(\vec{w}, b)$ et un maximum par rapport à λ, détermine la solution du POQ.

Si nous appelons λ^* la solution du problème, le paramètre optimal $\vec{w}^*$ est obtenu à partir de l'équation :

$$\vec{w}^* = \sum_{i=1}^{n} {\lambda_i}^* y_i \vec{x_i} \tag{2.17}$$

Et b^* est obtenu à partir des conditions de Karush-Kuhn-Tucker (KKT) :

$${\lambda_i}^*.[y_i(\vec{w}^*.\vec{x_i} + b^*) - 1] = 0 \tag{2.18}$$

Si ${\lambda_i}^*$ est non nul, le vecteur correspondant $\vec{x_i}$ est positionné sur le bord du tube (à une distance égale à la moitié de la marge de l'HSO : $y_i(\vec{w}^*.\vec{x_i} + b^*) = 1$). Sinon, le vecteur correspondant $\vec{x_i}$ est positionné du bon côté du tube et sa distance par rapport à l'HSO est supérieure à la moitié de la marge. Remarquons que $\vec{w}^*$ est une combinaison linéaire des vecteurs $\vec{x_i}$ tels que les multiplicateurs de Lagrange ${\lambda_i}^*$ soient non nuls. Ces vecteurs sont appelés Vecteurs Supports et sont les éléments de la base d'apprentissage les plus proches de l'HSO.

La fonction de décision f est donnée par :

$$f(\vec{x}) = \vec{w}^*.\vec{x_i} + b^* = \sum_{i=1}^{n} {\lambda_i}^* y_i \vec{x_i}.\vec{x} + b^* \tag{2.19}$$

La classification de nouveaux objets $\vec{x}$ nécessite le calcul du produit scalaire entre $\vec{x}$ et les Vecteurs Supports. [18] [15]

b. Cas des données non linéairement séparables

Deux méthodes essentielles permettent la classification binaire pour les données non linéairement séparables : la marge souple et les méthodes de Kernel.

- La marge souple

Dans cette partie, nous allons généraliser le cas des bases d'apprentissage non-linéairement séparables. En effet, l'hypothèse de cas linéairement séparable est trop restrictive pour de nombreuses applications réelles, surtout lorsque les données sont bruitées.

L'analyse précédente est étendue afin de permettre des erreurs sur la base d'apprentissage en introduisant des variables dites souples $\{\xi_i\}_{i\in\langle 1,n\rangle}$ dans le modèle précédent (voir Figure 2.02).

$$\xi_i(\vec{w}, b) = \begin{cases} 0 \quad si\ y_i(\vec{w}.\vec{x_i} + b) \geq 1 \\ 1 - y_i(\vec{w}.\vec{x_i} + b)\ si\ y_i(\vec{w}.\vec{x_i} + b) \leq 1 \end{cases} \tag{2.20}$$

Ces variables quantifient les erreurs réalisées sur les éléments de la base d'apprentissage.

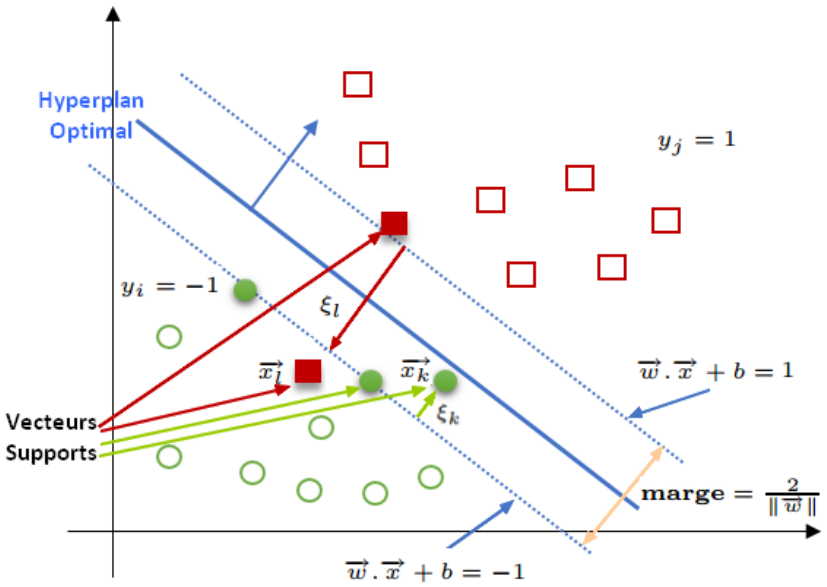


Figure 2.02 : *La marge souple*

Les SVM à marge souple ont pour but de maximiser la marge et minimiser l'erreur d'apprentissage, ce qui implique un nouveau POQ :

$$\min_{(\vec{w},b,\xi)} \frac{\|\vec{w}\|^2}{2} + Q\sum_{i=1}^{n} \xi_i \qquad (2.21)$$

tel que : $y_i(\vec{w}.\overrightarrow{x_i} + b) \geq 1 - \xi_i, \xi_i \geq 0, \forall i \in \langle 1, n\rangle$

où Q est appelé paramètre de régularisation et est positif. Lorsque Q est grand, le problème pénalise plus l'erreur d'apprentissage alors qu'un paramètre Q plus petit autorise une marge et une erreur d'apprentissage plus grande. Ce paramètre permet donc de définir un compromis entre une grande marge et un faible nombre d'erreurs sur la base d'apprentissage.

Comme ce qui précède, (à l'équation 2.15), introduisons les multiplicateurs de Lagrange $\{\lambda_i\}_{i\in\langle 1,n\rangle}$ et $\{\mu_i\}_{i\in\langle 1,n\rangle}$ associés respectivement aux contraintes : $y_i(\vec{w}.\overrightarrow{x_i} + b) \geq 1 - \xi_i$ et $\xi_i \geq 0$.

$$L(\vec{w}, b, \vec{\xi}, \vec{\lambda}, \vec{\mu}) = \frac{\|\vec{w}\|^2}{2} + Q\sum_{i=1}^{n} \xi_i - \sum_{i=1}^{n} \lambda_i.[y_i(\vec{w}.\overrightarrow{x_i} + b) - 1 + \xi_i] \sum_{i=1}^{n} \mu_i \xi_i \qquad (2.22)$$

La fonction à minimiser est identique à celle du problème dans le cas linéairement séparable, nous avons néanmoins une contrainte supplémentaire : λ doit être borné supérieurement par Q. Après avoir résolu ce problème, nous retrouvons la solution $\vec{w}$ à partir de la même équation 2.08.

Remarquons que les conditions KKT permettent de calculer b :

$$\begin{gathered}\lambda_i{}^*.[y_i(\vec{w}^*.\overrightarrow{x_i}+b^*)-1+\xi_i]=0\\ Q-\lambda_i{}^*-\mu_i{}^*=0\\ \mu_i{}^*\xi_i{}^*=0\end{gathered} \qquad (2.23)$$

- si $\lambda_i{}^* = 0$, alors $\mu_i{}^* = Q$, $\xi_i{}^* = 0$ et $[\, y_i(\vec{w}^*.\overrightarrow{x_i}+b^*)-1] > 0$; le vecteur $\overrightarrow{x_i}$ est du bon côté du tube,

- si $0 < \lambda_i{}^* < Q$, alors $0 < \mu_i{}^* < Q$, $\xi_i{}^* = 0$ et$[y_i(\vec{w}^*.\overrightarrow{x_i}+b^*)-1] = 0$; le vecteur $\overrightarrow{x_i}$ est sur la frontière du tube et est un Vecteur Support,

- si $\lambda_i{}^* = Q$ alors $\mu_i{}^* = 0$, $\xi_i{}^* > 0$ et $[y_i(\vec{w}^*.\overrightarrow{x_i}+b^*)-1+\xi_i{}^*] = 0$; $\overrightarrow{x_i}$ n'est pas du bon côté du tube (il est soit bien classifié du bon côté de l'HSO mais à l'intérieur du tube, soit du mauvais côté de l'HSO et donc mal classifié). Dans ce cas, il est toujours un Vecteur Support et intervient dans la définition de $\vec{w}$ et donc dans la position de l'HSO. [15]

- Les méthodes de Kernel

Dans la plupart des cas, la base d'apprentissage nécessite une surface de décision plus compliquée qu'un simple hyperplan linéaire. Pour pouvoir prendre en compte un séparateur non linéaire, les SVM linéaires peuvent être généralisés par l'introduction d'une fonction Φ qui transfère les données de R^m vers un ensemble de dimension supérieure F dans lequel elles deviennent linéairement séparables (voir Figure 2.03) :

$\Phi = R^m \rightarrow F$

$\vec{x} \rightarrow \Phi(\vec{x})$

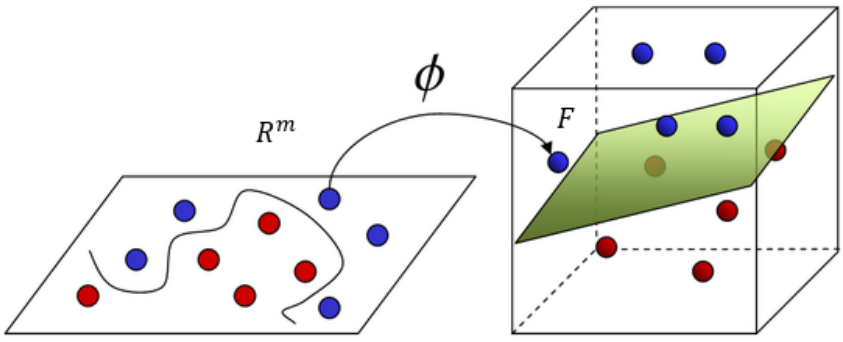


Figure 2.03 : *Rôle de la fonction Φ*

L'idée est donc de séparer linéairement les données transférées dans l'espace F de dimension supérieure grâce à la fonction Φ : $\{(\Phi(\overrightarrow{x_i}), y_i)\}_{i\in\langle 1,n\rangle}$.

Par analogie avec l'analyse précédente, les paramètres de l'HSO donnés par la résolution du problème d'optimisation sont les suivants :

$$\vec{w}^* = \sum_{i=1}^{n} \lambda_i{}^* \, y_i \Phi(\vec{x_i}) \tag{2.24}$$

$$f(\vec{x}) = \vec{w}^*.\vec{x_i} + b^* = \sum_{i=1}^{n} \lambda_i{}^* \, y_i \Phi(\vec{x_i}) \,.\, \Phi(\vec{x}) + b^* \tag{2.25}$$

Ces formules sont obtenues en remplaçant les vecteurs $\vec{x}$ par leur valeur dans l'ensemble des images $\Phi(\vec{x})$. Il suffit donc de connaître la valeur du produit scalaire pour n'importe quel couple de points, qui est appelé noyau :

$$Ke(\vec{x}, \vec{x}') = \Phi(\vec{x}).\Phi(\vec{x}') \tag{2.26}$$

Par conséquent, si l'on trouve une fonction Ke qui s'écrit comme un produit scalaire, la connaissance explicite de Φ n'est pas nécessaire : il s'agit du « truc du noyau ». Le théorème de Mercer fournit un critère très efficace afin de savoir si une fonction peut être considérée comme un noyau : une fonction symétrique K est un noyau si et seulement si, quelle que soit la famille de vecteurs $\{\vec{x_i}\}_{i\in<1,N>}$, $Ke(\vec{x_i}, \vec{x_j})$ est une matrice définie positive. Ce théorème permet de calculer directement le produit scalaire $Ke(\vec{x_1}, \vec{x_2})$ sans avoir à calculer $\Phi(\vec{x_1}) et\ \Phi(\vec{x_2})$. Ainsi, le calcul et la définition de Φ sont donc évités. Un noyau correspond en fait à un produit scalaire (fonction symétrique définie positive) dans l'espace des images. Le problème d'optimisation reste donc quadratique convexe avec des contraintes d'égalité linéaires puisque le noyau est défini positif.

Les noyaux les plus classiques sont :

- le noyau linéaire

$$Ke(\vec{x}, \vec{x}') = \vec{x} \,.\, \vec{x}' \tag{2.27}$$

- le noyau polynomial (plus le degré q est élevé, plus la forme de l'HSO est complexe)

$$Ke(\vec{x}, \vec{x}') = (\vec{x} \,.\, \vec{x}' + 1)^q \tag{2.28}$$

- le noyau RBF (Radial Basis Function)

$$Ke(\vec{x}, \vec{x}') = \exp(-\frac{\|\vec{x} - \vec{x}'\|}{\sigma}) \tag{2.29}$$

- le noyau Gaussien

$$Ke(\vec{x}, \vec{x}') = \exp(-\frac{\|\vec{x} - \vec{x}'\|^2}{2\sigma^2}) \tag{2.30}$$

Presque toutes les formes peuvent être obtenues à partir de ces deux derniers noyaux. Plus σ est proche de zéro, plus les gaussiennes centrées sur les Vecteurs Supports seront pointues et plus

complexe sera l'HSO. Ces noyaux correspondent à un transfert dans des espaces de "VC-dimensions" infinies. [15]

2.3.5.2 SVM multiclasse

La plupart des problèmes ne se contente pas de deux classes de données. Il existe plusieurs méthodes pour faire la classification multiclasse. Cette section présente la première approche mise en œuvre pour effectuer des tâches de classification au moyen de SVM multiclasse : l'emploi de méthode de décomposition dont la méthode Un contre Tous ainsi que la méthode Un contre Un.

a. Approche Un Contre Tous

L'approche Un contre Tous est la plus simple et la plus ancienne des méthodes de décomposition. Elle consiste à utiliser un classifieur binaire (à valeurs réelles) par catégorie. Le $k - i\grave{e}me$ classifieur est destiné à distinguer la catégorie d'indice k de toutes les autres. Pour affecter un exemple, on le présente donc à M classifieurs, et la décision s'obtient en appliquant le principe « winner-takes-all » : l'étiquette retenue est celle associée au classifieur ayant renvoyé la valeur la plus élevée. Il convient cependant de souligner qu'elle implique d'impliquer des apprentissages aux répartitions entre catégories très déséquilibrées, ce qui soulève souvent des difficultés pratiques. [19]

b. Approche Un Contre Un

Une autre méthode de décomposition très naturelle est la méthode Un contre Un. Elle consiste à utiliser un classifieur par couple de catégories. Le classifieur indicé par le couple (k, l), est destiné à distinguer la catégorie d'indice k de celle de l'indice l (avec $1 \leq k < l \leq M$). Pour affecter un exemple, on le présente donc C_M^2 à classifieurs, et la décision s'obtient habituellement en effectuant un vote majoritaire (« max-wins voting »). La voix de chaque classifieur peut être pondérée par une fonction de la valeur de la sortie calculée. [19]

2.4 Apprentissage non supervisé

L'apprentissage non-supervisé, comme son nom l'indique, consiste à apprendre sans superviseur. A partir d'une population, il s'agit d'extraire des classes ou groupes d'individus présentant des caractéristiques communes, le nombre et la définition des classes n'étant pas donnés à priori. De nombreux algorithmes ont été développés pour modéliser un système d'apprentissage non supervisé. Celui qui nous intéresse est le « clustering » qui peut se traduire par regroupement. Cette

section introduit la notion de clustering, ses principales étapes et les différentes méthodes de clustering, où sera spécialement développé la méthode des K-means.

2.4.1 Notions sur le clustering

On considère un ensemble de n objets $X = \{x_1, \dots, x_n\}$, ainsi qu'une matrice de dissimilarité D sur cet ensemble, telle que $d(x_i; x_j)$ représente la dissimilarité entre les deux objets x_i et x_j. La matrice D est de taille $n \times n$ et à valeurs dans $[0\,;1]$. Lorsque les données sont uniquement décrites par une telle matrice, on parle parfois de « données relationnelles ». La plupart des méthodes de classification non-supervisées utilisent une description vectorielle des données. On parle alors de « données objets » et on considère un ensemble fini $V = \{v_1, \dots, v_p\}$, de variables descriptives, telles que $v_j(x_i)$ désigne la valeur de l'objet $x_i \in X$ pour la variable $v_j \in V$.

La tâche de clustering permet de générer un ensemble de t clusters $C = \{C_1, \dots, C_t\}$ tel que chaque cluster C_a est un sous-ensemble de X ($C_a \subset X$) et l'union des clusters couvre l'ensemble des objets de départ ($\bigcup_{a=1}^{t} C_a = X$). [10]

2.4.1.1 Partitions, pseudo-partitions et partitions floues

Définition 2.01 :

C est une partition de X si et seulement si (*ssi*) C vérifie les propriétés suivantes :

$$(C_a \subset X) pour\ tout\ C_a \in C \tag{2.31}$$

$$(\bigcup_{a=1}^{t} C_a = X) \tag{2.32}$$

$$C_a \cap C_b = \emptyset\ pour\ (a, b)\ tel\ que\ a \neq b \tag{2.33}$$

L'équation 2.33 exprime le fait que les clusters constitués sont disjoints, chaque objet de X ne peut donc appartenir qu'à un seul cluster de C. Une telle partition sera également appelée « partition stricte ».

Définition 2.02 :

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

$$(C_a \subset X) pour\ tout\ C_a \in C \tag{2.34}$$

$$(\bigcup_{a=1}^{t} C_a = X) \tag{2.35}$$

$$C_a \subseteq C_b = \emptyset\ ssi\ a = b \tag{2.36}$$

Dans le cas d'une pseudo-partition, les intersections entre clusters ne sont pas nécessairement vides. Cependant, l'équation 2.36 interdit qu'un cluster soit inclus dans un autre.

Dans les deux définitions précédentes, chaque objet x_i appartient ou non à un cluster C_a donnée. On peut alors formaliser le processus de construction d'une partition ou d'une pseudo-partition par la donnée de t fonctions à valeurs binaires :

$$u_a : X \rightarrow \{0,1\}, a = 1 \dots t \; avec \; u_a(x_i) = \begin{cases} 1 & si \; x_i \in C_a \\ 0 & sinon \end{cases} \quad (2.37)$$

Cette formalisation peut être généralisée au cas de fonctions à valeurs réelles. Dans ce cas, les partitions généralisées sont dites « floues ».

Définition 2.03 :

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

$$u_a : X \rightarrow [0,1], a = 1 \dots t \quad (2.38)$$

u_a représente alors le degré d'appartenance de l'objet x_i au cluster C_a.

2.4.1.2 Hiérarchies et pseudo-hiérarchies

Certaines méthodes de clustering conduisent à un arbre hiérarchique aussi appelé dendrogramme. De même que l'on distingue les partitions strictes des pseudo-partitions, cette même distinction est faite entre hiérarchies et pseudo- hiérarchies. [10]

Définition 2.04 :

Soit ψ un ensemble de parties non vides sur X et h une partie, ψ est une hiérarchie si les propriétés suivantes sont vérifiées :

i) $X \in \psi$;

ii) pour tout $x_i \in X, \{x_i\} \in \psi$;

iii) pour tout $h, h' \in \psi, h \cap h' \in \{\emptyset, h, h'\}$;

iv) pour tout $h \in \psi, \cup \{h' \in \psi : h' \subset h\} \in \{h, \emptyset\}$.

Les propriétés *i)* et *ii)* expriment le fait que la racine de l'arbre est constituée de l'ensemble X et que les feuilles de l'arbre correspondent aux singletons.

Les propriétés *iii)* et *iv)* assurent que deux clusters ne s'intersectent que si l'un est inclus dans l'autre et que chaque cluster contient tous ses successeurs (« fils ») et est contenu dans son unique cluster prédécesseur (« père »).

Définition 2.05 :

Soit ψ un ensemble de parties non vides sur X et h une partie, ψ est une pseudo-hiérarchie si les propriétés suivantes sont vérifiées :

v) $X \in \psi$;

vi) pour tout $x_i \in X, \{x_i\} \in \psi$;

vii) pour tout $h, h' \in \psi, h \cap h' = \emptyset \; ou \; h \cap h' \in \psi$;

viii) il existe un ordre (total) Θ sur X compatible avec P.

Définition 2.06 :

Un ordre Θ est compatible avec un ensemble P de parties de X, si tout élément de $h \in P$ est connexe selon Θ.

Définition 2.07 :

Une partie h est connexe selon Θ si x et y étant les bornes (i.e. le plus petit et le plus grand élément) de h selon Θ, on a la condition : $\{z \; compris \; entre \; x \; et \; y \; selon \; \Theta\} \Leftrightarrow \{z \in h\}$

Par ces définitions, une pseudo-hiérarchie est telle que chaque cluster peut avoir plusieurs prédécesseurs. De plus, la propriété *viii)* concernant l'existence d'un ordre Θ sur X, permet la visualisation d'une telle pyramide.

2.4.1.3 Centroïdes et médoïdes

De nombreux algorithmes de clustering assimilent chaque cluster à un point, ceci pour des raisons pratiques de complexité notamment. Ce « point », censé être représentatif du cluster peut être un centroïde ou un médoïde. [10]

Définition 2.08 :

Le centroïde x^* d'un cluster C_a est le point défini dans V par :

$$\forall j = 1, \dots, p, v_j(x^*) = \frac{1}{|C_a|} \sum_{x_i \in C_a} v_j(x_i) \tag{2.39}$$

Dans le cas où toutes les variables descriptives sont quantitatives (continues ou discrètes), le centroïde est défini, sur chaque composante, par la valeur moyenne des objets du cluster pour cette même composante. En ce sens, le centroïde correspond au centre de gravité du cluster. Le centroïde d'un cluster ne fait généralement pas partie des objets constituant ce cluster.

En revanche, lorsque certaines variables descriptives sont de nature qualitative (exemples : couleur, forme, etc.), la définition précédente n'a plus de sens puisque les opérateurs classiques (addition, division, etc.) ne sont pas applicables sur ce type de variable. On cherche alors l'objet le plus représentatif du cluster que l'on appelle aussi parfois « prototype » ou plus formellement médoïde.

Définition 2.09 :
Soient un cluster C_a d'objets définis sur V, et d'une mesure de dissimilarité sur V, le médoïde du cluster C_a est l'objet $x^* \in C_a$ tel que :

$$x^* = \underset{x_i \in C_a}{\operatorname{argmin}} \frac{1}{|C_a|} \sum_{x_j \in C_a} d(x_i, x_j) \tag{2.40}$$

Par cette définition, le médoïde d'un cluster est l'objet du cluster tel que la dissimilarité moyenne de tous les objets du cluster avec le médoïde est minimale. Inversement, le médoïde d'un cluster est l'objet en moyenne le plus similaire aux autres.

2.4.1.4 Concavité et convexité

On est parfois amené à s'intéresser à la « forme » des clusters obtenus par un algorithme de clustering. Notons que cette notion de « forme » est difficile à définir formellement et pourrait laisser penser que les objets doivent systématiquement être représentés dans un espace. Par exemple, dans un espace à deux dimensions, les formes convexe et concave peuvent être illustrées par la figure ci-dessous :

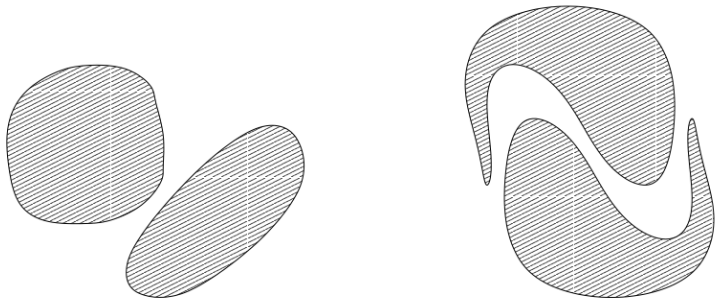


Figure 2.04 : *Exemples de clusters convexes (gauche) et concaves (droite), dans R^2*

On peut néanmoins parler de convexité ou concavité des clusters, pour des objets sur lesquels on ne connaît pas d'espace de représentation a priori. De façon générale, un cluster est convexe lorsque les objets qui le composent sont organisés autour d'un centre (centroïde ou médoïde). [10]

2.4.2 Etapes du clustering

Le processus de clustering se divise en trois étapes majeures (figure 2.05) : la préparation des données, l'algorithme de clustering et l'exploitation des résultats de l'algorithme. [10]

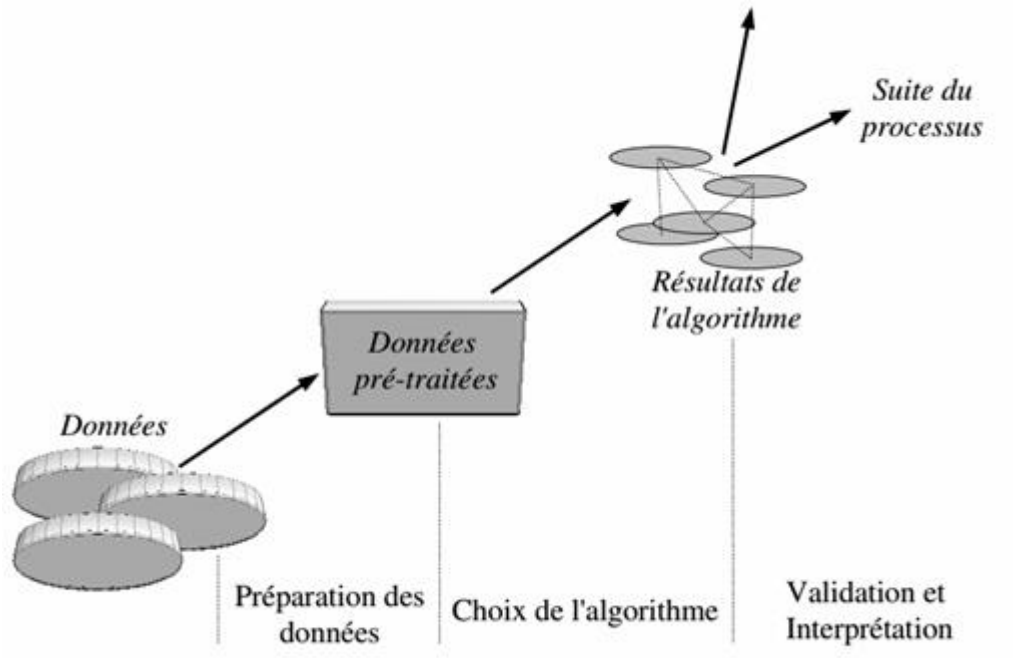


Figure 2.05 : *Les différentes étapes du processus de clustering.*

2.4.2.1 Préparation des données

a. Variables et sélections

Les objets sont décrits par des variables, aussi appelées attributs, descripteurs ou traits. Ces variables sont de différentes natures :

- variables quantitatives : continues (ex : la taille d'une personne), discrètes (ex : le nombre de personnes) ou sous forme d'intervalles (ex : la période de vie d'une personne) ;
- variables qualitatives : non-ordonnées (ex : la « couleur » des cheveux) ou ordonnées (ex : la taille : « petit », « moyen », « grand », etc.) ;
- variables structurées : par exemple la forme d'un objet (polygone, parallélogramme, rectangle, ovale, cercle, etc.)

L'étape de préparation consiste à sélectionner et/ou pondérer ces variables, voire à créer de nouvelles variables, afin de mieux discriminer entre eux les objets à traiter. [10]

b. Distances et similarités

La plupart des algorithmes de clustering utilise une mesure de proximité entre les objets à traiter. Cette notion de « proximité » est formalisée à l'aide d'une mesure (ou indice) de similarité, dissimilarité ou encore par une distance. Chaque domaine d'application possédant ses propres données, il possède également sa propre notion de « proximité » ; il faut concevoir alors une mesure différente pour chaque domaine d'application, permettant de retranscrire au mieux les différences (entre les objets) qui semblent importantes pour un problème donné. [10]

2.4.2.2 Le choix de l'algorithme

Le choix de l'algorithme de clustering doit donner lieu à une analyse globale du problème : quelle est la nature (qualitative et quantitative) des données ? Quelle est la nature des clusters attendus (nombre, forme, densité, etc.) ? L'algorithme doit être choisi de manière à ce que ses caractéristiques répondent convenablement à ces deux dernières questions. Les critères de décision peuvent être : la quantité de données à traiter, la nature de ces données, la forme des clusters souhaités ou encore le type de schéma attendu (pseudo-partition, partition stricte, dendrogramme, etc.). [10]

2.4.2.3 L'exploitation des clusters

Les clusters obtenus ne sont généralement ni remis en cause ni évalués en termes de disposition relative, dispersion, orientation, séparation, densité ou stabilité. Pourtant, il est sans aucun doute utile de distinguer les classes pertinentes obtenues, des autres. De même, cette étape d'analyse permet d'envisager le recours à une autre approche de clustering plus adaptée. Deux situations sont possibles : soit la tâche de clustering s'inscrit dans un traitement global d'apprentissage, soit les clusters générés par clustering constituent un résultat final.
Dans le premier cas, l'analyse des clusters obtenus (mesures statistiques de qualité) peut aider à orienter le traitement suivant. Une description des clusters n'est pas nécessaire dans cette situation. En revanche, dans le cas où le clustering constitue à lui seul un processus global de découverte de classes, l'exploitation des clusters pour une application donnée passe par une description de ces derniers. [10]

2.4.3 Différentes méthodes de clustering

Il existe de nombreuses méthodes de clustering. Cependant, il est difficile de proposer une classification « logique » de ces méthodes. Cela est due, d'une part, au fait que les classes

d'algorithmes se recouvrent et d'autre part, diffèrent selon que l'on s'intéresse plutôt aux résultats du clustering ou à la méthode utilisée pour parvenir à ce résultat. Nous proposons ici une classification en présentant d'abord les méthodes hiérarchiques puis de partitionnement.

2.4.3.1 Le clustering hiérarchique

Le principe des algorithmes hiérarchiques est de construire un arbre de clusters (ou dendrogramme) tel que présenté en figure 2.06 et formalisé par la définition *2.04* :

- la racine de l'arbre est formée par le cluster X contenant l'ensemble des objets ;
- chaque nœud de l'arbre constitue un cluster $C_i \subset X$;
- les feuilles de l'arbre correspondent aux singletons $\{x_1\}, \ldots, \{x_n\}$;
- l'union des objets contenus dans les fils d'un nœud donnée, correspond aux objets présents dans ce nœud ;
- les « paliers » sont indicés relativement à l'ordre de construction.

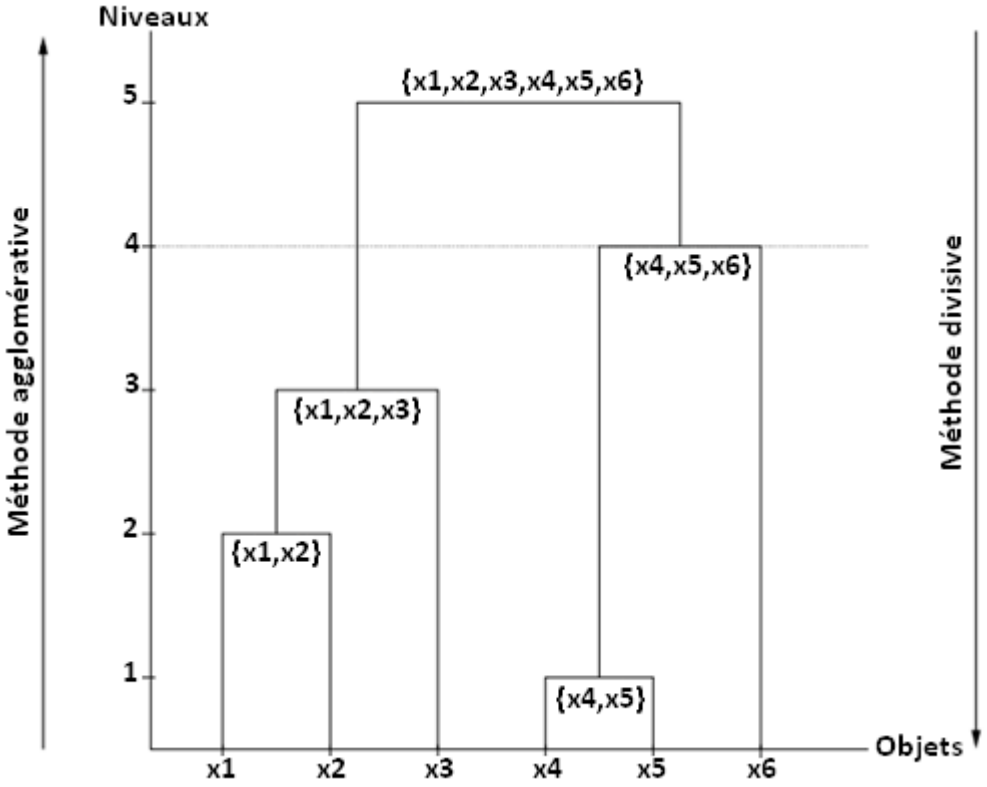


Figure 2.06 : *Exemple de dendrogramme*

A partir de ce dendrogramme, il est possible d'obtenir une partition de X en coupant l'arbre à un niveau donné. [10]

2.4.3.2 Le clustering par partitionnement

Contrairement aux approches hiérarchiques précédentes, les algorithmes de partitionnement proposent, en sortie, une partition de l'espace des objets plutôt qu'une structure organisationnelle du type dendrogramme. Le principe est alors de comparer plusieurs schémas de clustering (plusieurs partitionnements) afin de retenir le schéma qui optimise un critère de qualité. En pratique il est impossible de générer tous les schémas de clustering pour des raisons évidentes de complexité. On cherche alors un bon schéma correspondant à un optimum (le plus souvent local) pour ce critère. Cet optimum est obtenu de façon itérative, en améliorant un schéma initial choisi plus ou moins aléatoirement, par réallocation des objets autour de centres mobiles. Nous étudierons, dans la section suivante, les différentes techniques de réallocation à partir de l'algorithme bien connu des K-means.[10]

2.4.4 L'algorithme des K-means

Cet algorithme se présente comme suit :

Algorithme K-moyennes

Entrées : k le nombre de clusters désiré, d une mesure de dissimilarité sur l'ensemble des objets à traiter X

Sortie : Une partition $C = \{C_1, \dots, C_k\}$

Etape 0 : 1. Initialisation par tirage aléatoire dans X, de k centres $x^*_{1,O}, \dots, x^*_{k,0}$

2. Constitution d'une partition initiale $C_0 = \{C_1, \dots, C_k\}$ par allocation de chaque objet $x_i \in X$ au centre le plus proche :

$$C_l = \left\{ x_i \in X | d\left(x_i, x^*_{l,0}\right) = \min_{h=1,\dots,k} d\left(x_i, x^*_{l,0}\right) \right\}$$

3. Calcul des centroïdes des k classes obtenues $x^*_{1,1}, \dots, x^*_{k,1}$

Etape t : 4. Constitution d'une nouvelle partition $C_t = \{C_1, \dots, C_k\}$ par allocation de chaque objet $x_i \in X$ au centre le plus proche :

$$C_l = \left\{ x_i \in X | d\left(x_i, x^*_{l,t}\right) = \min_{h=1,\dots,k} d\left(x_i, x^*_{l,t}\right) \right\}$$

5. Calcul des centroïdes des k classes obtenues $x^*_{1,t+1}, \dots, x^*_{k,t+1}$

6. Répéter les étapes 4 et 5 tant que des changements s'opèrent d'un schéma C_t à un schéma C_{t+1} jusqu'à un nombre τ d'itérations.

7. Retourner la partition finale C_{final}.

L’algorithme des K-moyennes (K-means) est sans aucun doute la méthode de partitionnement la plus connue et la plus utilisée dans divers domaines d’application. Ce succès est dû au fait que cet algorithme présente un rapport cout/efficacité avantageux. Il fait partie des algorithmes utilisés par les méthodes de clustering par partitionnement strict.

A partir d’un tirage aléatoire de k « graines » dans X, l’algorithme des K-moyennes procède par itérations de super-étapes d’allocations des objets aux centres (initialement les graines), suivies du calcul de la position des nouveaux centres, dits « mobiles ».

L’algorithme des K-moyennes ne peut être utilisé que sur des données décrites par des attributs numériques permettant ainsi le calcul des centroïdes. [10]

2.5 Conclusion

L’apprentissage automatique a été conçu pour permettre aux machines d’être « intelligentes ». C’est un domaine qui se situe à l’intersection des statistiques et de l’intelligence artificielle. Dans ce chapitre, nous avons étudié deux méthodes d’apprentissage automatique : supervisé et non supervisé. De nombreux algorithmes ont été développés pour remplir l’objectif principal de l’apprentissage qui est l’extraction automatique de connaissances à partir d’exemples. Nous avons développé particulièrement les algorithmes de SVM pour le cas de la classification supervisée et le clustering par partitionnement K-means pour le cas non supervisé étant donné que nous allons les utiliser pour modéliser notre système de Data Mining.

CHAPITRE 3
MODELISATION D'UN SYSTEME DE CLASSIFICATION D'IMAGES

3.1 Introduction

L'objectif de la classification d'images est d'élaborer un système capable d'affecter une classe automatiquement à une image. Il est alors nécessaire de proposer des systèmes qui traitent des données images comme variable de prédiction. Dans le chapitre précédent, nous avons parlé d'apprentissage automatique et nous avons développé spécialement un algorithme pour chaque méthode supervisée et non supervisée. C'est dans ce troisième chapitre que nous allons utiliser ces algorithmes afin de concevoir un système de classification d'images. Nous allons d'abord énoncer quelques généralités, puis décrire une à une les étapes à suivre pour modéliser un tel système.

3.2 Généralités sur la classification d'images

La classification automatique d'image est une application de la reconnaissance de formes consistant à attribuer automatiquement une classe à une image à l'aide d'un système de classification. On retrouve ainsi la classification d'objets, de scènes, de textures, la reconnaissance de visages, d'empreintes digitales et de caractères parmi les applications courantes. Il existe deux approches pour la classification d'images. Dans l'approche non-supervisée (ou clustering), les données disponibles ne possèdent pas d'étiquettes ; il appartient alors au système d'extraire une règle d'appartenance de chaque image à un groupe donné. Dans l'approche supervisée chaque image est associée à une étiquette qui décrit sa classe d'appartenance. Cette dernière est celle qui sera détaillée dans ce chapitre. [20]

3.3 Etapes de création d'un système de classification d'image

Un système de classification automatique d'images est composé des étapes suivantes : l'acquisition des images ; le prétraitement permettant de « nettoyer » les images ; la phase d'extraction de caractéristiques permettant de décrire l'information pertinente contenue dans l'image à l'aide d'opérateur ou de descripteurs discriminants ; la phase d'apprentissage permettant de construire une frontière de décision pour identifier la classe d'une image présentée à l'entrée du système. Ces trois phases sont essentielles dans la construction du système de classification. Nous représentons sur la

figure 3.01 les différentes étapes à suivre pour respecter ces trois phases et nous les détaillerons dans les sections suivantes. [20]

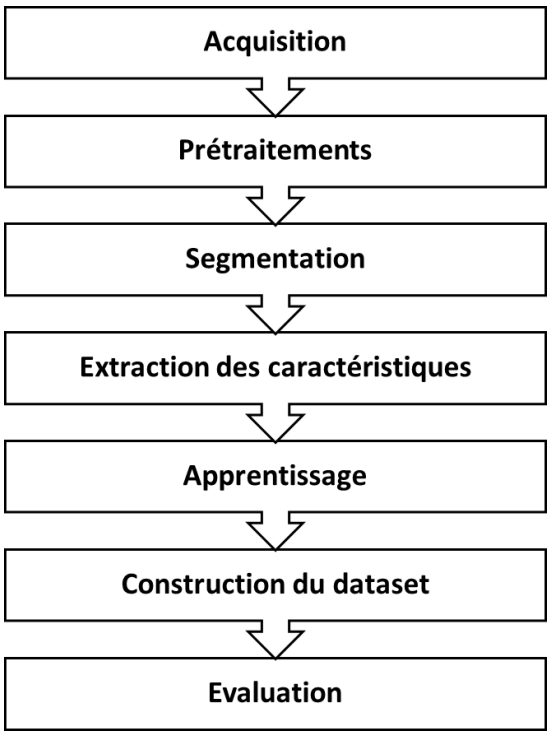


Figure 3.01 : *Etapes de création d'un système de classification*

3.4 Acquisition d'image

L'acquisition d'image constitue un des maillons essentiels de toute chaîne de conception et de production d'images. Pour pouvoir manipuler une image sur un système informatique, il est avant tout nécessaire de lui faire subir une transformation qui la rendra lisible et manipulable par ce système. L'image est considérée comme la variation de l'intensité lumineuse en fonction de la position sur un plan. La méthode d'acquisition se fait en convertissant ces informations lumineuses en signaux électriques grâce à des capteurs. Ces signaux sont ensuite convertis en numérique puis sauvegardés dans des éléments mémoire numériques (cartes mémoires, Random Access Memory ou RAM, …) pour une manipulation ultérieure. Ce passage de l'objet externe (l'image réelle) à sa représentation interne (image numérique) est le processus de numérisation. Ainsi, il existe divers types de capteurs, qui permettent de numériser les images. Les plus usuels sont :

- les appareils photo et caméras numériques monocanal et multicanal. L'acquisition d'images dans les domaines spectraux du rouge, du vert et du bleu permet de reconstituer la vision humaine ;

- les scanners médicaux : ils donnent des images 3-D sous la forme de séries d'images 2-D ;
- les scanners et micro densitomètres, ils numérisent des images ou négatifs sous forme analogique ;
- les microscopes : en modifiant leur focale, ils permettent d'observer la nature 3-D des objets étudiés ;
- les capteurs de télédétection : systèmes rigides, une barrette de 6000 CCD (Couple Charge Device) pour SPOT (Système probatoire d'observation de la Terre) ou à balayage (Thematic Mapper), avec un nombre très variable de bandes spectrales. [21] [22]

3.5 Prétraitement

Avant d'être analysées, les images doivent être prétraitées. Les principales opérations du prétraitement sont la suppression du bruit, la correction des erreurs, l'homogénéisation et la réduction des données. Le bruit est souvent dû aux appareils de mesure et aux capteurs du fait de la quantification. Il se manifeste par la présence d'informations résiduelles qui perturbent les données. Les erreurs sont dues soit aux capteurs (mauvais réglage de l'objectif de la caméra, mauvais calibrage de la caméra, …), soit à l'environnement (bruit ambiant, mauvais éclairage, …). L'homogénéisation des données consiste à débarrasser l'image d'informations redondantes et inutiles pour l'application que l'on désire réaliser. Enfin, les données fournies par le capteur sont souvent trop importantes et ne sont pas toutes utiles à l'analyse de l'image. Il convient donc de réduire cette masse d'information pour ne pas surcharger l'espace mémoire et pour éviter des temps et complexité de traitements trop élevés. [23]

3.6 Segmentation

La segmentation constitue, après la phase de prétraitement, l'étape essentielle de l'analyse des images qui fournit les éléments nécessaires à la description et à l'interprétation de leur contenu. Elle est souvent l'étape la plus critique du processus de reconnaissance des formes. En effet, une mauvaise segmentation ne pourra jamais être compensée par les traitements ultérieurs aussi sophistiqués soient-ils (extraction des attributs, classification automatique, etc, …).
La segmentation consiste à partitionner l'image étudiée en régions disjointes avec des couleurs homogènes. La segmentation peut être basée sur différents critères comme la couleur, la texture ou les niveaux de gris d'une image. Il existe plusieurs types d'approches pour réaliser une segmentation que nous allons classer en deux, à savoir :

- l'approche contours pour la recherche de frontières, cette méthode fait appel à des concepts de discontinuité ;
- l'approche région pour l'extraction de région, cette méthode fait appel à des concepts de similarité et de connexité.

L'approche qui nous intéresse directement ici est la segmentation par clustering qui est une approche région, mais avant de développer cette dernière, nous allons nous focaliser tout d'abord sur l'approche contours, afin d'avoir une présentation complète de la segmentation. [24] [25]

3.6.1 Approches contours

La segmentation par approche contours s'intéresse aux contours de l'objet dans l'image. La plupart des algorithmes qui lui est associé sont locaux, c'est-à-dire qu'ils fonctionnent au niveau du pixel. Des filtres détecteurs de contours sont appliqués à l'image. Les contours extraits sont souvent morcelés et peu précis, il faut alors utiliser des techniques de reconstruction de contours par interpolation ou connaître à priori la forme de l'objet recherché. Les contours dans une image proviennent des discontinuités de la fonction de réflectance (texture, ombre) et des discontinuités de profondeur (bords de l'objet).

Pour les détecter, il existe deux types d'approches :

- Approche formelle (approche gradient : détermination des extrema locaux dans la direction du gradient et approche laplacien : détermination des passages par zéro du laplacien.)
- Approche analytique proposée par Canny.

Un exemple de résultat obtenu par une approche formelle est donné sur la figure 3.02. [26]

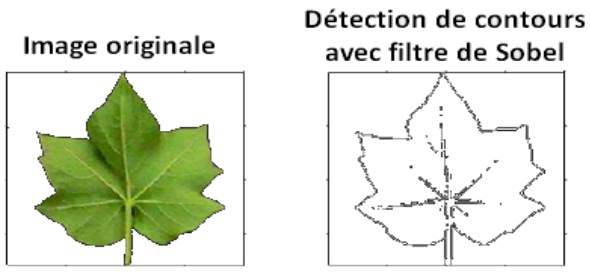


Figure 3.02 : *Exemple de détection de contours*

3.6.2 Approche région

Il existe quatre méthodes pour l'approche région : le seuillage, la croissance de région (region growing), la division-fusion (split and merge) et le clustering. Dans cette section, nous ne verrons

que les trois premières méthodes citées ci-dessus puisque la méthode de clustering sera détaillée en particulier dans une autre section.

3.6.2.1 Le seuillage

Le choix d'un seuil pour binariser une image, afin d'isoler les objets de leur contexte, est toujours un problème délicat. Il est alors préférable d'analyser d'abord la distribution d'intensité lumineuse dans l'image. L'histogramme des niveaux de gris permet de modéliser cette distribution. Dans le cas où les objets et l'arrière-plan est suffisant, on obtient un histogramme bimodal (voir figure 3.03).

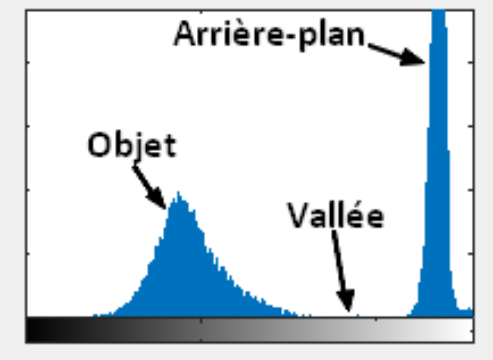


Figure 3.03 : *Histogramme bimodal*

Chacun des deux « pics » correspond à une des composantes de la distribution des intensités relatives d'une part à l'arrière-plan, d'autre part aux objets. Pour de tels histogrammes, en choisissant le seuil au fond de la « vallée » qui sépare les deux « pics », on obtient une binarisation très satisfaisante de l'image séparant nettement les objets de leur contexte. Un exemple est donné sur la figure ci-dessous :

Figure 3.04 : *Seuillage pour histogramme bimodal*

Cependant, de telles situations sont rares et, en général, l'histogramme n'est pas aussi « typé » que celui de la figure précédente. Il en résulte des difficultés pour ajuster le seuil, ce qui a donné lieu à d'autres types de seuillage dont : le seuillage global automatique (le plus connu est la méthode d'Otsu) et le seuillage local adaptatif. [24]

3.6.2.2 Le region-growing

L'idée de ce type de segmentation est de partir d'un point d'amorce (appelé seed) que l'on étend en ajoutant les points de la frontière qui satisfont le critère d'homogénéité afin de constituer des régions. Voici une illustration montrant ce principe de croissance de régions :

Figure 3.05 : *Principe de la croissance de région*

Cette méthode de croissance par régions a pour avantage d'être rapide et conceptuellement très simple mais elle ne permet pas d'avoir une vision globale du problème et il est difficilement d'avoir une bonne séparabilité des régions. En effet, dans une image il y a toujours plus ou moins de similitude entre des pixels adjacents au niveau de la couleur donc il est très difficile de trouver ces fameuses frontières. L'algorithme est très sensible au bruit et réalise en général une mauvaise segmentation si la variation des couleurs se fait progressivement. Cet algorithme est donc sujet au problème de dégradé des couleurs ou problème du gradient. [26]

3.6.2.3 Le split and merge

L'idée de cette méthode qui se divise en deux phases, le split (division) et le merge (fusion), est de regrouper les pixels de l'image originale en zones homogènes pré-calculées sur l'image. La première phase : le split, consiste à diviser chaque région non uniforme en quatre parties tandis que la seconde phase : le merge, consiste à fusionner toutes les régions uniformes adjacentes. On itère les deux étapes tant qu'il y a encore des régions non homogènes. Voici une illustration de ce concept : [26]

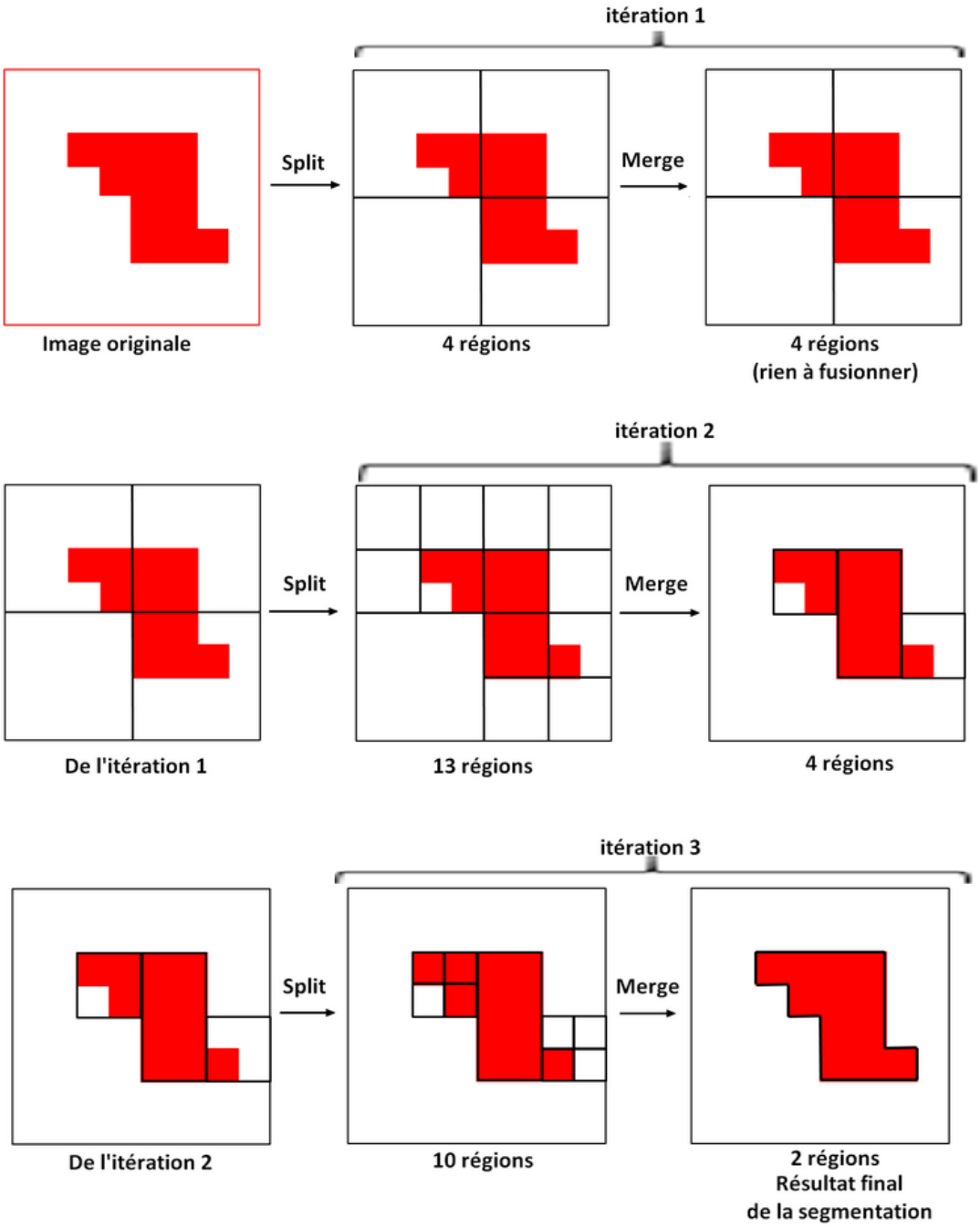


Figure 3.06 : *Principe du split and merge*

3.6.3 Segmentation par clustering

Le clustering, en segmentation d'image, a pour but de séparer différentes zones homogènes d'une image, afin d'organiser les objets en groupes (clusters) dont les membres ont en commun diverses propriétés (intensité, couleur, texture, etc). On peut diviser cette méthode de segmentation en deux catégories : la segmentation non supervisée, qui vise à séparer automatiquement l'image en clusters

naturels, c'est-à-dire sans aucune connaissance préalable des classes ; et la segmentation supervisée, qui s'opère à partir de la connaissance de chacune des classes définies par une approche probabiliste. Dans le cadre de notre étude, nous nous limiterons à méthode de segmentation non supervisée basée sur les K-means.

L'algorithme des K-means est une méthode que nous avons déjà introduite dans le chapitre précédent. Cependant, il est nécessaire de développer son rôle pour la segmentation d'image. En traitement d'image, l'algorithme des K-means a pour objectif de regrouper les observations en K-classes (ou clusters) dans lesquelles chaque observation (ici, un pixel caractérisé par son niveau de gris) appartient à la partition avec la moyenne la plus proche. Plus précisément, si on note S l'ensemble de toutes les partitions possibles des pixels en K ensembles $S_1, \dots, S_K$, on veut minimiser S sur la fonctionnelle :

$$\varepsilon\,(S_1, \dots, S_K) = \sum_{i=1}^{K} \sum_{x_j \in S_i} \left\| x_j - M_i \right\|^2 \tag{3.01}$$

où M_i est la moyenne des points de S_i. L'algorithme des K-means pour une image est la suivante :

Algorithme

Initialisation : Ensemble de K-moyennes $m_1^1, \dots, m_k^1$
(par exemple générées aléatoirement) ; $n = 1$

Affectation : On affecte chaque pixel à la classe dont la moyenne est la plus proche

$$S_i^n = \left\{ x_p : \|x_i - m_i^n\| \leq \left\| x_j - m_j^n \right\| \; \forall \; 1 \leq j \leq K \right\},$$

où chaque x_p est affecté à exactement une classe de S_n, même s'il peut être dans plusieurs classes.

Actualisation : On calcule les nouvelles moyennes, qui sont les centres des nouvelles classes.

$$m_i^{n+1} = \frac{1}{\left| S_i^n \right|} \sum_{x_j \in S_i^n} x_j$$

Arrêt quand les affectations ne changent plus.

Il y a un nombre fini de partitions possibles à K-classes (qui peut être très grand) et la fonctionnelle ε n'est pas convexe. On ne peut donc obtenir à priori qu'un minimum local. Cela permet d'affirmer que l'algorithme converge toujours et en temps fini (vers un minimum local). Toutefois, la convergence peut être lente et on peut rajouter des limiteurs au nombre d'itérations. De plus, la

solution fournie par cet algorithme dépend fortement de l'initialisation choisie. Les méthodes d'initialisation les plus utilisées sont des méthodes Forgy et le partitionnement aléatoire. La méthode Forgy effectue un choix aléatoire de K observations des données et les utilise comme moyennes (centres) initiales. Le partitionnement aléatoire assigne aléatoirement une classe à chaque observation et effectue l'étape d'actualisation c'est-à-dire le calcul des moyennes des éléments des classes ainsi définies. La méthode Forgy est néanmoins préférable pour l'algorithme des K-means. Le fait de devoir choisir à priori le paramètre K peut être aussi un inconvénient. La figure ci-dessous illustre les objets contenus dans les clusters à la fin de la segmentation ($K = 3$). [27] [28]

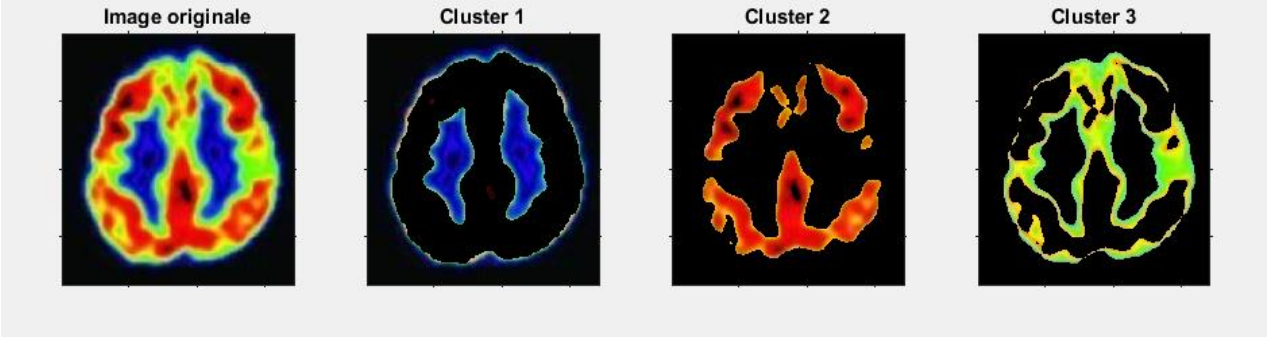


Figure 3.07 : *Segmentation K-means pour K=3*

3.7 Extraction des caractéristiques

Suite au processus de segmentation, la zone de l'image où se trouve véritablement le sujet d'intérêt est isolée. Les pixels appartenant à cette zone peuvent maintenant être analysés d'un point de vue de leur morphologie, de leur texture et de leur couleur en vue d'établir une relation entre ces caractéristiques et leur label correspondant.

La phase d'extraction de caractéristiques constitue généralement l'une des phases les plus importantes dans l'élaboration du système de classification d'images. Il s'agit en effet de déterminer un espace numérique de description dans lequel les données images seront projetées ce qui permet une séparation optimale des classes. Nous retrouvons des descripteurs de bas niveau s'intéressant à l'information contenue dans l'image au niveau du pixel et des descripteurs de plus haut niveau nécessitant une représentation intermédiaire de l'image plus adaptée. Cette description peut être locale (description de motifs de textures) ou globale (histogramme des orientations de la distribution des gradients de toute l'image) selon la nature de l'information à prélever et se fait à l'aide d'opérateurs ou de descripteurs. [20]

3.7.1 Extracteurs de bas niveau

Les extracteurs bas niveau permettent de traduire l'information présente au niveau du pixel, sans tenir compte des formes ou des patterns présents dans l'image. Parmi les caractéristiques extraites, nous retrouvons l'intensité du pixel brut, l'histogramme des intensités de pixels, les statistiques sur cet histogramme (moyenne, entropie, variance, coefficient d'aplatissement, asymétrie), la densité de pixels. On rencontre également des extracteurs de niveau intermédiaire traduisant des informations comme des liaisons entre les pixels, des distances, leur localisation, le contraste dans l'image. C'est le cas des statistiques à partir des matrices de cooccurrence, de simples gradients de l'image ou des statistiques sur des sorties de filtres appliquées à l'image via une caractérisation spectrale. Ce sont ces matrices de cooccurrence que nous utiliserons dans notre système. Nous présentons dans cette section les statistiques d'histogrammes et des matrices de cooccurrence.

3.7.1.1 Les statistiques d'histogramme

L'histogramme de l'image représente la distribution des intensités de pixels ; à chaque valeur d'intensité est associée le nombre de pixels ayant cette valeur dans l'image. Cette approche simple et générique est souvent utilisée pour l'analyse de texture. Nous donnons ci-dessous quelques mesures statistiques couramment utilisées, g étant la valeur de l'intensité (que l'on prend par exemple entre $[0: 255]$), a le nombre de niveaux de gris, P_g étant la fréquence de l'intensité g dans une région R de l'image : [20]

- la mesure de l'intensité moyenne :

$$\mu_R = \sum_{g=0}^{a-1} g.P_g \tag{3.02}$$

- la variance mesurant le contraste moyen :

$$\sigma_R^2 = \sum_{g=0}^{a-1} (g - \mu_R)^2 . P_g \tag{3.03}$$

- l'entropie mesurant le degré d'incertitude dans la distribution :

$$T_R = -\sum_{g=0}^{a-1} P_g \log_2 P_g \tag{3.04}$$

- l'énergie :

$$E_R = -\sum_{g=0}^{a-1} {P_g}^2 \tag{3.05}$$

- la dissymétrie :

$$D_R = \frac{1}{\sigma_R^3} \sum_{g=0}^{a-1} (g - \mu_R)^3 . P_g \tag{3.06}$$

- l'aplatissement :

$$A_R = \frac{1}{\sigma_R^4} \sum_{g=0}^{a-1} (g - \mu_R)^4 . P_g \tag{3.07}$$

3.7.1.2 Les statistiques des matrices de cooccurrence

La construction d'une matrice de cooccurrence (ou GLCM pour Gray Level Cooccurrence Matrix) des niveaux de gris sur une image permet de caractériser la texture par plusieurs statistiques. Les caractéristiques utilisées sont dérivées des travaux d'Haralick, Shanmugam et Dinstein (1973) sur cette matrice. Parmi les métriques utilisées on compte treize fonctions calculées à partir d'une matrice de dépendance des niveaux de gris P pour l'image I à traiter. Chaque élément de la matrice de dépendance P est fonction de quatre paramètres, le ton de gris de départ i, le ton de gris d'arrivée j, l'angle de voisinage θ et la distance de voisinage d.

Voici une brève présentation des calculs à effectuer pour extraire les caractéristiques liées à la matrice P. D'abord, il faut effectuer le calcul de la matrice de dépendance des niveaux de gris pour une distance d et un angle θ. La matrice P de dépendance des tons de gris est une matrice qui compte les occurrences des passages d'un ton de gris i vers un niveau de gris j au cours d'une distance d le long de l'angle θ. La matrice est généralement calculée pour des paramètres de distances et d'angles fixes. Ces paramètres sont donc à optimiser pour maximiser les performances de discrimination. Suite au dénombrement des transitions de niveau de gris, on obtient une matrice de dépendance spatiale des tons P qui couvre toutes les combinaisons de i et j possibles. Cette matrice est donc carrée et possède autant de lignes et de colonnes qu'il y a de niveaux de gris dans l'image considérée. Dans l'exemple qui suit, la matrice sera simplement de taille 4 par 4 puisque l'image considérée

possède une profondeur de couleur de 2 bits. La figure 3.08 présente un exemple simpliste de matrice P.

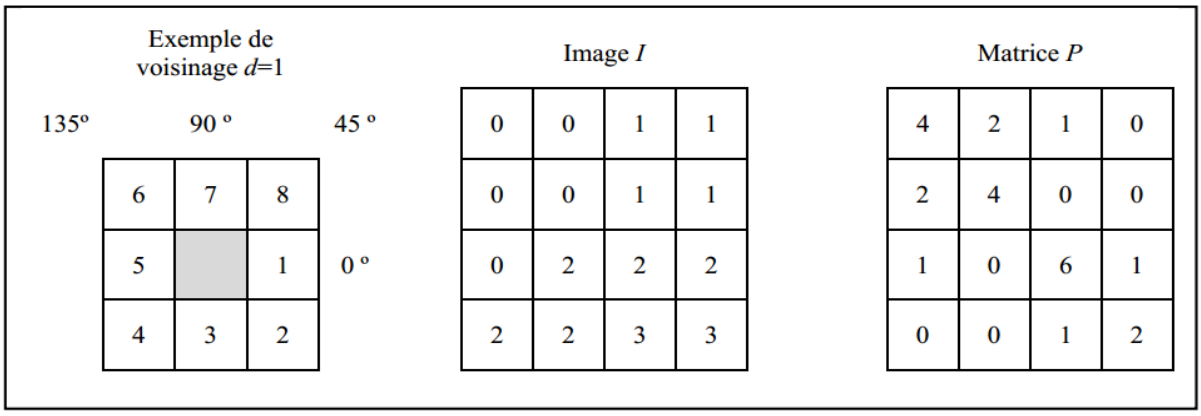


Figure 3.08 : *Exemple de matrice de cooccurrence avec d = 1 et θ = 0°*

Une fois la matrice de dépendance spatiale des niveaux de gris calculée, elle est normalisée de façon à ce que ses valeurs tiennent entre [0 ;2[. Il est maintenant possible d'extraire diverses statistiques qui permettent de caractériser de façon plus précise une texture donnée à partir de P. Les lignes suivantes présenteront une description très sommaire de ces statistiques. Une convention sera utilisée pour la présentation des fonctions. Voici la liste des variables utilisées :

- La variable $p(i,j)$ réfère aux éléments de la matrice P ;
- La variable $p_x(i)$ réfère à la probabilité marginale (à priori) obtenue en sommant les lignes de la matrice. Cette convention est généralisable pour $p_y(j)$ qui somme les colonnes de la matrice. Enfin, d'autres variables $p_{x+y}(k)$, $p_{x-y}(k)$ réfèrent à la somme et à la différence des deux vecteurs $p_x(i)$ et $p_y(j)$;
- La variable N_g réfère au nombre de niveaux de gris utilisés dans l'image ;
- Les variables μ_x, σ_x réfèrent à la moyenne et à l'écart type respectivement mesurés pour le vecteur $p_x(i)$. Ces nomenclatures sont généralisables pour $p_y(j)$ également ;
- Les variables HX et HY réfèrent à l'entropie mesurée sur $p_x(i)$ et $p_y(j)$ respectivement ;
- Les variables $HXY1$ et $HXY2$ sont calculées comme suit :

$$HXY1 = -\sum_i \sum_j p(i,j) \log\{p_x(i)p_y(j)\} \tag{3.08}$$

$$HXY2 = -\sum_i \sum_j p_x(i)p_y(j) \log\{p_x(i)p_y(j)\} \tag{3.09}$$

- La variable ϵ réfère à un très petit nombre réel positif. Elle est utilisée pour éviter qu'un logarithme ne soit calculé pour la valeur 0, qui est non définie pour cette fonction.

Voici maintenant les fonctions qui permettent l'extraction des descripteurs : [29]

- Le moment angulaire du second degré :

$$f_1 = \sum_i \sum_j (p(i,j))^2 \tag{3.10}$$

- Le contraste :

$$f_2 = \sum_{i=0}^{N_g-1} n^2 \left(\sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i,j) \right) \tag{3.11}$$

- La corrélation :

$$f_3 = \frac{\sum_i \sum_{j(ij)} p(i,j) - \mu_x \mu_y}{\sigma_x \sigma_y} \tag{3.12}$$

- La variance :

$$f_4 = \sum_i \sum_j (i - \mu) p(i,j) \tag{3.13}$$

- Le moment inverse :

$$f_5 = \sum_i \sum_j \frac{1}{1 + (i,j)^2} \, p(i,j) \tag{3.14}$$

- La moyenne de la somme :

$$f_6 = \sum_{i=2}^{2N_g} i p_{x+y}(i) \tag{3.15}$$

- La variance de la somme :

$$f_7 = \sum_{i=2}^{2N_g} \left(i - f_g\right)^2 p_{x+y}(i) \tag{3.16}$$

- L’entropie de la somme :

$$f_8 = \sum_{i=2}^{2N_g} p_{x+y}(i) \log(p_{x+y}(i) + \epsilon) \tag{3.17}$$

- L’entropie :

$$f_9 = \sum_{i} \sum_{j} p(i,j) \log(p(i,j) + \epsilon) \tag{3.18}$$

- La variance de la différence :

$$f_{10} = variance(p_{x+y}) \tag{3.19}$$

- L’entropie de la différence :

$$f_{11} = \sum_{i=2}^{2N_g} p_{x-y}(i) \log(p_{x-y}(i) + \epsilon) \tag{3.20}$$

- Les deux mesures de corrélation :

$$f_{12} = \frac{f_9 - HXY1}{\max(HX, HY)} \tag{3.21}$$

$$f_{13} = [1 - \exp(-2.0(HXY2 - f_9)]^2 \tag{3.22}$$

3.7.2 Extracteurs de plus haut-niveau

Les méthodes d’extraction de plus haut niveau tiennent compte des formes et des structures dans l’image, des relations spatiales entre les pixels ou ces structures. Les propriétés les plus recherchées dans ces extracteurs sont, outre leur pouvoir descriptif, l’invariance et la robustesse a différentes transformations pouvant affecter l’image. Ainsi, la description obtenue demeure relativement inchangée face à ces transformations pouvant plus ou moins affecter des contenus identiques ou similaires dans les images. On retrouve couramment les invariances au changement d’échelle, de perspective, aux transformations affines comme la translation, la rotation et la robustesse au changement de luminosité ou de contraste. Les approches standards de la littérature d’extraction de caractéristiques dans l’image sont principalement à base de descripteurs de textures. On retrouve particulièrement les extracteurs issus du détecteur de Harris, des approches Local Binary Pattern

(LBP) et Scale Invariant Feature Transform (SIFT), présentées dans la littérature comme des approches génériques et parmi les plus performantes. [20]

3.8 Apprentissage

A ce stade, nous avons isolé les régions de l'image qui nous intéressent grâce à la segmentation, et extrait les attributs caractérisant la texture des images. Pour notre système, nous avons choisi la matrice de cooccurrence pour cette extraction. Le SVM exige que chaque instance de données soit représentée comme un vecteur de nombres réels. C'est pourquoi nous avons normalisé les valeurs issues du GLCM entre l'intervalle [0 ;2[.

Nous devons aussi choisir la manière d'optimiser la performance relativement à la taille des données d'apprentissage et le nombre de caractéristiques. Ceci se fait en choisissant quel noyau de Kernel (vu dans le chapitre 2) il faut utiliser en premier. Ce noyau représente la mesure de similarité que l'on souhaite utiliser. Des études ont montré qu'il est mieux d'utiliser le noyau RBF pour obtenir des résultats acceptables. Cependant, si le nombre d'instances est inférieur au nombre de caractéristiques, ou si les deux sont extrêmement larges, il est préférable d'utiliser le SVM linéaire.

Dans la partie apprentissage, nous allons faire « apprendre » au système l'étiquette de chaque ensemble d'apprentissage à partir du SVM, étant un algorithme pour la classification supervisée. Ainsi, lorsque toutes les données ont été « apprises », nous pouvons construire notre dataset. Elle est de la même forme que la table suivante :

Id	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	Classe
1	f_1^1	f_2^1	f_3^1	f_4^1	f_5^1	f_6^1	f_7^1	f_8^1	f_9^1	f_{10}^1	f_{11}^1	f_{12}^1	f_{13}^1	$Classe_a$
2	f_1^2	f_2^2	f_3^2	f_4^2	f_5^2	f_6^2	f_7^2	f_8^2	f_9^2	f_{10}^2	f_{11}^2	f_{12}^2	f_{13}^2	$Classe_b$
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
$n > card(f_i)$	f_1^n	f_2^n	f_3^n	f_4^n	f_5^n	f_6^n	f_7^n	f_8^n	f_9^n	f_{10}^n	f_{11}^n	f_{12}^n	f_{13}^n	$Classe_a$

Tableau 3.01: *Forme du dataset*

avec f_i les caractéristiques issues de la matrice de cooccurrences ; $f_i^n \in [0; 2[$ et $Classe_x$, l'étiquette de chaque donnée image.

Notre dataset servira de modèle d'apprentissage. Il sera utilisé pour évaluer le comportement de notre système en lui présentant de nouvelles images, c'est-à-dire, à vérifier s'il arrive à attribuer la bonne étiquette à l'image dont des caractéristiques qui lui sont inconnues sont présentées. [30] [31]

3.9 Evaluation

La dernière étape nécessaire à la mise en place d'un système de classification d'images est l'évaluation de ce dernier. Toutefois, il ne s'agit pas encore de la mise en service du système comme telle. Avant de procéder, plusieurs autres calculs doivent être effectués pour traiter la sortie des algorithmes. Une question se pose : comment savoir si notre système classifie avec fiabilité les images qui lui sont présentées ? Les solutions proposées se basent sur le fait que les données utilisées pendant la phase d'apprentissage doivent être différentes de celles utilisées pour l'évaluation du résultat de cet apprentissage. Après avoir présenté notre ensemble de test au classifieur SVM, nous obtenons les étiquettes attribuées aux images test. Nous présentons dans cette partie un panorama des différents critères d'évaluation usuels.

3.9.1 Evaluation scalaire

Nous allons évoquer, dans cette partie, les méthodes et les métriques scalaires qui permettent d'évaluer et d'analyser les performances d'un système de classification. Parmi les méthodes les plus populaires, nous retrouvons le taux de bonne classification sans coût et le taux de bonne classification avec coût.

3.9.1.1 Taux de bonne classification sans coût

Il s'agit de l'indicateur le plus naturel et le plus évident permettant d'évaluer les performances d'un système de classification. Cette valeur, simple à calculer, correspond au nombre d'éléments correctement identifiés par le système. La définition du taux de bonne classification simplifié, sans la prise en compte du rejet est :

$$tbc_s = \frac{Nombre\ d'éléments\ correctement\ identifiés}{Nombre\ d'éléments\ total} \tag{3.23}$$

On obtient le taux d'erreur par :

$$te_s = 1 - tbc_s \tag{3.24}$$

Lorsque le rejet d'une forme est possible, un troisième taux, le « taux de rejet » est intégré. Il mesure le nombre d'éléments sur lesquels le système n'a pas pris de décision. Nous obtenons ainsi :

$$te_s = 1 - tbc_s - t_r \tag{3.25}$$

Le problème rencontré avec le taux de bonne classification sans coût, est qu'il s'agit d'une mesure faible, car elle ne tient pas compte de la distribution des classes et des coûts de classification. Pour remédier à cela, nous introduisons le taux de bonne classification avec coût. [32]

3.9.1.2 Taux de bonne classification avec coût

Il s'agit de l'évolution de la mesure précédente avec cette fois la prise en compte de la répartition des classes mais également des coûts de bonne et mauvaise classification. Plusieurs définitions existent en fonction des éléments qui sont pris en compte mais toutes utilisent la matrice de confusion (tableau 3.02), et la matrice des coûts classification (tableau 3.03).

Pour un système à C classes, nous définissons :

- les indices $i, j \in \{1, \dots, C\}$;
- w_i l'étiquette de la classe ;
- N_{w_i} le nombre d'éléments dans la classe i présents dans la base ;
- $\varepsilon_{i,j}$ le nombre d'éléments étiquetés de la classe i et identifiés comme des éléments de la classe j (dont la valeur est appelée score) ;
- $Cost_{i,j}$ le coût associé à $\varepsilon_{i,j}$.

Une matrice de confusion met en relation les décisions prises par le classifieur et les étiquettes des exemples. Associée à cette matrice, la matrice des coûts fait correspondre à chaque élément de la matrice de confusion un coût.

		Décision				
		w_1	w_2	…	w_C	
Etiquettes	w_1	$\varepsilon_{1,1}$	$\varepsilon_{2,1}$	…	$\varepsilon_{C,1}$	N_{w_1}
	w_2	$\varepsilon_{1,2}$	$\varepsilon_{2,2}$	…	$\varepsilon_{C,2}$	N_{w_2}
	…	…	…	…	…	…
	w_C	$\varepsilon_{1,C}$	$\varepsilon_{2,C}$	…	$\varepsilon_{C,C}$	N_{w_C}

Tableau 3.02: *Matrice de confusion*

		Décision			
		w_1	w_2	…	w_C
Etiquettes	w_1	$Cost_{1,1}$	$Cost_{2,1}$	…	$Cost_{C,1}$
	w_2	$Cost_{1,2}$	$Cost_{2,2}$	…	$Cost_{C,2}$
	…	…	…	…	…
	w_C	$Cost_{1,C}$	$Cost_{2,C}$	…	$Cost_{C,C}$

Tableau 3.03: *Matrice des coûts*

Nous définissons à partir ces matrices, le taux de bonne classification et le taux d'erreur avec la prise en compte de la distribution des classes par les équations suivantes :

$$tbc = \sum_{i=1}^{C} \left[\frac{\varepsilon_{i,j}}{N_{w_i}} \right] \times \sum_{i=1}^{C} N_{w_i} \times \frac{1}{C} \tag{3.26}$$

$$te = \sum_{i=1}^{C} \left[\frac{(1 - \varepsilon_{i,j})}{N_{w_i}} \right] \times \sum_{i=1}^{C} N_{w_i} \times \frac{1}{C} \tag{3.27}$$

$$te = 1 - tbc - tr \tag{3.28}$$

Pour aller plus loin, nous utilisons l'équation *3.26* en incorporant les coûts de mauvaise classification définis dans la matrice de coût.

$$tbc = \sum_{i=1}^{C} \left[\frac{\sum_{j=1, j \neq i}^{C} \varepsilon_{i,j} Cost_{i,j}}{N_{w_i}} \right] \times \sum_{i=1}^{C} N_{w_i} \times \frac{1}{C} \tag{3.29}$$

En incorporant les coûts de bonne classification :

$$tbc = \sum_{i=1}^{C} \left[\frac{(\varepsilon_{i,i} Cost_{i,i}) \sum_{j=1, j \neq i}^{C} \varepsilon_{i,j} Cost_{i,j}}{N_{w_i}} \right] \times \sum_{i=1}^{C} N_{w_i} \times \frac{1}{C} \tag{3.30}$$

Le problème qui se pose alors est la connaissance des coûts de classification. La solution proposée est d'utiliser des courbes plutôt que des valeurs scalaires pour évaluer les performances des classifieurs. Il y a deux types de courbes usuels à savoir : la courbe Précision–Rappel et la courbe ROC (« Receiver Operator Characteristic ») toutes basées sur la matrice de confusion. [32]

3.9.2 Evaluation multicritères

Les mesures que nous allons évoquer utilisent la matrice de confusion, du tableau 3.04, qui permet la différenciation des erreurs selon chaque classe en vue d'évaluer un classifieur.

	Décisions Positifs	Décisions Négatifs	
Etiquettes Positifs	**Vrai Positifs, TP**	**Faux Négatifs, FN**	Pos
Etiquettes Négatifs	**Faux Positifs, FP**	**Vrai Négatifs, TN**	Neg
	$PPos$	$PNeg$	N

Tableau 3.04: *Matrice de confusion*

Avec Pos : Nombre d'éléments étiquetés positifs dans la base ;

Neg : Nombre d'éléments étiquetés négatifs dans la base ;

$PPos$: Nombre d'éléments classés positifs ;

$PNeg$: Nombre d'éléments classés négatifs ;

N : Nombre d'éléments dans la base.

Définissons maintenant plusieurs mesures de manière formelle :

- Le taux de vrais positifs (« True positive rate ») :

$$tpr = \frac{TP}{Pos} = \frac{TP}{TP + FN} \tag{3.31}$$

- Le taux de vrais négatifs (« True negative rate ») :

$$tnr = \frac{TN}{Neg} = \frac{TN}{TN + FN} \tag{3.32}$$

- Le taux de faux positifs (« False positive rate ») :

$$fpr = \frac{FP}{Neg} = \frac{FP}{FP + TN} \tag{3.33}$$

- Le taux de faux négatifs (« False negative rate ») :

$$fnr = \frac{FN}{Pos} = \frac{FN}{FN + TN} \tag{3.34}$$

- Le taux de bonne classification ou l'exactitude (accuracy) :

$$acc = tbc = \frac{TP + TN}{N} \tag{3.35}$$

- La précision :

$$prec = \frac{TP}{PPos} = \frac{TP}{TP + FP} \tag{3.36}$$

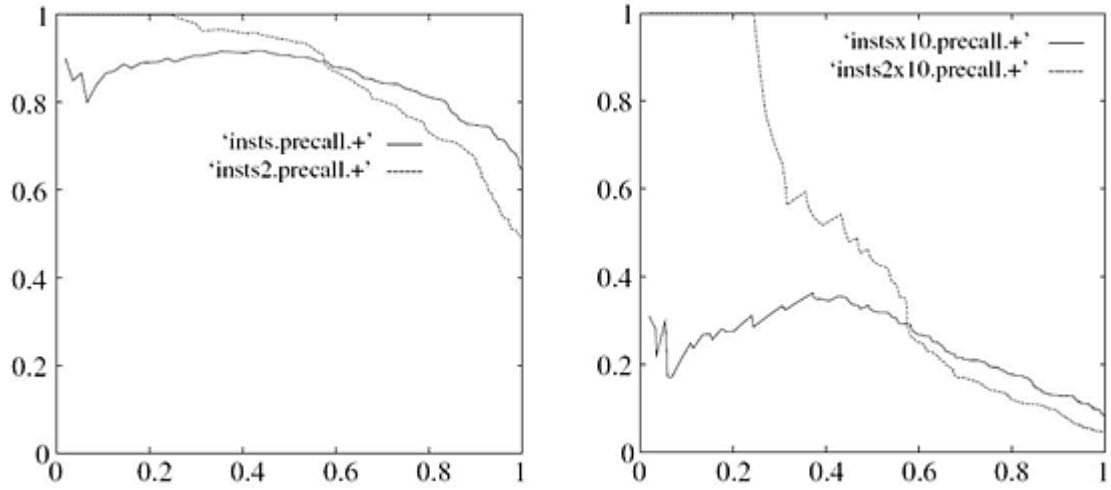
- Le rappel (recall) :

$$rec = tpr = \frac{TP}{Pos} = \frac{TP}{TP + FN} \tag{3.37}$$

Maintenant que nous avons caractérisé notre problème (estimation des taux de bonne et mauvaise classification, évaluation du type d'erreur, ...) via la matrice de confusion, nous allons représenter les performances des systèmes de classification à l'aide des courbes. [32]

3.9.2.1 La courbe Précision-Rappel

Cette courbe représente le rappel en fonction de la précision. Elle est utilisée pour visualiser la capacité du classifieur à attribuer la bonne étiquette aux données de test. Considérons la figure 3.09 qui présente deux classifieurs évalués par la courbe Précision-Rappel. Dans la figure 3.09a, la base de test est équilibrée dans la distribution des classes (1 : 1). Pour la courbe 3.09b, les mêmes classifieurs sont utilisés, mais cette fois, le nombre d'éléments négatifs est dix fois plus important (1 : 10).



(a) Courbes précision-Rappel, 1 :1 (b) Courbes précision-Rappel, 1 :10

Figure 3.09 : *Courbes Précision-Rappel vis-à-vis de la distribution des classes*

Nous constatons que ces courbes Précision-Rappel diffèrent sensiblement, ce qui indique une sensibilité de la représentation par rapport à la distribution. Il s'agit donc d'un problème important mettant en évidence la faiblesse de la courbe Précision–Rappel. De ce fait, nous devons faire appel à une autre représentation plus robuste vis-à-vis de la structure des bases de test.

3.9.2.2 La courbe ROC

La méthode d'analyse performante qui a été développée dans des environnements mal définis est le ROC. Elle s'est révélée très efficace pour donner une évaluation des classifieurs lorsque les coûts

associés à la matrice de confusion ne sont pas connus au moment de la construction du classifieurs. Considérons maintenant la figure suivante :

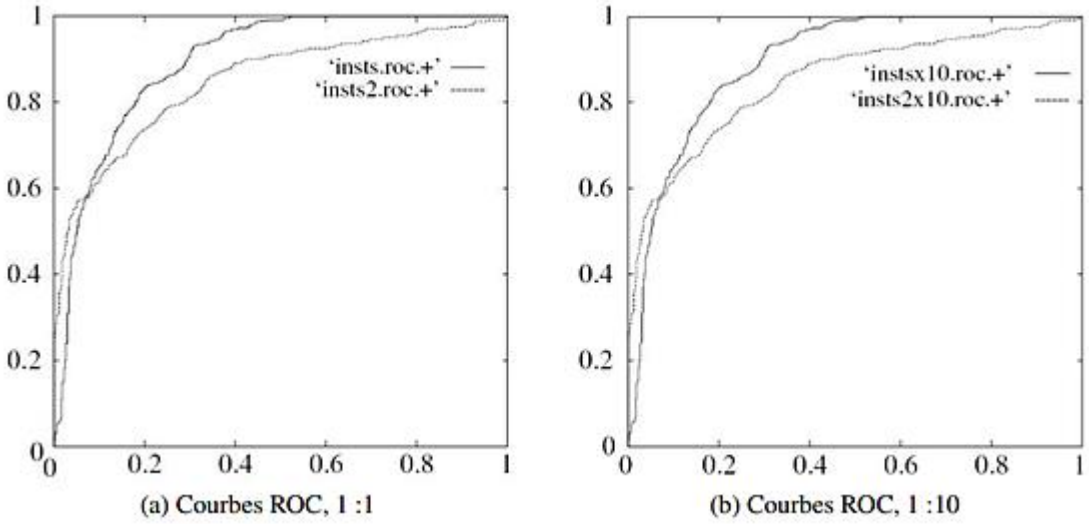


Figure 3.10 : *Courbes ROC vis-à-vis de la distribution des classes*

Dans la figure 3.10a, la base de test est équilibrée dans la distribution des classes (1 : 1). Pour la courbe 3.10b, les mêmes classifieurs sont utilisés, mais cette fois, le nombre d'éléments négatifs est dix fois plus important (1 : 10). Nous constatons que les courbes ROC sont identiques pour les deux cas. Si les proportions d'éléments positifs et/ou négatifs changent dans la base de test, la courbe ROC, elle, ne changera pas. Cette dernière est insensible aux modifications dans la distribution des classes. Elle est surtout utilisée pour comparer des classifieurs. [32]

3.10 Conclusion

Pour conclure, le SVM est une technique essentielle pour la classification d'images. Bien qu'il soit plus facile à utiliser que d'autres classifieurs, les utilisateurs qui ne sont pas familiers avec lui obtiennent souvent des résultats insatisfaisants au début. Ici, nous décrivons une « recette » qui donne généralement des résultats raisonnables. D'après les « ingrédients » proposés, nous pouvons déduire que les étapes de la conception d'un système de classification d'images correspondent à la méthode standard du CRISP-DM. Toutefois, il reste encore une étape qui n'a pas été « cuisinée » dans ce chapitre : c'est la phase déploiement de notre système. Nous attribuons cette tâche au chapitre suivant pour la simulation et l'interprétation des résultats obtenus à partir de notre système de classification d'images.

CHAPITRE 4
OUTIL DE CLASSIFICATION D'IMAGES APPLIQUE A L'IMAGERIE MEDICALE

4.1 Introduction

Dans le domaine de la santé et de l'imagerie médicale, les moyens d'acquisition et de visualisation d'images du corps humain sont de plus en plus variés et utilisés à grande échelle. Cela entraine une masse de données image importante dans les systèmes d'information médicales. Les techniques manuelles de photo-interprétation deviennent alors insuffisantes pour analyser et extraire de la connaissance à partir de ces images. Pour prendre en compte les difficultés engendrées par ces volumes d'informations, une automatisation de l'analyse d'image médicale est donc nécessaire. Dans le contexte de nos travaux, l'objectif est de classifier des images cérébrales pour l'aide au diagnostic de la Maladie d'Alzheimer (MA). Nous allons donc appliquer le système de classification que nous avons modélisé précédemment, au domaine de l'imagerie médicale.

4.2 Présentation de la réalisation

4.2.1 Architecture des systèmes d'information

Le système d'information est un ensemble d'éléments (personnel, matériel, logiciel...) permettant d'acquérir, traiter, mémoriser et communiquer des informations. Une application informatique est constituée de trois couches principales : l'interface avec les utilisateurs, les traitements et les données. Chaque couche peut se décomposer en deux modules distincts. Ainsi, six modules se dégagent d'une application informatique comme nous le montre la figure 4.01. Ces six modules peuvent être imbriqués ou répartis de différentes manières entre plusieurs machines physiques. La répartition de ces modules permet de distinguer les architectures applicatives suivantes :

- l'architecture à un niveau ou 1-tiers ;
- l'architecture à deux niveaux ou 2-tiers ;
- l'architecture à trois niveaux ou 3-tiers ;
- l'architecture à n niveau ou n-tiers.

De plus amples détails sur les caractéristiques de chacune des architectures peuvent être consultés en annexe. [33]

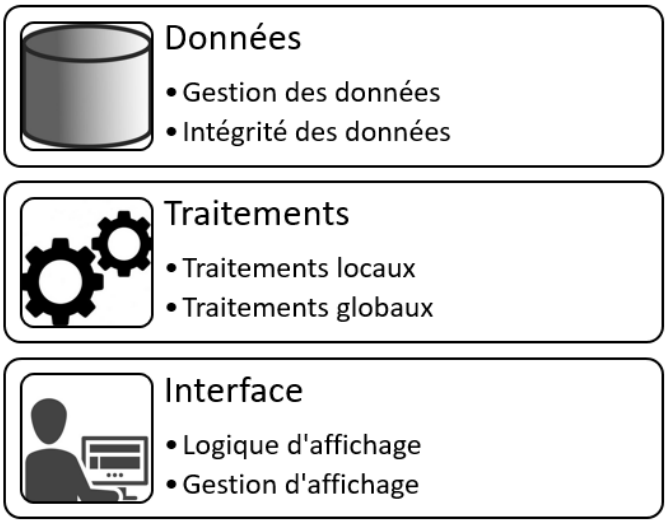


Figure 4.01 : *Décomposition d'une application*

4.2.2 Contexte

La Maladie d'Alzheimer constitue la cause la plus fréquente de démence ; elle représente un véritable problème de santé publique. La démence est une affection cérébrale qui entraine une altération de la mémoire et autres fonctions cognitives avec une répercussion fonctionnelle et sociale. Selon les dernières statistiques fournies par le ministère de la Santé publique, environ 500 personnes âgées sont atteintes de la MA à Madagascar. En outre, l'accroissement des nouvelles stratégies thérapeutiques suscite l'intérêt de détecter précocement les patients à risque d'attraper une MA. [34][35]

4.2.3 Objectifs

Dans le cadre de ce mémoire, nous n'allons pas réaliser la mise en place de toute une architecture matérielle avec les équipements nécessaires à fournir pour créer un système d'information. Pour la réalisation, l'architecture que nous avons adoptée est l'architecture 1-tiers où les couches applicatives se trouvent toutes sur une même machine. Nous nous limitons donc à la mise à disposition d'une base de données à laquelle les systèmes d'informations traitant l'imagerie médicale peuvent être reliés ; mais aussi à la conception d'un outil performant de classification d'images capable d'interagir avec la base pour la prédiction de la MA à partir d'images cérébrale. Pour atteindre ces objectifs, notre système aura l'obligation d'émettre un diagnostic différentiel, et d'éviter du mieux possible une erreur de classification qui mettra en danger la santé du patient concerné. De cette analyse d'image découlera une prise de décision par les spécialistes de l'imagerie médicale.

4.2.4 Données

Les données que nous allons utiliser sont issues d'un atlas d'enseignement de la neuro-imagerie intitulé : « The Whole Brain Atlas » qui est maintenu par le Dr K. A. Johnson, professeur de radiologie et de neurologie à la Harvard Medical School. Elles comprennent des images cérébrales de différents patients obtenues par diverses modalités et disponibles au public à des fins non commerciales. Les images que nous avons collectées concernent principalement les patients âgés entre 71 et 81 ans, ayant un risque de développer la MA mais aussi des volontaires du même âge et en bonne santé. Nous avons collecté 73 images représentant la coupe axiale du cerveau, que nous avons divisées en deux catégories : 43 images utilisées pour l'apprentissage contre 30 images utilisées pour le test. [36]

4.2.5 Logiciels utilisés

4.2.5.1 MATLAB

MATLAB (« MATrix LABoratory ») est un langage de programmation de haut niveau émulé par un environnement de développement du même nom. Développé par la société The MathWorks, MATLAB permet de manipuler des matrices, d'afficher des courbes et des données, de mettre en œuvre des algorithmes, de créer des interfaces utilisateurs, et peut s'interfacer avec d'autres langages comme le C, C++, Java, et Fortran. Il dispose de plusieurs fonctions mathématiques, scientifiques et techniques et comprend des bibliothèques, appelés Toolboxes (ou Boîtes à Outils) qui sont des collections de fonctions étendant l'environnement MATLAB pour résoudre des problèmes spécifiques. Les domaines couverts sont très variés et comprennent notamment le traitement du signal, l'automatique, le Machine Learning, la vision artificielle, les statistiques, etc. Notre application de classification d'images a été développée avec MATLAB. [37]

4.2.5.2 MySQL Workbench Community Edition

MySQL est un Système de Gestion de Bases de Données Relationnelles, qui utilise le langage SQL (Structured Query Language). C'est un des SGBDR les plus utilisés. Il est aujourd'hui développé par Oracle Corporation.

Il existe deux versions disponibles pour MySQL :

- la version open source, appelée Community Edition
- et la version commerciale payante : la Standard Edition.

MySQL Workbench permet de :

- modéliser des bases de données sous forme de schéma relationnel de type ER ;
- générer les tables de la base MySQL à partir du schéma ;
- élaborer des vues et des routines d'accès aux données de la base ;
- concevoir des scripts SQL pour traiter les données de la base ;
- documenter tous les modèles, routines et scripts ainsi réalisés.

Nous utilisons MySQL Workbench pour stocker, modifier et mettre à jour notre dataset et également pour gérer l'authentification des utilisateurs.

Il est à remarquer que notre application ne fonctionne que si les conditions suivantes sont remplies sous MySQL :

- Nom d'utilisateur : root ;
- Mot de passe : 1234 ;
- Adresse et port utilisés : localhost 3306 ;
- Nom du schéma : utilisateur ;
- Nom de la table : utilisateur ;
- Colonnes :
    - idutilisateur (de type int, à incrémenter automatiquement) ;
    - login (de type varchar) ;
    - password (de type varchar). [38]

4.3 Modèle mathématique

Pour la préparation de notre outil de classification d'images, nous procédons en deux phases :

- la phase d'apprentissage pour la construction du dataset,
- et la phase de test pour évaluer notre système.

Les modèles mathématiques utilisés sont représentés sous forme d'algorithmes et seront détaillés dans les sections qui suivent.

4.3.1 Phase d'apprentissage

La phase d'apprentissage comprend plusieurs étapes qui sont représentées sur la figure ci-dessous :

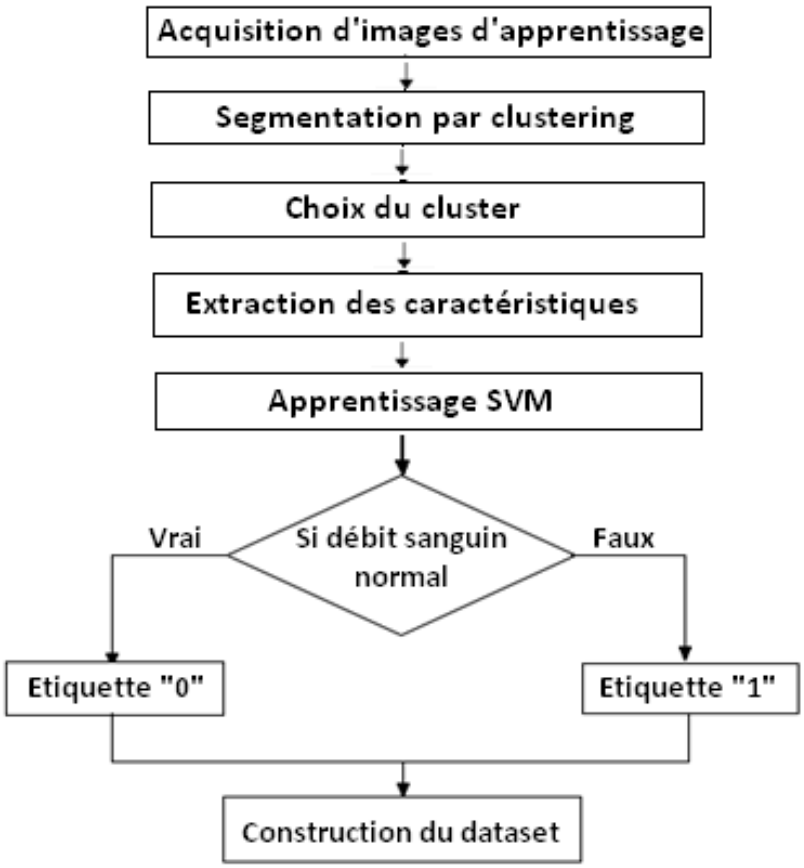


Figure 4.02 : *Phase d'apprentissage*

4.3.1.1 Acquisition des images

Les images collectées pour l'apprentissage sont issues de la scintigraphie cérébrale acquise par la Tomographie d'Emission Mono Photonique (TEMP). La scintigraphie cérébrale est une branche de l'imagerie nucléaire (appelée également imagerie fonctionnelle) qui a pour but d'étudier le fonctionnement du cerveau. Elle est fondée sur la détection des radiations émises par une substance radioactive (un radiotraceur) introduite dans l'organisme et présentant une affinité particulière pour un organe ou un tissu, ici le cerveau. Après administration du médicament, le patient est placé sous une caméra appelée caméra à scintillation ou encore gamma-caméra, qui détecte et enregistre l'émission d'un seul rayonnement : la détection est donc monophotonique. Elle permet l'obtention d'images planaires. Ces concepts sont illustrés par la figure 4.03. Les indications à la scintigraphie cérébrale sont la détection et le diagnostic différentiel de démences dégénératives telle que la maladie d'Alzheimer. [39]

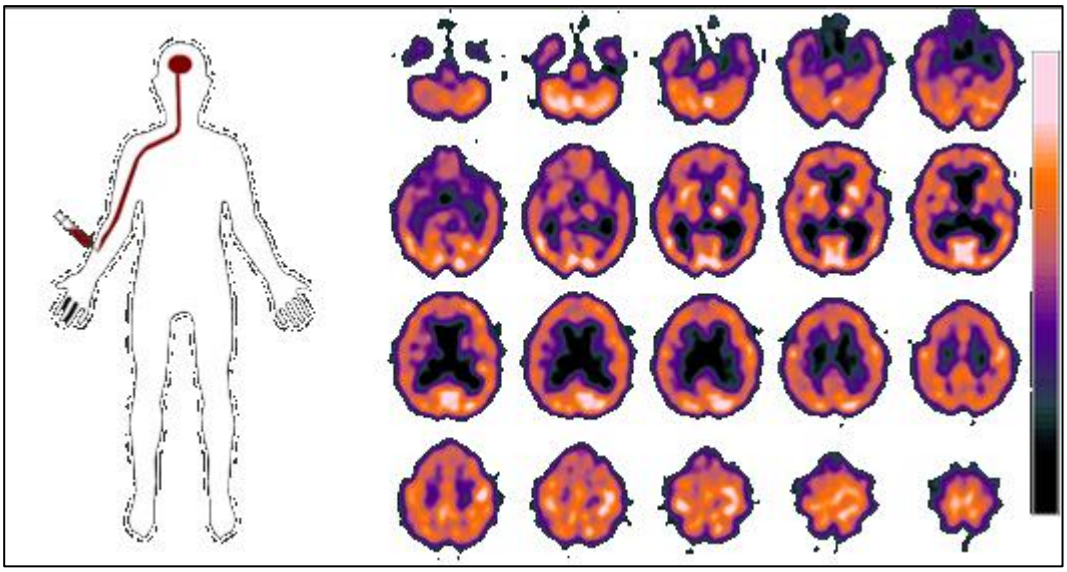


Figure 4.03 : *Administration du radiotraceur (à gauche) et distribution du traceur dans le cerveau (à droite)*

4.3.1.2 Segmentation et choix du cluster

La MA est caractérisée par une hypoperfusion, c'est-à-dire une diminution du débit sanguin cérébral (DSC) appelée ischémie. Dans l'imagerie TEMP, le flux sanguin vers le cerveau est représenté sur une échelle de couleur, où les zones sombres symbolisent une ischémie et les zones claires, un bon débit sanguin (voir figure 4.04). La segmentation par clustering à partir de l'algorithme des K-means est utilisée pour séparer les régions qui présente un bon DSC de celles présentant un DSC anormal.

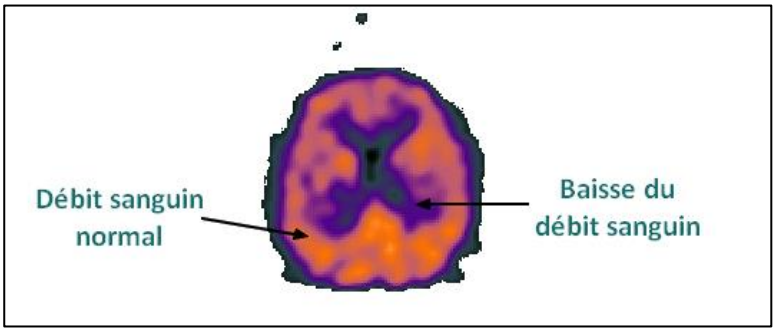


Figure 4.04 : *Zones claires vs zones sombres*

Après la segmentation, les objets contenus dans le cluster pour une perfusion cérébrale normale et celle anormale sont représentés sur la figure suivante :

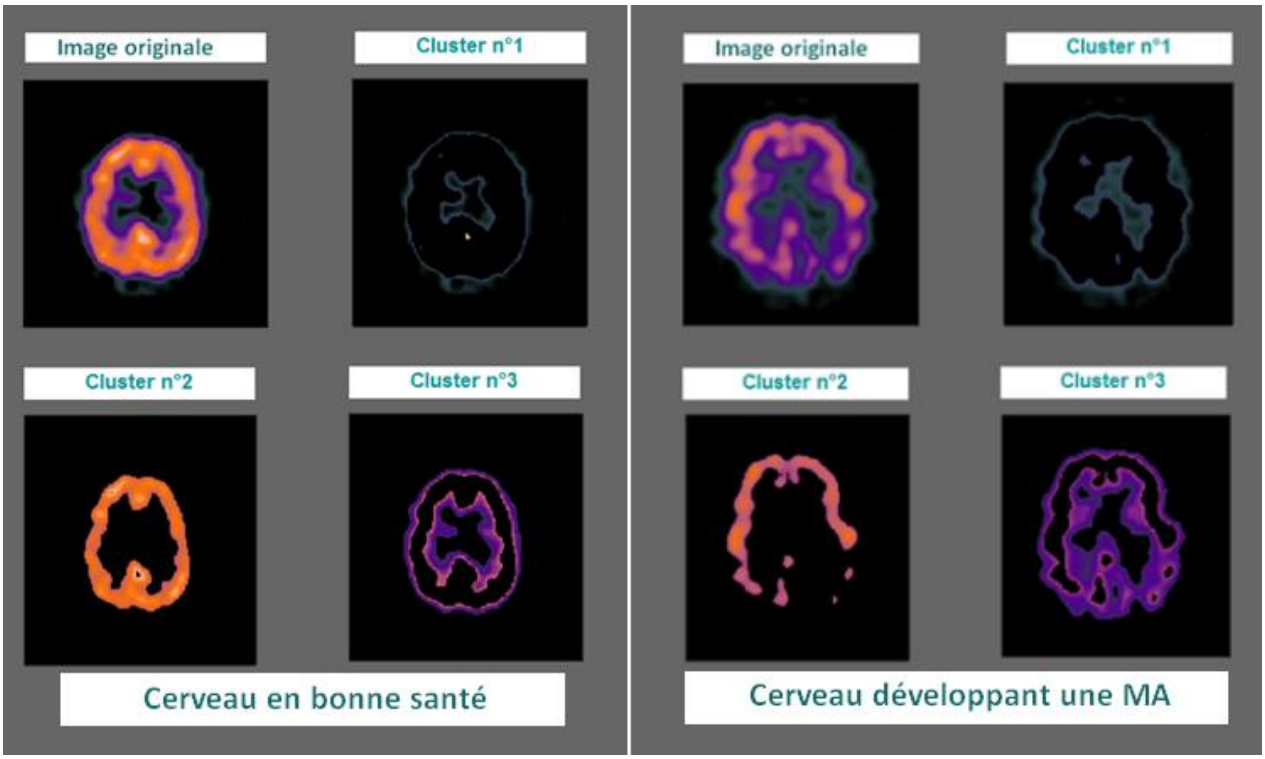


Figure 4.05 : *Perfusion cérébrale normale (à gauche) et anormale (à droite)*

D'après la figure 4.05, le cluster qui nous intéresse est celui qui contient la zone sombre : ici, il s'agit du cluster n°3 pour chaque cas. Nous remarquons que la diminution du DSC est très importante pour un cerveau affecté par la MA par rapport à celle du cerveau en bonne santé.

4.3.1.3 Extraction des caractéristiques, apprentissage et dataset

Après avoir effectué le choix du cluster, les caractéristiques (utilisées pour l'apprentissage par le classifieur) sont extraites à partir des statistiques des matrices de cooccurrence des niveaux de gris. Sur les 43 images collectées : 21 présentent la MA tandis que les 22 restantes représentent un cerveau en bonne santé. Les étiquettes « 0 » et « 1 » sont respectivement attribuées pour la classe en bonne santé et celle présentant une MA. Chaque image est alors décrite par la valeur des 13 fonctions issues du GLCM.

Notons f_i les treize fonctions descripteurs de chaque image. Nous avons constaté que f_1, f_4, f_{10} et f_{13} ont les mêmes valeurs pour les deux classes. La distinction des deux classes par le SVM se fera alors à partir des neuf fonctions restantes, grâce à la fonction « svmtrain » et « fitcsvm » sous MATLAB. La valeur de ces neuf fonctions est présentée sur le tableau ci-dessous :

Fonctions	Etiquette "0"		Etiquette "1"	
	Valeur minimale	Valeur maximale	Valeur minimale	Valeur maximale
f_2	0.0008	0.0009	0.0009	
f_3	0.0007	0.0148	0.0155	0.0239
f_5	0.0073	0.0008	0.0008	0.0008
f_6	0.0290	0.0381	0.0376	0.0469
f_7	0.0009	0.0016	0.0021	0.0028
f_8	0.0028	0.0048	0.0052	0.0066
f_9	0.7356	1.2042	1.0893	1.6919
f_{11}	0.0085	0.0205	0.0047	0.0078
f_{12}	0.0026	0.0042	0.0018	0.0024

Tableau 4.01: *Valeurs des caractéristiques pour l'apprentissage*

Pour de nouvelles images présentées à notre classifieur, le SVM leur attribuera l'étiquette « 0 » ou « 1 » selon que la valeur des caractéristiques extraites soit plus proche de celle d'une classe ou de l'autre. On peut désormais construire notre dataset. Un extrait est donné par le tableau 4.02 suivant, mais l'intégrale est mise en annexe.

0.0001	0.0009	0.0008	0.0010	0.0111	0.0345	0.0014	0.0039	1.0199	0.0010	0.0097	0.0032	0.2550	0
0.0001	0.0009	0.0007	0.0010	0.0135	0.0378	0.0016	0.0044	1.2042	0.0010	0.0098	0.0028	0.2550	0
0.0001	0.0009	0.0007	0.0010	0.0148	0.0381	0.0019	0.0048	1.1588	0.0010	0.0085	0.0026	0.2550	0
0.0001	0.0009	0.0006	0.0010	0.0177	0.0404	0.0022	0.0057	1.2863	0.0010	0.0063	0.0022	0.2550	1
0.0001	0.0009	0.0006	0.0010	0.0188	0.0411	0.0024	0.0057	1.2346	0.0010	0.0058	0.0021	0.2550	1
0.0001	0.0009	0.0008	0.0010	0.0103	0.0325	0.0014	0.0037	0.8667	0.0010	0.0119	0.0032	0.2550	1

Tableau 4.02: *Notre dataset*

Il est composé d'une matrice de 43 lignes correspondant au nombre d'images et 14 colonnes correspondant aux 13 caractéristiques plus l'étiquette de chaque image. Ce dataset sera enregistré dans le SGBDR MySQL Workbench et les systèmes d'informations pourront y accéder à partir de notre outil de classification.

4.3.2 Phase de test

Les 30 images utilisées pour le test correspondent à 11 images issues de perfusions cérébrales normales et 19 images provenant de perfusions caractéristiques de MA. Voici donc l'algorithme utilisé pour effectuer le test :

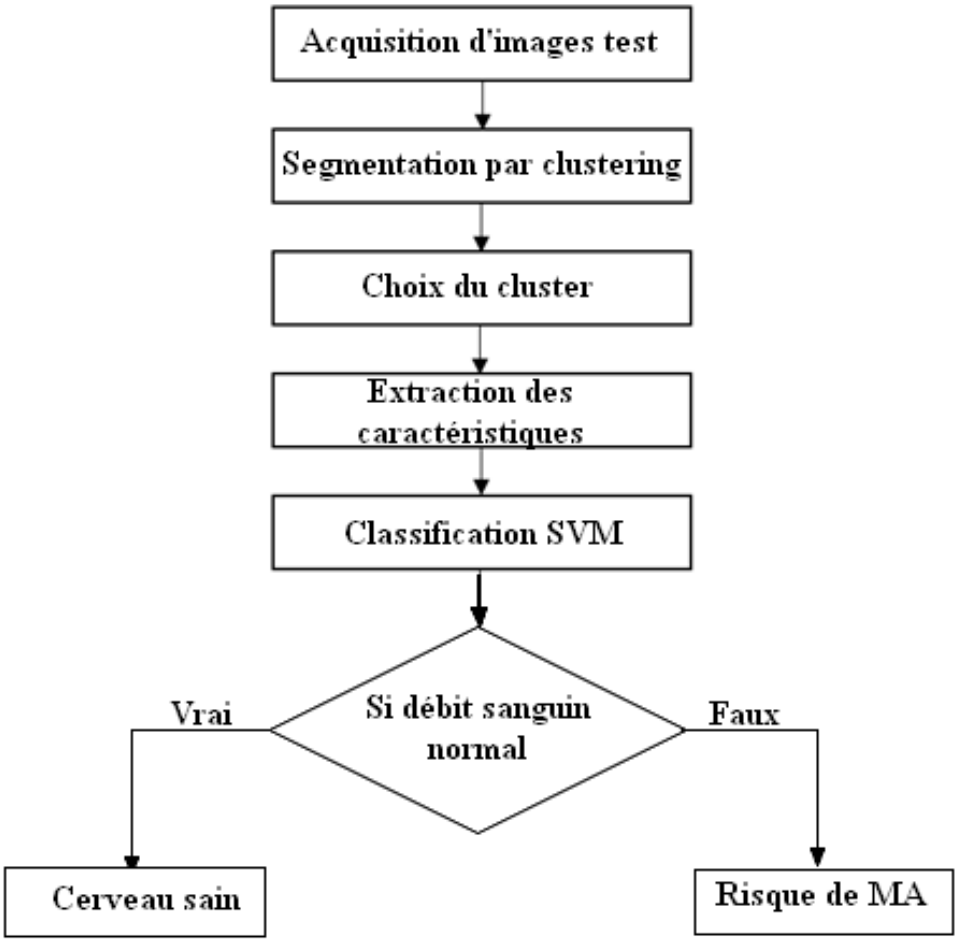


Figure 4.06 : *Phase de test*

On effectue la même procédure que la phase d'apprentissage mais au lieu de l'apprentissage, on procède à la classification par le SVM. Cela se fait à partir des fonctions « svmclassify » et « predict » sous MATLAB. A la sortie, une étiquette est attribuée à chaque image test selon la classe prédite par le classifieur.

4.4 Résultats

Après avoir classifié toutes les images de la base de test, seulement 4 images étiquetées « 1 » ont été classées « 0 » ce qui donne un taux de bonne classification de 86.67%. Notons que le danger engendré par l'attribution de la classe « 0 » à une image étiquetée « 1 » est moins fatal par rapport à l'attribution de la classe « 1 » à une image étiquetée « 0 ». Cela reviendrait à diagnostiquer une MA chez un patient en bonne santé. La matrice de confusion obtenue est la suivante :

	Décisions « 1 »	Décisions « 0 »	
Etiquettes « 1 »	**TP** = 15	**FN** = 4	$Pos = 19$
Etiquettes « 0 »	**FP** = 0	**TN** = 11	$Neg = 11$
	$PPos = 15$	$PNeg = 15$	N = 30

Tableau 4.03: *Notre matrice de confusion*

Voyons maintenant les mesures de performances de notre système :

- Le taux de vrais positifs :

$$tpr = \frac{TP}{Pos} \times 100 = 78.95\% \quad (4.01)$$

- Le taux de vrais négatifs (« True negative rate ») :

$$tnr = \frac{TN}{Neg} \times 100 = 100\% \quad (4.02)$$

- Le taux de faux positifs (« False positive rate ») :

$$fpr = \frac{FP}{Neg} \times 100 = 0\% \quad (4.03)$$

- Le taux de faux négatifs (« False negative rate ») :

$$fnr = \frac{FN}{Pos} \times 100 = 21.05\% \quad (4.04)$$

- Le taux de bonne classification ou l'exactitude (accuracy) :

$$acc = tbc = 86.67\% \quad (4.05)$$

Ce qui nous donne la courbe de précision-rappel (PR) sur la figure 4.07 issue des fonctions suivantes et dont la construction sera expliquée en annexe :

- La précision :

$$prec = \frac{TP}{PPos} \quad (4.06)$$

- Le rappel (recall)

$$rec = tpr = \frac{TP}{Pos} \tag{4.07}$$

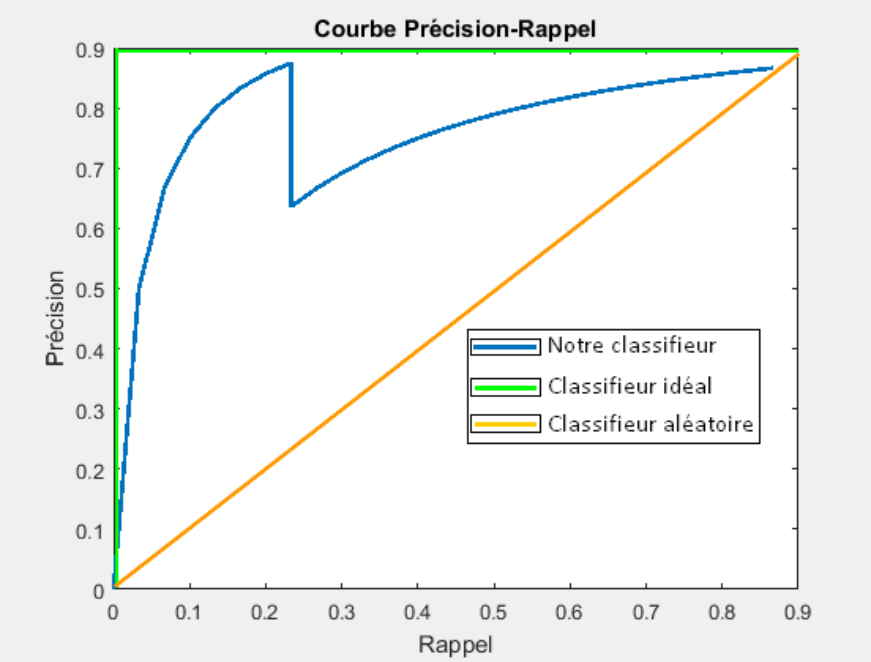


Figure 4.07 : *Courbe de Précision-Rappel*

4.5 Interprétations

Il faut savoir qu'un modèle qui classe les individus au hasard dans l'un ou l'autre classe, aura une courbe PR équivalente à celle en orange sur la figure ci-dessus. De l'autre côté, un modèle qui prédit très bien, aura une courbe équivalente à celle en vert. Ainsi, plus l'air sous la courbe PR d'un modèle est grand, plus ce modèle est bon. Or, la courbe PR de notre classifieur se rapproche étroitement de la courbe du classifieur idéal. Nous avons donc modélisé un système de classification valide.
De plus, le tpr et tnr étant largement supérieurs aux fpr et fnr indiquent que la probabilité qu'une image appartienne à une classe sachant qu'elle devrait y appartenir est supérieur à la probabilité qu'une image n'appartienne pas à la classe à laquelle elle a été attribuée. Il y a donc un risque moindre que notre système classifie une image issue d'un cerveau sain en une image développant la MA. Les obligations qui ont été fixées à la première étape de la conception du système sont ainsi atteintes, nous pouvons maintenant déployer notre système.

4.6 Mise en service de l'outil de classification

Pour mettre en service notre outil de classification, nous avons créé une application exécutable sous Windows. MATLAB permet de déployer les applications sous la forme d'un exécutable grâce à un compilateur d'application intégré à celui-ci. Après l'exécution de notre application, la fenêtre d'accueil suivante s'affiche :

Figure 4.08 : *Fenêtre d'accueil*

La fenêtre d'accueil comporte trois boutons :

- Le bouton Quitter affiche une boîte de dialogue pour confirmer la fermeture de l'application comme nous pouvons voir sur la figure suivante :

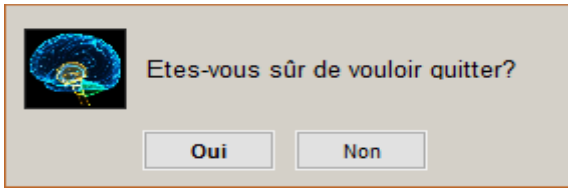


Figure 4.09 : *Boîte de dialogue « quitter »*

En cliquant sur Oui, l'application se ferme, tandis qu'en cliquant sur Non ou en appuyant la touche « Esc », l'activité précédente reprend.

- Le bouton A propos affiche dans une autre fenêtre (voir figure 4.10) une description de l'application avec le contact pour tout renseignement sur l'outil.

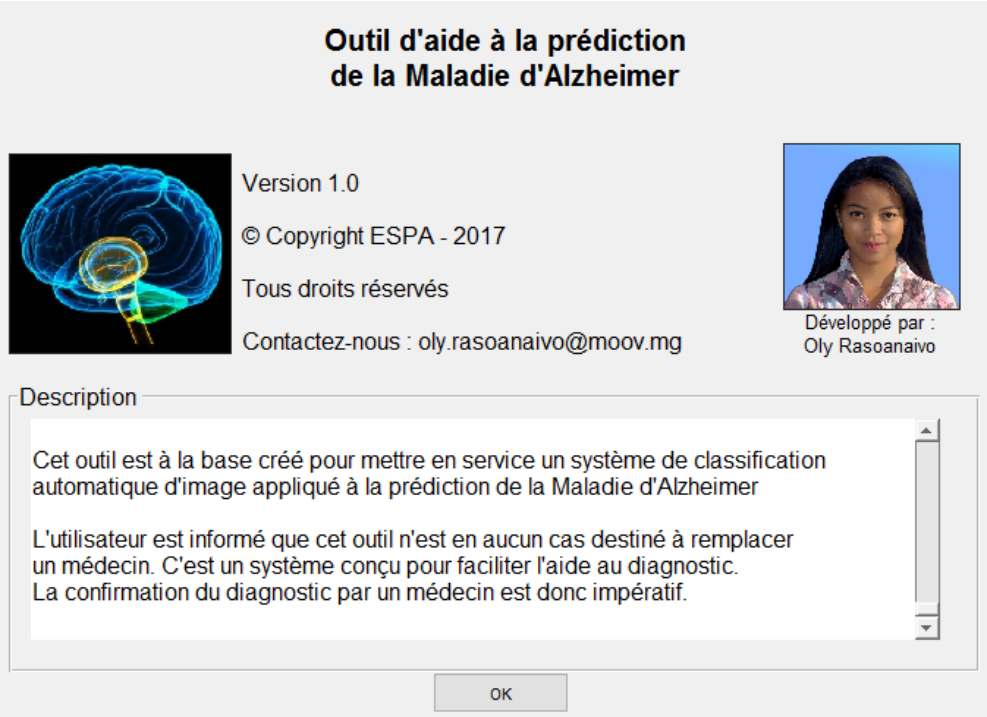


Figure 4.10 : *Fenêtre à propos*

- Le bouton S'authentifier renvoie une fenêtre pour s'identifier comme illustré sur la figure 4.11. Si l'identifiant saisi n'existe pas dans notre base de données « utilisateur » ou si le mot de passe est incorrect, il faut entrer l'identifiant et/ou le mot de passe correct. Sinon, il faut créer un nouveau compte en cliquant sur le bouton Créer un compte. Autrement, on peut accéder à la fenêtre de prédiction.

Figure 4.11 : *Fenêtre authentification*

Pour créer un nouveau compte, il faut appuyer sur le bouton Créer un compte et une autre fenêtre s'affiche (voir figure 4.12). Si l'identifiant saisi existe déjà dans notre base, il faut en choisir un autre. Et si le mot de passe à confirmer est différent du mot de passe saisi en premier, l'application demande de ressaisir le mot de passe.

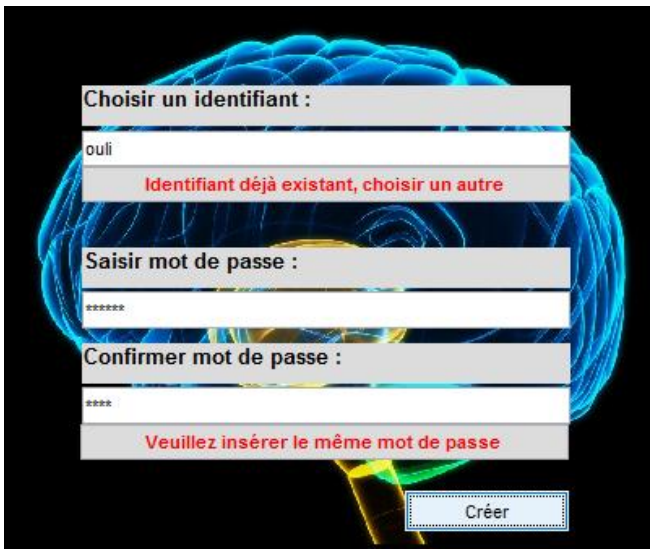


Figure 4.12 : *Fenêtre de création de compte*

Lorsque toutes les conditions sont remplies, on peut avoir accès à la fenêtre de prédiction de la figure 4. 13 après avoir cliqué sur Créer. Les valeurs entrées dans les champs sont alors enregistrées dans la base de données.

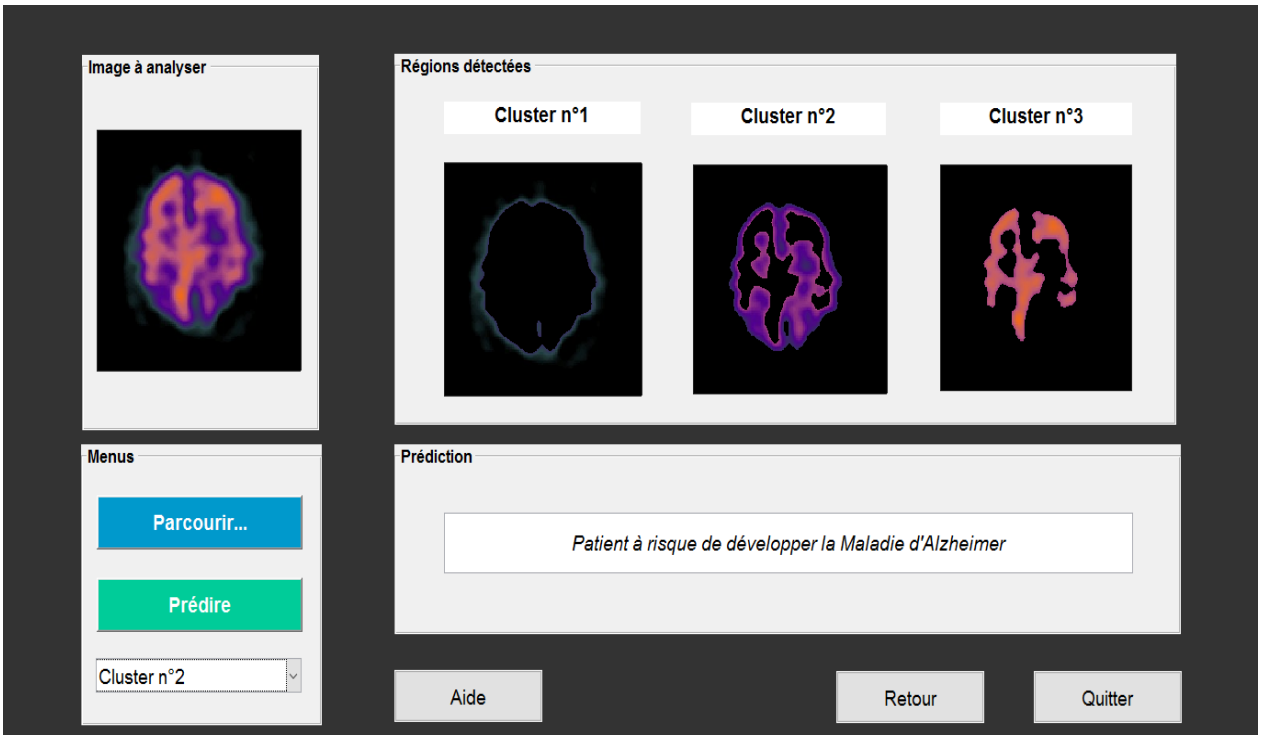


Figure 4.13 : *Fenêtre prédiction*

Cette fenêtre est composée de plusieurs boutons, de panneau de visualisation d'image, d'un menu pop-up et d'un éditeur de texte.

- Le bouton 

permet de choisir une image qui s'affiche sur le panneau « Image à analyser » après sélection ;
- Le bouton 

segmente l'image sélectionnée et affiche les objets contenus dans chaque cluster sur le panneau « Régions détectées » ;
- Le menu pop-up 

sert à choisir quel cluster contient la zone présentant une diminution du débit sanguin cérébral ;
- Le résultat de la prédiction s'affiche sur l'éditeur de texte.
Si l'image est classifiée normale, il affiche « Patient en bonne santé », sinon, il affiche « Patient à risque de développer la Maladie d'Alzheimer » ;
- Le bouton 

affiche un tutoriel sur l'utilisation de l'outil de prédiction représenté sur la figure suivante :

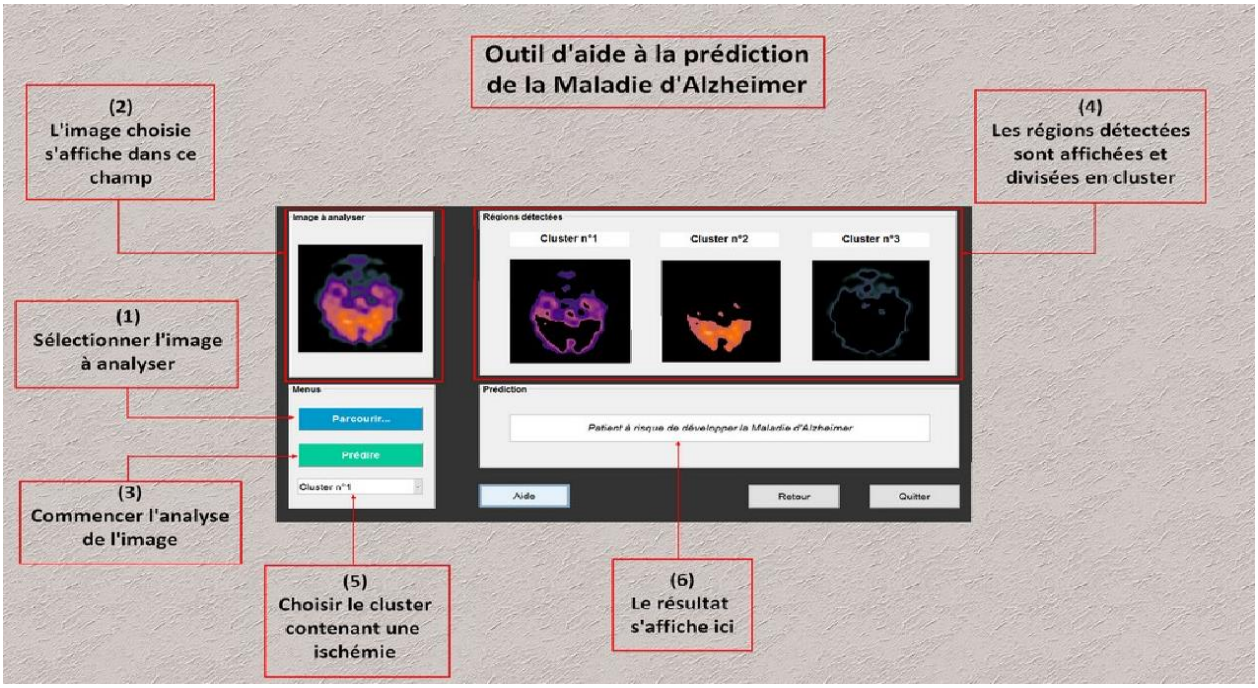


Figure 4.14 : *Fenêtre aide*

- Le bouton Retour permet de revenir à la fenêtre d'acceuil ;
- Le bouton Quitter affiche la même boîte de dialogue comme sur la figure 4.09 pour confirmer la fermeture de l'application.

4.7 Conclusion

Pour conclure, dans ce chapitre nous avons pu appliquer notre système de classification d'images à la prédiction de la Maladie d'Alzheimer. Etant donné qu'elle affecte la vie quotidienne des personnes âgées et de celles qui les entourent, il devient nécessaire de trouver un moyen pour la détection précoce de cette maladie. Dans notre cas, il s'agit de détecter les zones présentant une forte diminution du DSC en couplant le traitement d'image avec la classification par SVM. Le nombre d'images dans la base d'apprentissage, pour chaque classe est bien balancé ce qui a entrainé un comportement adéquat du classifieur envers les images de test. D'ailleurs, les indicateurs de performance issus de l'évaluation montrent que l'outil conçu répond aux contraintes de fiabilités qui sont prédéfinies et permet d'atteindre les objectifs fixés. Notre outil peut donc être intégré aux systèmes d'information disposant de grande base d'images médicales.

CONCLUSION GENERALE

En somme, donner du sens aux montagnes de données collectées jour après jour n'est pas chose facile. Ce mémoire nous permet de conclure que le couplage des techniques d'apprentissage automatique avec le Data Mining simplifie et accélère l'analyse des données. Nous avons choisi l'algorithme des Séparateurs à Vaste Marge qui offre une solution appropriée au problème de classification automatique de données, et particulièrement des données image.

La précision de ce classifieur permet d'éviter de se fier à l'instinct d'un expert sur un volume important de données à analyser et de diminuer la durée de cette analyse. Ceci dit, l'automatisation du Data Mining ne signifie en aucun cas la disparition des experts. Ils ont pour rôle de superviser les tâches effectuées lors du Data Mining.

Notre système de classification a été appliqué à l'analyse d'images médicales issues de Tomographie d'Emission Mono Photonique pour la prédiction de la Maladie d'Alzheimer. Etant donné que cette maladie constitue un fléau pour la personne atteinte et son entourage, l'application qu'on a développée permet de détecter automatiquement si une image présente les caractéristiques de la maladie.

Dans l'outil que nous avons proposé, après la segmentation par l'algorithme des K-means, il faut choisir quel cluster représente la région de l'image qui nous intéresse, ce qui entraine une automatisation partielle du processus. En outre, l'outil que nous avons conçu se base sur un algorithme supervisé nécessitant la connaissance à priori des classes de chaque image constituant la base d'apprentissage et de test.

Les perspectives et futurs travaux relatifs à notre thème sont donc : la recherche d'autres méthodes pour extraire les régions d'intérêts d'une image afin d'automatiser totalement le processus de classification ; l'utilisation d'un algorithme d'apprentissage non supervisé pour pouvoir classifier les images sans connaissance préalable des classes, ce qui diminuerai le temps de préparation des données à analyser ; le déploiement de l'application sur Internet pour cibler plus d'utilisateurs ; l'utilisation de l'outil de classification dans d'autres domaines que celui de la santé, notamment pour la classification d'images satellitaires, l'indexation d'image, la reconnaissance faciale, la reconnaissance de caractères et bien d'autres.

ANNEXE 1
ARCHITECTURE DES SYSTEMES D'INFORMATION

A1.1 L'architecture à un niveau

Dans une application à un niveau ou 1-tiers, les trois couches applicatives s'exécutent sur le même ordinateur et elles ne peuvent être facilement distinguées ; on parle alors d'informatique centralisée. Historiquement, les applications sur site central furent les premières à proposer un accès multi-utilisateur. Dans ce type d'architecture, les utilisateurs se connectent aux applications exécutées par le serveur central (le mainframe) à l'aide de terminaux passifs se comportant en esclaves (Figure A.01). C'est le serveur central qui prend en charge l'intégralité des traitements, y compris l'affichage qui est simplement déporté sur des terminaux passifs.

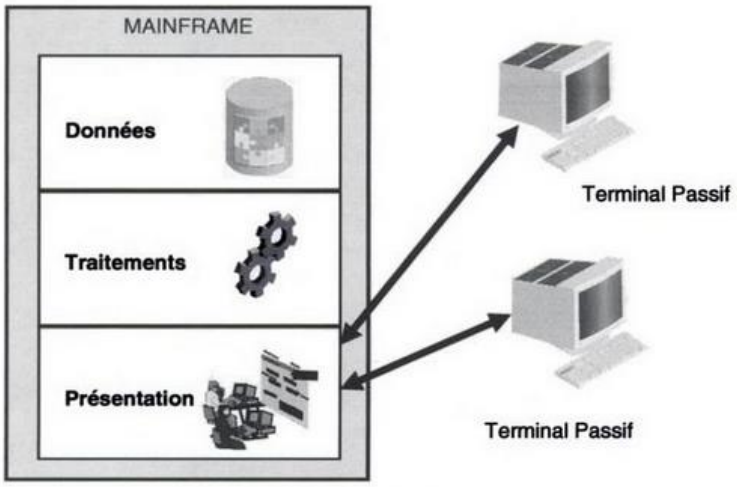


Figure A1.01 : *Architecture à un niveau*

Ce type d'organisation permet une administration facile et offre une haute disponibilité.
L'émergence des interfaces utilisateur de type multifenêtrage, a démodé les applications mainframe car elles exploitaient exclusivement une interface utilisateur en mode caractère.

A1.2 L'architecture à deux niveaux

L'organisation du client-serveur de données est la suivante. Diverses stations de travail, les clients peuvent dialoguer avec un serveur de données sur lequel sont installés le moteur de la base de données et les données à gérer. Les postes clients ne possèdent donc pas la base de données en local, mais transmettent au serveur de données des requêtes (le plus fréquemment en utilisant le langage SQL). Ce type d'application offre à l'utilisateur une interface riche, tout en garantissant la cohérence des données qui restent gérées de façon centralisée.

L'architecture à deux niveaux ne peut fonctionner que grâce à une couche particulière, le logiciel médiateur ou middleware qui assure la liaison transparente entre le client et le serveur à travers un réseau : transport des requêtes, harmonisation des types de données, respect des protocoles, gestion de la performance…

A1.3 L'architecture à trois niveaux

Les limites de l'architecture à deux niveaux proviennent en grande partie de la nature du client utilisé et du middleware :

- Le poste client est complexe et non standard (même s'il s'agit presque toujours d'un PC sous Windows) ;
- Le middleware entre un client et serveur n'est pas standard.

La solution consiste à utiliser un poste client simple communicant avec le serveur par le biais d'un protocole standard.

Dans ce but, l'architecture à trois niveaux applique les principes suivants :

- Les données sont toujours gérées de façon centralisée ;
- La présentation est toujours prise en charge par le poste client ;
- La logique applicative est prise en charge par un serveur intermédiaire.

L'architecture à trois niveaux sépare l'application en trois niveaux de service distincts :

- Premier niveau : le poste client gère l'affichage et les traitements locaux (contrôles de saisie, mise en forme de données, …).
- Deuxième niveau : le service applicatif s'occupe des traitements applicatifs globaux.
- Troisième niveau : le SGBD prend en charge les services de base de données.

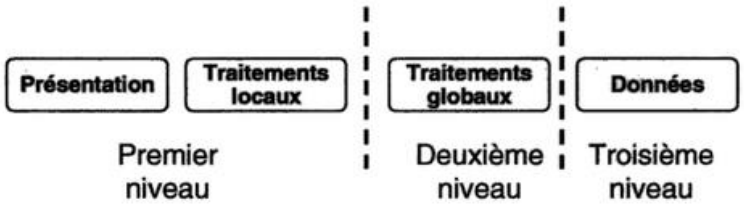


Figure A1.02 : *Architecture à trois niveaux*

Tous ces niveaux étant indépendants, ils peuvent être implantés sur des machines différentes, donc :

- Le poste client est moins sollicité et donc moins évolué car il ne supporte plus l'ensemble des traitements ;

- Les traitements applicatifs peuvent être partagés ou regroupés (le serveur d'application peut s'exécuter sur la même machine que le SGBD) et ainsi les ressources présentes sur le réseau sont mieux exploitées ;
- La centralisation de certains traitements permet d'améliorer leur fiabilité et leurs performances.

Dans le cadre d'un Intranet, le poste client prend la forme d'un simple navigateur Web, le service applicatif est assuré par un serveur http et la communication avec le SGBD met en œuvre les mécanismes bien connus des applications client-serveur de la première génération (figure A.03).

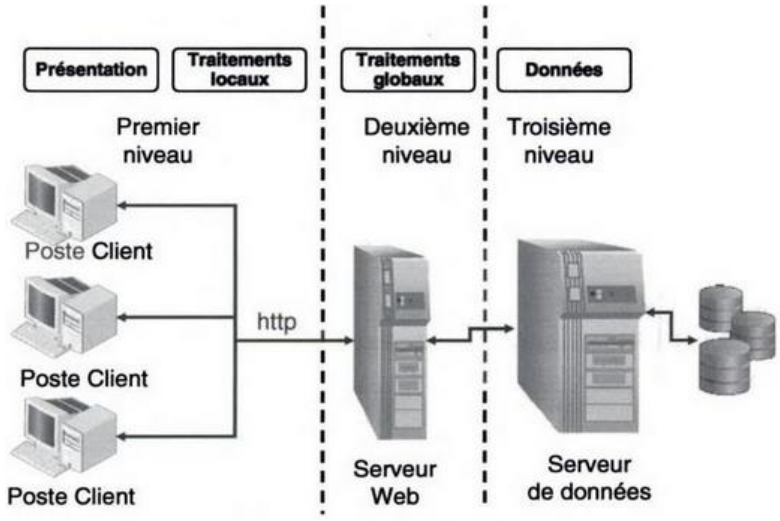


Figure A1.03 : *Architecture 3 tiers (intranet)*

A1.4 Les architectures à n-niveaux

L'architecture à n niveaux met en œuvre une approche objet pour offrir une plus grande souplesse d'implémentation et faciliter la réutilisation des développements. Théoriquement, ce type d'architecture supprime tous les inconvénients des architectures précédentes :

- Elle permet l'utilisation d'interfaces utilisateurs riches ;
- Elle sépare nettement tous les niveaux de l'application ;
- Elle offre de grandes capacités d'extension.

Contrairement à ce que pourrait laisser penser sa terminologie, une architecture à n niveaux n'a pas pour but de multiplier les couches applicatives, qui restent au nombre de trois (données, traitement, interface) mais de distribuer l'application sur plusieurs services s'appuyant sur des composants. En séparant, par exemple, les services métiers, des services de persistance, de suivi de sessions, … il est alors possible de mieux maitriser l'évolution du système d'information.

ANNEXE 2

INTEGRALE DU DATASET

Voici l'intégrale de notre dataset composée de 43 lignes représentant chaque image de la base d'apprentissage et 14 colonnes correspondant aux 13 fonctions issues des statistiques du GLCM et aux étiquettes de chaque image.

0.0001	0.0009	0.0007	0.0010	0.0135	0.0378	0.0016	0.0044	1.2042	0.0010	0.0098	0.0028	0.2550	0
0.0001	0.0009	0.0007	0.0010	0.0148	0.0381	0.0019	0.0048	1.1588	0.0010	0.0085	0.0026	0.2550	0
0.0001	0.0009	0.0008	0.0010	0.0117	0.0354	0.0015	0.0042	1.0853	0.0010	0.0115	0.0031	0.2550	0
0.0001	0.0009	0.0008	0.0010	0.0108	0.0341	0.0014	0.0040	1.0142	0.0010	0.0127	0.0033	0.2550	0
0.0001	0.0009	0.0007	0.0010	0.0129	0.0374	0.0015	0.0042	1.1970	0.0010	0.0106	0.0029	0.2550	0
0.0001	0.0009	0.0007	0.0010	0.0129	0.0365	0.0016	0.0043	1.1138	0.0010	0.0100	0.0029	0.2550	0
0.0001	0.0009	0.0007	0.0010	0.0122	0.0355	0.0016	0.0041	1.0165	0.0010	0.0103	0.0029	0.2550	0
0.0001	0.0009	0.0007	0.0010	0.0122	0.0355	0.0016	0.0041	1.0165	0.0010	0.0103	0.0029	0.2550	0
0.0001	0.0008	0.0008	0.0010	0.0110	0.0336	0.0014	0.0039	0.9319	0.0010	0.0118	0.0031	0.2550	0
0.0001	0.0009	0.0008	0.0010	0.0097	0.0326	0.0012	0.0034	0.8831	0.0010	0.0140	0.0035	0.2550	0
0.0001	0.0009	0.0008	0.0010	0.0100	0.0318	0.0014	0.0035	0.8124	0.0010	0.0132	0.0033	0.2550	0
0.0001	0.0008	0.0008	0.0010	0.0109	0.0340	0.0014	0.0038	0.9765	0.0010	0.0128	0.0033	0.2550	0
0.0001	0.0008	0.0008	0.0010	0.0091	0.0321	0.0011	0.0034	0.9071	0.0010	0.0163	0.0037	0.2550	0
0.0001	0.0009	0.0008	0.0010	0.0101	0.0334	0.0013	0.0036	0.9640	0.0010	0.0142	0.0035	0.2550	0
0.0001	0.0008	0.0008	0.0010	0.0098	0.0325	0.0013	0.0036	0.8967	0.0010	0.0143	0.0035	0.2550	0
0.0001	0.0008	0.0008	0.0010	0.0103	0.0332	0.0013	0.0036	0.9198	0.0010	0.0137	0.0034	0.2550	0
0.0001	0.0008	0.0008	0.0010	0.0093	0.0316	0.0012	0.0035	0.8594	0.0010	0.0153	0.0036	0.2550	0
0.0001	0.0008	0.0008	0.0010	0.0091	0.0314	0.0012	0.0034	0.8517	0.0010	0.0157	0.0036	0.2550	0
0.0001	0.0008	0.0008	0.0010	0.0093	0.0317	0.0012	0.0034	0.8607	0.0010	0.0151	0.0036	0.2550	0
0.0001	0.0009	0.0008	0.0010	0.0091	0.0317	0.0012	0.0033	0.8546	0.0010	0.0155	0.0036	0.2550	0
0.0001	0.0009	0.0008	0.0010	0.0079	0.0299	0.0010	0.0030	0.7611	0.0010	0.0183	0.0040	0.2550	0
0.0001	0.0009	0.0008	0.0010	0.0073	0.0290	0.0009	0.0028	0.7356	0.0010	0.0205	0.0042	0.2550	0
0.0001	0.0009	0.0007	0.0010	0.0155	0.0376	0.0021	0.0052	1.0893	0.0010	0.0078	0.0024	0.2550	1
0.0001	0.0009	0.0006	0.0010	0.0178	0.0408	0.0022	0.0058	1.3449	0.0010	0.0066	0.0022	0.2550	1
0.0001	0.0009	0.0006	0.0010	0.0221	0.0457	0.0026	0.0063	1.6605	0.0010	0.0053	0.0019	0.2550	1
0.0001	0.0009	0.0006	0.0010	0.0219	0.0457	0.0025	0.0063	1.6919	0.0010	0.0054	0.0019	0.2550	1
0.0001	0.0009	0.0005	0.0010	0.0239	0.0469	0.0028	0.0066	1.7114	0.0010	0.0047	0.0018	0.2550	1
0.0001	0.0009	0.0006	0.0010	0.0231	0.0458	0.0027	0.0064	1.5717	0.0010	0.0048	0.0018	0.2550	1
0.0001	0.0009	0.0006	0.0010	0.0203	0.0437	0.0024	0.0059	1.4593	0.0010	0.0056	0.0020	0.2550	1
0.0001	0.0009	0.0006	0.0010	0.0181	0.0413	0.0022	0.0056	1.3521	0.0010	0.0063	0.0022	0.2550	1
0.0001	0.0009	0.0007	0.0010	0.0155	0.0378	0.0021	0.0052	1.1428	0.0010	0.0075	0.0024	0.2550	1
0.0001	0.0009	0.0006	0.0010	0.0187	0.0418	0.0023	0.0058	1.3743	0.0010	0.0061	0.0021	0.2550	1
0.0001	0.0009	0.0007	0.0010	0.0172	0.0412	0.0021	0.0055	1.4135	0.0010	0.0074	0.0024	0.2550	1
0.0001	0.0009	0.0006	0.0010	0.0179	0.0422	0.0021	0.0055	1.4667	0.0010	0.0071	0.0023	0.2550	1
0.0001	0.0009	0.0006	0.0010	0.0202	0.0444	0.0024	0.0059	1.5853	0.0010	0.0061	0.0021	0.2550	1
0.0001	0.0009	0.0006	0.0010	0.0197	0.0429	0.0024	0.0059	1.4097	0.0010	0.0059	0.0021	0.2550	1
0.0001	0.0009	0.0006	0.0010	0.0194	0.0437	0.0023	0.0056	1.4818	0.0010	0.0059	0.0021	0.2550	1
0.0001	0.0009	0.0006	0.0010	0.0199	0.0428	0.0024	0.0059	1.3908	0.0010	0.0055	0.0020	0.2550	1
0.0001	0.0009	0.0006	0.0010	0.0183	0.0416	0.0023	0.0056	1.3666	0.0010	0.0062	0.0022	0.2550	1
0.0001	0.0009	0.0006	0.0010	0.0198	0.0429	0.0024	0.0059	1.4288	0.0010	0.0056	0.0020	0.2550	1
0.0001	0.0009	0.0006	0.0010	0.0177	0.0404	0.0022	0.0057	1.2863	0.0010	0.0063	0.0022	0.2550	1
0.0001	0.0009	0.0006	0.0010	0.0188	0.0411	0.0024	0.0057	1.2346	0.0010	0.0058	0.0021	0.2550	1
0.0001	0.0009	0.0008	0.0010	0.0103	0.0325	0.0014	0.0037	0.8667	0.0010	0.0119	0.0032	0.2550	1

Tableau A2.01 : *Dataset*

ANNEXE 3
OBTENTION D'UNE COURBE DE PRECISION-RAPPEL POUR UN SYSTEME DE CLASSIFICATION BINAIRE

La courbe de Précision-Rappel est une mesure de performance d'un système de classification binaire où les classes sont divisées en deux : la classe positive et la classe négative.
Soit une base de 5 images toutes étiquetées « 1 » (classe positive) représentant par exemple des images médicales de patients atteints de la Maladie d'Alzheimer. La classe négative représentant des images médicales de patients en bonne santé est étiquetée « 0 ».
Représentons les étiquettes et le résultat de la classification automatique de ces images par un système de classification sur le tableau ci-dessous :

Images	**Etiquette**	**Classe**
Image1	« 1 »	« 1 »
Image2	« 1 »	« 0 »
Image3	« 1 »	« 1 »
Image4	« 1 »	« 1 »
Image5	« 1 »	« 0 »

Tableau A3.01 : *Etiquettes et classes de chaque image*

On considère d'abord la première image. A ce stade, le système a bien classé une image parmi les cinq. Donc, on a un taux de rappel de $0{,}2$ pour $\frac{1}{5}$. La précision est alors 1 image étiquetée positive et classée positive sur 1 image $\left(\frac{1}{1}\right)$ qui donne 1. Le point à mettre sur la courbe est de coordonnées $(0.2\ ;\ 1)$.

On considère ensuite les deux premières images présentées à l'entrée du système (Image1 et Image2). Le système a bien classé Image1 parmi les cinq (tandis qu'il n'a pas bien classé Image2). Donc à ce point, on a le même rappel (toujours $\frac{1}{5} = 0{,}2$), mais la précision devient $\frac{1}{2}$ puisqu' on a 1 image étiquetée positive et classée positive sur 2 images. Ainsi le point a comme coordonnées $(0.2\ ;\ 0.5)$.

Puis considérons les trois premières images à l'entrée du système (Image1, Image2 et Image3). Ici, on a un rappel de $\frac{2}{5}$ puisque Image1 et Image3 sont bien classées. La précision devient $\frac{2}{3}$ puisqu' on a 2 image étiquetées positive et classées positive sur 3 images. Le point est de coordonnées $(0.4\ ;\ 0{,}67)$.

Considérons maintenant les images (Image1, Image2, Image3 et Image4). On obtient un rappel de $\frac{3}{5}$ puisque Image1, Image3 et Image4 sont bien classées. La précision est de $\frac{3}{4}$. Le point est placé aux coordonnées $(0{,}6\ ;\ 0{,}75)$.

Enfin, considérons les 5 images de la base. Le rappel est toujours de $\frac{3}{5}$ puisque 3 images sur les 5 ont été bien classées. La précision est aussi de $\frac{3}{5}$ donc le point se trouve aux coordonnées $(0{,}6\ ;\ 0{,}6)$.

On obtient ainsi la courbe de précision en fonction du rappel sur la figure suivante :

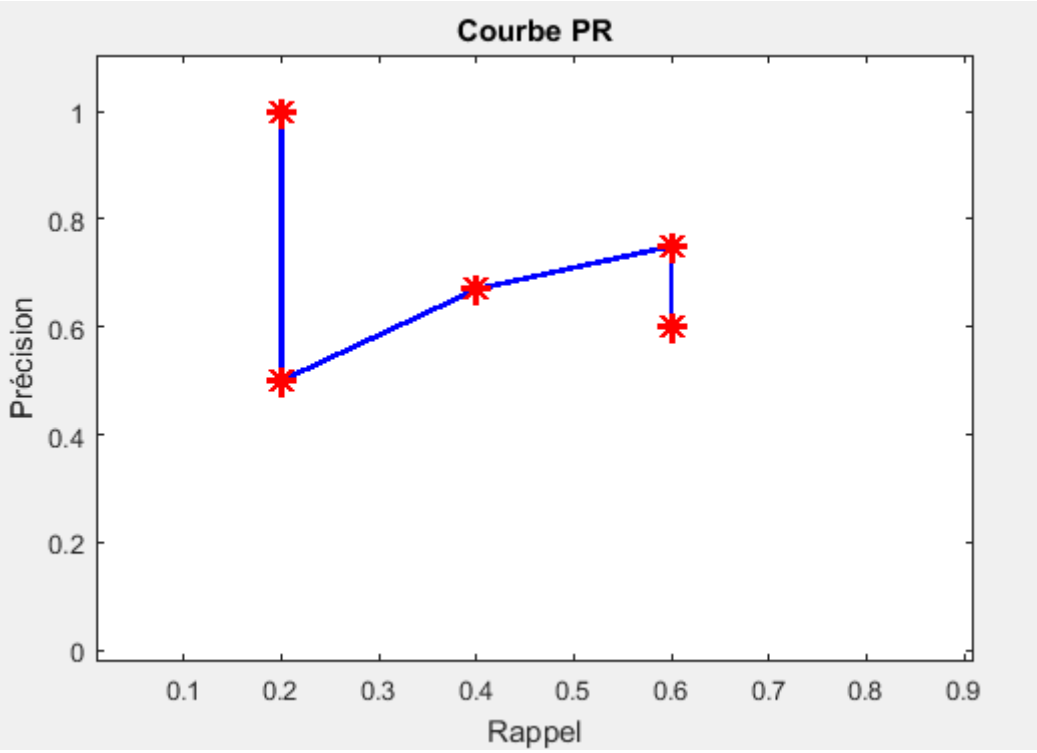


Figure A3.01 : *Exemple de courbe de Précision – Rappel*

ANNEXE 4

EXTRAIT DU CODE SOURCE DE L'APPLICATION

```
seg_img = segmented_images{1}; %choix du cluster 1 après segmentation
if ndims(seg_img) == 3
   img = rgb2gray(seg_img); %conversion de l'image couleur en niveaux de gris
end

%extraction des caractéristiques
glcms = graycomatrix(img);
stats = graycoprops(glcms,'Contrast Correlation Energy Homogeneity');
Contrast = stats.Contrast;
Correlation = stats.Correlation;
Energy = stats.Energy;
Homogeneity = stats.Homogeneity;
Mean = mean2(seg_img);
Standard_Deviation = std2(seg_img);
Entropy = entropy(seg_img);
RMS = mean2(rms(seg_img));
Variance = mean2(var(double(seg_img)));
a = sum(double(seg_img(:)));
Smoothness = 1-(1/(1+a));
Kurtosis = kurtosis(double(seg_img(:)));
Skewness = skewness(double(seg_img(:)));
m = size(seg_img,1);
n = size(seg_img,2);
in_diff = 0;
for i = 1:m
    for j = 1:n
        temp = seg_img(i,j)./(1+(i-j).^2);
        in_diff = in_diff+temp;
    end
end
IDM = double(in_diff);
caracteristiques = [Contrast,Correlation,Energy,Homogeneity, Mean,
Standard_Deviation, Entropy, RMS, Variance, Smoothness, Kurtosis, Skewness,
IDM];
caracteristiques= caracteristiques/1.0e+03;

%classification SVM
load ('data.mat','dataset');%base d'apprentissage
training=dataset;
load ('data.mat','label'); %étiquettes
groupe=label;
svmModel=fitcsvm(training,groupe); %phase d'apprentissage
[classe,~]=predict(svmModel,caracteristiques); %phase de prédiction
```

BIBLIOGRAPHIE

[1] J. Han, M. Kamber, *« Data Mining : Concepts and Techniques »*, Editions Morgan Kaufmann, 2006

[2] N.R. Pal, L. Jain, *« Advanced Techniques in Knowledge Discovery and Data Mining »*, Editions Springer, 2005

[3] K. El Himdi, *« Introduction au Data Mining »*, www.fsjesr.ac.ma/actuariat/pdf/cours/DMining_Partie1.pdf, Février 2017

[4] S. Saitta, *« Data Mining, des données à la connaissance »*, flashinformatique.epfl.ch/IMG/pdf_10-7-page4, 2007

[5] I. Witten, E. Frank, *« Data Mining, Practical Machine Learning Tools and Techniques »*, Editions Morgan Kaufmann, 2005

[6] M. Charrad, « *Techniques d'extraction de connaissances appliqués aux données du web »*, Mémoire de Mastère, Université de la Manouba – Tunis, Décembre 2005

[7] A.V. Ralambomahay, « *Conception d'un arbre de décision appliqué au Data Mining »*, Mémoire d'Ingénieur, ESPA – Vontovorona, Septembre 2013

[8] M. El Hadi Benelhadj, « *Entrepôt de Données et Fouille de Données - Un Modèle Binaire et Arborescent dans le Processus de Génération des Règles d'Association »*, Thèse de Doctorat, Université Mentouri – Constantine, Avril 2012

[9] P. Chapman, J. Clinton, R. Kerber, T. Khabaza, T. Reinartz, C. Shearer, R. Wirth, « *CRISP-DM 1.0 – Step-by-step data mining guide »*,1996

[10] A. Cornuéjols, L. Miclet, Y. Kodratoff, *« Apprentissage artificiel - Concepts et algorithmes »*, Eyrolles, 2003

[11] T. M. Mitchell, « *Machine Learning* », Editions McGrawHill Science/Engineering/Math, Mars 1997

[12] A. Dobra, « *Introduction to classification and regression* », https://www.cise.ufl.edu/~adobra/datamining/classif-intro.pdf, Février 2017

[13] A. Djeffal, « *Utilisation des méthodes Support Vector Machine (SVM) dans l'analyse des bases de données* », Thèse de Doctorat, Université Mohamed Khider – Biskra, 2012

[14] J.F. Mari, A. Napoli, « *Aspects de la classification* », RR-2909, INRIA, pp.97, 1996

[15] O. Zammit, « *Détection de zones brûlées après un feu de forêt à partir d'une seule image satellitaire Spot5 par technique SVM* », Thèse de Doctorat, Université de Nice – Sophia Antipolis, Septembre 2008

[16] F. Denis, H. Kadri, C. Capponi, « *Apprentissage automatique* », pageperso.lif.univ-mrs.fr/~francois.denis/IAAM1/chap1.pdf, Février 2017

[17] R. Caruana, A. Niculescu-Mizil « *An Empirical Comparison of Supervised Learning Algorithms* », https://www.cs.cornell.edu/~caruana/ctp/ct.papers/caruana.icml06.pdf, Février 2017

[18] Y. Dong, *« Modélisation probabiliste de classifieurs d'ensemble pour des problèmes à deux classes »*, Thèse de Doctorat, Université de Technologie de Troyes, Juillet 2013

[19] Y. Guermeur, « *SVM Multiclasses Théorie et Applications* », Université Nancy I, Novembre 2007

[20] C. Desir, « *Classification Automatique d'Images, Application à l'Imagerie du Poumon Profond* », Thèse de Doctorat, Université de Rouen, Juillet 2013

[21] H. Essid, « *Modélisation spatio-temporelle à base de modèles de Markov cachés pour la prévision des changements en imagerie satellitaire : cas de la végétation et de l'urbain* », Thèse de Doctorat, Université Blaise Pascal – Clermont II, Décembre 2012

[22] J.P. Gastellu-Etchegorry, « *Acquisition et traitement d'image numérique* », *www.cesbio.ups-tlse.fr/data_all/pdf/TI08.pdf*, Avril 2008

[23] N. Naffakhi, « *Apprentissage supervisé pour la Classification des images à l'aide de l'algèbre P-tree* », Université de Tunis, Février 2004

[24] A. Milloud, « *Segmentation des images numériques par seuillages multiples, Application à la découpe automatique dans les ateliers flexibles* », https://ori-nuxeo.univ-lille1.fr/nuxeo/site/esupversions/b2cb61eb-5149-4865-a4c4-f921a26fd1d8, Février 2017

[25] L. Macaire, S. Philipp-Foliguet, « *Segmentation d'images couleur* », www-lagis.univ-lille1.fr/ehinc2005/cours/ehinc05_seg_ludo_02_01.pdf, Février 2017

[26] J. Dubois, « *Segmentation par approche contour* », alpageproject.free.fr/doc/RAPPORT_TER_final_1.pdf, Février 2017

[27] H. Frédéric, E.K. Brahim, « *Etude de méthodes de Clustering pour la segmentation d'images en couleurs* », https://tcts.fpms.ac.be/cours/1005-07-08/speech/projects/2005/dhondt_elkhayati.pdf, Février 2017

[28] M. Bergounioux, « *Introduction au traitement mathématique des images - méthodes déterministes* », Editions Springer, 2015

[29] J. Bouchard, « *Méthodes de vision et d'intelligence artificielles pour la reconnaissance de spécimens coralliens* », Université du Québec, Avril 2011

[30] C.W. Hsu, C.C. Chang, and C.J. Lin, « *A Practical Guide to Support Vector Classification* », Mai 2016

[31] A. Cornuéjols, « *Méthodes à noyaux* », https://www.lri.fr/perso/~antoine/Courses/Master-ISI/chap-14-svm.pdf, Février 2017

[32] Y. Oufella, « *Evolution du concept de front ROC et combinaison de* classifieur », Université de Rouen, Septembre 2008

[33] M. Bigand, J. P. Bourey, H. Camus, D. Corbeel, « *Conception des systèmes d'information, modélisation des données, études de cas* », Editions Technip, Paris, 2006

[34] L. Marfai, « *Les difficultés rencontrées lors du développement de nouvelles molécules thérapeutiques dans l'indication de la maladie d'Alzheimer* », Thèse de Doctorat, Université De Lorraine, Novembre 2013

[35] Dimisoa, « *Fléau social Alzheimer: maladie laissée dans l'oubli* », http://madagascar-actualites.com/fleau-social-alzheimer-maladie-laissee-dans-loubli/, Septembre 2015

[36] K.A.Johnson, « *The Whole Brain Atlas* », *http://www.med.harvard.edu/aanlib/home.html*, Février 2017

[37] The MathWorks, « *Le langage du calcul* technique », https://fr.mathworks.com/products/matlab.html, Février 2017

[38] Oracle Corporation, « *General Information* », https://dev.mysql.com/doc/workbench/en/wb-intro.html, Février 2017

[39] J. Potdevin-verdier, *« Évaluation des pratiques professionnelles en radiopharmacie et amélioration de la sécurité du médicament radiopharmaceutique au CHR de Metz-Thionville »*, Thèse de Doctorat, Université de Lorraine, Octobre 2013

RENSEIGNEMENTS

Nom: RASOANAIVO

Prénom(s): Lucia Olihanta

Adresse: Lot VV 9 bis Ambohimitsimbina – Antananarivo 101

oly.rasoanaivo@moov.mg

+261 (0) 34 72 162 30

Titre du mémoire :

OUTIL DE CLASSIFICATION D'IMAGES
PAR METHODE D'APPRENTISSAGE AUTOMATIQUE

Nombres de pages : 84

Nombres de tableaux : 10

Nombre de figures : 39

Mots clés : Data Mining, Apprentissage, Classification, SVM, K-means, GLCM

Directeur de mémoire: M. RAKOTOMALALA Mamy Alain

rakotomamialain@gmail.com

+261 (0) 33 12 036 09

RESUME

L'émergence du Big Data a provoqué la nécessité d'analyser les données stockées dans de larges bases pour en extraire des informations. Les données images sont en particulier concernées par cette nécessité. Des systèmes capables de classifier correctement des images contenues dans ces bases sont alors proposés. Ils se basent sur des algorithmes d'apprentissage automatique, le plus souvent supervisé. Une phase de préparation des données précède la phase de conception de tels systèmes. Cela comprend : l'isolation des régions d'intérêt par une méthode de segmentation d'image, l'extraction des caractéristiques dont le classifieur a besoin pour l'apprentissage et la mise à disposition d'une base d'apprentissage. Une évaluation des performances du système est effectuée avant de déployer l'outil aux utilisateurs. La modélisation de ces systèmes de classification automatique d'images peut ainsi être envisagée dans plusieurs domaines d'application.

Mots clés : Data Mining, Apprentissage, Classification, SVM, K-means, GLCM

ABSTRACT

The emergence of Big Data has caused the need to analyze the data stored in large databases for information extraction. Images are particularly concerned with this necessity. A system that is able to correctly classify images stored in databases is then proposed. They are based on a machine learning algorithm that is most of the time supervised. A data groundwork phase comes before designing phase of such systems. This includes: regions of interest isolation via image segmentation method, features extraction needed by the classifier for the training and the provision of a dataset. The performances of the built system are evaluated before deploying the tool to the users. The modeling of these automatic image classification systems can thus be considered in several fields of application.

Keywords: Data Mining, Machine learning, Classification, SVM, K-means, GLCM