Université de Tunis
Institut Supérieur de Gestion de Tunis

Apprentissage supervisé pour la classification des images à l'aide de l'algèbre P-tree

Présenté par :

Najeh NAFFAKHI

Encadré par :

Pr. Khaled MELLOULI

Dr. Rim FAIZ

Pour l'obtention du

Diplôme des études approfondies
en informatique appliquée à la gestion

Année de soutenance :
Février 2004

Remerciements

En premier lieu, je tiens à remercier Monsieur Khaled Mellouli et Madame Rim Faiz qui ont su me communiquer leurs passions pour le Data Mining. Madame Rim Faiz m'a initié à la recherche et j'en suis très reconnaissant. Je tiens à témoigner ma profonde gratitude envers Monsieur Khaled Mellouli qui a donné une impulsion décisive à mon travail et m'a guidé attentivement tout au long de mes recherches. Sa rigueur, ses conseils éclairés et ses suggestions pertinentes m'ont permis de mener à bien ce travail.

Je remercie vivement Madame Amel Borgi de l'université de Reims (laboratoire LIP6) pour ses précieux conseils.

Le laboratoire LARODEC m'a accueilli durant cette année de recherche et m'a permis de réaliser ce travail dans de bonnes conditions. Les membres du laboratoire LARODEC ne seront jamais assez remerciés pour la bonne ambiance qu'ils ont su instaurer, et la motivation qu'ils m'ont communiqué.

Je tiens aussi à remercier les enseignants et le personnel de l'Institut Supérieur de Gestion de Tunis qui ont contribué à notre formation ainsi que les membres du jury pour avoir accepté de nous prêter leur attention et évaluer notre travail.

Ma famille et mes amis m'ont soutenue par leur affection et leur confiance tout au long de ce travail. Je pense en particulier à mes parents qui m'ont permis de faire des études et m'ont appris la valeur et le respect du savoir.

Table des matières

Introduction générale **6**

Chapitre 1: Le Data Mining : Généralités **9**

1.1 Définition et concepts 9
1.2 Le processus du Data Mining 11
1.2.1 L'exposition du problème 11
1.2.2 La recherche des données 11
1.2.3 La sélection des données pertinentes 11
1.2.4 Le nettoyage des données 12
1.2.5 Les actions sur les variables 12
1.2.6 La recherche du modèle 12
1.2.7 L'évaluation du résultat 12
1.2.8 L'intégration de la connaissance 13
1.3 Les techniques du Data Mining 14
1.3.1 Le raisonnement à base de cas 15
1.3.2 Les associations 16
1.3.3 Les arbres de décision 16
1.3.4 Les régressions 17
1.3.5 Les réseaux bayésiens 17
1.3.6 Les algorithmes génétiques 17
1.3.7 Les réseaux de neurones 19
1.4 Le Data Mining appliqué aux données multimédia 19
1.4.1 Le Text Mining 20
1.4.2 Le Sound Mining 21
1.4.3 L'Image Mining 21
1.4.4 Le Video Mining 22
1.5 Conclusion 23

Chapitre 2: Classification et apprentissage **24**

2.1 Introduction 24
2.2 L'apprentissage 25
2.2.1 L'apprentissage non supervisé 26
2.2.2 L'apprentissage supervisé 26
2.3 Présentation de quelques méthodes de classification 27
2.3.1 L'approche statistique 27
2.3.2 L'approche probabiliste 35
2.3.3 Classification par génération de règles 36
2.4 Méthode proposée pour la conception d'un système de classification des images 41

2.5 Conclusion 42

Chapitre 3: L'algèbre de P-tree.....43

3.1 Introduction.....43
3.2 Structure de P-trees.....44
3.2.1 Structure de base.....44
3.2.2 Variations de P-tree 45
3.3 L'algèbre P-tree 45
3.3.1 Opération de complémentation.....45
3.3.2 Opération d'intersection (AND) 46
3.3.3 Opération d'union (OR).....47
3.3.4 Valeur et Tuple de P-trees 47
3.4 Propriétés de P-trees 48
3.5 Utilisation et Performance de P-tree.....49
3.6 Conclusion 50

Chapitre 4: Méthode de l'apprentissage supervisé appliquée à la classification des images.....51

4.1 Introduction.....51
4.2 Présentations générale des données images.....51
4.2.1 Sélection des attributs 53
4.2.2 La classification 54
4.3 Méthode proposée.....54
4.3.1 Organisation relationnelle.....54
4.3.2 Conversion de la base de données en table binaire.....55
4.3.3 Stockage de P-trees.....56
4.3.4 Classification par arbre de décision en utilisant la structure P-tree.....57
4.3.5 Méthodologie de conception.....63
4.4 Conclusion 68

Chapitre 5: Implémentation et validation de la méthode 69

5.1 Introduction.....69
5.2 Environnement logiciel et matériel utilisé.....69
5.2.1 Environnement logiciel.....69
5.2.2 Environnement matériel.....70
5.3 Description de la base de données utilisée.....70
5.3.1 Structure de la base d'images 70
5.3.2 Les attributs de la base d'images 71
5.4 Comparaison 71
5.5 Les interfaces de l'application 73
5.6 Conclusion 77

Conclusion et perspectives 78

Bibliographie 80

Annexe : Extraits du code de l'application.....85

Liste des figures

FIG. 1.1 - Data Mining : étape clé dans le processus KDD 13
FIG. 1.2 - Le compromis entre lisibilité et prédiction..... 14
FIG. 1.3 - Principe de raisonnement d'un RBC 15
FIG. 1.4 - Fonctionnement des algorithmes génétiques 18
FIG. 2.1 - Extrait de la classification taxinomique de Linné. 26
FIG. 2.2 - L'apprentissage supervisé. 27
FIG. 2.3 - Recherche du meilleur axe par l'AFD 30
FIG. 2.4 - Un exemple de nuages de points concentriques. 31
FIG. 2.5 - Projection des exemples sur le 1er axe factoriel. 31
FIG. 2.6 - Règle géométrique avec la distance euclidienne 32
FIG. 2.7 - Résultat de l'analyse discriminante. 33
FIG. 2.8 - Un exemple de fonction discriminante linéaire. 34
FIG. 2.9 - Discrimination linéaire par morceaux 34
FIG. 2.10 - Un exemple simple d'arbre de décision. 38
FIG. 3.1 - Un P-tree pour un fichier (une bande) bSQ 8x8. 44
FIG. 3.2 - Un PM-tree 45
FIG. 3.3 - P1-tree et P0-tree. 45
FIG. 3.4 - Complément d'un P-tree. 46
FIG. 3.5 - L'intersection de deux P-trees 46
FIG. 3.6 - L'union de deux P-trees. 47
FIG. 4.1 - Repérage des pixels. 52
FIG. 4.2- Conversion des données relationnelles en une représentation REL pour les attributs de domaine binaire 55
FIG. 4.3 - Conversion des données relationnelles en une représentation REL pour les attributs de domaine non binaire. 55
FIG. 4.4 - Génération de P-tree de base pour l'Item a_1. 57
FIG. 4.5 - Résultats des intersections des P-trees (ANDing) 61
FIG. 4.6 - Classification suivant l'attribut couleur. 63
FIG. 4.7 - Diagramme des cas d'utilisation. 64
FIG. 4.8 - Diagramme de classe 65
FIG. 4.9 - Diagramme de séquence relatif au remplissage des bandes. 66
FIG. 4.10 -Diagramme de séquence relatif aux calculs des probabilités. 67
FIG. 4.11-Diagramme de séquence relatif à la génération des classes..... 67
FIG. 5.1 - Structure de la base d'images utilisée. 71
FIG. 5.2 - Coût de classification. 72
FIG. 5.3 - Interface de la base de données des images. 73
FIG. 5.4 - Boite de dialogue permettant de charger une autre image à partir de la base 73
FIG. 5.5 - La base des images convertie en base binaire. 74
FIG. 5.6 - Calcul des probabilités et des entropies des attributs des images. 75
FIG. 5.7 - Classification d'images suivant les attribut. 76

Liste des tableaux

TAB. 2.1 - Un exemple de base d'apprentissage. 38
TAB. 3.1 - Les règles de P-tree ANDing. 46
TAB. 3.2 - Les règles de P-tree ORing. 47
TAB. 4.1 - Table binaire de l'attribut couleur. 60
TAB. 5.1 - Comparaison entre algorithmes. 71

Introduction générale

Aujourd'hui, il devient de plus en plus difficile de comprendre, analyser et prendre des décisions face à une masse de données chaque jour plus importante. La quantité de données dans le monde est estimée doubler tous les 20 mois [Kodratoff 97]. Face à cette explosion de données, la détection de structures et liens particuliers, l'organisation et la recherche de connaissances exploitables dans cette masse d'information deviennent un enjeu stratégique pour la prise de décision mais aussi à des fins de prédiction. Dans cette optique, un nouveau domaine d'étude a émergé : c'est l'Extraction de Connaissances à partir de Données (ECD)[1]. L'ECD se situe à l'intersection de plusieurs disciplines, elle fait appel notamment à l'apprentissage et aux systèmes experts, aux statistiques et à l'analyse de données, mais également aux Bases de Données et à la visualisation des données [Fayyad et al. 96].

D'une manière générale, il ne suffit pas d'avoir les données, encore il faut savoir les traiter pour les synthétiser, les analyser, les rechercher, les mettre en perspective d'une action bien définie ou d'un objectif à atteindre. C'est ainsi que s'introduit naturellement le fouille de données (*Data Mining*), désignant, l'ensemble des moyens d'exploitation intelligent des données en vue d'une action bien déterminée.

Le *Data Mining* est un processus d'acquisition progressif de connaissances basé sur la combinaison et l'enchaînement de plusieurs techniques. Mais, c'est l'intelligence de l'exploitation de données qui valorise l'information au moyen de ces techniques.

Les méthodes du *Data Mining* sont variées ; parmi ces méthodes qui sont largement appliquées, nous retrouvons les règles d'association, l'identification des séquences et la classification. Leur objectif est l'extraction non triviale des informations utiles et potentielles non connues d'avance à partir de données structurées. Mais, cet objectif est difficile à atteindre lorsqu'on est devant une base de données multimédia. Autrement dit, les méthodes du *Data Mining* doivent évoluer vers des méthodes qui traitent toutes sortes de données: structurées et non structurées comme des textes, des images, des signaux, des séquences vidéo, etc.

Nous nous sommes intéressés à l'un des aspects de ce problème complexe, celui de l'extraction des connaissances à partir de données multimédia. Il constitue aujourd'hui

[1] En anglais Knowledge Discovery in Databases ou bien *Data Mining*.

un domaine de recherche très dynamique et présente de multiples facettes. Notre travail s'inscrit précisément dans ce domaine de recherche.

Nous nous intéressons en premier lieu à un type de données multimédia particulier qui est l'image. Notre travail vise à générer de règles de production à partir d'une base d'images de fraise. Ces exemples sont décrits par une liste d'attributs. Ils sont représentés par un ensemble de couples (attribut, valeur). Les règles générées sont des règles de classification. Il s'agit plus précisément de la classification supervisée.

Dans le cadre de ce mémoire, nous désignons par le terme classification, la tâche qui consiste à *assigner une image à un groupe parmi un nombre fini de groupes prédéterminés*. La classification constitue un problème très répandu et largement étudié aussi bien dans le cadre du *Data Mining* que dans d'autres domaines tels que l'Analyse de données ou la Reconnaissance des Formes. Elle correspond à une tâche de traitement d'information, à laquelle est associée une forte sémantique cognitive. En effet, la classification semble être une puissante stratégie humaine pour organiser les connaissances à des fins de compréhension et d'action. La classification est une activité omniprésente qui sous-tend les processus de raisonnement humains.

Notre méthodologie se base sur la génération de règles de production à partir d'une base d'images convertie en base binaire et sur une utilisation de l'algèbre du P-tree [Qin Ding et al. 02] pour accélérer la classification.

Problématique

Les applications actuelles traitent plutôt des bases de données structurées bien que l'extraction des connaissances issues de données multimédia (image, son, vidéo,…) est devenue récemment un axe de recherche actif, notamment pour le traitement d'informations recueillies lors des recherches sur le web. Les documents images ont comme caractéristique principale le manque d'un langage permettant d'en exprimer la sémantique. Il en résulte une très grande difficulté pour proposer des systèmes de classification et de recherche à partir d'une base d'images.

Dans ce mémoire, nous proposons une nouvelle méthode de classification en se basant sur l'algorithme C4.5 dévéloppé par Quinlan [Quinlan 93]. Pour la classification d'images, le temps est un facteur important. Cependant, ces données sont aussi volumineux (taille de données) et leurs classification par les méthodes existantes nécessite beaucoup du temps. Dans ce travail, nous avons développé une nouvelle méthode de classification basée sur l'arbre de décision en utilisant une structure de donnée appelée P-tree (arbre de Peano Count). Cette méthode permet de réduire le temps de classification. En effet, la structure P-tree fournit une représentation binaire des données. Le calcul des probabilités pour choisir le meilleur attribut a été basé sur des règles de comparaison des P-trees (P-tree_i ANDing P-tree_j). Cette méthode, d'une part, évite le balayage direct de la base de données qui est une opération assez coûteuse en mémoire et en temps d'exécution et, d'autres parts, elle offre un gain de comparaison des attributs/items car la comparaison s'effectue par bloc. Nous comparons la classification par arbre de décision basée sur P-tree et la méthode classique d'arbre de décision en terme de temps.

Plan du mémoire

Le premier chapitre présente le contexte général du *Data Mining*. Il est consacré à la description du processus du *Data Mining* ainsi que ses principales techniques. Le deuxième chapitre fournit un état de l'art sur l'apprentissage et la classification, ainsi qu'une description détaillée des arbres de décision. Le troisième chapitre décrit l'algèbre de P-tree, ses domaines d'applications et les performances réalisées. Dans le quatrième chapitre, nous présentons les caractéristiques et la conception de notre méthode de classification. Nous présentons brièvement le domaine de traitement d'images ainsi que les différentes méthodes de classification des images. Dans le cinquième chapitre, nous développons notre méthode de classification à l'aide de l'algorithme C4.5 basée sur la structure P-tree.

Enfin, nous terminons ce mémoire en concluant par une présentation des avantages de notre méthode par rapport aux méthodes de classification existantes. Les perspectives de recherche ouvertes par ce travail et les points à approfondir sont également exposés dans la conclusion.

Chapitre 1

Le Data Mining : Généralités

1.1 Définition et concepts

Aujourd'hui, des milliards de données sont collectées chaque jour dans le monde. Néanmoins, si l'essor de l'informatique a rendu possible le stockage de volumes de données toujours plus importants à des coûts toujours plus faibles, la quantité d'information fournie aux utilisateurs n'a quant à elle que peu évolué. Les entreprises ont plus que jamais besoin de transformer leurs données en connaissance directement valorisable.

Pour générer des informations riches et découvrir des modèles implicites dans des ensembles de données multidimensionnels, un ensemble de démarches, de méthodologies et d'outils ont été regroupés sous le terme *extraction des connaissances à partir des données*[2]. Il existe probablement autant de définitions différentes du terme *Data Mining*. Il dépasse aujourd'hui le cercle restreint de la communauté scientifique pour susciter un vif intérêt dans le monde des affaires [Lefebure et al. 01]. La littérature spécialisée et la presse ont pris le relais de cet intérêt et proposent plusieurs définitions du *Data Mining*. Nous avons sélectionné quelques-unes :

- *Un processus non élémentaire de mises à jour de relations, corrélations, dépendances, associations, modèles, structures, tendances, classes, facteurs obtenus en naviguant à travers de grands ensembles de données* [Mitchell 83].

- *Un processus d'aide à la décision où les utilisateurs cherchent des modèles d'interprétation dans les données* [Piatetsky-Shaprio 93].

[2] Nous utilisons dans la suite de ce mémoire le terme *Data Mining* pour décrire l'ensemble du processus d'extraction de connaissances à partir de données contenues dans une base de données.

- *L'exploration et l'analyse, par des moyens automatiques ou semi-automatiques, d'un large volume de données afin de découvrir des tendances ou des règles* [Michie 94].

- *L'extraction d'informations originales, auparavant inconnues et potentiellement utiles, à partir de données* [Gartner 95].

- *La découverte de nouvelles corrélations, tendances et modèles par le tamisage d'un large volume de données* [Gaines 96].

D'autres, plus poétiques, parlent de *« torturer l'information disponible jusqu'à ce qu'elle avoue »* [Fayyad et al. 96].

Plus généralement, les spécialistes du domaine considèrent que la découverte de connaissances dans les bases de données ou *Knowledge Discovery in Data-base* (KDD), englobe tout le processus d'extraction de connaissances à partir des données.

Le terme *Data Mining* est souvent utilisé pour désigner l'ensemble des outils permettant à l'utilisateur d'accéder aux données et de les analyser. *Data Mining* consiste alors à générer des informations riches, de préférences à partir des données historiées, à découvrir des modèles, des règles, des exceptions entre les données.

Les outils d'aide à la décision, qu'ils soient relationnels ou *OLAP*[3], laissent l'initiative à l'utilisateur, qui choisit les éléments qu'il veut observer ou analyser. Au contraire, dans le cas du *Data Mining*, le système implémenté découvre lui-même les associations entre les données, sans que l'utilisateur ait à lui dire de rechercher plutôt dans telle ou telle direction ou à poser des hypothèses. Il est possible de prédire l'avenir, par exemple le comportement d'un client, et de détecter, dans le passé, les données inusuelles et exceptionnelles.

Le but est de pouvoir extraire un élément : la connaissance « *Une connaissance décrit une catégorie abstraite. Chaque catégorie peut couvrir plusieurs exemples. Des experts sont nécessaires pour recueillir et formaliser la connaissance.* » [Lefebure et al. 01], ce concept s'appui sur le constat qu'il existe au sien de chaque entreprise des informations cachées dans le gisement de données. Le *Data Mining* permet, grâce à un certain nombre de techniques spécifiques, de faire apparaître des connaissances.

Littéralement, il s'agit d'exploiter des ensembles de données brutes comme on exploiterait une mine d'or, c'est à dire pour découvrir les trésors qui y sont enfouis. Concrètement, il s'agit d'automatiser les processus de découverte des tendances présentes dans de gros ensembles de données.

De manière pragmatique, nous retiendrons que l'objet des techniques du *Data Mining* est d'identifier les relations complexes existantes entre diverses données, d'établir des règles de comportement, et de mettre à jour des modèles fonctionnels permettant ainsi la prévision du comportement futur. L'idée sous-jacente reste celle d'un processus, le plus automatisé possible, qui permet de passer de données brutes à la prise de décision, en apportant à chaque étape une plus-value informationnelle.

---

[3] On Line Analytical Processing désigne une catégorie d'outils d'exploration de données qui permettent de visualiser des valeurs dans plusieurs dimensions.

Dans la partie qui suit, nous allons fournir un tour d'horizon des étapes du processus du *Data Mining*. Dans la section 3 nous présentons les techniques du *Data Mining* en discutons les avantages et les limites de chacune. La section 4 présente l'application du *Data Mining* aux données multimédia.

1.2 Le processus du Data Mining

Le *Data Mining* s'inscrit dans le courant, aujourd'hui irréversible, de la gestion des connaissances. Il constitue un outil qui facilite la mise en évidence de classes ou des règles à partir de l'observation de données. Il n'est donc qu'un élément du processus, beaucoup plus global, de transformation des données en connaissances, « *Knowledge Discovery in Data bases* » (KDD) que nous appelons en français « Extraction de Connaissances à partir de Données » (ECD). Le *Data Mining* est un des maillons de la chaîne de traitement pour la découverte des connaissances à partir des données. Il s'intègre dans le processus KDD (huit étapes), que nous allons le présenter en bref ci-après.

1.2.1 L'exposition du problème

Cette première phase est celle où l'on expose le problème et où l'on définit les objectifs, le résultat attendu ainsi que les moyens de mesurer le succès de l'étape du *Data Mining*. Il s'agit de comprendre le contexte de la recherche en vue de donner une signification logique aux variables. Dans cette phase introductive, il est intéressant de recueillir les intuitions et la connaissance des experts afin d'orienter le processus de découverte ou tout simplement pour identifier les variables les plus pertinentes susceptibles d'expliquer les phénomènes analysés.

1.2.2 La recherche des données

Il s'agit dans cette phase de déterminer la structure générale des données ainsi que les règles utilisées pour les constituer. Il faut identifier les informations exploitables et vérifier leur qualité et leur facilité d'accès : documents papiers, supports électroniques, fichiers internes ou externes, fichiers multiples ou bases de données de type *Data Warehouse*[4] ou *Data Mart*[5].

1.2.3 La sélection des données pertinentes

Le meilleur moyen de créer un modèle est de rechercher des événements similaires dans le passé. Il faut donc constituer, à partir de système d'information, cette base d'informations qui va permettre de construire l'apprentissage. La récupération des données peut être plus ou moins facilitée par les technologies en place (bases de données ouvertes, existence d'un *Data Warehouse* exhaustif, etc.).

Cette phase de collecte et de sélection constitue, lorsque les systèmes informatiques source sont très hermétiques (peu documentés, hétérogènes ou basés sur des technologies obsolètes), une charge de travail considérable, qui peu représenter jusqu'à 80 % de la charge de travail global du processus du *Data Mining*. Face au

[4] *Data Warehouse* : collection de données orientées sujet, intégrées, non volatiles et historiées, organisées pour le support du processus d'aide à la décision.
[5] *Data Mart* : sous-ensemble logique et physique d'un *Data Warehouse*.

sentiment de perte de temps et d'inutilité éprouvée par les utilisateurs du système du *Data Mining* pendant cette étape, le responsable du projet doit mettre une politique d'animation et de suivi de la collecte des données de façon à motiver les intervenants. De plus, il faut éviter le piège *GIGO (Garbage In, Garbage Out)*, dans lequel les erreurs en entrée entraînent des erreurs en sortie [Gartner 96].

La tendance à l'intégration des outils du *Data Warehouse*, des produits du *Data Mining* et des systèmes d'aide à la décision contribue à rendre les données plus accessibles.

1.2.4 Le nettoyage des données

La définition de la taille de la base de données et le choix de son mode d'élaboration passent par un diagnostic de la qualité potentielle de données. Une mauvaise qualité des données (erreurs de saisie, champs nuls, valeurs aberrantes) impose généralement une phase de nettoyage des données. Celle-ci a pour objectif de corriger ou de contourner les inexactitudes ou les erreurs qui se sont glissées dans les données.

1.2.5 Les actions sur les variables

Maintenant que les variables sont pertinentes et que les données sont fiables, il faut les transformer pour réparer le travail d'analyse. Il s'agit d'intervenir sur les variables pour faciliter leur exploitation par les outils de modélisation.

1.2.6 La recherche du modèle

L'étape de recherche du modèle consiste à extraire la connaissance utile d'un ensemble de données et à la présenter sous une forme synthétique. Il s'agit de la phase le plus souvent décrite sous le terme *Data Mining* et qui repose sur une recherche exploratoire, c'est à dire dépourvue de préjugés concernant les relations entre les données.

1.2.7 L'évaluation du résultat

L'évaluation du résultat permet d'estimer la qualité du modèle, c'est à dire sa capacité à déterminer correctement les valeurs qu'il est censé avoir appris à calculer sur des cas nouveaux. Cette évaluation prend généralement une forme qualitative et une forme quantitative.

La forme qualitative consiste à restituer la connaissance sous forme graphique ou textuelle. Cette contribution permet d'améliorer la compréhension des résultats et faciliter le partage de la connaissance.

La forme quantitative nécessite une certaine expertise fonctionnelle tels que l'intervalle de confiance, la validation par test, etc. pour interpréter les résultats.

1.2.8 L'intégration de la connaissance

La connaissance ne sert à rien tant qu'elle n'est pas convertie en décision puis en action. Cette phase d'intégration de la connaissance consiste à implanter le modèle ou ses résultats dans les systèmes informatiques.

Sous forme imagée, nous pourrions dire que le KDD est un véhicule dont le *Data Mining* est le moteur. En effet, le KDD est un processus complexe qui se déroule suivant une suite d'opérations. Des étapes de prétraitement ont lieu avant le *Data Mining* proprement dit. Le prétraitement porte sur l'accès aux données en vue de construire des *Data Marts*, des corpus de données spécifiques. Le prétraitement concerne la mise en forme des données entrées selon leur type (numérique, symbolique, image, texte, son), ainsi que le nettoyage des données, le traitement des données manquantes, la sélection d'attributs. Cette première phase est cruciale car du choix des descripteurs et de la connaissance précise de la population va dépendre la mise au point des modèles de prédiction. L'information nécessaire à la construction d'un bon modèle de prévision peut être disponible dans les données mais un choix inapproprié de variables ou d'échantillons d'apprentissage peut faire échouer l'opération. Le schéma ci-dessous récapitule les grandes étapes du processus KDD (cf. figure 1.1) [HAN 00].

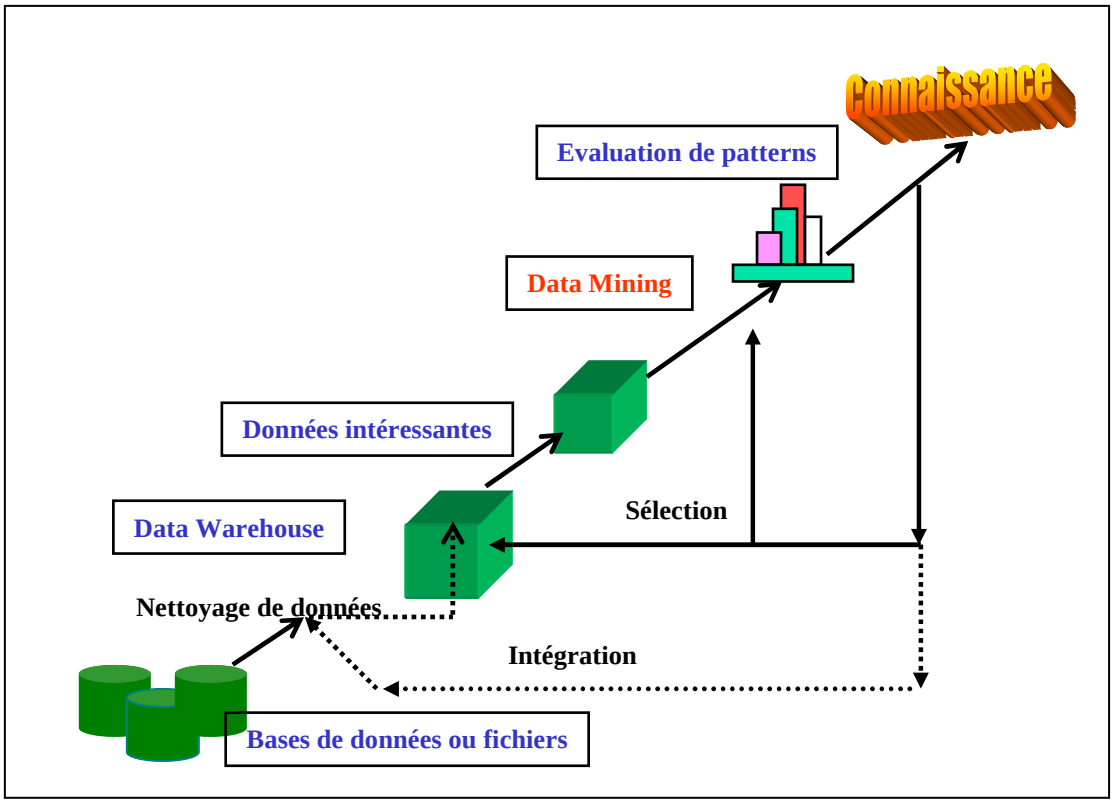


FIG. 1.1 - *Data Mining : étape clé dans le processus KDD.*

L'ensemble du processus décrit ci-dessus n'insiste pas suffisamment sur le rôle primordial des utilisateurs et des experts, mais aussi sur la configuration du problème et la technique du *Data Mining* la plus adéquate.

Dans la section suivante, nous discutons les avantages et les limites des techniques les plus usuels du *Data Mining*.

1.3 Les techniques du Data Mining

Avant de présenter les différentes techniques du *Data Mining*, il est important de préciser le compromis entre clarté du modèle et pouvoir prédictif. Plus un modèle est simple, plus il sera facile à comprendre, mais moins il sera capable de prendre en compte des dépendances subtiles ou trop variées. Le schéma ci-dessous illustre ce compromis (cf. figure 1.2) [Lefebure et al. 01].

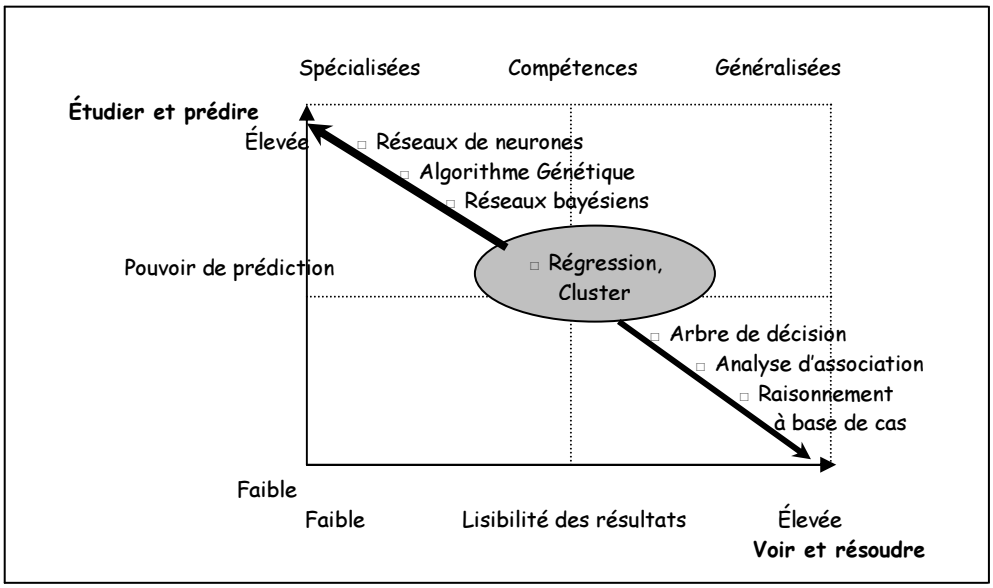


FIG. 1.2 - *Le compromis entre lisibilité et prédiction.*

Les arbres de décision et les bases de règles sont très faciles à interpréter. Néanmoins, ces techniques ne reconnaissent que des frontières nettes de discrimination. Les régressions sont un peu plus fines. Toutefois, compte tenu du caractère additif de leurs fonctions (de type ax + by), elles ne peuvent prendre en compte les relations multi-variables. L'existence d'une relation d'interdépendance entre les variables conduit à une diminution de la performance du modèle. Les réseaux de neurones, par leur capacité à intégrer les relations entre les variables présentent un pouvoir prédictif élevé. Néanmoins, ce progrès entraîne une perte de lisibilité, compte tenu de la complexité du modèle mathématique sous-jacent.

Ce découpage fondé sur trois axes présente un certain degré de simplification ; il servira néanmoins de fil conducteur dans la description de différents outils.

Il existe deux grandes classes de techniques : celles qui permettent de décrire un phénomène, et celle qui permettent de le prévoir. Nous abordons successivement les techniques les plus usuelles, suivant un degré de complexité qui augmente petit à petit, le raisonnement à base de cas, les associations, les arbres de décision, les algorithmes génétiques, les réseaux bayésiens, pour terminer par les réseaux de neurones.

1.3.1 Le raisonnement à base de cas

Les systèmes de RBC [Turban 01] (Raisonnement à Base de Cas), en anglais CBR (Case Based Reasoning), résolvent des problèmes par la comparaison d'exemples proches puisés dans un ensemble de cas préalablement stockés. Avec cette méthode de résolution, si une expérience passée et une nouvelle situation sont suffisamment similaires, toutes les conclusions appliquées à l'expérience passée restent valides et peuvent être appliquées à la nouvelle situation.

Le RBC mettent en œuvre une procédure de recherche pour comparer les descriptifs du cas à traiter avec ceux des cas existants dans leur base interne. À ce titre, la capacité de résolution augmente au fil de l'arrivée de nouveaux exemples dans la base de références. Plus le nombre d'exemples sera important, plus le RBC aura de chances de retrouver un exemple proche (cf. figure 1.3) [Turban 01].

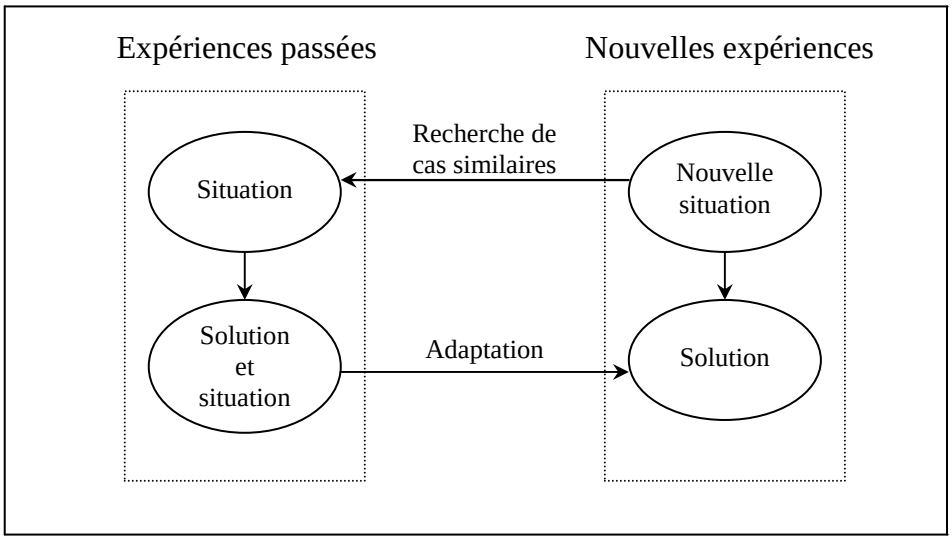


FIG. 1.3 - *Principe de raisonnement d'un RBC.*

Les applications des systèmes de RBC sont multiples ; cette technique rencontre un gros succès dans les domaines du service après-vente ou du diagnostic de panne, notamment dans les centres d'appels.

L'un des problèmes d'optimisation des RBC est lié à la gestion de données textuelles non structurées. Dans ce cas, la recherche des similarités est construite à partir de l'identification des mots-clés. En plus, l'ajout successif de plusieurs clés d'indexation conduit fatalement à réduire l'efficacité du RBC. La performance a tendance à se dégrader au fur et à mesure de la croissance de la base de cas lorsque celle-ci atteint plusieurs milliers d'exemples. Il est nécessaire de revoir les processus de classification et d'indexation avec des techniques d'arbre de décision.

Le temps de développement d'un RBC est de l'ordre de 3 à 6 mois pour un problème normal. Cet ordre de grandeur est comparable à celui d'un système expert, sauf si la BD est déjà structurée. En revanche, compte tenu de leur capacité d'évolution, les RBC présentent des charges de maintenance plus faibles.

1.3.2 Les associations

La recherche d'association vise à construire un modèle fondé sur des règles conditionnelles à partir d'un fichier de données. Une règle conditionnelle se définit sous la forme d'une suite « Si conditions, alors résultats ».

L'exemple le plus immédiat est l'analyse des tickets de caisse : chaque transaction est constituée d'une liste d'articles et le but de l'analyse d'associations est de déterminer l'existence de règles conditionnelles du type {achat article(s) **A**}=> {achat article(s) **B**}. En fonction des résultats, un certain nombre d'actions pourront être entreprises : repenser la configuration des linéaires, optimiser la gestion des stocks, mieux cibler les promotions...

Une variante du problème précédent consiste à introduire une composante temporelle en recherchant des comportements séquentiels du type {**A**} => {**B** dans x jours}. Le champs des applications potentielles est alors beaucoup pus vaste : marketing, anticipation des pannes dans l'industrie ou encore prévention médicale.

La force d'une association est généralement mesurée par un niveau de confiance indiquant le pouvoir prédictif de la règle, et un niveau de support correspondant au nombre de cas pour lesquels la règle a été vérifiée.

Le choix de données et des échantillons (des volumes de données monstrueux, …) joue un rôle primordial dans la fiabilité du calcul des associations.

1.3.3 Les arbres de décision

Les arbres de décision suscitent un large engouement en *Data Mining* car ils sont simples et rapides tout en restituant de manière compréhensible les relations existantes entre les variables d'entrée et le phénomène à modéliser.

Formellement, un arbre de décision est un enchaînement hiérarchique de règles logiques qui permet de diviser la base d'exemples en sous-groupes, en fonction de la valeur des variables d'entrées. L'arbre est construit en recherchant à chaque niveau le paramètre le plus discriminant pour classifier un exemple. Différents algorithmes, qui se distinguent essentiellement par le critère de segmentation appliqué pour définir la variable la plus significative, peuvent être utilisés pour développer l'arborescence.

Les avantages indéniables des arbres de décision sont leur simplicité et leur lisibilité. En effet, les produits à base d'arbres de décision sont simples d'utilisation, très visuels et leur mise en main est très intuitive. En plus, le modèle construit est structuré sous la forme d'une règle du type « si […], alors … » facilite le travail de validation et de communication du modèle.

Les arbres construits automatiquement sont souvent très compliqués. Une surabondance de branches dégrade la lisibilité du modèle. De plus, un arbre trop détaillé peut présenter un risque de non généralisation du modèle.

1.3.4 Les régressions

Les régressions sont des méthodes utilisées pour découvrir des modèles fonctionnels mettant en relations une variable à expliquer et un ensemble de variable explicatives.

Classiquement, une régression, qu'elle soit simple ou multiple, consiste à construire la droite (régression linéaire) ou la courbe (kernel régression) qui permet de représenter au mieux les k exemples de la base d'apprentissage pour laquelle les couples (entrées, sorties) sont connus.

Ces dernières années, l'émergence des régressions régularisées permettent de rechercher automatiquement le modèle fonctionnel dans l'ensemble des polynômes à *n* variables. Ce type de technique permet une meilleure maîtrise des capacités de génération des modèles, une meilleure lisibilité et un temps d'exécution bien plus faible.

1.3.5 Les réseaux bayésiens

Les réseaux bayésiens sont une méthode classique utilisée pour associer une probabilité d'apparition d'un événement étant donnée la connaissance de certains autres événements. Un réseau bayésien est un modèle graphique qui encode les probabilités entre les variables les plus pertinentes.

Les réseaux bayésiens recherchent le meilleur graphe de connexion entre les variables et apportent à l'utilisateur une connaissance des variables pertinents et des liens qui unissent ces variables. Les réseaux bayésiens se révèlent donc performants pour intégrer l'incertitude et les données manquantes. En revanche, il est crucial de construire l'apprentissage sur des données de qualité. En effet, l'apprentissage sur des données bruitées est généralement déconseillé car le bruit se propage à tous les niveaux lorsqu'il n'est pas filtré.

Les réseaux bayésiens, malgré leurs qualités, n'échappent pas à la règle *GIGO (Garbage In, Garbage Out)*. En plus, la recherche du meilleur réseau est une tâche consommatrice de puissance informatique, car le nombre de combinaisons variables-arcs possibles est de nature combinatoire.

1.3.6 Les algorithmes génétiques

Les algorithmes génétiques sont relativement récents par rapport aux autres concepts d'acquisition de la connaissance. Ils ont été introduits par John Holland en 1975, avec la présentation d'une méthode d'optimisation inspirée de l'observation des capacités d'adaptation et d'évolution des espèces (cf. figure 1.4) [Turban 01].

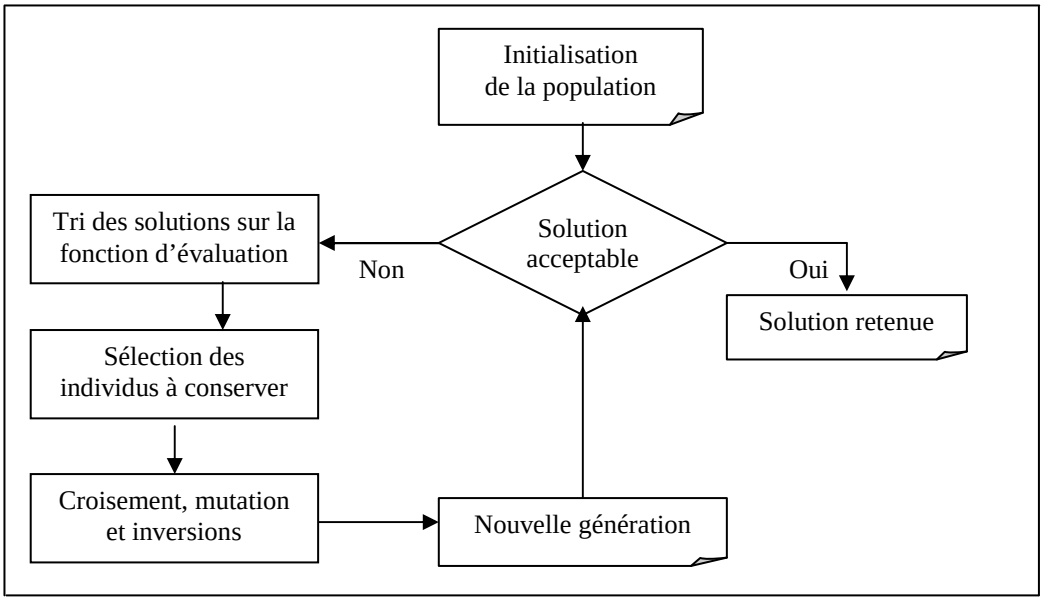


FIG. 1.4 - *Fonctionnement des algorithmes génétiques.*

Les cycles d'évolution se répètent jusqu'à ce que la population converge, c'est à dire jusqu'à ce que la diversité devienne très faible ou que la fonction d'évaluation ne progresse plus.

Les algorithmes génétiques trouvent des applications dans de nombreux domaines. Ils sont utilisés dans l'industrie, en complément des autres techniques, pour résoudre des problèmes d'optimisation ou de contrôle de processus complexes (optimisation de la température d'un four ou de la pression d'un cylindre, par exemple) et dans le domaine des données spatiales pour optimiser des positions dans l'espace et du géomarketing (optimisation d'un plan d'affichage, choix des implantations d'automates bancaires, par exemple).

Certains problèmes d'optimisation sont difficiles à résoudre car l'espace des solutions est très vaste. Les algorithmes génétiques possèdent un gros avantage sur les autres techniques d'optimisation. En effet, le nombre de possibilités explorées par les techniques des algorithmes génétiques sur une population de taille n est de l'ordre n^3. En d'autres termes, un algorithme génétique sur 1000 éléments accomplit, en quelques centaines de générations, le même résultat que l'évaluation d'un milliard de solutions prises au hasard.

Le codage des données sous formes de 0 et de 1 est une contrainte. En effet, le codage sous un format binaire pose un problème pour représenter la proximité des valeurs numériques. En plus, l'augmentation du taux d'hybridation augmente la recombinaison des chaînes, mais accroît le risque de détruire des bonnes solutions. L'augmentation du taux de mutation augmente l'espace des solutions, avec le risque d'altération d'une bonne solution. En fin, l'augmentation de la taille de la population augmente la diversité et réduit le risque de se trouver dans un optimum local, mais entraîne une forte croissance des temps de calcul.

1.3.7 Les réseaux de neurones

Les réseaux de neurones proposent une simulation du fonctionnement de la cellule nerveuse à l'aide d'un automate : le neurone formel. Les réseaux neuronaux sont constitués d'un ensemble de neurones (nœuds) connectés entre eux par des liens qui permettent de propager les signaux de neurone à neurone.

Grâce à leur capacité d'apprentissage, les réseaux neuronaux permettent de découvrir des relations complexes non-linéaires entre un grand nombre de variables, sans intervention externe. De ce fait, ils sont largement utilisés dans de nombreux problèmes de classification (ciblage marketing, reconnaissance de formes, traitement de signal,…) d'estimation (modélisation de phénomènes complexes,…) et prévision (bourse, ventes,…). Il existe un compromis entre clarté du modèle et pouvoir prédictif. Plus un modèle est simple, plus il sera facile à comprendre, mais moins il sera capable de prendre en compte des dépendances trop variées. La figure 1.2 (ci-dessus) illustre le compromis entre lisibilité et prédiction.

Bien que d'utilisations moins répandues, plusieurs autres méthodes peuvent être utilisées pour faire du *Data Mining*. Nous pouvons citer principalement :

- Les techniques de projection telles que les analyses factorielles ou les analyses en composantes principales.

- Les outils de visualisation, qui permettent à l'utilisateur de visualiser les relations trouvées et de les explorer de manière interactive.

Il convient de signaler que les différentes techniques de modélisation présentées ci-dessus peuvent être utilisées conjointement pour aboutir à un modèle optimal et améliorer la qualité des prédictions. Dans notre travail, nous allons combiner arbre de décision et algèbre de P-tree pour faire la classification d'une base d'images de fraise.

Généralement, les applications du *Data Mining* concernent le domaine de l'exploitation de données structurées. Cependant, l'explosion des données multimédia tels que le texte, l'image, le son, le vidéo à partir de différentes sources (Internet, Satellite, Agence de photo de presses, etc.) a orienté les recherches actuelles vers un nouveau domaine : le *Multimedia Data Mining* [Zaïane 00]. Un certains nombre de recherche ou de produits émergents traitent du *Data Mining* appliqué à des informations multimédia [Lefebure et al. 01]. C'est ce que nous allons discuter dans la partie qui suit.

1.4 Le Data Mining appliqué aux données multimédia

Le *Data Mining* est l'art d'extraire des connaissances à partir des données. Les données peuvent être stockées dans des *Data Warehouses*, dans des bases de données distribuées ou sur Internet « *Web Mining* ». Le *Data Mining* ne se limite pas au traitement des données structurées sous formes de tables numériques ; il offre des moyens pour aborder les corpus en langage naturel, les images, le son ou la vidéo et dans ce cas en parle alors plus généralement du « *Multimedia Mining* ».

Le multimédia a été le centre d'intérêt pour beaucoup de chercheurs. Plusieurs définitions ont été proposées. Nous avons sélectionné quelques-unes :

- *Le Multimédia Data Mining est l'extraction des informations de haut niveau à partir des bases de données multimédia* [HAN 99].

- *Le Multimédia Data Mining est un sous-domaine de Data Mining qui traite en plus l'extraction de connaissances implicites, des rapports entre de données multimédia, ou d'autres modèles non explicitement stockés dans des bases de données multimédia* [Zaïane 00].

La définition la plus communément utilisée est la suivante : *Le Multimédia Data Mining est l'application des techniques de Data Mining sur tout type de média : Image, Son, Vidéo, Texte, etc.* [Lefebure et al. 01]. De ce fait, les chercheurs s'orientent vers le *Text Mining*, le *Sound Mining*, l'*Image Mining*, le *Video Mining*, etc., nous les décrivons respectivement dans les sections suivantes.

1.4.1 Le Text Mining

Les travaux de caractérisation du texte étaient moins ambitieux. Ils visent à faciliter le classement et la recherche des documents. Toutefois, la présence des pluriels, des abréviations, des synonymes, des polysémies, etc., complique le travail de recherche ; l'utilisateur doit imaginer tous les mots susceptibles de graviter autour de son thème de recherche. Le regroupement automatique sous des concepts et la mesure des associations ont représenté une étape supplémentaire vers la maîtrise des données textuelles.

Le *Text Mining* actuel est issu majoritairement de ces travaux. Il s'attache à construire une description objective et quantitative d'un texte en vue de dégager les traits marquants et de construire un résumé textuel ou graphique. En fait, il y a plusieurs définitions du terme *Text Mining*, nous avons sélectionné quelques-unes :

- *Le Text Mining regroupe l'ensemble des techniques issues du traitement automatique de la langue et de l'analyse de données permettant de retrouver des informations cachées dans de larges bases de données textuelles* [Bara 99].

- *Le Text Mining est l'extraction automatique de connaissances à partir de fichiers contenant du Texte, essentiellement en vue de procéder à une classification automatique* [Lefebure et al. 01].

Généralement, le *Text Mining* ne cherche pas à mettre au point des systèmes décrivant les langues dans leur ensemble. Il identifie certaines régularités de contenu et de forme pour effectuer une analyse simplifiée du texte.

En fait, un texte n'est pas seulement un sac de mots, mais bien véritablement un ensemble fortement structuré de termes qui permet de communiquer des informations d'une grande précision. Cette richesse et cette complexité sont prises en compte par plusieurs travaux de recherche.

Dans [Faiz 02], FAIZ R. et BISKRI I. proposent deux chaînes de caractères : Numexco et Gramexco. Ces deux chaînes privilégient un parcours des éléments du texte selon une perspective surtout numérique. Leur objectif consiste essentiellement à assister la classification du lexique d'un corpus textuel.

Dans d'autres travaux, CHEVALLET J-P. et HADDAD H.[6] pensent que l'utilisation d'index structurés permettra d'augmenter de manière significative la qualité des Systèmes de Recherche d'Information (SRI). La difficulté de cette approche tient dans la puissance d'expression du langage d'indexation qu'il convient de mettre en place, et, particulièrement, son adéquation à la recherche d'informations orientée vers la précision des réponses. Ils proposent pour aller vers plus de sémantique de modéliser les index en des arbres issues des relations homo syntagmatiques extraites automatiquement avec un degré de réussite suffisant pour être pris en compte dans un SRI.

Généralement, l'outil du *Text Mining* apporte une nouvelle puissance de traitement en identifiant des concepts et des relations au cœur des textes. Il apporte également une aide précieuse dans le cadre de la recherche d'informations, mais une validation par l'homme de la pertinence du contenu est encore indispensable [Lefebure et al. 01].

Nous allons maintenant quitter l'univers de données textuelles pour nous intéresser aux données de type son.

1.4.2 Le Sound Mining

Le développement de capacité de stockage de données presque illimitée combiné avec l'utilisation éclosante de l'Internet ont fait possible d'avoir accès aux énormes volumes de texte avec le clic d'une souris. Comme la technologie mûrit, les volumes même plus grands de données de type son tels que des discours capturés de la télévision, de la radio, des coups de téléphone, des réunions, des conférences, des présentations, … peuvent être examinés, fouillés, recherchés et extraits comme le texte.

Le *Sound Mining* consiste à rechercher des éléments communs ou à classer des sons en fonction de leur contenu. Les applications potentielles sont l'indexation et la recherche des pièces musicales à partir des bases de données. Ces applications[7] permettent aux utilisateurs de fouiller et recouvrer la musique, non seulement au moyen des requêtes textuelles (comme le titre, l'orchestre, le conducteur, la chanson de texte, le compositeur), mais aussi en fonction de contenu (comme la liste des pièces musicales de type 'energitic', 'regee', etc.) ou bien par une combinaison du son et du texte.

Nous allons maintenant quitter l'univers de données de type son pour nous intéresser aux données de type image.

1.4.3 L'Image Mining

Il s'agit de rechercher des relations entre les images ou des séquences d'images. L'*Image Mining* consiste à réaliser un système intelligent capable de fouiller dans les bases de données multimédia afin de fournir des images définis par l'utilisateur. Un des défis principaux dans la compréhension d'image est de développer un système flexible, adaptable, capable d'exécuter des tâches d'analyse d'image complexes et d'extraire des connaissances [Lefebure et al. 01].

---

[6] http://www-mrim.imag.fr/realisations/
[7] http://www.ipem.rug.ac.be/

Les images ont comme caractéristique principale, le manque d'un langage permettant d'en exprimer la sémantique, à l'inverse des textes et de la parole. Il en résulte une très grande difficulté pour proposer des systèmes de recherche d'images à base de concepts. En fait, le but de l'indexation d'un document image est d'extraire et de représenter le contenu nécessaire et suffisant pour qu'il soit trouvé par un utilisateur. Cette indexation se base donc sur une représentation et sur un processus d'extraction. Afin d'extraire des informations pertinentes « *Knowledge* » dans un contexte donnée, ZHANG J., HSU W. et LEE L. M. [HSU 01] proposent une procédure d'analyse d'images formée de quatre niveaux d'information : niveau pixel, niveau objet, niveau sémantique et niveau connaissance. Les techniques d'indexage et de recherche sont aussi inclues dans cette procédure pour soutenir le flux d'informations parmi les quatre niveaux.

Le défi fondamental dans l'*Image Mining* est comment identifier des objets, des relations entre ces objets, ainsi que des connaissances à partir de le plus bas niveau : niveau pixel.

Il est à signaler qu'une image est considérée comme une mosaïque de différentes régions de textures et les caractéristiques d'image associées aux régions peuvent être employées pour la recherche et la récupération dans des bases d'images.

Actuellement, les recherches s'orientent vers l'utilisation des descripteurs uniques pour un domaine particulier tels que la couleur, la forme, la texture, etc. Quoiqu'il y ait des diverses descripteurs disponibles, l'objectif est d'identifier les bons descripteurs afin d'extraire des informations pertinentes à partir des images. Les techniques actuelles de recherche et de classification sont concentrées sur les descripteurs visuels (couleur, texture, forme, etc.). Plusieurs systèmes tels que QBIC, RetievalWare [Dowe 93] et PhotoBook [Pentland 96] ont tous une variété des descripteurs visuels, mais sont toujours employés dans des domaines particuliers. Jain et al [Jain 96] utilisent la couleur avec la forme pour la classification, Ma et al [Manjunath 97] utilisent la couleur et la texture pour la recherche d'images.

Nous allons maintenant passer à un univers de données qui regroupe à la fois le texte, l'image et le son : c'est la vidéo.

1.4.4 Le Video Mining

Le *Video Mining* est l'extension de l'*Image Mining* au domaine de la vidéo. Très théorique pour l'instant compte tenu des performances de la machine qu'il requiert, le *Video Mining* consiste à rechercher des éléments communs ou à classer des vidéos en fonction de leur contenu. Les applications potentielles sont l'indexation de bases de films ou l'optimisation des grilles de programmes des opérateurs de télévision.

Actuellement, les systèmes de recherche de vidéo assurent une indexation par le contenu de documents vidéo. Ces systèmes sont basés sur une indexation automatique partielle dont les résultats peuvent être corrigés ou complétés par l'utilisateur en fonction de compromis coût-qualité recherché. Plusieurs briques de base de ces systèmes sont développés (segmentation en plans, recherche des mouvements de caméras, détection et suivi d'objets mobiles, caractérisation par vecteurs de caractéristiques, détection de personnages). Parallèlement, à ces traitements effectués sur la bande image, une indexation à partir de la transcription de la bande son ou des

sous-titres lorsqu'ils sont disponibles sera intégrée et couplée à des systèmes d'indexation d'image [Doermann 00]. Mais la réutilisation de techniques d'indexation de textes ou d'images conduit à un système d'indexation incomplet ne tenant pas compte de toutes les caractéristiques propres aux documents vidéo (synchronisation son et image, par exemple). Dans ce contexte, QUENOT G. et MULHEM P.[8] proposent deux approches : d'une part, une approche conduisant à la proposition d'un modèle du contenu sémantique des documents vidéos et d'autres part, une approche permettant d'étudier les caractéristiques extraites du signal utile pour la recherche d'information.

Pour le modèle du contenu sémantique de documents vidéo, QUENOT G. et MULHEM P. utilisent une approche par strate qui ne possède pas, à priori, les limitations de la représentation structurelle, car elle peut représenter des éléments qui ne coïncident pas avec la structure de la vidéo. Ces travaux permettent de représenter finement des éléments tels que les flash-back. Les graphes conceptuels de Sowa utilisés dans leurs modèles permettent de représenter des descriptions précises : on peut exprimer formellement le fait que la même personne réalise deux actions dans deux parties de la vidéo, alors qu'avec une description textuelle une ambiguïté peut toujours exister. De plus, les graphes conceptuels supportent des algorithmes rapides de recherche des parties de vidéo pertinents malgré la complexité des descriptions.

Le deuxième objectif est d'extraire, automatiquement, au niveau du signal de la bande un certain nombre de caractéristiques utiles pour l'indexation des documents vidéo. QUENOT G. et MULHEM P. ont développé des outils permettant la segmentation en plans en utilisant des techniques basées sur des histogrammes de couleur, des suivis de contours et des analyses du mouvement. La similarité entre plans permet de structurer les documents en séquences ayant un sens au niveau sémantique.

1.5 Conclusion

Dans ce chapitre, nous avons exploré le processus du *Data Mining* et les principales méthodes utilisées. En fait, nous avons précisé les principes de fonctionnement, ainsi que ses domaines d'application, ses avantages et ses limites.

Nous avons proposé aussi une vision de l'évolution du *Data Mining* vers le traitement des données multimédia tels que le texte, le son, l'image et la vidéo.

Nous allons se concentrer dans la suite de ce travail les différentes méthodes de classification.

[8] http://www-mrim.imag.fr/presentation/video.php

Chapitre 2

Classification et apprentissage

2.1 Introduction

Le besoin de décider à quel groupe parmi plusieurs possibles un objet appartient est un problème étonnamment général. Dans la plus part des cas, sauf ceux pour lesquels la classification est évidente et triviale, la tâche de classification dépend encore d'une expertise humaine basée sur des observations. Par exemple, un médecin diagnostique des maladies en se basant sur des années d'expérience et de pratiques médicales. De manière similaire, un botaniste identifie les espèces des plantes, un psychologue reconnaît des types de personnalités …

Avant même que l'utilisation des ordinateurs ne soit répandue, des statisticiens et d'autres ont développé des méthodes de classification objective basées sur la théorie classique de probabilités. Cependant, comme la classification a constitué un problème important dans de multiples et divers domaines d'applications, elle a souffert d'avoir été résolue de nombreuses fois. A chaque fois qu'une discipline a étudié le sujet de la classification, elle a introduit sa propre terminologie, ses propres notations et ses propres méthodes favorites. Ainsi la classification se retrouve dans la reconnaissance des formes, dans l'analyse discriminante, dans la théorie de la décision,… La plus récente et la plus importante réutilisation de la classification est probablement dans le domaine des « systèmes experts », en Intelligence Artificielle.

Dans ce chapitre, nous situons la classification dans le cadre plus général de l'apprentissage ainsi que ses principaux aspects. Sans prétention d'exhaustivité, nous présentons ensuite des méthodes de classification issues des domaines différents et qui nous paraissent représentatives d'un courant scientifique ou caractéristiques d'une démarche théorique, telles que:

- Régression et analyse discriminante,
- Discrimination fonctionnelle,

- Méthodes probabilistes : théorie bayésienne de la décision, discrimination par boules, k plus proches voisins, classifieurs bayésiens,…
- Classification par génération de règles : les arbres de décision, les méthodes polythétiques de génération de règles, le système d'inférence Floue,...

Enfin, nous concluons cette partie en présentant les grandes lignes de la méthode de classification que nous proposons dans ce mémoire.

2.2 L'apprentissage

L'apprentissage est une problématique ancienne qui a été abordée dans de nombreux domaines dont la Psychologie, la Statistique, la Didactique, l'Intelligence Artificielle,… Donner une définition précise de l'apprentissage reste un problème délicat et difficile, comme le note Stéphane Boucheron dans [Boucheron 92] : *« l'apprentissage je sais ce que c'est tant qu'on ne me demande pas de définir »*.

Le terme apprentissage ou apprentissage automatique correspond à une caractéristiques des machines. Il s'agit plus précisément de leur capacité à organiser, construire et généraliser des connaissances pour une utilisation ultérieure, de leur capacité de tirer profit de l'expérience pour améliorer la résolution d'un problème. L'apprentissage peut être appréhendé par la définition que donne Herbert Simon [Simon 84] : *« learning denotes changes in the system that are adaptative in the sense they enable the system to do the same task or tasks drawn from the same population more efficiently and more efficiently the next time»*.

L'apprentissage automatique est utilisé pour remplacer l'acquisition des règles de classification auprès d'un expert du domaine, il permet ainsi de contourner la difficile tâche de l'acquisition des connaissances. Une règle de classification permet de prédire la classe d'un nouvel exemple, la classe étant une variable discrète d'une importance particulière. A titre d'illustration, un exemple peut correspondre à un patient décrit par un ensemble d'attributs comme l'age, le sexe, la pression sanguine, … et la classe pourrait être un attribut binaire concluant ou non à l'affectation du patient par une maladie particulière. L'apprentissage automatique consisterait ici à « apprendre » des règles de classification à partir d'un ensemble de descriptions de patients.

Nous nous intéressons à un type d'apprentissage particulier qui est l'apprentissage inductif encore appelé apprentissage à partir d'exemples ou d'observations. Le processus de l'apprentissage inductif peut être considéré comme une recherche de descriptions générales plausibles qui expliquent les données d'entrée et qui sont utiles pour en prédire de nouvelles (résultats, données de sortie, etc.) [Dietterich et al. 83]. En d'autres termes, l'inférence inductive essaie de dériver une description complète et correcte d'un phénomène donné à partir d'observations spécifiques de ce phénomène ou de parties de lui.

Deux approches distinctes de l'apprentissage existent. La première qualifiée d'apprentissage non supervisée vise à regrouper en classes des objets en se basant sur des ressemblances ou similarités entre eux. Le deuxième type d'apprentissage est l'apprentissage supervisé qui se base quand à lui sur un ensemble d'apprentissage constitué d'objets dont la classe est connu ***a priori***.

2.2.1 L'apprentissage non supervisé

L'apprentissage non supervisé, encore appelé apprentissage à partir d'observations ou découverte, consiste à déterminer une classification « sensée » à partir d'un ensemble d'objets ou de situations données (des exemples non étiquetés). On dispose d'une masse de données indifférenciées, et l'on désire savoir si elles possèdent une quelconque structure de groupes. Il s'agit d'identifier une éventuelle tendance des données à être regroupées en classes. Ce type d'apprentissage, encore appelé ***Clustering*** ou ***Cluster Analysis***, se trouve en classification automatique et en taxinomie numérique.

Cette forme de classification existe depuis des temps immémoriaux. Elle concerne notamment les sciences de la nature (**FIG. 2.1**), les classifications des documents et des livres mais également la classification des sciences élaborées au cours des siècles par les philosophes [Parrochia 97].

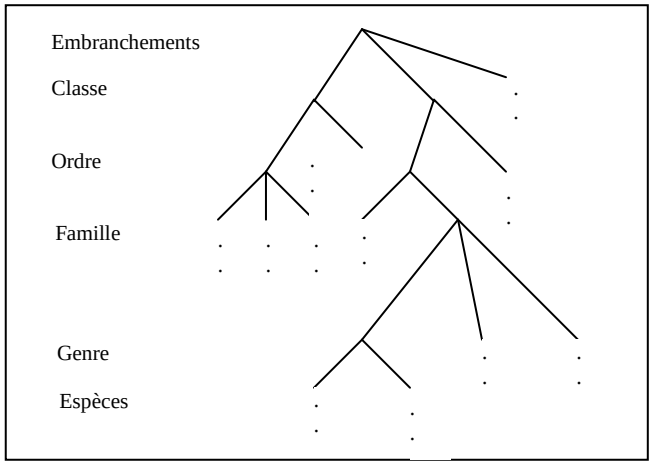


FIG. 2.1 - *Extrait de la classification taxinomique de Linné.*

L'automatisation de la construction de classification constitue aujourd'hui un véritable domaine de recherche. La notion clé utilisée pour créer des classes d'objets est une mesure de la similarité entre les objets. Les classes ou concepts sont construits de façon à maximiser la similarité intra-classes et à minimiser la similarité inter-classes.

L'apprentissage non supervisé correspond également à la classification conceptuelle, où une collection d'objets forme une classe si cette classe peut être décrite par un concept, compte tenu d'un ensemble de concepts prédéfinis [Stepp et al. 83].

2.2.2 L'apprentissage supervisé

Dans le cas de l'apprentissage supervisé, on dispose d'un ensemble de données étiquetées, ou d'exemples qui se sont vus associés une classe par un professeur ou un expert. Cet ensemble d'exemples constitue la base d'apprentissage. Les méthodes d'apprentissage supervisé se donnent alors comme objectif général de construire à partir de la base d'apprentissage, des classifieurs, ou fonctions de classement. Une telle fonction permet, à partir de la description d'un objet, de reconnaître un attribut particulier, la classe (**FIG. 2.2**) [Borgi et al. 01].

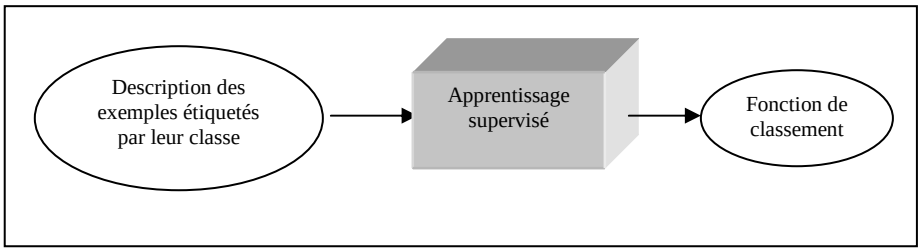


FIG. 2.2 - *L'apprentissage supervisé.*

Dans [Michalski et al. 83], l'inférence inductive est définie comme un processus qui à partir d'une connaissance spécifique observée sur certains objets et d'une hypothèse inductive initiale, permet d'obtenir une assertion inductive impliquant ou rendant compte fortement ou faiblement des observations. Dans le cas de l'apprentissage inductif supervisé, qui est un sous domaine de l'inférence inductive, la connaissance spécifique consiste en un ensemble d'objets appartenant à des classes connues. L'assertion inductive est exprimée par une *règle de classification* qui assigne une classe à chaque objet. L'implication forte est satisfaite si la règle classe correctement tous les objets connus.

2.3 Présentation de quelques méthodes de classification

La classification, qui consiste à assigner un objet à un groupe parmi un ensemble de groupes prédéterminés, est d'une importance fondamentale dans de nombreux domaines scientifiques et techniques. Sa grande importance a conduit chaque discipline à développer ses propres méthodes et sa propre terminologie, ce qui a abouti à faire de la classification une collection d'îlots, plutôt isolés les uns des autres. Ainsi les termes « Reconnaissance des Formes », « Analyse Discriminante » et « Apprentissage Supervisé » font tous référence à la classification.

Les méthodes les plus anciennes viennent du champ de la statistique, d'autres sont issues de la Reconnaissance des Formes ou d'Intelligence Artificielle, domaines plus récents, et dont certaines approches leur sont spécifiques. Nous présentons ci-après des approches de classification issues des différents domaines.

2.3.1 L'approche statistique

La statistique et l'analyse de données sont parmi les premières disciplines à s'être intéressées aux problèmes de classification, par classification nous entendons ici classification supervisée et non pas automatique. Ces disciplines ont étudié et proposé de multiples méthodes de classification, comme par exemple la régression, le modèle linéaire ou encore l'analyse discriminante. Cependant, certaines de ces techniques imposent des contraintes difficilement interprétables par l'utilisateur qui n'est en général ni un spécialiste des mathématiques ni des statistiques. Par exemple la fonction discriminante de Fisher ou la régression supposent un modèle linéaire sous-jacent, en analyse discriminante les méthodes paramétriques nécessitent d'émettre des hypothèses sur les lois de probabilités des variables étudiées, … De plus les solutions proposées sont souvent exprimées sous forme d'équations, de combinaisons linéaires et sont par conséquent peu intelligibles pour un utilisateur quelconque.

Les exemples sont représentés par des points dans un espace de dimension p (IR^P), p étant le nombre d'attribut ou de variables décrivant un exemple. Les généralisations sont le plus souvent représentées par des surfaces de décisions dans cet espace (hyperplans, polynômes d'un certain degré, …).

Les méthodes les plus caractéristiques, les plus classiques, sont la régression et l'analyse discriminante. Cette dernière englobe les méthodes probabilistes (règles bayésienne, modèle normal multidimensionnel) que nous aborderons dans la partie qui suit.

2.3.1.1 La régression

L'objectif de la régression est de prédire une variable **Y**, dite variable expliquée ou critère, à l'aide d'un ensemble de variables $\mathbf{X_1}$, …, $\mathbf{X_p}$ dites variables explicatives ou prédicteurs. Toutes ces variables sont quantitatives continues. De manière générale, on désire déterminer une fonction $\boldsymbol{\varphi}$ telle que $\boldsymbol{\varphi}(\mathbf{X_1}, \ldots, \mathbf{X_p})$ soit aussi proche que possible de **Y**.

Le modèle de régression le plus répandu et le plus important dans la pratique est le modèle linéaire (dépendance linéaire entre **Y** et les variables explicatives). Rappelons que dans le cas de la régression linéaire simple, c'est-à-dire avec une unique variable explicative **X** (p=1), le modèle recherché est de la forme :

$$Y = \alpha + \beta X + \varepsilon$$

Le problème est d'estimer $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ ainsi que la variance du résidu $\boldsymbol{\varepsilon}$. Cette estimation se fait sur l'échantillon d'apprentissage $\boldsymbol{\Omega}$ en utilisant la méthode des moindres carrés [Saporta 90].

La régression linéaire multiple généralise la régression linéaire simple, et sa pratique reste assez délicate. L'objectif d'un modèle de régression linéaire multiple est de trouver l'hyperplan qui ajuste au mieux le nuage de points constitué par les valeurs des exemples pour toutes les variables ($\mathbf{X_1}$, …, $\mathbf{X_p}$ et **Y**). Par conséquent le modèle va chercher les coefficients de l'équation tels que le carré des écarts des points à l'hyperplan soit minimal, c'est toujours le critère des moindres carrés qui est utilisé. Le lecteur intéressé par ces méthodes trouvera de plus amples détails et explications dans les ouvrages [Tomassone et al. 88] et [Saporta 90].

Les méthodes de régression sont très utilisées en matière de prédiction, mais leur particularité est que toutes les variables sont quantitatives continues. Une extension à des variables explicatives qualitatives est possible, mais la variable expliquée **Y** reste quant à elle toujours quantitative. Dans le cadre d'un problème de Reconnaissance de Formes, le nombre de classes est fini, ce qui correspond à une variable expliquée **Y** qualitative, cette situation non prise en compte par la régression est traitée par l'analyse discriminante.

2.3.1.2 L'analyse discriminante

On peut considérer l'analyse discriminante comme une extension du problème de la régression dans le cas où la variable à expliquer est (**Y**) qualitative. On distingue classiquement deux aspects en analyse discriminante :

- L'analyse discriminante à but descriptif, appelée l'Analyse Factorielle Discriminante. Son objectif est de déterminer les combinaisons linéaires de variables qui permettent de séparer au mieux les différentes classes, et de donner une représentation graphique qui rende compte de cette séparation,

- L'analyse discriminante à but décisionnel, son objectif est de prédire la classe d'un nouvel objet décrit par la valeur de ces attributs ou variables explicatives.

a) L'analyse discriminante à but descriptif

L'Analyse Factorielle Discriminante (**AFD**) a pour but de trouver les axes factoriels qui discriminent au mieux les **C** classes de l'ensemble d'apprentissage, c'est-à-dire tels que les projections (sur ces axes) des exemples d'une même classe soient les plus regroupés possibles tandis que celles des exemples de classes différentes soient le plus éloignés possible.

Lorsque le nombre d'attributs est élevé, supérieur à deux ou trois, il est difficile d'avoir une représentation graphique simple de données. L'**AFD** cherche alors des variables synthétiques qui permettent de représenter au mieux les données et la discrimination des exemples selon les classes. En analyse discriminante, les matrices d'inertie et inter classes sont utilisées pour formuler le critère de séparabilité des classes [Saporta 90].

Etant donné une distribution de **N** exemples selon **C** classes dans un espace de dimension p, l'espace de représentation, l'**AFD** consiste donc à déterminer de nouvelles variables, dites variables discriminantes, correspondant à des directions de IR^p qui séparent le mieux possible en projection les **C** classes. Les variables discriminantes sont des combinaisons linéaires des variables explicatives, c'est-à-dire des attributs initiaux. Pour la première variable discriminante, on cherche la meilleure combinaison ***U*** de la forme :

$$U = a_1X_1 + a_2X_2 \ \ldots + a_iX_i + \ldots + a_pX_p$$

Avec	U	la meilleur combinaison linéaire
	X_i	les p variables explicatives centrées, i = 1…p
	a_i	les coefficients de la combinaison
On note	$\mathbf{g}_j$	le centre de gravité du nuage de points de classe y_j, j = 1…**C**

Chercher la meilleure combinaison linéaire, revient à chercher l'axe **a** le plus discriminant en projection : les C centres de gravité de chaque classe doivent être aussi séparés que possible en projection sur cet axe **a**, tandis que chaque sous- nuage (correspondant à chaque classe) doit se projeter de manière groupée autour de la projection de son centre de gravité. En d'autres termes, l'inertie inter-classes doit être la plus grande possible, et l'inertie intra-classes doit être minimale. Le critère retenu consiste à maximiser le rapport de l'inertie inter-classes sur l'inertie intra-classes. Ce maximum est atteint si **a** est le vecteur propre de $V^{-1}B$ associé à sa plus grande valeur propre λ_1 :

$$V^{-1}B\mathbf{a} = \lambda_1\mathbf{a}$$

Avec E la matrice N*p pour les N exemples des valeurs centrées des p attributs
N est le nombre d'exemples, i.e. le cardinal de **Ω**
B la matrice de variance ou d'inertie inter-classes
W la matrice de variance ou d'inertie intra-classes
V la matrice de variance ou d'inertie totale

$$V = \frac{1}{N} E^t E$$

$$V = W + B$$

a le premier vecteur propre de $V^{-1}B$
λ_1 la première valeur propre de $V^{-1}B$

La figure ci-dessus (**Fig. 2.3**) illustre le meilleur axe, l'axe **a**, pour un échantillon réparti selon deux classes et décrit par deux attributs.

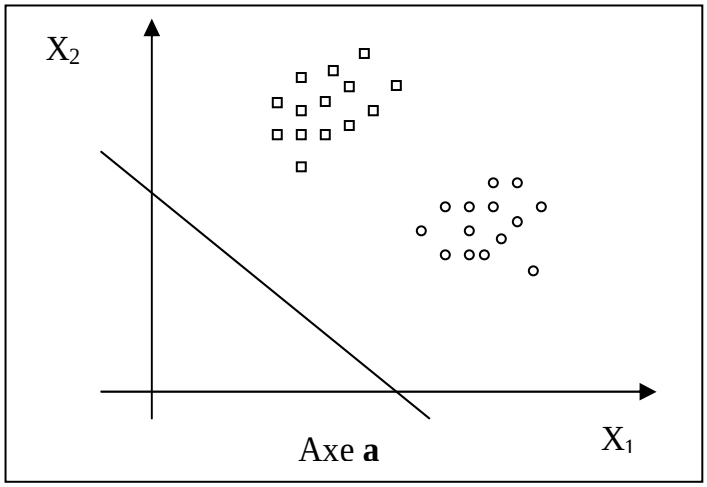


Fig. 2.3 - *Recherche du meilleur axe par l'AFD.*

Les coefficients de la meilleure combinaison linéaire **U** des attributs correspondent aux composantes du vecteur propre **a** = (a_1, …, a_p) :

$$\boldsymbol{U = a_1X_1 + a_2X_2 \ \ldots + a_iX_i + \ldots + a_pX_p}$$

Par analogie avec l'analyse factorielle en composantes principales, le pouvoir discriminant de cette combinaison est donné par λ_1**,** la première valeur propre de la matrice $V^{-1}B$. Cette valeur propre associée à l'axe le plus discriminant permet d'évaluer la séparabilité linéaire des classes, elle constitue toutefois une mesure pessimiste du pouvoir discriminant d'un axe. λ_1**,** qui varie entre 0 et 1, peut s'interpréter de la façon suivante :

- λ_1= 0 correspond au cas où le meilleur axe ne permet pas de séparer les centres de gravité $\mathbf{g}_j$, c'est le cas où ils sont confondus. Les nuages sont donc concentriques et aucune séparation linéaire n'est possible (**Fig. 2.4**).

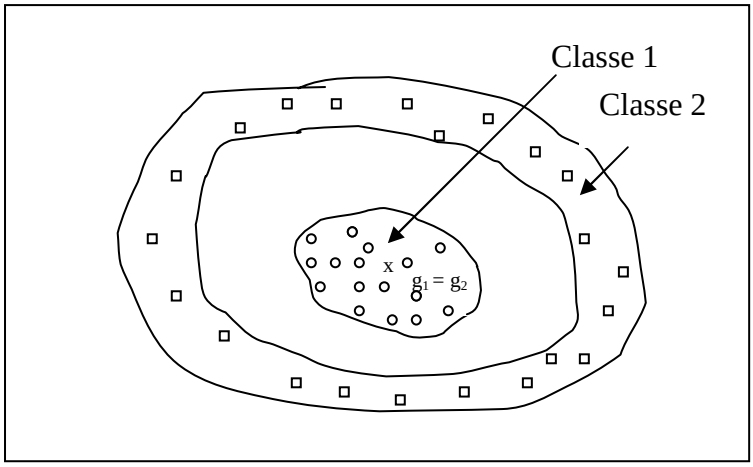


FIG. 2.4 - *Un exemple de nuages de points concentriques*.

- λ_1= 1, les classes sont linéairement séparables, il y a discrimination parfaite si les centres de gravité se projettent en des points différents (**FIG. 2.3**).

La projection des exemples sur le premier axe factoriel permet de visualiser la répartition des exemples selon leur classe. La Figure 2.5 représente la projection des exemples de la Figure 2.3 sur le premier axe factoriel, dans notre illustration cet axe permet de discriminer les exemples parfaitement.

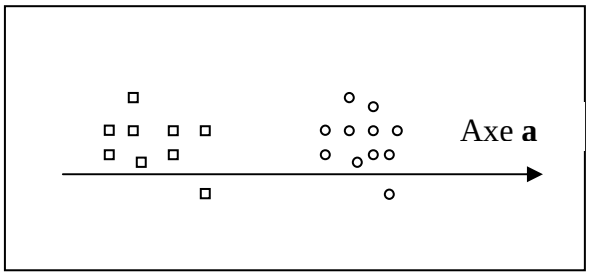


FIG. 2.5 - *Projection des exemples sur le 1er axe factoriel*.

Quand le nombre de classes est supérieur à deux, il se peut qu'un seul axe ne suffise pas pour visualiser la séparation des classes. L'AFD permet de trouver d'autres axes factoriels et permet ainsi de définir un sous espace discriminant. Le nombre d'axes discriminants est égal à C-1, C étant le nombre de classes. Il est donc possible de chercher la deuxième, puis la troisième, … meilleure variable discriminante en choisissant le vecteur propre associé à la deuxième, troisième, … valeur propre de $V^{-1}B$. Dans le cas de deux axes discriminants, le nuage des N exemples peut être représenté en projection sur le plan défini par ces deux axes, ce plan est alors celui qui permet de visualiser le mieux la séparation des exemples en classes.

b) L'analyse discriminante à but décisionnel

Ayant trouvé la meilleure représentation de la séparation en C classes des n exemples, on peut alors chercher à affecter un nouvel objet ou individu Π à l'une des classes. Cela nécessite de définir des règles d'affectation, ces règles peuvent être de deux sortes : géométriques ou probabilistes.

Les règles géométriques

Les règles géométriques d'affectation consistent à calculer les distances de l'observation à classer Π à chacun des C centres de gravité g_1, …, g_i, …, g_c. La classe attribuée à Π est celle de plus faible distance, i.e. la classe y_k vérifiant :

$$d(x,g_k) = \min_{J=1,\ldots,c} (d(x,g_j))$$

Avec x vecteur représentatif de l'observation Π constitué des valeurs centrées des p attributs, x correspond au point associé à Π dans un espace à p dimensions.
g_j centre de gravité des exemples de classe y_j, dans un espace à p dimension.

La figure 2.6 illustre l'utilisation de la règle géométrique d'affectation dans le cas d'exemples répartis selon deux classes et représentés par deux attributs.

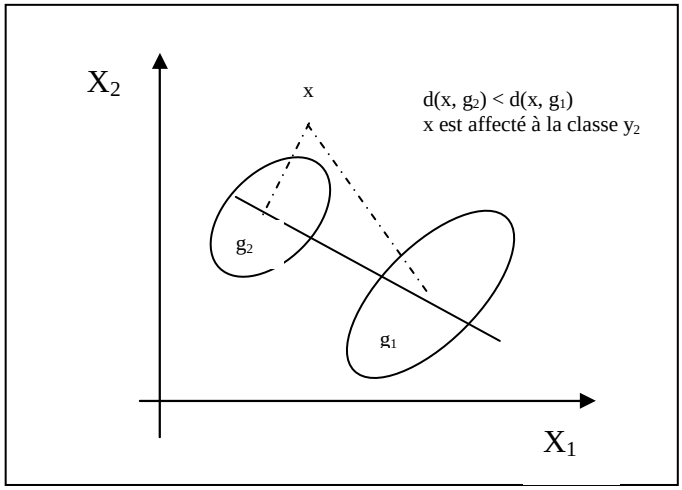


FIG. 2.6 - *Règle géométrique avec la distance euclidienne.*

Il reste à définir la distance ***d*** à utiliser. La distance généralement retenue est celle de Mahalanobis définie par :

$$d^2(x, g_j) = (x- g_j)^t V^{-1} (x- g_j)$$

Avec V matrice de variance ou d'inertie totale.
En développant cette quantité, on trouve :

$$d^2(x, g_j) = x^t V^{-1} x + g_j^t V^{-1} g_j - 2 x^t V^{-1} g_j$$

Comme $x^tV^{-1}x$ ne dépend pas de la classe y_j, minimiser la distance de Mahalanobis revient à chercher le minimum de $g_j^t V^{-1} g_j - 2 x^t V^{-1} g_j$, quantité linéaire par rapport aux coordonnées de x. On est donc ramené à maximiser une fonction discriminante linéaire $f_{gj}(x)$

$$f_{gj}(x) = 2 x^t V^{-1} g_j - g_j^t V^{-1} g_j,$$

Il y a donc C fonctions discriminantes linéaires et les surfaces séparent les classes sont des hyperplans. En représentant ces fonctions discriminantes sur le premier plan factoriel défini par l'AFD, il est possible de les visualiser (voir la figure 2.7).

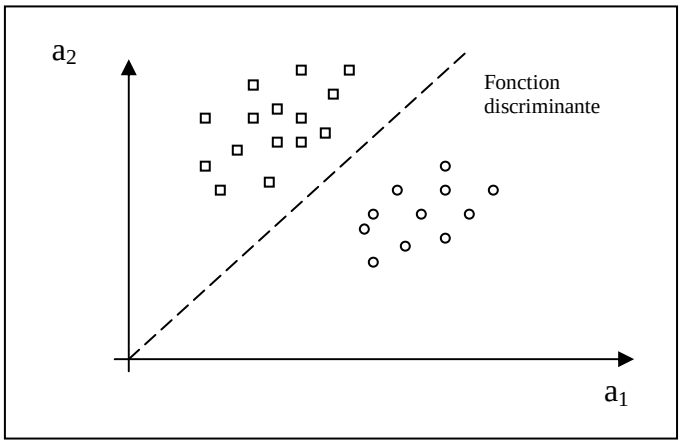


FIG. 2.7 - *Résultat de l'analyse discriminante.*

Dés 1936 Fisher proposait de séparer deux classes par un « hyperplan », il est équivalent de trouver un axe sur lequel les projections des deux classes sont séparées au mieux. La séparation linéaire, pour laquelle la frontière de séparation entre deux classes est linéaire, ne possède guère de justification théorique ou physique mais la simplicité et la rapidité des algorithmes de décision sont incontestables.

Les règles probabilistes

Dans ce cas, la règle d'affectation dépend de la probabilité qu'un individu Π représente par le vecteur X, provienne de la classe y_j :

$$P(y_j/X)$$

L'observation X est affectée à la classe qui a la probabilité à posteriori maximale : c'est la règle de Bayes.

Il est donc nécessaire d'estimer $P(y_j/X)$. Diverses méthodes existes, et n'ont pas toujours été développées ni utilisées dans le cadre de l'analyse discriminante. C'est la raison pour laquelle, nous préférons consacrer un paragraphe indépendant aux méthodes de classification probabilistes (voir la partie **2.3.2 L'approche probabiliste**).

c) La discrimination fonctionnelle

De manière plus générale, la discrimination fonctionnelle consiste à définir des fonctions permettant de séparer partiellement ou totalement des classes représentées par les vecteurs attributs de leurs exemples [Tou et al. 74]. La forme de la fonction de discrimination dépend de la forme des classes, et n'est pas toujours linéaire.

La figure 2.8 illustre une fonction de discrimination linéaire dans le cas de deux classes : séparation de deux classes par une droite [Bonnet 97]. La procédure consiste à

chercher la droite d'équation $w_1.x_1 + w_2.x_2 + w_3 = 0$ telle que tous les points de la classe y_1 soient d'un côté ($W.X<0$, avec $W = (w_1, w_2, w_3)$ et $X = (x_1, x_2, 1)$) et tous les points de la classe y_2 soient de l'autre côté ($W.X>0$). En pratique, à chaque présentation d'un nouvel exemple, on modifie les coefficients (w_1, w_2, w_3) de la droite si le signe du produit scalaire W.X n'est pas correct.

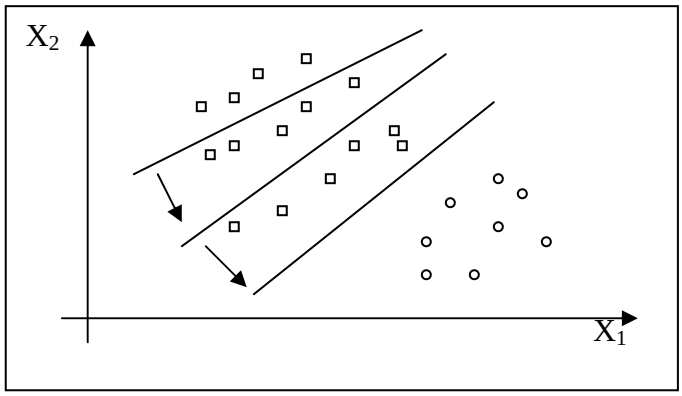


FIG. 2.8 - *Un exemple de fonction discriminante linéaire.*

Cette fonction de discrimination peut se généraliser aux cas de plusieurs classes (hyperplans) dans un espace de représentation de dimension p [Bélaïd et al. 92]. Le classifieur correspondant est alors dit linéaire [Fukunaga 72].

Plusieurs méthodes de détermination des fonctions de discrimination linéaires existent : l'analyse discriminante bien sûr, mais aussi l'algorithme du Perceptron, l'algorithme de Arkadev et Braveman [Arkadev et al. 67] … Ces techniques ainsi que d'autres méthodes de discrimination fonctionnelle sont présentées dans [Bélaïd et al. 92].

Lorsque les classes ne sont pas linéairement séparables mais sont formées de sous-classes qui, elles, sont linéairement séparables comme l'illustre la Figure 2.9, la solution consiste à traiter chaque sous classe comme une classe distincte. La discrimination est alors dite linéaire par morceaux.

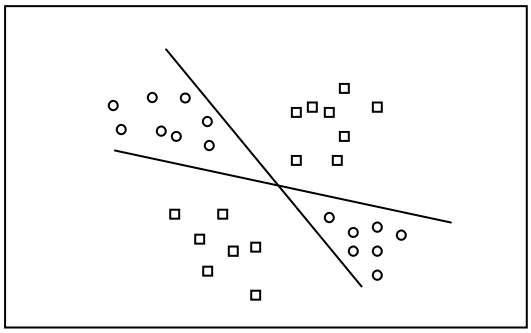


FIG. 2.9 - *Discrimination linéaire par morceaux.*

Lorsque les classes ne sont pas linéairement séparables, il est possible de définir des fonctions de discrimination non linéaires, par exemple de forme polynomiale, exponentielle, …

De nombreuses méthodes de discrimination statistique se basent sur la théorie des probabilités pour l'affectation d'une observation à une classe. C'est le cas par exemple de l'analyse discriminante à but décisionnel lorsqu'elle met en œuvre des règles probabilistes. Lorsque les hypothèses sur la forme générale de la distribution des exemples conditionnellement à leur classe d'appartenance sont faites, le recours aux statistiques pour estimer certains paramètres de cette distribution devient incontournable. Les statistiques et les probabilités sont des disciplines étroitement liées. Après avoir brièvement exposé certaines approches statistiques de discrimination, nous présentons maintenant des méthodes de classification supervisée issue d'une approche probabiliste.

2.3.2 L'approche probabiliste

L'apprentissage supervisé vise à inférer à partir d'un ensemble limité d'exemples des règles générales s'appliquant avec des bonnes chances de succès à une situation nouvelle. Le cadre naturel pour obtenir ce type de garantie est celui de la théorie des probabilités. Une littérature abondante traite depuis de nombreuses années la théorie des probabilités. Nous ne prétendons nullement, dans cette partie, faire le point sur la question mais nous nous limitons à une brève présentation des principales notions probabilistes utilisées dans le domaine de la classification supervisée.

La règle de décision bayésienne consiste à associer à chaque nouvel individu à classer Π la classe la plus probable, la fonction de classement φ est alors définie par :

$$\varphi : R \text{ ----------> } C$$
$$\chi(\Pi) = X \text{ ----------> } y_j \,/\, \forall y \in C \;\; P(y_j/X) \geq P(y/X)$$

$P(y_j/X)$ est une probabilité ***a posteriori***. Cette règle est assez simple mais difficile à mettre en œuvre en pratique car les probabilités ***a posteriori*** sont généralement inconnues. Elles doivent dons être estimées.

La probabilité conditionnelle que l'individu Π, représenté par X, provienne de la classe y_j est donné par la formule de Bayes :

$$\mathbf{P}(yj/X) = \frac{P(X/yj)P(yj)}{P(X)}$$

ce qui s'écrit encore :

$$\mathbf{P}(yj/X) = \frac{P(X/yj)P(yj)}{\sum_{i=1}^{C} P(X/yi)P(yi)}$$

Avec l'hypothèse que l'ensemble $\{y_1, y_2, \ldots, y_C\}$ constitue un système complet d'événement, en d'autres termes que chaque individu Π de la population π appartient nécessairement à une classe et une seule. Cette hypothèse est supposée être vérifiée dans le cadre d'un problème de Reconnaissance des Formes.

Le problème revient donc à déterminer **P(X/*yj*)** pour chaque classe y_j, j = 1…C. Cette probabilité conditionnelle peut être échantillonnée par expérience. Une statistique permet alors de l'estimer. Deux types de solutions sont possibles :

- soit on se donne a priori des lois de probabilités paramétrées et on estime les paramètres en utilisant l'ensemble d'apprentissage Ω, ces méthodes sont dites paramétriques. Une méthode paramétrique est donc une méthode qui fait appel à une hypothèse a priori sur l'expression analytique de la fonction de représentation ; par exemple, l'analyse factorielle discriminante suppose que la fonction de répartition des classes est gaussienne. Une abondante littérature existe sur ce sujet que nous ne développerons pas dans le cadre de ce mémoire, pour plus de précision voir [Duda et al. 73] [Fukunaga 72] [Simon 85] [Saporta 90].

- soit on cherche à interpoler la fonction de décision à partir des échantillons donnés par l'ensemble d'apprentissage Ω. On ne fait donc pas d'hypothèse spécifique sur la famille de loi de probabilités. Il s'agit alors de techniques non paramétriques dont nous présentons succinctement les principales approches [Duda et al. 73].

Des méthodes probabilistes de discrimination sont présentées dans [Caraux et al. 96] [Duda et al. 73] [Fukunaga 72] et la théorie bayésienne de la décision y est étudiée en détails. Le lecteur intéressé pourra y trouver notamment la règle de décision de Bayes d'erreur minimale, la règle du maximum qui introduit la notion de coût associé à un mauvais classement.

2.3.3 Classification par génération de règles

Les méthodes de classification sont souvent considérées comme charnières entre les approches numériques et symboliques car elles intègrent des représentations symboliques et numériques, elles ont un fort pouvoir explicatif et sont des méthodes non paramétriques. Elles consistent en général à générer directement ou indirectement des règles de classification. Il s'agit de règles de production de la forme :

Si [prémisse] Alors [conclusion]

Nous nous intéressons, dans cette partie, aux méthodes fournissant des classifieurs composés d'une base de règles de cette forme. La prémisse est une conjonction de descripteurs logiques du type Attribut = valeur, et la conclusion de type Classe = modalité.

L'interprétation d'une règle *Si [prémisse] Alors [conclusion]* est que si une situation satisfait la *[prémisse]*, alors on peut inférer la *[conclusion]*. Typiquement, la *[prémisse]*, encore appelée condition, est une conjonction de propositions. Elle prend la forme d'une expression quelconque exprimée dans un certain langage formel. De manière générale, la conclusion est une action, une décision ou une attribution de valeur à des variables, mais dans le cas de règles de classification, la conclusion est de la forme «appartenance à une classe parmi les classes de $\boldsymbol{C} = \{y_1, y_2,\ldots, y_C\}$».

Les méthodes d'apprentissage par inductions développées dans les cadres de l'IA sont souvent des méthodes performantes quand les attributs sont symboliques, c'est-à-

dire qu'ils prennent leurs valeurs dans un ensemble discret et de cardinalité faible. Par contre, ces méthodes rencontrent souvent des difficultés à traiter des attributs numériques ou numériques-symboliques. Un attribut numérique prend ses valeurs dans un ensemble continu. Un attribut numérique-symbolique est un attribut dont les valeurs peuvent être numériques ou symboliques.

L'objectif de l'apprentissage inductif par génération de règles est d'identifier un nombre minimal de règles qui généralisent les exemples de la base d'apprentissage. Une règle peut être considérée comme une combinaison de variables et de valeurs. Le problème de recherche d'un sous ensemble minimal de combinaisons est un problème NP-Complet et les méthodes existantes sont toujours heuristiques [Bonnet 84]. Ces heuristiques diffèrent par leur façon de construire les combinaisons minimales. La plupart des méthodes de génération de règles construisent les prémisses en ajoutant un par un des descripteurs correspondant à des optima locaux, ce sont des méthodes mono-attributs ou monothétiques. On distingue les approches mono-attributs des approches multi-attributs (voir les sections suivantes : 2.3.3.1, 2.3.3.2 et 2.3.3.3).

Nous nous proposons de présenter les arbres de décision, qui correspondent à une approche de classification mono-attribut, ainsi qu'une méthode de classification multi-attributs : la méthode IPR. Enfin, nous présentons les grandes lignes de notre méthode de classification.

2.3.3.1 Les arbres de décision

La construction des arbres de décision, cas particulier des graphes d'induction [Rakotomalala 97], n'est pas une technique récente de l'Intelligence Artificielle, puisque des travaux d'analyse de données et de théorie des questionnaires [Picard 72] ont étudié la construction d'arbres de décision dans les années 60 [Hunt et al. 66]. Trois systèmes ont plus particulièrement marqué les travaux sur les arbres de décision : ID3 et C4.5 [Quinlan 83] [Quinlan 93] dans la communauté de l'IA, et CART [Breiman et al. 84] dont l'origine est statistique. Plus récemment, des travaux ont étudiés les arbres de décision flous [Ramdani 94] [Marsala 98], qui permettent de traiter des connaissances imprécises. D'autres ont étudiés les arbres de décision basés sur la croyance [Elouedi et al. 00] [Mellouli et al. 01], qui traitent des incertitudes.

Un arbre de décision est constitué de trois types d'éléments : les nœuds, les arcs et les feuilles. Chaque nœud est associé à un attribut et représente un test (par exemple taille=grand). Chaque arc issu de ce nœud correspond à l'une des valeurs de cet attribut, les arcs sont étiquetés par les différents résultats des tests. Enfin, les feuilles sont des nœuds où un ensemble d'exemples vérifie un critère d'arrêt, par exemple, tous les exemples du nœud possèdent la même valeur pour la classe. Pour illustrer ce processus, considérons l'exemple donné par Quinlan [Quinlan 83]. Chaque individu est décrit par 3 attributs :

- *taille* dont les valeurs possibles sont {petit, grand}
- *cheveux* avec les valeurs {noir, roux, blond}
- *yeux* avec les valeurs {bleu, brun}

Attributs			Classe
Taille	*Cheveux*	*Yeux*	
petit	blond	bleu	+
grand	blond	brun	-
grand	roux	bleu	+
petit	noir	bleu	-
grand	noir	bleu	-
grand	blond	bleu	+
grand	noir	brun	-
petit	blond	brun	-

TAB. 2.1 - *Un exemple de base d'apprentissage*.

Les exemples sont libellés par deux classes notées + et -, la base d'apprentissage est décrite par le tableau 2.1. Le choix de l'attribut *cheveux* pour former la racine de l'arbre, puis de l'attribut *yeux* conduit à l'arbre illustré par la figure ci-dessus (**FIG. 2.10**).

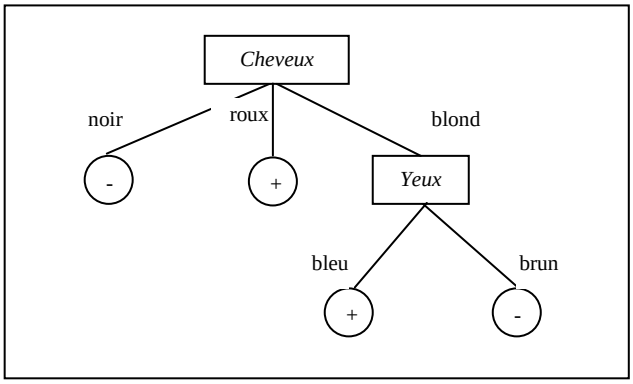


FIG. 2.10 - *Un exemple simple d'arbre de décision*.

La construction d'un arbre de décision se fait, en général, de manière descendante, de la racine vers les feuilles, et est qualifiée dans la littérature anglaise par le terme TDIDT (Top Down Induction of Decision Trees). Les algorithmes de construction d'arbres de décision procèdent par divisions successives de la base d'apprentissage, l'algorithme générique formalisé dans [Marsala 98] est le suivant :

1. Choisir le « meilleur » attribut X_k, de modalités $\{v_{k,1},\ldots, v_{k,i},\ldots, v_{k,mk}\}$, au sens d'un critère donné, pour partitionner la base d'apprentissage. Un nœud $\eta(X_k)$ est créé dans l'arbre, pourtant un test sur la valeur de X_k.

2. Partitionner la base d'apprentissage courante (au départ de l'algorithme, c'est la base complète Ω) avec cet attribut en créant autant de sous-bases Ω_1 que l'attribut possède $\Omega = \bigcup_{i=1,\ldots,mk} \Omega i$ avec $\Omega i = \{\psi i \in \Omega \,/\, X_k(\psi i) = v_{k,1}\}$.

   Chaque valeur de l'attribut libelle un arc issu de ce nœud $\eta(X_k)$ et associé à la sous- base Ω_1.

3. A l'aide d'un critère d'arrêt, vérifier la condition d'arrêt sur chaque sous-base Ω_1 créée. Une sous-base qui vérifie la condition d'arrêt donne naissance à une feuille dans l'arbre.

4. Recommencer en 1 avec les sous-bases qui ne vérifient pas le critère d'arrêt.

Les paramètres qui conditionnent la structure et le fonctionnement des différents algorithmes sont relatifs aux solutions adoptées pour choisir un attribut à l'étape1, pour partitionner la base à l'étape 2 et pour mesurer le critère d'arrêt à l'étape 3.

Le choix du « meilleur » attribut se fait à l'aide d'une mesure de discrimination qui rend compte du pouvoir discriminant d'un attribut relativement aux classes pour un ensemble d'exemples donné, et qui permet de choisir l'attribut qui réduit au maximum l'incertitude de prédiction des classes. De nombreux travaux se sont intéressés aux mesures de discrimination. Les mesures généralement utilisées dans la construction des arbres de décisions proviennent de la théorie de l'information ou sont issues des statistiques. Dans le cas des premières, nous pouvons citer le gain d'information (ou information mutuelle) issu de la mesure d'entropie de Shannon et cité par Picard [Picard 72] pour ordonner des questions selon leur pouvoir discriminant et construire un questionnaire. Le gain d'information a été utilisé par Quinlan [Quinlan 83], qui a également proposé le ratio de gain dans C4.5 pour pallier certains défauts du gain d'information [Quinlan 86a] [Quinlan 93]. On peut citer aussi, une mesure de discrimination sensible aux effectifs introduite par Zighed dans la méthode SIPINA [Zighed et al. 92]. A titre d'exemple de mesures de discrimination issues des statistiques, citons le critère de Gini utilisé dans le système CART et qui mesure l'impureté d'un attribut au regard d'une classe [Breiman et al. 84][9], ou encore un critère basé sur la distance de Kolmogorov-Smirnov utilisé dans [Celeux et al. 82]. Les propriétés générales d'une mesure de discrimination, utiles pour valider et comparer la qualité de ces mesures, sont étudiées dans [Zighed et al. 92]. Une étude détaillée de différentes mesures de discrimination est faite dans [Rakotomalala 97]. Christophe Marsala propose une modélisation des mesures de discrimination par une approche constructive.

Le second paramètre caractérisant un algorithme de construction d'arbres est la stratégie de partitionnement. Elle consiste à décomposer la base d'apprentissage en sous-bases, chacune induite par une modalité de la liste des modalités de l'attribut traité.

[9] Un nœud d'un arbre de décision est dit pur si le sous-intervalle qui lui est associé ne contient que des exemples d'une même classe. Inversement un sous-intervalle est d'impureté maximum quand les C classes sont équiprobables dans ce sous-intervalle.

Dans le cas d'attributs numériques, une étape de discrétisation est nécessaire. Les méthodes usuelles de prise en compte de ce type d'attributs discrétisent leurs univers de valeurs en calculant des seuils qui serviront dans les tests des nœuds de l'arbre de décision [Catlett 91].

Le troisième paramètre qui distingue les différents algorithmes de construction d'arbres de décision est le critère d'arrêt. Ce critère est souvent lié à la mesure de discrimination utilisée pour chercher le meilleur attribut. Si tous les exemples de la sous-base courante possèdent la même classe, la construction de l'arbre peut s'arrêter, comme dans ID3. mais, on peut aussi choisir d'arrêter la construction en fonction d'autres critères comme la taille de la base, quand par exemple le nombre d'exemples est trop faible pour continuer à partitionner.

Un arbre de décision représente une connaissance structurée, intelligible lorsque la taille de l'arbre est raisonnable. Divers méthodes d'élagage permettent de simplifier un arbre déjà construit [Breiman et al. 84] [Rakotomalala 97].

Le classement d'un nouvel individu est réalisé en le présentant séquentiellement aux nœuds de l'arbre. Il est d'abord présenté à la racine où un test est réalisé sur la valeur qu'il possède pour l'attribut libellant ce nœud. Selon le résultat du test, l'individu suit l'arc libellé par ce résultat pour atteindre soit un nouveau nœud où le processus de comparaison est réitéré, soit une feuille. Dans ce dernier cas, la classification est terminée : le nouvel individu est affecté à la classe libellant cette feuille.

Chaque chemin dans un arbre de décision effectue une série de tests sur les valeurs des attributs pour déduire la classe à affecter aux valeurs testées. Un chemin est donc équivalent à une règle de production *Si [prémisse] Alors [conclusion]*. La prémisse de cette règle est une conjonction des tests réalisés sur le parcours en question, et la partie conclusion contient la classe libellant la feuille du chemin, atteinte en fin de parcours. Ainsi, un arbre de décision correspond à une base de règles de classification, et toute nouvelle observation peur être classée par l'usage de la méthode d'inférence classique de raisonnement par déduction. Un arbre de décision peut également être vu comme une table de décision.

2.3.3.2 Les méthodes polythétiques de génération des règles

Les méthodes *mono-attributs* ou *monothétiques*, ne permettent pas de prendre en compte simplement l'éventuel pouvoir prédictif sur l'ensemble des exemples d'une conjonction d'attributs pris simultanément.

D'autres méthodes, qualifiés de *multi-attributs* ou *polythétiques* [DiPalma et al. 97] présentent l'avantage de sélectionner en bloc les attributs qui apparaissent dans les prémisses des règles. Parmi ces méthodes recensées dans [Rabaséda 96] et [DiPalma et al. 97], nous pouvons citer :

- des travaux récents sur les arbres de décision, dont le principe est de sélectionner l'attribut courant en tenant compte de deux ou trois sélections futures. Cela permet de remettre en cause à chaque stade le choix réalisé à l'étape précédente [d'Alché-Buc 93], mais cette méthode demeure très coûteuse en complexité.

- certaines méthodes d'apprentissage supervisé par algorithmes génétiques comme la méthode SIA [Venturini 96], dont le principe est de généraliser des expressions (propositions logiques) couvrant un exemple. A partir d'un exemple de la base, l'algorithme chercha une règle qui le généralise en utilisant les opérations de mutation et de croisement qui peuvent respectivement modifier ou échanger des conditions de la règle initiale couvrant cet exemple.

Le lecteur intéressé par d'autres méthodes *polythétiques* peut consulter les articles suivants [Mitchell 82] [Ganascia 87] et [Maddouri et al. 97].

2.3.3.3 Les Systèmes d'Inférence Floue ou SIF

Les systèmes à base de règles floues ont été appliqués à des multiples domaines, les règles étaient généralement construites à l'aide d'experts humains. Les travaux portant sur la génération des règles floues à partir de données numériques ont commencé dans le domaine du contrôle flou, et du traitement du signal.

La construction des règles floues à partir de données numériques consiste en deux phases : une partition floue de l'espace des entrées puis la construction d'une règle floue pour chaque espace flou issu de cette partition. Précisons que l'espace des entrées correspond à l'espace de représentation.

Le lecteur intéressé par les SIF peut consulter les articles suivants [Ishibuchi et al. 92] [Wang et al. 92].

2.4 Méthode proposée pour la conception d'un système de classification des images

Après avoir décrit quelques principes généraux de l'apprentissage supervisé et avoir présenté un certain nombre de méthodes de classification supervisée, nous résumons dans cette partie les grandes lignes de la méthode que nous proposons pour concevoir et développer un système de classification supervisée. La méthodologie générale de notre modélisation se base d'une part sur la génération semi-automatique de règles de classification à partir des concepts de Peano Count Tree (P-tree) [Qiang Ding et al. 02] et d'autre part sur une utilisation originale des arbres de décision.

Dans notre cadre, l'apprentissage consiste à générer des règles qui formalisent les connaissances extraites à partir des données étiquetées par un expert : cette expertise se limite à associer une classe aux observations.

Notre démarche sera illustrée sur des données réelles dans le cadre de traitement d'images. Il s'agit plus précisément une application sur la classification des images. Dans ce cadre chaque image sera représenter par un vecteur d'attributs qui sont des informations significatives sur l'image à classer. Contrairement à d'autres méthodes qui assurent la représentation et la classification des pixels à partir des valeurs relatives à chaque point sans être nécessairement liées à la réalité qu'elles représentent, sans comprendre leur signification. Par exemple, la méthode de détection de contours (traitement de bas niveau) ne correspond pas forcément à la frontière d'un objet dans la réalité. L'utilisation des mots du vocabulaire de l'application (traitement de haut niveau) permet de mieux exprimer la compréhension et l'interprétation de l'image. Par

exemple des régions d'une image aérienne peuvent être étiquetés avec les termes routes, bâtiments, … Les traitements de bas niveau sont généralement de nature numérique et ceux de haut niveau plutôt symbolique [Coquerez et al. 95].

Les principales caractéristiques de la méthode développée dans le cadre de ce mémoire sont les suivantes :

- La méthode développée est *mono-attribut* ou *monothétique*, cela permet de prendre en compte d'un fort pouvoir explicatif. En effet, cette méthode consiste à décomposer la base d'apprentissage en sous-bases, chacune induite par une modalité de la liste des modalités de l'attribut traité.

- La construction des prémisses des règles se fait étape par étape. A chacune d'elles, on ajoute une condition sur un meilleur attribut, et le choix du meilleur attribut se fait selon le gain d'information. L'élu attribut est celui possédant le gain le plus important.

- Les règles générées sont des règles de classification : leurs conclusions sont de la forme appartenance à une classe. Un chemin dans un arbre de décision est équivalent à une règle de classification. La prémisse de cette règle est une conjonction des tests réalisés sur le parcours en question, et la partie conclusion contient la classe libellant la feuille du chemin, atteinte en fin de parcours.

- Afin de réduire la masse d'information pour ne pas surcharger l'espace mémoire et pour éviter des temps et complexité de traitements trop élevés, nous utilisons la structure P-tree pour comprimer les données originales et par conséquence faciliter et accélérer l'exploitation efficace de ces données.

2.5 Conclusion

Notre méthode se situe à la jonction des méthodes de classification *mono-attribut* et de la structure P-tree. Elle permet de combiner les avantages des classifieurs basés sur les règles de classification et la structure P-tree. La structure P-tree permet d'accélérer le processus de classification. En effet, les données d'origine sont converties sous format binaire ce qui permet de réduire le temps de classification. Nous proposons dans le chapitre suivant un résumé de l'algèbre de P-tree.

Chapitre 3

L'algèbre P-tree

3.1 Introduction

L'extraction des règles de classification à partir des données non structurées telles que les données images est une tâche importante et complexe (classer des images de l'environnement en deux groupes: environnement pollué et environnement dépollué, en médecine regrouper des images des poumons en deux classes: poumon contaminé par le concert ou non, etc.). Cependant, dans la plupart des cas, les tailles des données des images sont trop grandes pour être extraites dans un temps raisonnable en utilisant des méthodes standards. Autrement dit, les algorithmes de classification sont assez longs pour regrouper des images.

Une nouvelle organisation spatiale de données appelée bit Sequential organisation (bSQ)[10] et une nouvelle structure de données appelée Peano Count Tree (P-tree) sont proposées par [Perrizo et al. 02]. L'idée principale de P-tree est de diviser récursivement les données en quadrants et enregistrer le nombre de bit à 1 (1-bit) de chaque quadrant dans un arbre. En utilisant la structure P-tree, on peut accélérer le processus de calcul 1-bit. Ceci facilite de manière efficace la représentation et l'exploitation de données de type image [Perrizo et al. 02].

Dans ce chapitre, nous récapitulons les idées fondamentales de la structure de données P-tree et sa variation. Ensuite, nous décrivons brièvement l'algèbre de P-tree.

[10] bSQ est une format d'organisation de données proposée par [Qin Ding et al. 02]. Par rapport aux autres formats tels que Band Interleaved by Line (BIL), Band Interleaved by Pixel (BIP) et Band Sequential organization (BSQ), la format bSQ présente les avantages suivants : Premièrement, le représentation des données avec les bits apporte différentes contributions à la valeur. Deuxièmement, le format bSQ facilite la représentation d'une hiérarchie de précision (d'un bit jusqu' à la précision de huit bits). Enfin, le format bSQ donne une meilleure compression.

Les propriétés sont également abordées dans ce chapitre. Enfin nous présentons brièvement des mesures et des résultats expérimentaux.

3.2 Structure de P-trees

Les P-trees sont semblables dans la construction par rapport à d'autres structures de données, par exemple Quadtrees [Agrawal et al. 92] [Breiman 84].

3.2.1 Structure de base

Chaque fichier bSQ (B_{ij}) est organisé dans une structure arborescente appelé P-tree. Un P-tree est un arbre basé sur des quadrants. La racine de l'arbre contient le nombre de bits à 1 (1-bit) de la bande entière. Le deuxième niveau contient les 1-bit pour le quatre quadrants dans l'ordre de trame. La construction se continue récursivement en bas de chaque chemin d'arbre jusqu'à ce que le sous-quadrant soit pur (entièrement 1-bit ou entièrement 0-bits). Voici un exemple de P-tree de 8 lignes et de 8 colonnes (cf. figure 3.1) [Qin Ding et al. 02].

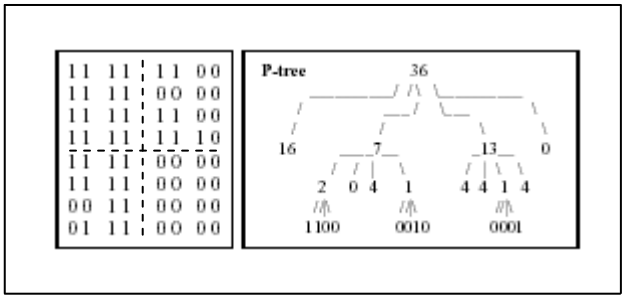


FIG. 3.1 - *Un P-tree pour un fichier (une bande) bSQ 8x8.*

Dans cet exemple, 36 est le nombre de bits à 1 dans ce fichier appelé compte de racine (1-bit). Le niveau de la racine est le niveau 0. Les nombres 16, 7, 13 et 0 du niveau suivant (niveau 1) sont les comptes à 1-bit pour les quatre quadrants principaux. Généralement, un P-tree possède les caractéristiques suivantes :

- un P-tree contient 1-count pour chaque quadrant de chaque dimension.
- un P-tree, pour n'importe quel sous-quadrant à n'importe quel niveau, est simplement le sous-arbre enraciné à ce sous-quadrant.
- une séquence de feuille d'un P-tree (profondeur-premier) est une version comprimée par longueur partielle de la bande originale.
- un P-tree de base peut être combiné pour reproduire les données originales.
- des P-trees peuvent être partiellement combinés pour produire les limites supérieures et inférieures sur tous les comptes de quadrant.
- des P-trees peuvent être employés pour lisser des données par la purification ascendante du quadrant (remplacement ascendant des comptes mélangés avec leurs comptes purs plus étroits).

La structure P-tree peut être générée rapidement et vue comme un format simple et facile pour l'enregistrement des données.

3.2.2 Variations de P-tree

La variation de la structure P-tree appelé aussi Peano Mask Tree (PM-tree ou bien PMT), est une structure semblable dans laquelle des masques plutôt que des comptes sont employés. Dans un PM-tree, on utilise trois valeurs logiques pour représenter des quadrants pur-1, pur-0 et mixtes. Voici un exemple de PM-tree de 8x8 (cf. figure 3.2).

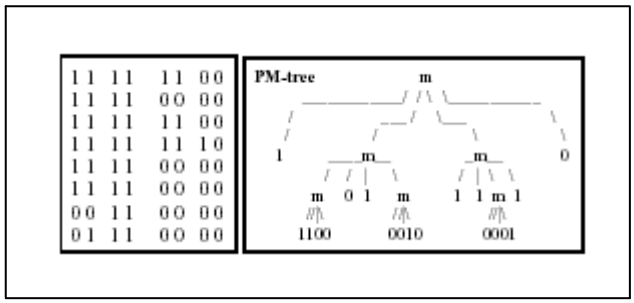


FIG. 3.2 - *Un PM-tree.*

On peut utiliser d'autres variations comme par exemple P1-tree et P0-tree. Dans P1-tree, on utilise 1 pour indiquer le quadrant pur-1 tandis qu'on utilise 0 pour indiquer des quadrants mixtes. Dans P0-tree, on utilise 1 pour indiquer le quadrant pur-0 tandis qu'on utilise 0 pour indiquer des quadrants mixtes. Les P1-tree et P0-tree de l'exemple précédent sont représentés dans la figure 3.3.

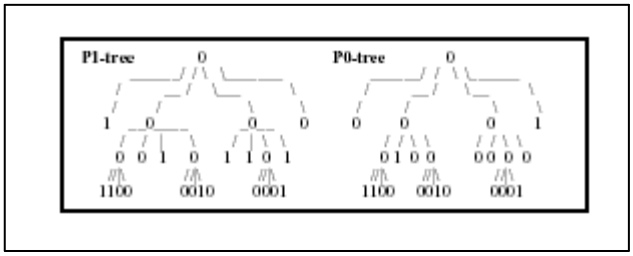


FIG. 3.3 - *P1-tree et P0-tree.*

On peut également utiliser non-pure-0-tree (NP0-tree) et non-pure-1-tree (NP1-tree). Dans non-pure-0-tree (NP0-tree), on utilise 1 pour indiquer le quadrant non-pur-0 qui couvre le quadrant pur-1 et le quadrant mixte tandis qu'on utilise 0 pour indiquer le quadrant pur-0. NP1-tree peut être défini d'une manière semblable.

3.3 L'algèbre P-tree

L'algèbre P-tree inclut trois opérations de base : Complement, AND et OR.

3.3.1 Opération de complémentation

Chaque structure de base d'un P-tree possède un complément noté **P-tree'**. Le complément peut être construit directement en complétant simplement les comptes de chaque niveau (soustrayant du compte pur-1 à ce niveau), comme montré dans l'exemple ci-dessous. Il est à noter que le complément d'un P-tree fournit le compte 0-bit pour chaque quadrant (cf. figure 3.4) [Qin Ding et al. 02].

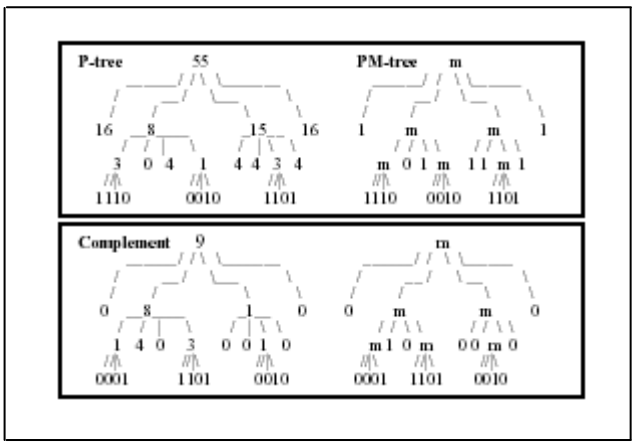


FIG. 3.4 - *Complément d'un P-tree.*

3.3.2 Opération d'intersection (AND)

C'est la plus importante et fréquente opération d'un P-tree. Le tableau 3.1 illustre les règles à respecter pour déterminer l'intersection entre deux P-trees ou sous-P-trees (opérande1 et opérande 2) avec racine X_1 et X_2 respectivement.

Opérande 1	Opérande 2	Résultat
1	X_2	Sous arbre avec la racine X_2
0	X_2	0
X_1	1	Sous arbre avec la racine X_1
X_1	0	0
m	m	**Si** les quatre quadrants ont comme conséquence 0 **alors** 0 ; **Sinon** m

TAB. 3.1 - Les règles de P-tree ANDing.

La figure 3.5 illustre l'opération AND appliquée sur deux arbres P-tree-1 et P-tree-2.

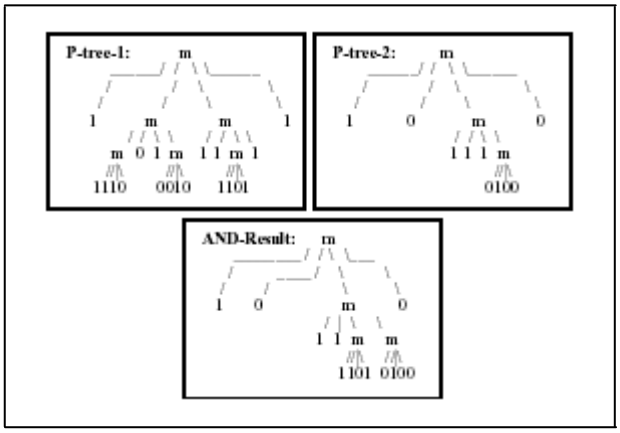


FIG. 3.5 - *L'intersection de deux P-trees.*

3.3.3 Opération d'union (OR)

L'opération OR peut être implémentée d'une manière semblable à l'opération AND. Le tableau 3.2 illustre les principales règles à respecter pour faire l'union de deux arbres P-tree-1 et P-tree-2 (opérande1 et opérande 2) avec racine X_1 et X_2 respectivement (cf. figure 3.6) [Qin Ding et al. 02].

Opérande 1	Opérande 2	Résultat
0	X_2	Sous arbre avec la racine X_2
1	X_2	1
X_1	0	Sous arbre avec la racine X_1
X_1	1	1
m	m	1

TAB. 3.2 - *Les règles de P-tree ORing.*

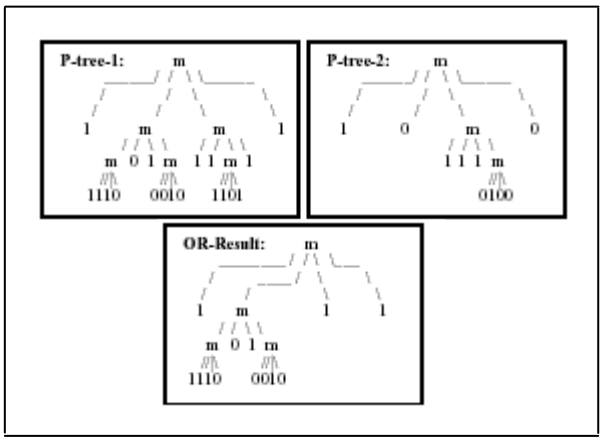


FIG. 3.6 - *L'union de deux P-trees.*

L'opération XOR est notée ⊕ (le ou exclusive) est dérivée à partir du groupe complet (Complement, AND et OR).

3.3.4 Valeur et Tuple de P-trees

Définition 1. *Une valeur de P-tree P_i (v) est un P-tree de valeur v à la bande i.*

La valeur de P-trees peut être construite à partir de P-trees ANDing ou leurs compléments. Par exemple, la valeur de P-tree P_i (110) donne le compte de la bande i bit 1 égal à 1, bit 2 égal à 1 et bit 3 égal à 0. On peut construire P_i (110) à partir de P-trees de base :

$$P_i\,(110) = P_{i,1}\ \textbf{AND}\ P_{i,2}\ \textbf{AND}\ P_{i,3}'$$

Les P-trees peuvent également représenter des données pour n'importe quelle combinaison de valeur de n'importe quelle bande appelée tuple. De la même manière nous pouvons construire un tuple de P-trees.

Définition 2. *Un tuple de P-tree P (v_1, v_2, …, v_n) est un P-tree de valeur v_i à la bande i, pour tout i de 1 à n.*

$$P(v_1, v_2, \ldots, v_n) = P_1(v_1) \textbf{ AND } P_2(v_2) \textbf{ AND } \ldots \textbf{ AND } P_n(v_n)$$

Si la valeur v_j n'est pas donnée cela veut dire qu'il pourrait être n'importe quelle valeur dans la bande j. Par exemple, P (110, , 101, 001, , , ,) représente un tuple P-tree de la valeur 110 dans la bande 1, 101 dans la bande 3, 001 dans la bande 4 et toute valeur dans toute autre bande.

3.4 Propriétés de P-trees

Dans cette section on va citer quelques propriétés de P-trees. On utilise les notations suivantes dans les corollaires qui suivent.

Soient P_1 et P_2 deux P-trees.

- P_1 and P_2 , P_1 & P_2 dénote P_1 AND P_2
- $P_1 \setminus P_2$ dénote P_1 OR P_2
- $P_1 \oplus P_2$ dénote P_1 XOR P_2
- P' dénote le complément de P
- rc (P) dénote le nombre de 1 de P-tree P.
- P^0 est un pure-0-tree
- P^1 est un pure-1-tree
- N est le nombre de bits à 1 (1-bit)

Corollaire 1. *Soient deux P-trees P_1 et P_2, rc ($P_1 \setminus P_2$) = 0 ssi rc (P_1) = 0 et rc (P_2) = 0.*

<u>Preuve</u> *: (Preuve par absurde) Soit rc (P_1) $\neq 0$ alors il y a nécessairement des 1 dans l'arbre P_1 et donc rc ($P_1 \setminus P_2$) $\neq 0$,* mais on a supposé *rc ($P_1 \setminus P_2$) = 0.* Par conséquent *rc (P_1) = 0.* De même nous pouvons prouver que *rc (p2) = 0.*
La preuve pour l'inverse, rc (P_1) = 0 et rc (P_2) = 0 si et seulement si rc ($P_1 \setminus P_2$) = 0 est insignifiant. Ceci suit immédiatement les définitions.

Corollaire 2. a) *rc (P_1) = 0 ou rc (P_2) = 0 ssi rc (P_1 & P_2) = 0*
b) *rc (P_1) = 0 et rc (P_2) = 0 ssi rc (P_1 & P_2) = 0*
c) *rc (P^0) = 0*
d) *rc (P^1) = N*
e) *P & P^0 = P^0*
f) *P & P^1 = P*
g) *P \ P^0 = P*
h) *P \ P^1 = P^1*
i) *P & P' = P^0*
j) *P \ P' = P^1*

Les preuves sont immédiates.

Corollaire 3. *$v_1 \neq v_2$ ssi rc {P_i (v_1) & P_i (v_2)} = 0 pour toute bande i.*

<u>Preuve</u> *: P_i (v) représente la valeur v dans la bande i. Si $v_1 \neq v_2$ alors on ne peut pas y avoir à la fois les valeurs v_1 et v_2 dans la même bande. Par conséquent, si la*

valeur de $P_i(v_1)$ vaut 1 pour une position (x, y) de la bande i alors on doit y avoir 0 dans $P_i(v_2)$ pour la même position et vice versa. Par conséquent $rc\{P_i(v_1) \& P_i(v_2)\} = 0$.

Corollaire 4. $rc(P_1 \setminus P_2) = rc(P_1) + rc(P_2) - rc(P_1 \& P_2)$.

<u>Preuve</u> : *Soient n_1 le nombre des points à 1 et à 0 respectivement dans P_1 et P_2, n_2 le nombre des points à 0 et à 1 respectivement dans P_1 et P_2 et n_3 le nombre des points à 1 P_1 et P_2.*
On a : $rc(P_1) = n_1 + n_3$, $rc(P_2) = n_2 + n_3$, $rc(P_1 \& P_2) = n_3$
et $rc(P_1 \setminus P_2) = n_1 + n_2 + n_3 = (n_1 + n_3) + (n_2 + n_3) - n_3$
$= rc(P_1) + rc(P_2) - rc(P_1 \& P_2)$.

Théorème. $rc\{P_i(v_1) \setminus P_i(v_2)\} = rc\{P_i(v_1)\} + rc\{P_i(v_2)\}$ *avec* $v_1 \neq v_2$.

<u>Preuve</u> :
$rc\{P_i(v_1) \setminus P_i(v_2)\} = rc\{P_i(v_1)\} + rc\{P_i(v_2)\} - rc\{P_i(v_1) \& P_i(v_2)\}$ **(Corollaire 4)**
Si $v_1 \neq v_2$, $rc\{P_i(v_1) \& P_i(v_2)\} = 0$. **(Corollaire 3)**
Par conséquent, $rc\{P_i(v_1) \setminus P_i(v_2)\} = rc\{P_i(v_1)\} + rc\{P_i(v_2)\}$.

3.5 Utilisation et Performance de P-tree

Quoique relativement récente, la structure de P-tree trouve des applications dans de nombreux domaines. Elle est utilisée dans le prétraitement de données (Data Preprocessing) pour faciliter la compression de données, comme par exemples: les P-trees pour les données de panier de le ménagère, les P-trees pour les données microarray[11] (traitement de l'ADN), les P-trees pour la partition équi-profondeur, etc.

En plus de la structure de données P-tree, il y a la structure Quadtrees qui consiste à décomposer l'univers au moyen des hyperplans orientés. Ces partitions ne doivent pas être de taille égale, bien que ce soit souvent le cas. La décomposition dans des sous-espaces est habituellement continuée jusqu'à ce que le nombre d'objets dans chaque partition soit au-dessous d'un seuil donné. Il y a aussi la structure HH-codes (Helical Hyperspatial Codes). Cette structure est une représentation binaire de diagonale de Riemannian. La division binaire de la diagonale forme un noeud dont huit cubes secondaires sont formés. Chaque cube secondaire a sa propre diagonale. Ces cubes sont constitués par des valeurs unidimensionnelles d'entrelacement codées en tant que codes de bit HH. Une fois assortis, ils groupent dans des classes le long de diagonale. Les classes sont ordonnées dans un modèle hélicoïdal.

Le P-tree, le Quadtree et le HHCode sont basées sur des quadrants. La différence est que la structure de P-tree se concentre sur le compte (1-bit). Les P-trees ne sont pas indexées, plutôt, elles représentent des ensembles de données elles-mêmes. Les P-trees sont particulièrement utiles pour le *Data Mining* parce qu'ils contiennent l'information

[11] La technologie microarray d'ADN est un outil important pour surveiller et analyser des profils d'expression de gène des milliers de gènes simultanément.

agrégée[12] [Qin Ding et al. 02] et que nous souhaitons, pour un gain de temps, par exemple, fusionner les arguments au fur et à mesure que nous les recevons.

La structure de P-tree peut être employée pour l'exploitation efficace de données, la classification et la génération de règle d'association. Avec l'information dans les P-trees, les comptes peuvent être rassemblés d'une manière rapide par ANDing P-trees approprié au lieu de balayer la base de données entière [Qin Ding et al. 02].

En bref, la structure P-tree permet de gagner énormément de temps du calcul des probabilités à l'aide des opérateurs d'agrégation (Complement, AND et OR), pour cette raison nous proposons dans la suite de ce mémoire une classification d'images par arbre de décision basée sur la structure P-tree.

3.6 Conclusion

Dans ce chapitre nous avons présenté la structure P-tree, l'algèbre de P-tree ainsi que les domaines d'applications de la structure P-tree.

Nous allons voir dans la suite de ce travail la proposition que nous formulons pour classer des images à l'aide de la technique de l'arbre de décision et la structure P-tree.

[12] Agrégation d'information « *c'est l'expression du résumé d'un ensembles de données quelconques* ». Les agrégateurs permettant de fusionner des objets pour gérer des poids (les objets peuvent avoir plus ou moins d'importance les uns par rapport aux autres) ou bien le fait qu'il intègre un comportement décisionnel (c'est-à-dire que l'on peut souhaiter obtenir un résultat optimiste ou, au contraire, pessimiste) Herman AKDAG et Isis TRUCK, thèse de doctorat, décembre 2002. http://www.univ-reims.fr/labos/LERI/

Chapitre 4

Méthode de l'apprentissage supervisé appliquée à la classification des images

4.1 Introduction

Dans le chapitre précédent nous avons présenté l'algèbre de P-tree que nous allons essayer de la combiner avec l'algorithme de classification C4.5 proposé par Ross Quinlan en 1993 [Quinlan 93]. Bien que cet algorithme présente une amélioration de l'algorithme ID3, il présente aussi des inconvénients. Nous mentionnons notamment le chevauchement lié au grand nombre de classes qui peut augmenter le nombre des branches dans l'arbre et compliquons par conséquent la compréhension et le temps de génération de l'arbre. Pour éviter ce problème de coût de classification, nous proposons une combinaison de l'algorithme C4.5 avec la structure P-tree. En fait, notre travail est inspiré de celui de Qiang Ding, Qin Ding et William Perrizo [Qiang Ding et al. 02]. Dans leurs travaux de recherche, ils ont appliqué la structure P-tree à l'algorithme ID3 pour classer d'images à partir du satellite. Le résultat obtenu présente un gain important au niveau de la classification. Dans notre travail, nous testons notre méthode sur une base d'images de fraise.

4.2 Présentation générale des données images

Une image est une représentation bidimensionnelle d'objets à trois dimensions de natures diverses (vues aériennes, images médicales, scènes naturelles, …). Elle contient en chaque point l'intensité lumineuse perçue par une caméra, ou tout autre système de mesure, en ce point. Elle rend compte de la variation de contraste de la scène. L'intensité lumineuse varie de façon uniforme entre le blanc (réflexion importante de lumière) et le noir (réflexion nulle de la lumière due à la présence d'un fort contraste). Sous cette forme, l'image est inexploitable par la machine. Il est nécessaire de la numériser. Cette opération se fait en 2 étapes. D'abord un échantillonnage du support de l'image la remplace par une grille de points uniformément répartis dans un système d'axes, ensuite une quantification de l'intensité lumineuse ou de la profondeur est réalisée.

Les images digitalisées ou numérisées sont obtenues suite à une opération de codage. Le codage consiste à transformer un ensemble de données analogiques en un ensemble de données numériques de manière à pouvoir les traiter par ordinateur. Cette transformation doit se faire de la manière la plus fidèle possible, sans perte d'informations pertinentes. Pour plus de détails sur les différents types d'échantillonnage et de quantification, vous pouvez se référer à l'ouvrage [Belaïd et al. 92].

Une image numérisée est considérée comme un ensemble fini d'un espace discret, elle est en général liée à un repère de travail dans lequel on précise la position d'un point par ses coordonnées x et y. Le couple (position, mesure) est appelé *pixel* (figure 4.1). L'information élémentaire associée à chaque point est en général transcrite en niveau de gris ou en couleur.

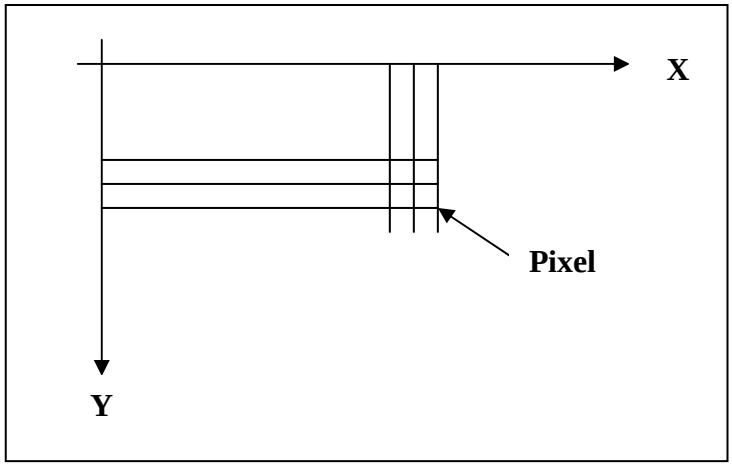


FIG. 4.1 - *Repérage des pixels*.

A partir d'une image numérique, il est intéressant d'*extraire* les informations pertinentes au regard de l'application concernée, de les *traiter* puis de les *interpréter*.

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.

En analyse d'images, on distingue les traitements de *bas niveau* et ceux de *haut niveau*. Cette distinction, comme c'est le cas en Intelligence Artificielle, est liée au contenu sémantique des objets traités des images, ainsi qu'aux aspects numériques et symboliques des représentations. Les traitements de bas niveau concernent en général les grandeurs calculées à partir des valeurs relatives à chaque point de l'image sans être

nécessairement liées à la réalité qu'elles représentent, sans « comprendre » leur signification. Par exemple la détection de contours est un traitement de bas niveau. Un contour détecté ne correspond pas forcément à la frontière d'un objet dans la réalité. Les traitements de haut niveau sont quant à eux relatifs à la compréhension et à l'interprétation de l'image exprimée par des mots du vocabulaire de l'application. Par exemple des régions d'une image aérienne peuvent être étiquetés avec les termes routes, bâtiments, … Les traitements de bas niveau sont généralement de nature numérique et ceux de haut niveau plutôt symbolique [Coquerez et al. 95]. Dans notre travail, nous choissions le traitement de haut niveau.

4.2.1 Sélection des attributs

La classification manuelle se base généralement sur un petit nombre de caractéristiques ou d'attributs. Chacune de ces caractéristiques apporte des informations significatives pour la classification et est sélectionnée suivant la sémantique du problème. De façon évidente, plus les attributs sont discriminants et plus le nombre d'entrées d'un classifieur est faible, plus la conception de ce dernier est facile. Par ailleurs, le coût en temps et en mémoire, d'une classification augmente avec le nombre d'attributs. De plus, il peut être avantageux d'un point réduction du nombre d'attributs intervenant dans la discrimination des exemples. L'utilisation des attributs non représentatifs peut être problématique. Le bruit engendré par ces attributs modifie la dispersion des classes et perturbe ainsi les résultats. Dans [Kanal et al. 72], l'auteur affirme que le pouvoir de classification vient souvent d'un choix réfléchi et judicieux des attributs fondé sur une bonne connaissance du domaine, plutôt que de la conception spécifique de l'algorithme de séparation.

Il est donc intéressant d'extraire ou de sélectionner les attributs importants à partir des exemples. La sélection consiste à déterminer les attributs les plus pertinents pour la tâche de classification, et l'extraction consiste à « construire » de nouveaux attributs par combinaison optimale des attributs initiaux [Rakotomalala 97]. Ces opérations reviennent à trouver une application de l'espace initial de représentation de dimension p vers un espace de dimension inférieur, cette opération doit conserver au mieux la séparabilité des classes selon les distributions des attributs initiaux. Elle permet une réduction de la dimensionnalité.

En général, le nombre d'attributs sélectionnés ou extraits doit être comparable au nombre de classes. Il serait illusoire de penser qu'un grand nombre d'attributs apporte une meilleure discrimination, car l'information qu'ajoute les attributs excédentaires est négligeable devant le bruit qui apportent [Coquerez et al. 95]. Après avoir augmenté en fonction du nombre d'attributs, la performance des classifieurs atteint en général un palier et parfois se met à décroître au dé la d'un certain nombre d'attributs comme cela a été démontré dans [Bonnet 98].

Le problème de la sélection d'attributs, ou réduction de dimension, a engendré de nombreux travaux. Plusieurs méthodes permettent de sélectionner les attributs les plus discriminants ou de déterminer de nouveaux attributs plus synthétiques [Rakotomalala 97]. Ainsi en analyse de données, l'analyse factorielle discriminante permet de rechercher de nouvelles variables qui séparent le mieux possible en projection les classes d'exemples [Saporta 90] [Fukunaga 72]. D'autres méthodes, issues de l'Intelligence Artificielle et de l'apprentissage, permettent de sélectionner un ou plusieurs attributs avec le plus grand pouvoir discriminant. Elles utilisent divers critères

pour estimer la qualité d'un attribut ou d'un ensemble d'attributs. Différentes méthodes de sélection d'attributs sont présentées dans [John et al 94], [Moulinier 97].

Parmi les différents attributs caractérisant une image nous pouvons citer le niveau de gris ou intensité, qui varie en général sur une échelle de 0 à 255, ainsi que les couleurs exprimées en général à travers le système RVB des trois couleurs primaires Rouge, Vert et Bleu, chaque couleur variant entre 0 et 255. Un autre exemple d'attribut est la texture, elle correspond, au niveau linguistique, à des termes tels que rugosité, finesse, régularité, etc. cependant, la richesse du contenu sémantique est liée au niveau de représentation. Au niveau élémentaire, l'élément de base est le pixel dont le contenu sémantique est pauvre. Ce niveau peut toutefois être enrichi par la prise en compte du voisinage d'un pixel (classer un pixel isolé n'a pas beaucoup de sens si l'on ne tient pas compte de son voisinage). Un niveau encore supérieur traduirait des relations entre régions regroupant plusieurs pixels. Une autre solution que nous proposons, consisterait à utiliser des attributs symboliques. Cette solution permet de conserver la sémantique des images à classer.

Dans le cadre de ce travail de recherche, nous ne nous intéressons pas au problème de la sélection des attributs. Nous supposerons que cette opération a été réalisée et que les attributs symboliques dont nous disposons pour représenter les exemples sont pertinents pour la tâche de classification. Nous nous intéressons dans la section suivante à quelques techniques de classification.

4.2.2 La classification

En analyse d'images, la classification s'effectue sur des objets qui peuvent être :

- les pixels qui sont alors locaux, comme le niveau de gris, ou calculés sur un voisinage du pixel, comme la moyenne des intensités ou un indice de texture calculé sur une fenêtre centrée sur le pixel,
- une sous-image de taille fixe, les attributs sont alors calculés pour cette sous-image. Par exemple, une image peut être découpée en fenêtres que l'on désire classifier,
- une région obtenue par une méthode de segmentation et que l'on cherche à classifier. On se situe ici en aval de la segmentation, dans une phase d'interprétation de l'image ou de Reconnaissance des Formes,
- une image, il s'agit alors de classifier plusieurs images différentes selon des attributs.

C'est à cette dernière classification que nous essayons de la développer plus en détail dans la section suivante parce qu'elle s'adapte bien au traitement de haut niveau.

4.3 Méthode proposée

4.3.1 Organisation relationnelle

L'organisation relationnelle (REL) est une organisation de base pour laquelle il y a uniquement un fichier représentant une relation par un ensemble de tuples et chaque tuple est un ensemble de valeurs d'attributs (formé par un ou plusieurs items). L'organisation REL est le point de départ pour les autres organisations de P-tree. Dans

une table REL, la première colonne représente l'attribut clé primaire 'Id' et les autres colonnes représentent les attributs binaires de REL c'est à dire un item pour les données originales (voir figure 4.2 et figure 4.3). En adoptant cette structure, on peut localiser chaque tuple par un accès direct vers la colonne appropriée, si un item cherché existe dans le tuple cherché, il y a '1', sinon il y a '0' dans la table REL.

4.3.2 Conversion de la base de données en table binaire

Dans notre travail, une base de données est convertie en une table binaire dont les colonnes sont des attributs du format REL de P-tree et chaque attribut possède un ensemble de valeurs (items). Dans cette base de données chaque attribut est associé à un ensemble de valeurs. Par exemple, l'attribut classe possède deux items (X et Y) transformés en bits binaires (1 ou 0 pour oui ou non). Donc, les attributs sont caractérisés par des valeurs binaires 0 et 1, représentant la présence d'un item ou son absence (voir figure 4.2). Il s'agit d'un ensemble d'items A = {a_1, a_2, …, a_n} tel que n est le nombre des items dans une base de données.

Id	Classe
1	x
2	y
3	x
4	y
…	…

Id	a_1
1	1
2	0
3	1
4	0
…	…

FIG. 4.2 - *Conversion des données relationnelles en une représentation REL pour les attributs de domaine binaire.*

Pour un attribut avec un domaine non binaire (multi-valeurs), nous associons à chaque valeur un item. Par exemple, pour une meilleure représentation de l'attribut couleur, nous lui affectons les valeurs suivantes {rouge vif, pâle, ambré}. Le résultat des items est défini comme suit : a_1 = rouge vif, a_2 = pâle, a_3 = ambré (voir figure 4.3).

Id	Couleur
1	rouge vif
2	pâle
3	ambré
4	rouge vif
5	pâle
…	…

Id	a_1	a_2	a_3
1	1	0	0
2	0	1	0
3	0	0	1
4	1	0	0
5	0	1	0
…	…	…	

FIG. 4.3 - *Conversion des données relationnelles en une représentation REL pour les attributs de domaine non binaire.*

4.3.3 Stockage de P-trees

D'une manière générale, la génération de P-tree à partir des trames de données se fait d'une manière ascendante et le stockage de chaque arbre se fait dans un vecteur des entiers. La génération de P-tree dépend du nombre de fan-out dans les nœuds internes de l'arbre P-tree et dans le nœud racine. Pour représenter P-tree avec différents fan-out, nous utilisons la notation suivante P-tree (r-i) ; où r = fan-out du nœud racine ; i = fan-out de tous les nœuds internes du niveau 1. Nous utilisons dans notre travail la représentation P-tree (4-4-n) c-à-d nous divisons le nombre de tuples composant la base de données par 4 blocs (les blocs doivent avoir au minimum quatre tuples). Si le nombre de tuples est inférieur à 16, nous complétons par des 0 pour obtenir le format P-tree sur 16 tuples. D'une manière générale, si le nombre de tuples est compris entre 2^{n+1} et 2^n alors le P-tree de base est stocké avec un nombre de tuples égale à 2^{n+1}.

4.3.3.1 Procédure de stockage de P-tree

Dans cette procédure, nous stockons les nœuds racines de P-tree dans un vecteur et les binaires (quadrant) dans un autre vecteur afin de faciliter l'accès aux données des sous quadrants de chaque bande, en vue de les comparer avec les sous quadrants d'autres bandes pour déterminer la probabilité de chaque item.

Soit N_t = nombres de tuples et I désigne le nombre total des items.

Procédure *Stockage_de_P-tree*

```
Pour (bj = 1 ; j <I ; j++)
 racine[j] := rootcount (1 ; bandj) // vecteur qui stocke les racines des P-trees.
     Si (2^n ≤ N_t et N_t < 2^(n+1)) alors
       Pour (i = N_t; i≤2^(n+1); i++)
         bandj[i+1]:= 0;
       FinPour
     FinSi

       Pour (i =1; i≤2^n /4; 2^n /4) // calcul des racines et sous-quadrants.
         rootbandj[i] := rootcount (1 ; bandj); // vecteur racines internes
       FinPour

       Pour (i =1; i≤2^n ; 2^n /4)
         Si (rootbandj[i] <>2^n ou rootbandj[i] <>0) alors
            Bitsbandj[i] := subquadrant_i;//vecteur des bits
         FinSi
       FinPour

FinPour
```

4.3.3.2 Génération des P-trees de base

Après avoir classé les données, la question qui se pose est la suivante : comment produire les P-trees ?

Nous avons n tuples et nous voulons choisir le fan-out de P-tree en tant que 4.

- balayer les données une fois pour chaque 4 tuples. Ajouter un nœud additionnel dans P-tree. Après chaque 4 nœuds produits, ajouter un nœud interne comme nœud parent et ainsi de suite.

- de cette façon, P-tree est produit du bas vers le haut et de gauche vers la droite tout en balayant les données de la base de données convertie.

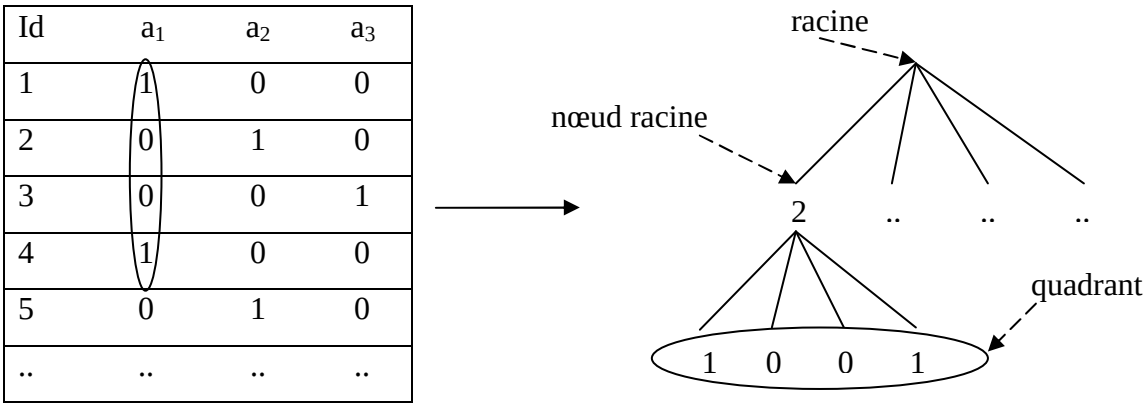


FIG. 4.4 - *Génération de P-tree de base pour l'Item a_1.*

4.3.4 Classification par arbre de décision en utilisant la structure P-tree

Une tâche de classification implique typiquement trois phases, une phase d'apprentissage, une phase de test et une phase d'application. Dans la phase d'apprentissage, les données sont analysées par un algorithme de classification. Chaque tuple dans un ensemble de données est choisi aléatoirement parmi la population témoin. Un attribut d'étiquette de classe est identifié, dont les valeurs sont employées pour marquer les classes. Le modèle ou le classificateur résultant de cette phase d'apprentissage, peut être sous forme de règles de classification ou d'arbre de décision ou des formules mathématiques. Puisque l'étiquette de classe de chaque échantillon de formation est fournie, cette approche est connue comme l'apprentissage supervisée. Dans la phase de test, les essais sont employés pour évaluer l'exactitude du classificateur. Si le classificateur est assez précis, il peut être employé pour prévoir l'étiquette de classe pour un nouveau tuple de données, qui est la phase d'application [Qin Ding et al. 02].

4.3.4.1 L'arbre de décision

Les mesures généralement utilisées dans la construction des arbres de décision proviennent de la théorie de l'information. Nous pouvons citer le gain d'information

issu de la mesure d'entropie de Shannon et cité par Picard [Picard 72]. Le gain d'information a été utilisé par Quinlan [Quinlan 83], qui a également proposé le ratio de gain dans l'algorithme C4.5 pour pallier certains défauts du gain d'information [Quinlan 86a], [Quinlan 93]. Nous décrivons dans ce qui suit les algorithmes d'apprentissage pour la construction des règles de classification sous forme d'arbre de décision ainsi que quelques limites.

4.3.4.2 Algorithmes ID3 et C4.5

L'algorithme ID3 est proposé par Ross Quinlan en 1979 [Quinlan 93]. Le but de ID3 est de construire un arbre optimal en utilisant uniquement les items qui figurent dans les prémisses des règles. L'arbre est construit récursivement noeud par nœud.

Algorithme ID3 ({E})

Si tout e ∈ E a la même classe C, alors
    Retourne la feuille "C"
Sinon
    1. A ← Choix Attribut({E})
    2. EPos ← {e ∈ Ej | A(e) = 1}
    3. ENeg ← {e ∈ Ej | A(e) = 0}
    4. N ← Nouveau Noeud()
    5. N.test ← A
    6. N.pos ← ID3(EPos)
    7. N.neg ← ID3(ENeg)
    8. Retourne N

FinSi

ID3 est un algorithme très efficace et beaucoup utilisé en pratique, mais il présente quelques limites:

- les arbres de classification qui en résultent sont souvent difficiles à interpréter ;
- certaines décisions peuvent être dues au hasard de la sélection des exemples ;
- la classification n'est pas toujours bonne pour de nouveaux exemples.

Une des étapes difficiles de cet algorithme est de choisir le prochain attribut. Si on choisit un attribut qui résulte en une partition dont tous les exemples se retrouvent dans le même sous-ensemble, alors on augmente la taille de l'arbre sans augmenter l'habileté à classer les exemples. En classant les éléments, on veut rejoindre une feuille contenant un ou des exemples en posant le moins de questions; on veut donc l'arbre le plus petit possible. Puisque c'est difficile de trouver l'arbre du chemin le plus court parmi tous les arbres possibles, on construit l'arbre de haut en bas en utilisant une heuristique permettant de choisir l'attribut le plus déterminant (le plus informatif). Il existe plusieurs techniques tels que Index Gini, X^2, ReliefF, théorie de l'information [Shannon 48],…Dans notre travail, nous allons utiliser une heuristique basée sur la théorie de l'information.

Soit une conclusion C pouvant valoir c_1, c_2, …, c_n. Dans notre exemple n = 2. Soit un attribut A pouvant valoir a_1, a_2, …, a_m. La probabilité que C vaut c_i sachant que A vaut a_j est $p(c_i \mid a_j)$. La quantité d'information de a_j sur c_i est notée : $-\log_2 (c_i \mid a_j)$.

L'entropie de a_j par rapport à la conclusion C est notée $E(a_j) = -\sum p(c_i \mid a_j) \log_2(c_i \mid a_j)$. Le gain d'information est calculé comme suit : Gain (A) = E(C) – E (A). Il s'agit de maximiser le gain. L'élu attribut est celui possédant le gain le plus important.

Un deuxième algorithme (C4.5) proposé par Ross Quinlan en 1993. Cet algorithme permet la construction de l'arbre de décision en divisant récursivement l'ensemble d'apprentissage A. On utilise la fonction entropie.

Algorithme C4.5

Si tous les éléments du nœud sont dans la même classe
Alors quitter.
Sinon /*Faire les étapes suivantes*/
1. Choisir un attribut et l'assigner au nœud /* Gain d'information*/
2. Partitionner les exemples dans des sous-ensembles selon les valeurs de cet attribut
3. Créer des nouveaux nœuds pour chaque sous-ensemble non vide de la partition
4. La liste de nouveaux noeuds devient les enfants du nœud.
5. Appliquer la procédure récursivement sur les nouveaux nœuds.

FinSi

Au début, l'arbre de décision est un noeud simple représentant l'ensemble entier d'apprentissage. Si tous les échantillons sont dans la même classe, ce noeud devient une feuille et est marqué avec cette étiquette de classe. Autrement, une mesure basée par entropie " gain de l'information ", est employée comme une heuristique pour sélectionner l'attribut qui sépare mieux les échantillons dans différentes classes (l'attribut de décision). Une branche est créée pour chaque valeur de l'attribut d'essai et des échantillons sont divisés en conséquence. L'algorithme avance périodiquement pour former l'arbre de décision pour le sous-échantillon réglé à chaque nouveau nœud. Une fois un attribut a été employé, il n'est pas considéré dans des noeuds de descendant. L'algorithme s'arrête quand tous les échantillons pour un noeud donné appartiennent à la même classe ou quand il n'y a aucun attribut restant. L'attribut choisi à chaque niveau d'arbre de décision est celui avec le gain le plus élevé de l'information.

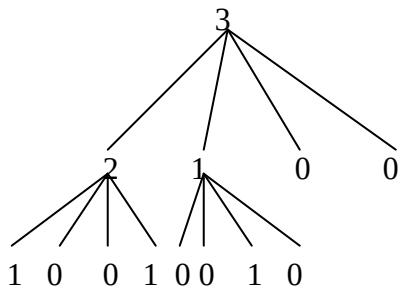
4.3.4.3 Calcul des probabilités et valeur de l'entropie à l'aide de la structure P-tree

Dans cet exemple, nous allons essayer de calculer le gain d'information pour l'attribut couleur. Il est à noter que cet attribut contient trois items (rouge vif, pâle, ambré). Ce travail sera répété pour les autres attributs qui sont : la texture et la forme. Une fois la phase de calcul de la valeur de gain d'information pour chaque attribut est achevée, l'attribut qui présente la plus grande valeur sera retenu comme meilleur attribut de classification (racine de l'arbre) et nous allons partitionner tous les tuples en sous-ensembles (nœuds) selon la valeur du meilleur attribut retenu.

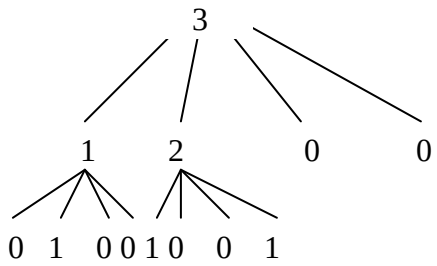
Id	rouge vif = a_1	pâle = a_2	ambré = a_3	Classe X = a_4	Classe Y = a_5
1	1	0	0	1	0
2	0	1	0	0	1
3	0	0	1	0	1
4	1	0	0	1	0
5	0	1	0	1	0
6	0	0	1	0	1
7	1	0	0	1	0
8	0	1	0	0	1
..	..	..	..	..	..

TAB. 4.1 - *Tableau binaire de l'attribut couleur*.

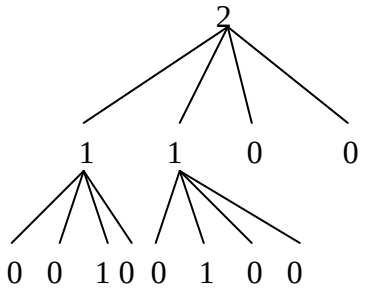
A partir de cette table binaire, nous allons générer des arbres P-trees associés chacun à un item bien défini. Le calcul des probabilités sera déduit automatiquement à partir des ces P-trees.



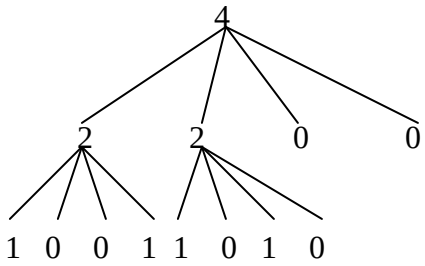
P-tree1 de base pour l'item rouge vif



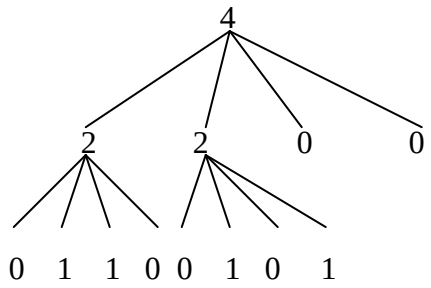
P-tree2 de base pour l'item pâle



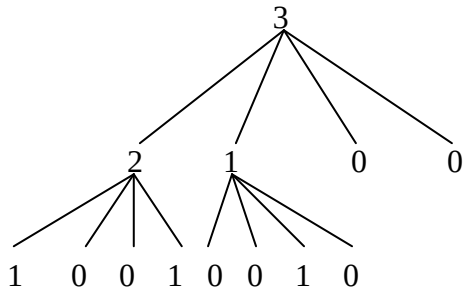
P-tree3 de base pour l'item ambré



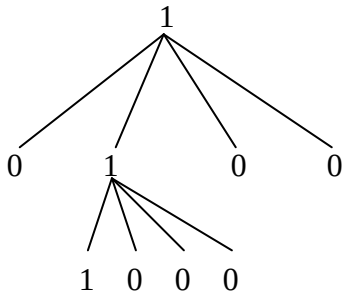
P-tree4 de base pour l'item classe X



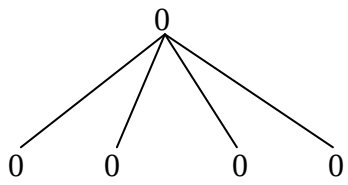
P-tree5 de base pour l'item classe Y



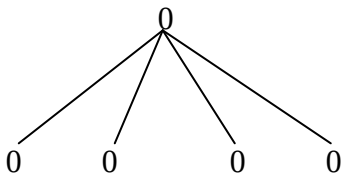
Résultat de P-tree1 ANDing P-tree4



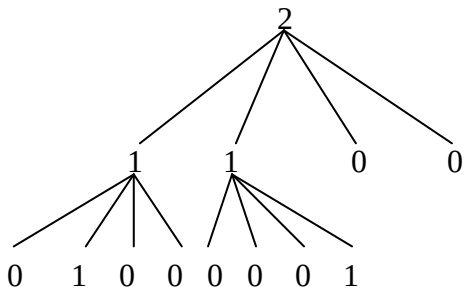
Résultat de P-tree2 ANDing P-tree4



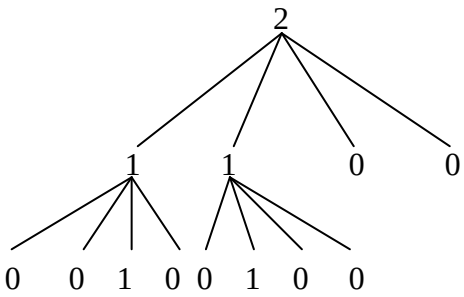
Résultat de P-tree3 ANDing P-tree4



Résultat de P-tree1 ANDing P-tree5



Résultat de P-tree2 ANDing P-tree5



Résultat de P-tree3 ANDing P-tree5

FIG. 4.5 - *Résultats des intersections des P-trees (ANDing).*

Probabilités de classes :

P(X) = 4/8 (P-tree4)
P(Y) = 4/8 (P-tree5)

Entropie de l'attribut classe C(X et Y) :

$E(C) = - (4/8 \log_2(4/8) + 4/8 \log_2 (4/8)) = 0,301$

Probabilités d'apparition :

P(rouge vif) = 3/8 (P-tree1 de base pour l'item rouge vif)
P(pâle) = 3/8 (P-tree2 de base pour l'item pâle)
P(ambré) = 2/8 (P-tree3 de base pour l'item ambré)

Probabilités conditionnelles :

P(X|rouge vif) = 3/4 (P-tree1 ANDing P-tree4)
P(X|pâle) = 1/4 (P-tree2 ANDing P-tree4)
P(X|ambré) = 0 (P-tree3 ANDing P-tree4)
P(Y|rouge vif) = 0 (P-tree1 ANDing P-tree5)
P(Ypâle) = 2/4 (P-tree2 ANDing P-tree5)
P(Y|ambré) = 2/4 (P-tree3 ANDing P-tree5)

Entropie de l'attribut couleur:

E(Couleur) = - [3/8 E(rouge vif) + 3/8 E(pâle) + 2/8 E(ambré)]

$$E(Couleur) = - [3/8 (3/4 \log_2(3/4) + 0) + 3/8 (1/4 \log_2(1/4) + 2/4 \log_2(2/4)) + 2/8 (1/4 \log_2(1/4) + 0)] = 0,185$$

Gain d'information de l'attribut couleur :

Gain (Couleur) = E(C) – E(Couleur)
= 0,301 – 0,185
= 0,116

La classification selon l'attribut couleur sera représentée dans la figure 4.6.

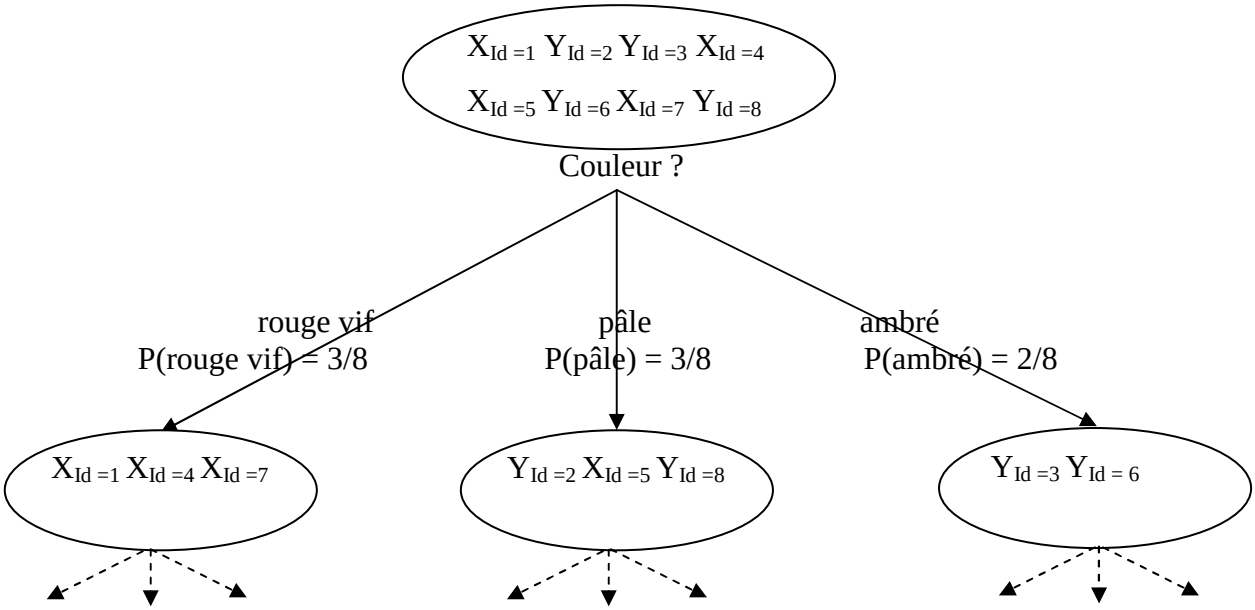


FIG. 4.6 - *Classification suivant l'attribut couleur.*

Ce travail se poursuit en adoptant la même démarche pour les autres attributs (forme et texture) afin de construire les autres branches de l'arbre de décision.

4.3.5 Méthodologie de conception

Dans cette partie, nous détaillons les diagrammes, ainsi qu'une vue de modélisation représentant des activités, des actions, des acteurs, des classes et des composants nécessaires pour le développement de notre application. Le méthodologie de conception que nous utilisons est l'UML (Unified Modeling Language) qui est un langage de modélisation fondé sur les concepts orientés objets ; UML n'est donc pas à l'origine des objets ; néanmoins, il en constitue une étape majeure, dans le sens où il unifie les trois méthodes de modélisation (OOD, OOA, HOOD).

4.3.5.1 Diagramme des cas d'utilisation

Les cas d'utilisation et les acteurs sont décrits ensembles dans un diagramme appelé *diagramme de cas d'utilisation*. Dans notre application, nous avons un acteur appelé *utilisateur* qui est responsable de remplissage de la table REL et un acteur *système* qui est responsable des opérations ANDing entre P-trees ainsi que la classification des tuples. La figure 4.7 représente le diagramme des cas d'utilisation pour notre application. Nous avons trois cas d'utilisation.

- le premier cas d'utilisation est le remplissage des bandes par des valeurs binaires qui est l'équivalent à la création de la base de données.

- le deuxième cas d'utilisation est le calcul des probabilités pour faire le choix du meilleur attribut.

- le troisième cas d'utilisation est la génération des classes.

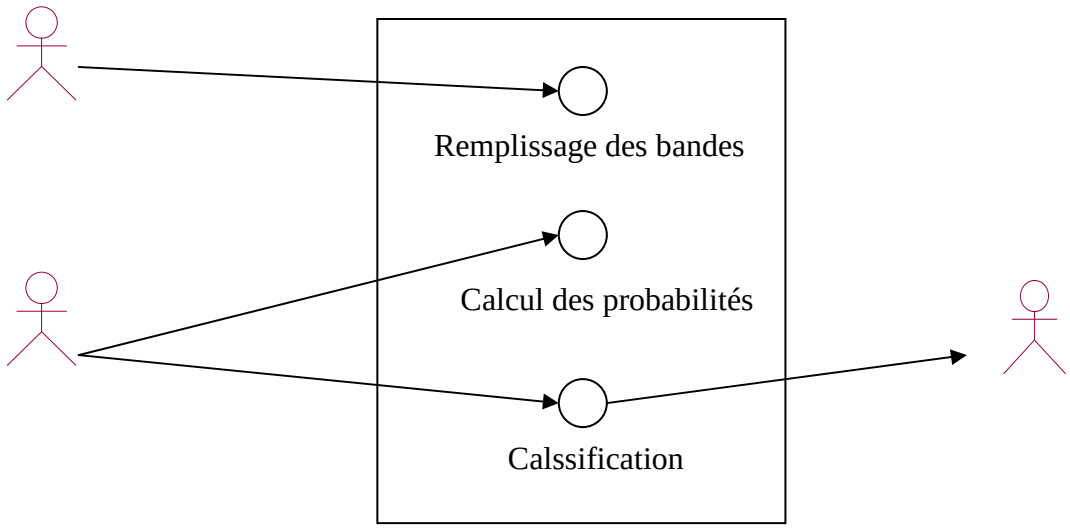


FIG. 4.7 - *Diagramme des cas d'utilisation.*

4.3.5.2 Diagramme de classe

Le diagramme de classe exprime la structure statique du système en terme de relations et de classes. Ce diagramme est spécifié par des attributs, des opérations et des identifiants. La figure 4.8 représente le diagramme de classe.

P-tree : cette classe représente l'état d'appartenance d'un item à tous les tuples d'une base de données. Elle contient un identifiant (*IdP-tree*). Elle est caractérisée par une racine représentant l'apparition du nombre des bits à 1 pour chaque item. Une bande de type liste des textes, ces textes sont des bits 0 et 1. un p-tree appartient à un attribut bien déterminé (*Nomattrib*) et défini par un nom (*Nomitem*). Un p-tree est formé par une liste des racines (*listracines*) et une liste des quadrants (*listquadrant*).

TItemSet : cette classe représente les *itemsets* de la base de données. Chaque *itemset* est relatif à un attribut. Il est caractérisé par un identifiant (*Iditemset*), une probabilité d'apparition, une probabilité conditionnelle, un Gain d'information. Tous ces calculs se font à l'aide de la liste des racines et la liste des quadrants.

Arbre-décison : c'est une classe qui représente l'ensemble des règles de classification. Elle est composée par les attributs suivants :

- idrègle : identifiant de chaque règle.
- ChoixAttribut : c'est un attribut de type liste des strings qui représente le meilleur attribut retenu pour commencer la création de l'arbre de décision.
- Attributcond : c'est un attribut de type liste des strings qui représente la liste des attributs constituant la condition d'une règle.
- Attributconc : c'est un attribut de type liste des strings qui représente la liste des attributs constituant la conclusion d'une règle.

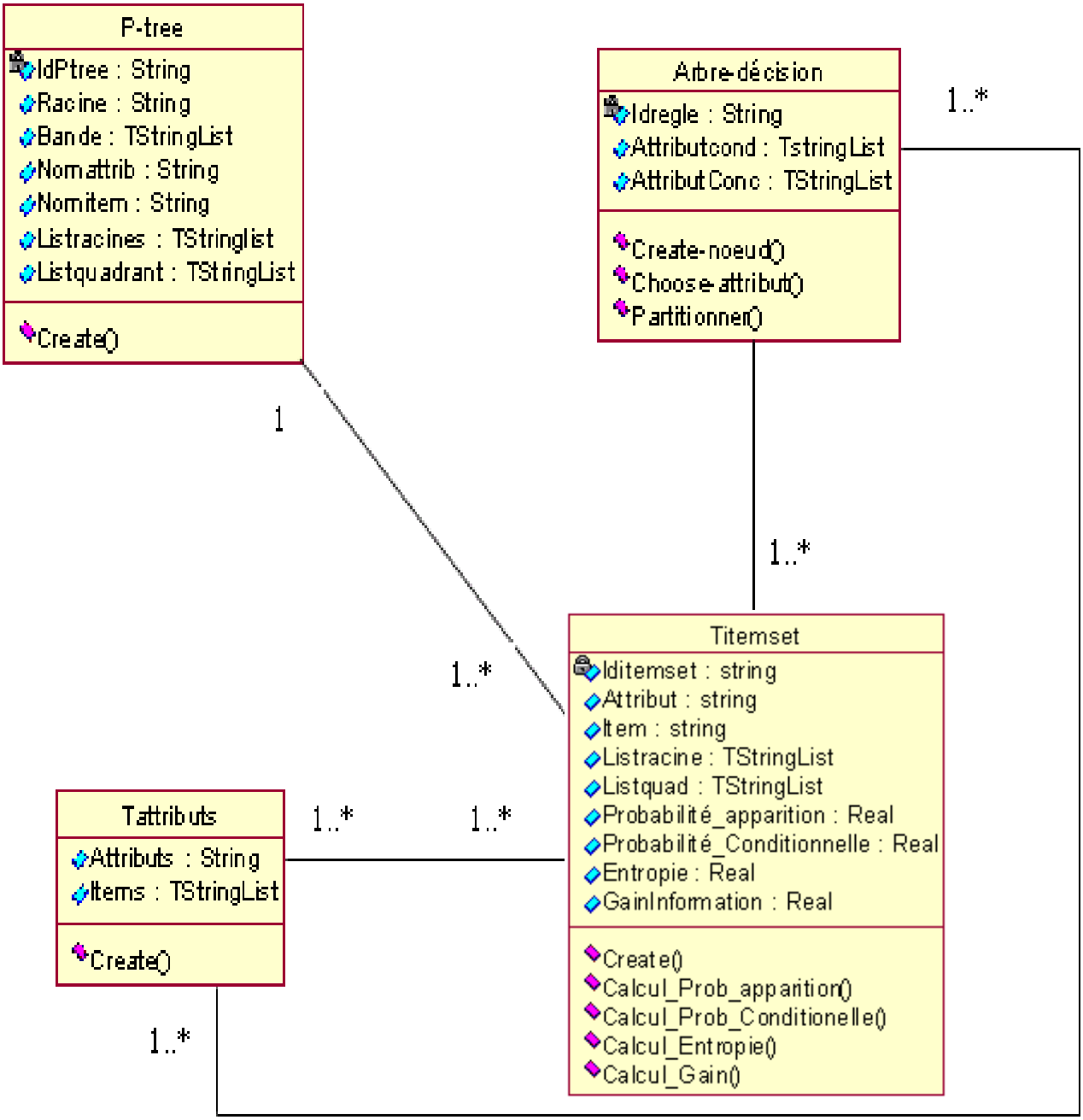


FIG. 4.8 - *Diagramme de classe.*

4.3.5.3 Diagramme de séquence

Un diagramme de séquence est représenté en deux dimensions : l'axe vertical correspondant au temps alors que l'axe horizontal recense les objets qui interagissent dans la séquence en question. Une ligne verticale est attachée à chaque objet de la collaboration et montre la durée de vie de l'objet en question.

Une activation, montre la période durant laquelle un objet exécute une action, soit directement, soit via des procédures.

Une activation est démarrée par un passage entrant représenté graphiquement par une flèche horizontale qui indique la direction du message. Un message est l'unique moyen de communication entre deux objets.

Pour chaque cas d'utilisation un diagramme de séquence est élaboré :

- le premier cas d'utilisation relatif au remplissage des bandes est représenté par la figure 4.9.
- le deuxième cas d'utilisation relatif au calcul des probabilités pour chaque item est représenté par la figure 4.10.
- le troisième cas d'utilisation relatif à la génération des classes est représenté par la figure 4.11.

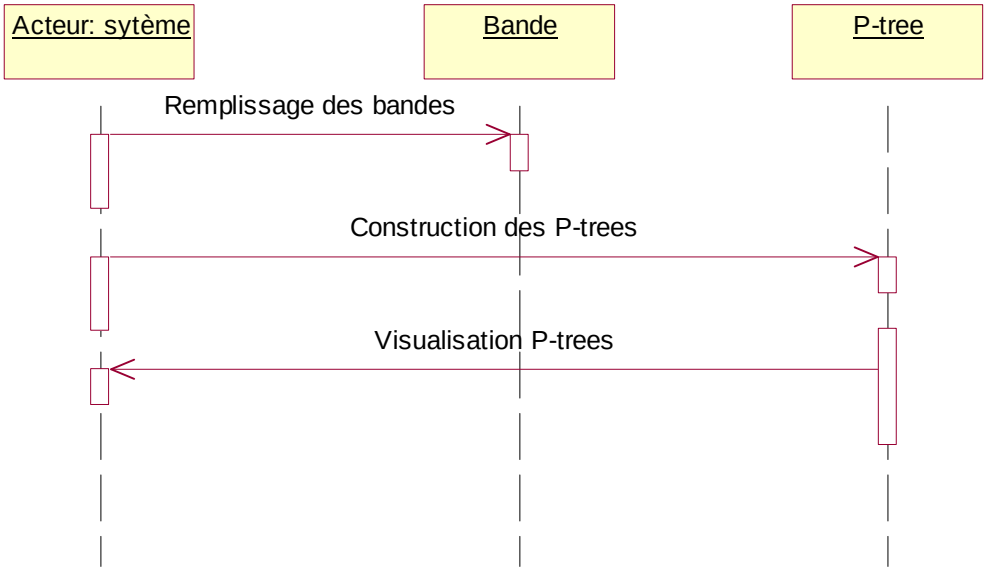


FIG. 4.9 - *Diagramme de séquence relatif au remplissage des bandes.*

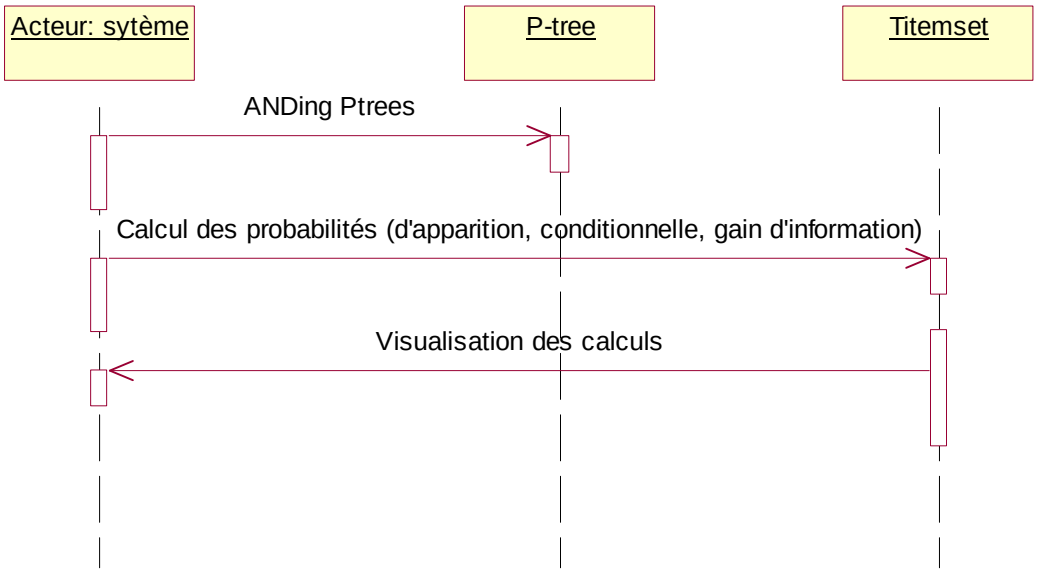


FIG. 4.10 - *Diagramme de séquence relatif aux calculs des probabilités.*

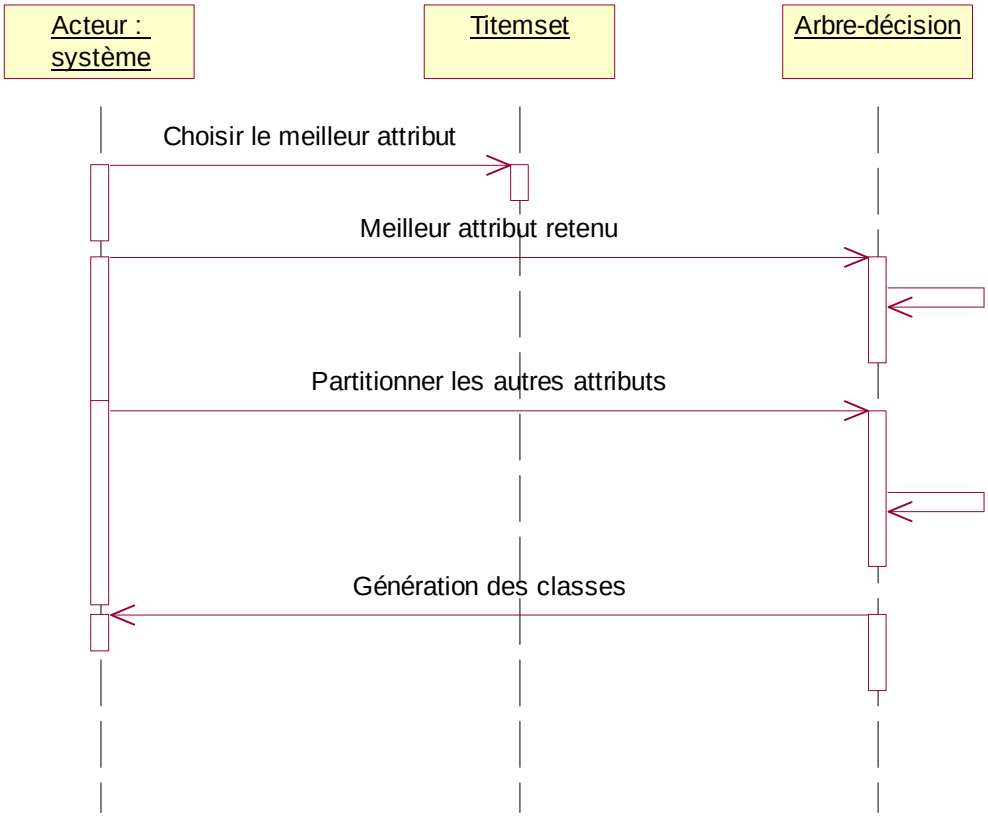


FIG. 4.11 - *Diagramme de séquence relatif à la génération des classes.*

4.4 Conclusion

Dans ce travail nous avons démontré que l'utilisation de la structure P-tree permet d'éviter le balayage de la base des données et le calcul des probabilités se fait directement sur des fichiers binaires à l'aide des bandes présentant des quadrants pure-1 ou pure-0. La synthèse d'information nécessite l'opérateur AND pour l'agrégation, par exemple, de la valeur rouge vif de l'attribut couleur de toutes les images qui ont la valeur 1 afin de pouvoir calculer la probabilité correspondante.

Dans le chapitre qui suit nous allons décrire l'implémentation et la validation de ce travail.

Chapitre 5

Implémentation et validation de la méthode

5.1 Introduction

Dans ce chapitre, nous présentons l'implémentation de notre travail, l'environnement matériel et logiciel utilisé, une comparaison de l'algorithme C4.5-Ptree avec l'algorithme C4.5 ainsi que les interfaces graphiques de notre application.

5.2 Environnement logiciel et matériel utilisé

5.2.1 Environnement logiciel

Nous avons développé une application, que nous avons appelé **C.I.A.D.P-tree**, pour **C**lassification des **I**mages par **A**rbre de **D**écision à l'aide de **P-tree**, comprenant deux modules: le premier "Remplissage des attributs des iamges" permettant de charger les images à partir de la base, de faire la saisie des attributs de chaque imagé entrée, de convertir ces attributs symboliques en valeurs binaires. Le deuxième module "Classification" comporte trois onglets. Le premier onglet appelé "BD Convertie" affiche la base de données binaire. Le deuxième onglet nommé "Structure Ptree" permet de calculer les propabilités en se basant sur la base de données binaire. Le troisième onglet baptisé "Arbre de décision" permet de classer les images en deux groupes (classe1: fraise à gros fruits et classe2: fraise à petits fruits).

Nous avons développé le système **C.I.A.D.P-tree** sous Windows avec le langage de programmation Delphi.

Delphi est un langage de programmation qui intègre des outils d'accès très divers des bases de données quelques soit la nature. Interbase est un SGBD relationnel qui permet la création des bases de données relationnelles très grandes, accessibles depuis Delphi et comprend des techniques robustes pour le stockage et la réconciliation de l'information. Nous présentons dans les parties qui suivent chaque logiciel à part.

5.2.1.1 Présentation de Delphi

Delphi est un environnement de programmation permettant de développer des applications pour Windows 3.x, Windows 98, Windows NT et même linux. Il incarne la suite logique de la famille Turbo Pascal avec ses nombreuses versions. Le Turbo Pascal (DOS et Windows) ne subira apparemment plus de mises à jour. Delphi est un outil moderne, qui fait appel à la programmation objet. De plus, il prend en charge le maintien automatique d'une partie du code source. Delphi permet de créer et de manipuler tout objet de Windows. Tous les objets et une grande partie de l'API de Windows sont encapsulés dans des composants Delphi. Nous avons choisi Delphi car c'est un langage robuste qui surpasse largement d'autres langages de développement comme Visual Basic et présente les caractéristiques suivantes :

- programmation objet ;
- outils visuels bidirectionnels ;
- compilateur produisant du code natif ;
- traitement complet des exceptions ;
- possibilité de créer des exécutables et des DLL ;
- bibliothèque de composants extensibles ;
- débogueur graphique intégré ;
- support de toutes les API de Windows : OLE2, DDE, VBX, OCX,…

5.2.1.2 Présentation de Interbase

Il n'existe pas un seul format de base de données sur le marché. Par exemple Interbase, Oracle, Access et Paradox sont des noms très répandus dans les entreprises. Delphi propose un très vaste choix d'accès aux données, soit en ouvrant des portes vers des systèmes existants comme ADO, soit en proposant une gestion directe de certains serveurs comme Interbase/FireBird, soit en fournissant une solution totalement intégrée comme MyBase. Nous avons choisi Interbase car ceci permet de gérer de très grandes bases de données.

5.2.2 Environnement matériel

Notre application a été testée sur un ordinateur ayant les caractéristiques suivantes :

- micro-porcesseur : Pentium III
- capacité RAM : 128 Mo
- capacité disque : 10 Go
- fréquence d'horloge : 500 Mhz
- Marque : IBM

5.3 Description de la base de données utilisée

5.3.1 Structure de la base d'images

La base de données sur laquelle nous avons travaillé s'intitule base d'images de fruits de fraise. Cette base de données a des caractéristiques relationnelles et contient une table unique représentant des informations sur des fruits de fraise (cf. figure 5.1).

Fraise (à petits fruits ou à gros fruits)

- Couleur — portrait de l'image
- Texture — composition et remplissage de l'image
- Forme — aspect et apparence de l'image

FIG. 5.1 - *Structure de la base d'images utilisée.*

5.3.2 Les attributs de la base d'images

La base de données d'évaluation des fruits de fraise contient des exemples réels représentant des images de fraise avec quatre attributs. Le nombre d'instance de la base d'images pour faire l'apprentissage est 14 images. Le nombre des attributs de la base de données est trois pour les attributs de la partie prémisse et un attribut pour l'attribut de la partie conclusion et dont les valeurs sont définies comme suit :

Valeurs des attributs de décision :
Classe : fraise à petits fruits, fraise à gros fruits.

Valeurs des attributs de prémisse :
Couleur : pâle, rouge vif, ambré.
Texture : lisse, caillé.
Forme : allongé, oblongue (au forme du cœur), conique.

5.4 Comparaison

Il existe des travaux proposés par Qin Ding, William Perrizo et Qiang Ding [Qin Ding et al. 02] qui ont comparé l'algorithme ID3 avec P-tree-classifier. P-tree-classifier est un ID3 qui utilise la structure P-tree.

La comparaison des résultats de classification obtenus avec l'algorithme C4.5 d'une part et l'algorithme C4.5(P-tree) d'autre part montre une amélioration des performances en utilisant la structure P-tree. Le tableau 5.1 reprend les attributs de la base d'images avec une évaluation de deux algorithmes.

Base de données	**Evaluation**	
Attribut de décision	Classe : fraise à petits fruits et fraise à gros fruits	
Attributs de prémisse	Couleur, texture et forme	
Algorithme appliqué	C4.5	C4.5(P-tree)
Taille de la base d'apprentissage	14	14
Temps CPU en seconde	13	9

TAB. 5.1 - *Comparaison entre algorithmes.*

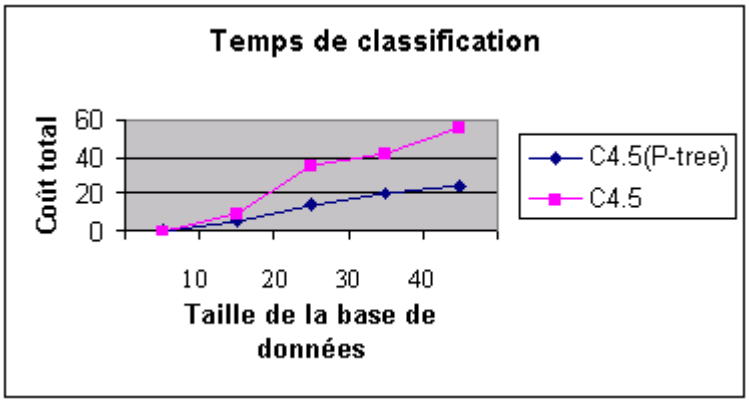


FIG. 5.2 – *Coût de classification.*

Une nette amélioration des performances est à noter. Cette amélioration atteint 30% du temps de classification ($\frac{13-9}{13}$ = 0,30) dans le cas d'une taille de la base de 14 images.

5.5 Les interfaces de l'application

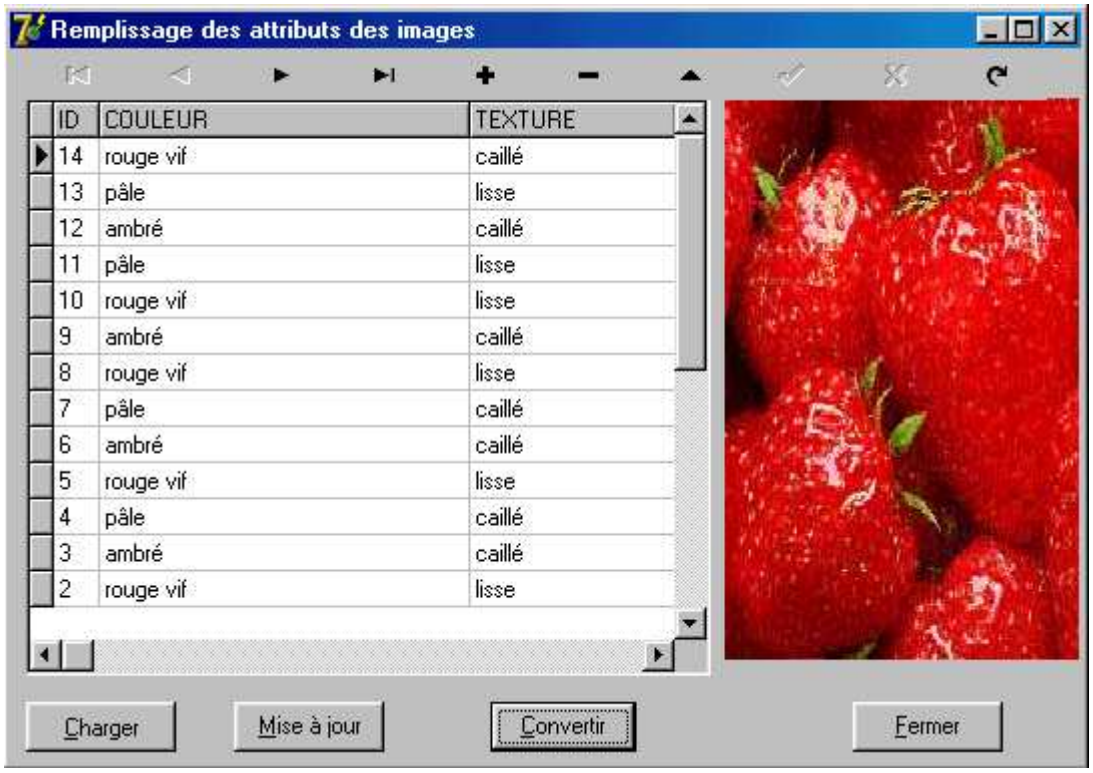


Fig. 5.3 – *Interface de la base de données des images.*

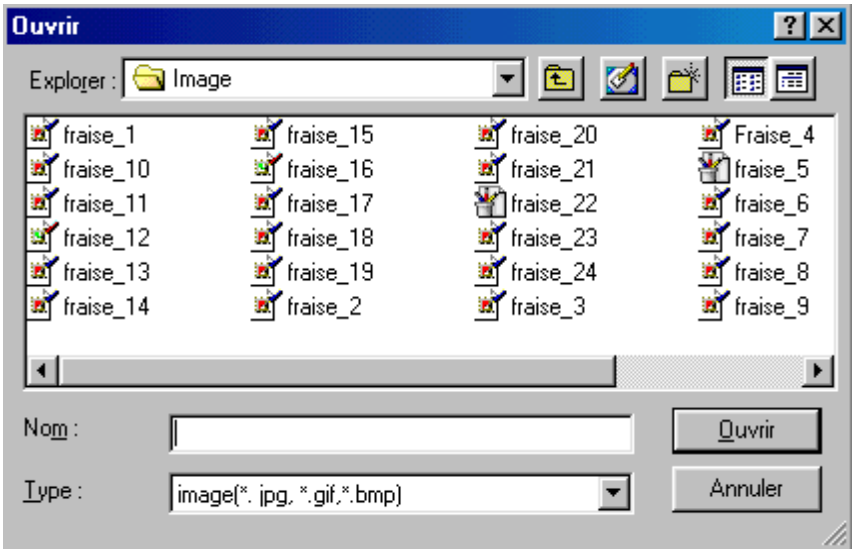
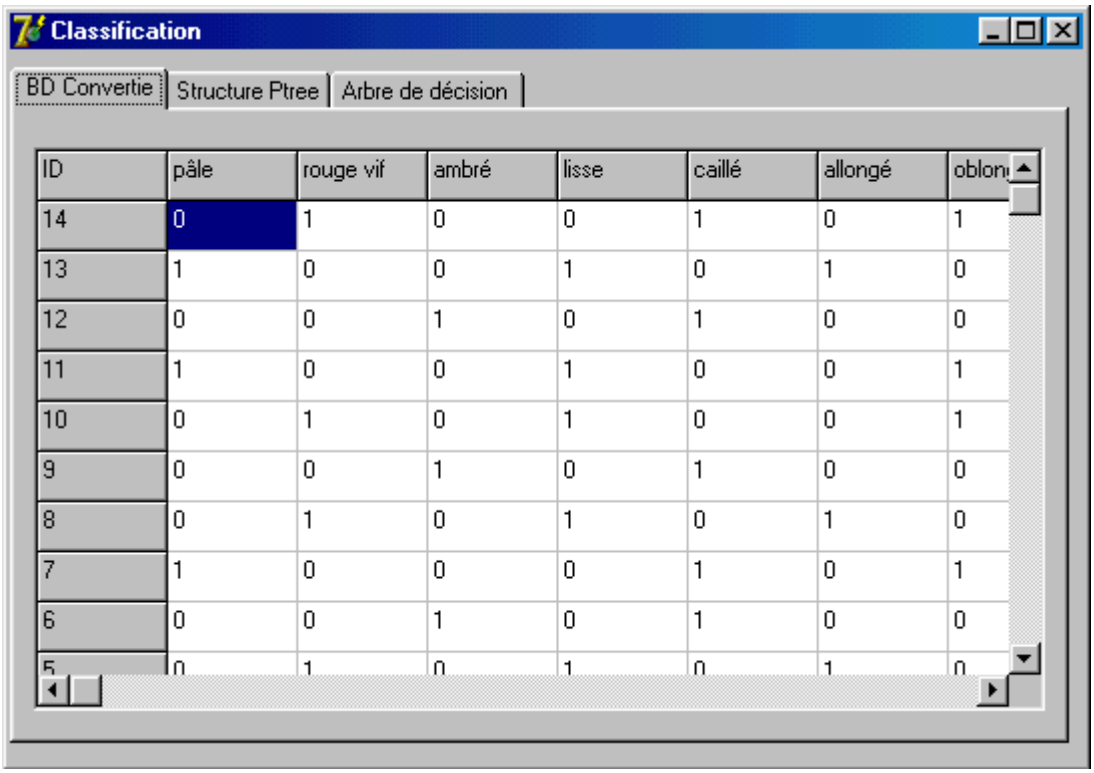


Fig. 5.4 – *Boite de dialogue permettant de charger une autre image à partir de la base.*



Classification

BD Convertie | Structure Ptree | Arbre de décision

ID	pâle	rouge vif	ambré	lisse	caillé	allongé	oblon
14	0	1	0	0	1	0	1
13	1	0	0	1	0	1	0
12	0	0	1	0	1	0	0
11	1	0	0	1	0	0	1
10	0	1	0	1	0	0	1
9	0	0	1	0	1	0	0
8	0	1	0	1	0	1	0
7	1	0	0	0	1	0	1
6	0	0	1	0	1	0	0
5	0	1	0	1	0	1	0

FIG. 5.5 – *La base des images convertie en base binaire.*

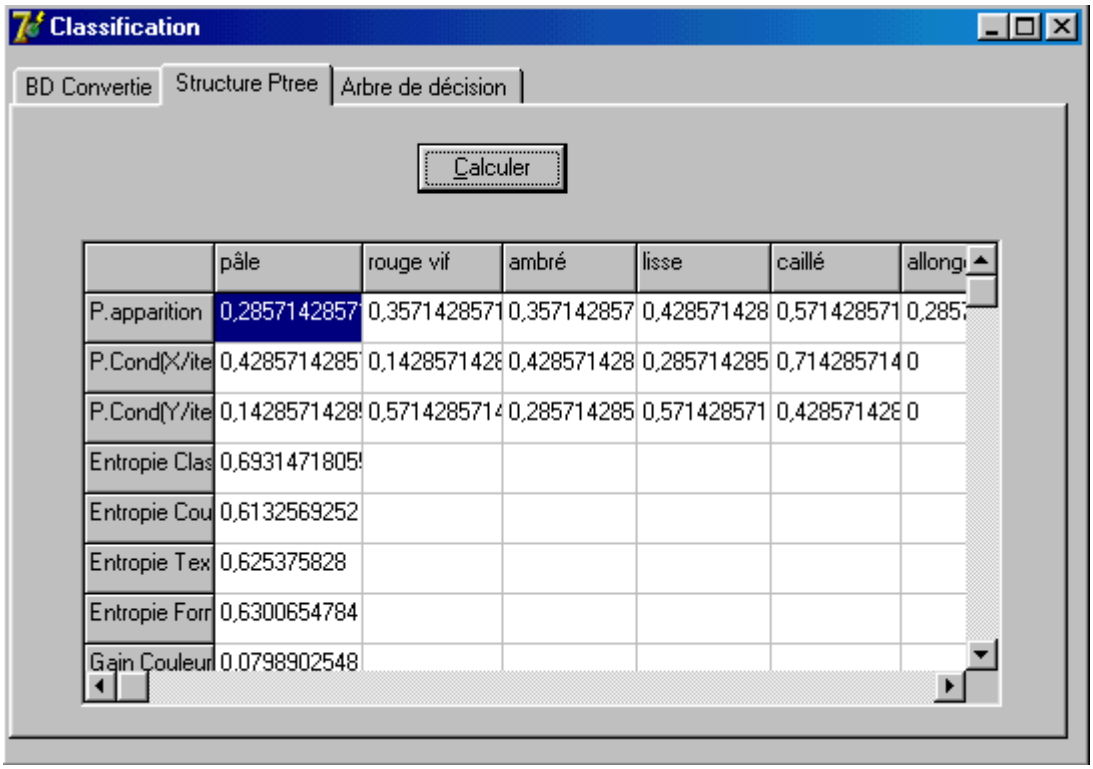


FIG. 5.6 – *Calcul des probabilités et des entropies des attributs des images.*

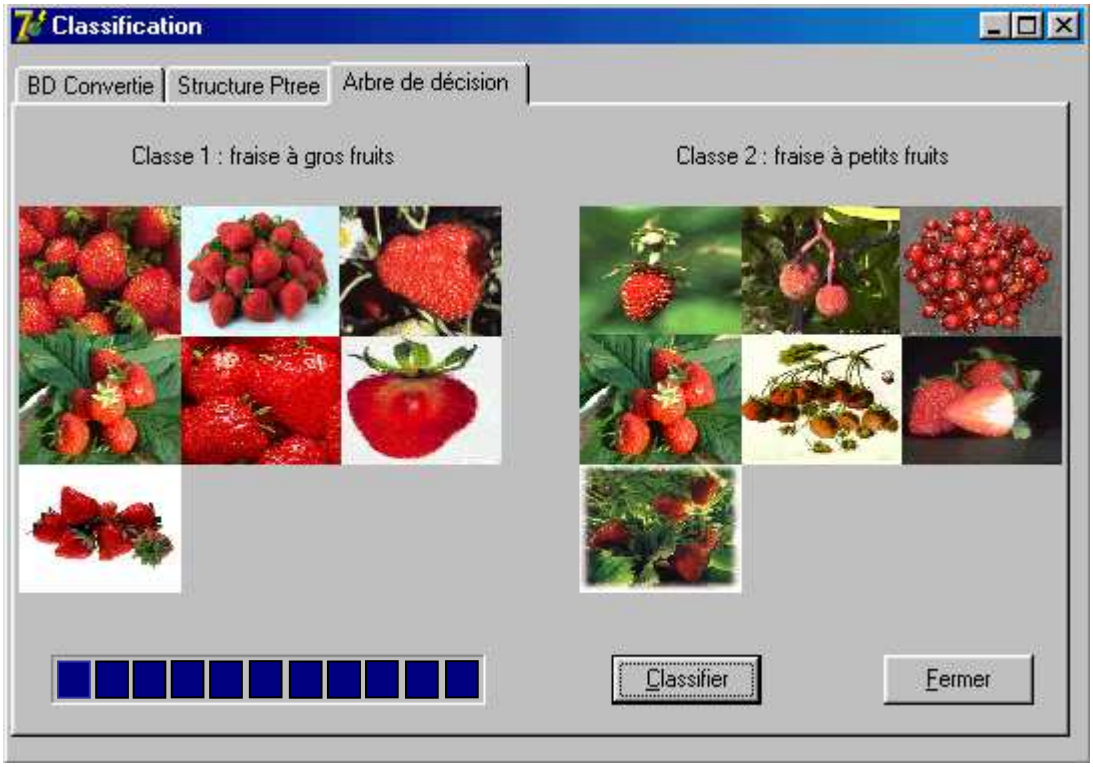


FIG. 5.7 – *Classification d'images suivant les attributs.*

5.6 Conclusion

A travers l'application que nous avons développé, nous avons montré que l'algorithme C4.5 basé sur la structure de donnée P-tree offre un gain de temps considérable par rapport à l'algorithme C4.5 en classification de la même base de données. Enfin, la technique P-tree combinée avec l'algorithme de classification C4.5 peut offrir un gain du temps assez élevé pour d'autres base de données.

Nous pouvons également appliquer notre système sur d'autres bases par exemple, la base Breast Cancer Wisconsin qui concerne une étude médicale réalisée sur 699 individus. Il s'agit de détecter à partir de 9 caractéristiques décrivant l'apparence des cellules, si un individu est atteint ou non du cancer du sein. La base Pima Indians Diabetes. Il s'agit cette fois de diagnostiquer à partir de caractéristiques cliniques si un sujet présente ou pas du diabète dans le sang[13], etc.

[13] ftp://ftp.ics.uci.edu/pub/machine-learning-databases/

Conclusion et perspectives

Dans ce travail, nous avons présenté une méthode d'apprentissage supervisé appliquée sur une base d'images. L'apprentissage consiste à générer des règles de classification à l'aide d'un arbre de décision basé sur l'algorithme C4.5. Nous avons également proposé de mettre en œuvre l'application **C.I.A.D.P-tree** basée sur la structure P-tree afin d'améliorer le temps de classification. Etant donnée une base d'images, cette méthode consiste dans un premier temps à représenter chaque image par un vecteur d'attributs (couleur, texture, forme) et associer à cette base d'images une base de données relationnelle. Dans un second temps, cette méthode consiste à convertir la base de données en une table REL formée par des bits (0 et 1), afin d'optimiser le temps de classification.

Nous avons présenté cette méthode binaire en terme des items (valeurs des attributs). Chaque chemin dans l'arbre de décision effectue une série de tests sur les items pour déduire la classe à affecter aux valeurs testées. Un chemin est donc équivalent à une règle de production *SI [prémisse] Alors [conclusion]*. L'algorithme utilisé dans **C.I.A.D.P-tree** est l'algorithme C4.5. Le calcul des probabilités pour choisir le meilleur attribut a été basé sur des règles de comparaison des P-trees (P-tree_i ANDing P-tree_j). Notre application présente plusieurs avantages parce que, d'une part, il évite le balayage direct de la base de données qui est une opération assez coûteuse en mémoire et en temps d'exécution et, d'autres part, il offre un gain de comparaison des attributs/items car la comparaison s'effectue par bloc des tuples de la table REL.

Si notre méthode présente de nombreux intérêts, soulignés ci-dessous, il est cependant clair qu'elle possède certaines limites. Celles-ci ouvrent de nouvelles voies d'études et posent de nouvelles questions à élucider. Nous développons ces différents points ci-dessous.

L'une des premières perspectives que nous nous proposons de réaliser est un travail complémentaire de validation. Il s'agit de tester notre application en utilisant d'autres bases d'apprentissage et de comparer les résultats avec ceux obtenus par d'autres méthodes de classification.

Les perspectives à court terme concernent également l'amélioration de notre système de classification. Il s'agit de développer une interface qui permettrait à l'utilisateur de visualiser directement les images classées au lieu des vecteurs d'attributs associés à ces images.

Il serait intéressant d'étendre l'utilisation de notre application à d'autres domaines d'applications. Nous pensons par exemple, à la segmentation d'images satellitaires ou encore d'images médicales, par exemple pour la détection de cellules anormales.

Bibliographie

[Agrawal et al. 92] R. Agrawal, S. Ghosh, B. Iyer, *An Interval classifier for database mining applications*, VLDB, 1992.

[Arkadev et al. 67] A.G. Arkadev, E.M Braveman,*Teaching Computer to Recognize Patterns*, Academic Press, London, 1997.

[Bara 99] BARA M., NANCERI C., *Neurotext, analyse de textes*, IEEE Intelligent Systems, juillet 1999.

[Bélaïd et al. 92] A. Bélaïd, Y. Bélaïd, *Reconnaissance des Formes, Méthodes et Applications*, Informatique InterEditions, 1992.

[Bonnet 84] A. Bonnet, *L'Intelligence artificielle : promesses et réalités*, Interédition, Paris, 1984.

[Bonnet 97] N. Bonnet, Intelligence Artificielle en analyse/traitement d'images, Séminaire d'informatique du LERI. Université de Reims Champagne Ardenne. La page des séminaires, http:// www.univ-reims.fr/Labos/SciencesExa/Leri/.Janvier 1997.

[Bonnet 98] N. Bonnet, *La madéldiction de la dimensionnalité, ses implications en reconnaissances des formes et les moyens d'y remédier*, Séminaire d'informatique du LERI, Université de Reims Champagne Ardenne, les pages des séminaires, http:// www.univ-reims.fr/Labos/SciencesExa/Leri/.12Mars 1998.

[Borgi et al. 01] BORGI A., AKDAG H., *Supervised Learning and Approximate Reasoning, The Hypothesis of Imperfections*, Revue d'intelligence artificielle, vol. 15-n°1/2001, p. 55-85, France.

[Boucheron 92] S. Boucheron. *Théorie de l'apprentissage, de l'approche formelle aux enjeux cognitifs*. Editions Hermes, Paris, 1992.

[Breiman 84] Breiman L., Friedman J.H., Olshen A., Stone C.J., *Classisfication of regression trees*, Champan and Hall, 1984.

[Caraux et al. 96] G. Caraux, Y. Lechevallier, *Règles de décision de Bayes et méthodes statistiques de discrimination*, Approches Symboliques et numériques de l'apprentissage, Revue d'Intelligence Artificielle, Vol. n°10, p 2-3, 1996.

[Coquerez et al. 95] J.P Coquerez et S. Philipp, analyse d'images : f*iltrage et segmentation*, Masson Editeur, Paris 1995.

[Dietterich et al. 83] DIETTERICH T.G., Michalski R.S., *A comparative review of selected methods for learning from examples*, Machine learning, An artificial intelligence Approach, vol.1, p. 41-81, Morgan Kaufman, 1983.

[Doermann 00] DOERMANN D., KOBLA V., FALOUTSOS C., *Representing and visualizing structure in video sequences*, Proceedings of Fifth ACM International Multimedia Conference, pages 335-346 November 2000.

[Dowe 93] DOWE J., *Content based retrieval in multimedia imaging*, In Proc. SPIE storage and retrieval for Image and Video Databases, 1993.

[Duda et al. 73] R.O. Duda, P.E. Hart, *Pattern Classification and Scene Analysis*, Standford Research Institute, California. A wiley-Interscience Publication, 1973.

[Elouedi et al. 00] ELOUEDI Z., MELLOULI K., SMETS P., *Decision trees using the belief function theory*, In Proceedings of the Eight International Conference on Information Processing and Management of Uncertainty, Vol. 1, pp. 141-148, Madrid, Spain: IPMU'2000.

[Faiz 02] FAIZ R., BISKRI I., *Hybrid Approach for the Assistance in the Events Extraction in Great Textuel Data Bases*, Proceedings of IEEE SMC 02, 6-9 October , Tunisia.

[Fayyad et al. 96] U. Fayyad, G. Piatetsky-Shapiro, P. Smyth. *From Data Mining to Knowledge Discovery in Databases*. AI Magazine, Vol. 17, Number. 3, pp.37-54 1966.

[Fukunaga 72] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, Academic Press, New York and London, 1972.

[Gaines 96] GAINES B., *Transforming rules and trees into comprehensible knowledge structure*, Advances in Knowledge Discovery and Data Mining, 1996.

[Gartner 95] Gartner Group, *Using data mining to gain a competetive edge, part 2* , Gartner Group this Week, 30 août 1995.

[Gartner 96] Gartner Group, *Peak performance, database marketing & data mining*, Gartner Group Report, mars 1996.

[HAN 99] HAN J., LI Z., CHEE S.H., CHIANG J. Y., *MultiMediaMiner: A System Prototype for Multimedia Data Mining*, Intelligent Database systems Research Laboratory and Vision and Media Laboratory, Burnaby, Canada, 1999.

[HAN 00] HAN J.and KAMBER M., *Data Mining : Concepts and techniques*, Edition Morgan Kaufmann, 2000.

[Mellouli et al. 01] MELLOULI K., ELOUEDI Z., SMETS P., Belief decision trees: Theoretical foundations, International Journal of Approximate Reasonning, pp. 91-124, Compiègne, France: AS-MDA 2001.

[Hunt et al. 66] E.B. Hunt, J. Marin, P.J. Stone, *Experiments in induction*, Academic Press, New York, 1996.

[Hsu 01] HSU W., ZHANG J., LEE L. M., *An Information-Driven Framework for Image Mining*, H.C. Mayer et al. (Eds): DEXA, LNCS 2113, pp. 232-242, 2001.

[Ishibuchi et al. 92] H. Ishibuchi, K. Nozaki, H. Tanaka, *Distributed representation of fuzzy rules and its application to pattern classification*, Fuzzy sets and systems, Vol. 52, pp 21-32, 1992.

[Jain 96] JAIN A., VAILAYA A., *Image Retrieval using Color and shape*, Pattern Recognition, 29(8): 1233-1244, August 1996.

[John et al. 94] G. H. John, R. Kohavi, *Irrelevant Features and the subset selection Problem*, Machine Learning, Proceedings of the Eleventh International Conference, pp121-129, 1994.

[Kanal et al.72] L. Kanal, B. Chandrasekaran, *On linguistic, statistical and mixed models for pattern recognition*, Frontiers of Pattern Recognition, New York, Academic Press, pp. 163-192, 1972.

[Kodratoff 97] Y. Kodratoff. *L'Extraction de Connaissances à partir des données : un nouveau sujet pour la recherche scientifique*. READ, Revue Electronique sur l'Apprentissage par les données, 1997.

[Lefebure et al. 01] LEFEBURE R., VENTURI G., *Data Mining : Gestion de la relation client, Personalisation de sites web*, Paris, Editions Eyrolles, Mars 2001.

[Manjunath 97] MANJUNATH B. S., MA W., DENG Y., *Tools for Texture/color based serach of images*. SPIE International conference – Human Vision and Electronic Imaging : 496-507, February 1997.

[Marsala 98] C. Marsala, Apprentissage inductif en presence de données imprécises: construction et utilisation d'arbres de décision flous, Thèse de doctorat de l'université Paris 6, Janvier 1998.

[Michalski et al. 83] R. S. Michalski, R.L. Chilausky. *Knowledge acquisition by encoding expert rules versus computer induction from examples : a case study involving soybean pathology*. Fuzzy Reasonning and its Applications, edited by E. H. Mamdani and B.R Gaines, Computers and people Series, pp 247-271, 1981.

[Michie 94] MICHIE D., *Methodologies from machine learning in data analysis and software*, The Computer Journal, 34, 6, 559-565, 1991.

[Mitchell 83] MITCHELL T. M., *Machine Leraning : An Artificial Intelligence Approach*, Iloga, 1983.

[Moulinier 97] I. Moulinier, *une approche de la catégorisation de textes par l'apprentissage symbolique*, Thèse de doctorat de l'université Paris 6, Avril 1997.

[Parrochia 97] *Classifications, histoire et problèmes formels*. Cinquièmes Rencontres de la société Francophone de classification SFC'97, pp 13-18, Lyon, Septembre 1997.

[Pentland 96] PENTLAND A., PICARD R.W., SLAROFF S. Photobook: *Content based manipulation of databases*. Int. J. Comput. Vis., 18(3): 233-254, 1996.

[Perrizo et al. 02] Perrizo W., Q. Ding, A. Roy, *On Mining satellite and other Remotely sensed Images*, in Proceedings of Workshop on Research Issues on Data Mining and Knowledge discovery, 2002, p 33-44.

[Piatetsky-Shaprio 93] PIATETSKY-SHAPRIO G., *Knowledge Discovery in databases*, IEEE Transactions on data and Knowledge Engineering, 5, 1993.

[Picard 72] C.F. Picard, Graphes et questionnaires, Gauthiers-Villards, 1972.

[Qiang Ding et al. 02] Qiang Ding, Qin Ding, William Perrizo, *Decision Tree Classification of Spatial Data Streams Using Peano Count Tree*, Technical Report NDSU-CSOR-TR-01-1, 2002.

[Qin Ding et al. 02] Qin Ding, Maleq Khan, Amalendu Roy, William Perrizo, *The P-tree Algebra*, Proceedings of ACM SIGKDD 2001.

[Quinlan 83] J.R. Quinlan, *Learning efficient classification procedures and their application to chess and games*: An Artificial Intelligence Approach, Vol.1, pp 463-482, Morgan Kaufman Publishers, 1983.

[Quinlan 86a] J.R. Quinlan, *Induction of decisions trees*, Machine Learning, Vol.1, p 82-106, Kluwer Academic Publishers, Boston, 1986.

[Quinlan 93] J.R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufman, 1993.

[Rakotomalala 97] R. Rakotomalala, *Graphes d'induction*, Thèse de doctorat de l'Université Claude Bernard, Lyon, décembre 1997.

[Ramdani 94] M. Ramdani, *Système d'Induction Formelle à base de connaissaces Imprécises*, Thèse de doctorat de l'université Paris 6, 1994.

[Saporta 90] G. Saporta. *Probabilités, Analyses des données et statistiques*, Editions technip, Paris, 1990.

[Shannon 48] C. E. Shannon, A mathematical theory of communication, *The Bell Systems Technical Journal*, pp. 379-423, 623-656.

[Simon 84] H. A. Simon. *Why Should Machines learn?* Machine Learning: an Artificial Intelligence Approach. Volume1, p25-37, Springer-Verlag, 1984.

[Simon 85] J. C. Simon. *La reconnaissance des formes par algorithmes*, Masson, 1985.

[Stepp et al. 83] R.E. Stepp, R.S. Michalski. *Learning from Observation: Conceptual Clustering*, Machine Learning, An Artificial Intelligence Approach, Vol.1, pp 331-363, Editors Morgan Kaufman Publishers, 1983.

[Tomassone et al. 88] R. Tomassone, MDanzart, J.J Daudin, J.P. Masson, *Discrimination et classement*, Masson, Paris, 1988.

[Tou et al. 74] J.T. Tou, R.C. Gonzalez, *Pattern Recognition Principles*. Addison-Wesley Publiishing Compagny, 1974.

[Turban 01] TURBAN E. and ARONSON Jay E., *Decision Support Systems and Intelligent Systems*, 6th ed, Copyright 2001, Prentice Hall, Upper Saddle River, NJ.

[Wang et al. 92] L-X. Wang, J.M. Mendel, *Generating Fuzzy Rules by Learning from Examples*, IEEE Transactions on systems, Man, and Cybernetics, Vol. 22, n°6, November/December 1992.

[Zaïane 00] ZAÏANE O. R., Simeon J., *Variations on Multimedia Data Mining*, Proceedings of the International Conference on Knowledge Discovery & Multimedia Data Mining, August 20-23, 2000, Boston, USA.

Sites web

http://vivaldi.ece.ucsb.edu/projects.html

http://www.engr.uconn.edu/~ibrahim/publications/image.html

http://www.irisa.fr/EXTERNE/afian

http://vision.ece.ucsb.edu/~jelena/reserach/

http://www.cs.ualberta.ca/~zaiane/mdmkdd2000/

http://db.cs.sfu.ca

http://citeseer.nj.nec.com/
http://www.datamodeling.com/fr/dmtechno.htm

Annexe

Extraits du code de l'application

A.1 Exemples de structures de données

```
type
Tattributs = record
            attributs:string[20];
            items:TStringList;
           end;
TFichier = file of Tattributs;

type
  Tbande = record
           bande:String[20];
           bits:array[1..100] of string[20] ;
         end;
       TFichb = file of Tbande;
```

A.2 Des exemples de procédures

```
procedure TForm1.ConvertirClick(Sender: TObject);
var
z,h,i,j,r,p,d:integer;
tex,x,y:String;
Lst:TStringList;
begin
 nomfb:='C:\WINDOWS\Bureau\ProjetNajeh\bandef';
 assignfile(fb, nomfb);
 rewrite(fb);
// reset(fb);
if  marquer=0 //filesize(fb)=0 //
```

```
then
begin
if marquer=0 then
begin
  nomf:= 'C:\WINDOWS\Bureau\ProjetNajeh\attribut';
   assignfile(f, nomf);
   rewrite(f);
  if nomf <> '' then begin
 // Lst:=TStringList.Create;
  Form2.StringGrid1.Visible:=false;
  Form2.Show;
  j:=DBGRID1.Columns.Count;
 // z:=0;
  Form2.StringGrid1.Cols[0].Add(DBGRID1.Columns[0].DisplayName);
  r:=2;
 Form2.StringGrid1.Rows[1].Add (DBGRID1.Fields[0].DisplayText);
 while(DBGRID1.Fields[0].DataSet.FindNext=true) do
   begin
   Form2.StringGrid1.Rows[r].Add(DBGRID1.Fields[0].DisplayText);
   r:=r+1;
  end;
  for h:=0 to j-1 do
   begin
   attrib.attributs := DBGRID1.Columns[h].DisplayName; {on enregistre les données }
   end;
 end;
 marquer:=1;
 end;
 nomfa:='C:\WINDOWS\Bureau\ProjetNajeh\att';
 assignfile(fa, nomfa);
  reset(fa);
  z:=1;
  d:=0;
  for i:=0 to filesize(fa) do
  if(not eof(fa)) then
  begin
  position:=filepos(fa);
  read(fa,attrib);
  y:=attrib.attributs;
  Lst:=TStringList.Create;
  for j:=0 to attrib.items.Count-1 do
  begin
  x:=attrib.items[j];
  Form2.StringGrid1.Cols[z].Add(x);
  z:=z+1;
  Lst.add(x);
  end;
   attrib.items:=Lst;
   write(f,attrib);
  end;
```

```
 reset(fa);
 for i:=0 to filesize(fa) do
 if (not eof(fa)) then
 begin
 position:= filepos(fa);
 read(fa, attrib);
 band.bande:=attrib.attributs;
// Form2.ProgressBar2.StepBy(14);
 y:=attrib.attributs;

DBGRID1.Fields[i].DataSet.First;
//if(DBGRID1.Fields[i].DataSet.FindNext=true) then
 x:=DBGRID1.Fields[i+1].DisplayText;
 for j:=0 to attrib.items.count-1 do
 begin
 tex:=attrib.items[j];
  d:=d+1;
  p:=1;
  //edit1.Text:=edit1.Text+ '&' +x+tex;
  if(p=1)then
  if(tex=x) then
 begin
 Form2.StringGrid1.Cells[d,p]:=IntToStr(1);
 end
 else
 begin
 Form2.StringGrid1.Cells[d,p]:=IntToStr(0);
 end;
 DBGRID1.Fields[i+1].DataSet.First;
//DBGRID1.Fields[i+1].DataSet.FindNext;
while (DBGRID1.Fields[i+1].DataSet.FindNext=true) do
begin
p:=p+1;
if(DBGRID1.Fields[i+1].DisplayText=tex) then
begin
Form2.StringGrid1.Cells[d,p]:=IntToStr(1);
end
else
begin
Form2.StringGrid1.Cells[d,p]:=IntToStr(0);
//band.bits.Add(IntToStr(0));
end;
end;
 DBGRID1.Fields[i+1].DataSet.First;
 Form2.StringGrid1.Visible:=true;
  write(fb,band);
 end;
 end;
 nomfb:='C:\WINDOWS\Bureau\ProjetNajeh\bandef';
 assignfile(fb, nomfb);
```

```
 rewrite(fb);
 for i:=0 to 125 do
 begin
  band.bande:=Form2.StringGrid1.Cells[i,0];
//  band.bits:=TStringList.Create;
//  Lst:=TStringList.Create;
  for j:=1 to 14 do
  begin
 // Lst.Add(Form2.StringGrid1.Cells[i,j]);
  band.bits[j]:=Form2.StringGrid1.Cells[i,j];
  end;
  write(fb,band);
  end;
 //Form1.Edit1.Text:=IntToStr(filesize(fb));
 end
else
begin
   showmessage('La base de données est déja convertie');
   end;
end;
```