THÈSE DE DOCTORAT DE
l'UNIVERSITÉ ABDELMALEK ESSAADI
FACULTÉ DES SCIENCES et TECHNIQUES
TANGER

Discipline
Analyse de Données

Spécialité
Mathématiques-Informatique

Centre d'Etudes Doctorales : « Sciences et Techniques de l'Ingénieur »
Formation Doctorale : « Sciences et Techniques de l'Ingénieur »

Présentée par
Abdellatif EL IDRISSI

Sujet de la thèse :

Processus Ponctuels Marqués Pour l'Analyse de Données Multidimensionnelles : Application à l'Analyse d'Images Couleur Texturées

Soutenue le 15 Juin 2015

devant le jury composé de :

Pr. Abderrahmane Sbihi, Tanger	Président
Pr. Abdelouahid Imlahi, Tanger	Rapporteur
Pr. Raja Touahni, Kenitra	Rapporteur
Pr. Rachid Oulad Haj Thami, Rabat	Rapporteur
Pr. Olivier Alata, France	Examinateur
Pr. Abdellatif Amrani-Ghacham, Larach	Directeur de thèse
Pr. Ahmed Moussa, Tanger	Directeur de thèse

Au nom d'Allah, le Tout Miséricordieux, le Très Miséricordieux.

" Lis, au nom de ton Seigneur qui a créé, qui a créé l'homme d'une adhérence. Lis ! Ton Seigneur est le Très Noble, qui a enseigné par la plume [le calame], a enseigné à l'homme ce qu'il ne savait pas."

Le Coran, Sourate 96 : L'adhérence (Al-Alaq), Versets 1-5

Ce travail n'aurait pas été bien accompli sans le soutien particulier de mes chers parents. Je me souviens toujours de leurs prières pour le bon aboutissement de ce projet de thèse, à chaque fois que je leur rendais visite. Mes sœurs et frères, également, m'avaient énormément supporté durant ces années à tout moments où je sentais avoir besoin de l'aide, en particulier *Mohammed, Meryem* et *Abdelhamid*. Je souhaite à vous tous de belles récompenses.

Et à vous, je dédie ce travail !

Cette thèse a bénéficié d'une bourse d'excellence octroyée par le Centre National pour la Recherche Scientifique et Technique et ce dans le cadre du programme des bourses de recherche initié par le Ministère de l'Education Nationale, de l'Enseignement Supérieur, de la Formation des Cadres et de la Recherche Scientifique.

Remerciements

C'est avec grande joie qu'à la fois je commence et je termine ce mémoire par des mots de remerciements et de reconnaissances envers celles et ceux qui ont contribué d'une manière ou d'une autre à l'aboutissement de ce travail de thèse.

Ainsi, je voudrais remercier d'abord le Professeur Ahmed Moussa, premièrement pour m'avoir proposé ce sujet de thèse et ensuite pour l'avoir dirigée et m'avoir encadré étape par étape, durant ces années de travail acharné. J'apprécie particulièrement son dynamisme, son exigence et son suivi minutieux. Telles attitudes m'ont donné de la motivation pour m'améliorer davantage tant sur le coté professionnel que personnel. Également, je le remercie pour les échanges agréables d'idées que nous avions entretenus durant ces années de recherche scientifique et pour m'avoir fait apprendre à travailler en collaborations.

Des remerciements spéciaux vont ensuite au Professeur Abdellatif Amrani Ghacham, qui a dirigé aussi cette thèse, je le remercie pour ses conseils et ses encouragements constants tout au long de ce projet. En particulier, dans les moments les plus difficiles, il était toujours là pour me redonner l'envie de persévérer. Il partageait de son expérience avec moi et j'en appris à chaque fois que nous nous rencontrions, surtout grâce à sa disponibilité et à son sens d'écoute.

J'aimerais également exprimer mes gratitudes à Monsieur Olivier Alata, professeur des universités et membre du Laboratoire Hubert Curien, Université Jean Monnet, pour les discussions entretenues sur les problématiques rencontrées dans ce travail, et pour l'apport de soutien à chaque fois que j'avais du blocage, que ce soit au niveau des idées ou au niveau de la méthodologie de raisonnement. Au delà de cela, j'ai apprécié énormément ses qualités d'écoute, de dialogue et de motivation.

Je tiens à remercier les professeurs Mme Raja Touahni, M. Abdelouhid Imlahi et M. Rachid Oulad Haj Thami pour avoir accepté de me faire l'honneur

de rapporter cette thèse. Je leur suis reconnaissant de l'attention qu'ils ont accordée à la lecture de ce manuscrit et de leurs remarques apportées à ce propos.

Je souhaite remercier aussi M. Abderrahmane Sbihi, Directeur de l'école Nationale des Sciences Appliquées de Tanger, pour les efforts incessibles déployés pour installer des conditions dignes de la recherche scientifique ainsi que pour avoir bien voulu me faire l'honneur de participer à mon jury de thèse.

Je m'adresse aussi avec de vifs remerciements au Professeur Yassine Ruichek, directeur du Laboratoire Système et Transport affilié à l'Université de Technologie de Belfort-Monbéliard, pour m'avoir accueilli au sein de son laboratoire lors d'un séjour de recherche et pour le suivi énergique du travail effectué pendant et après le séjour.

Je remercie aussi le jeune enseignant-chercheur Youssef El Merabet pour les jours et les nuits de recherche qu'on a passés ensemble, soit à Belfort en France lors de mon séjour ou bien à Kenitra au Maroc dans un travail d'équipe exemplaire. Merci pour les discussions fructueuses, les échanges d'idées et pour la bonne humeur.

Je ne saurais pas conclure cette page de remerciements sans que j'adresse mes gratitudes à mes amis et à mes collègues qui m'ont côtoyé au laboratoire pendant ces années, en particulier Asmae, Ouafae, Ismail, Ikram, Salah... Merci pour l'ambiance, pour les temps de convivialité, mais aussi pour les débats et discussions.

Résumé

Nous proposons dans cette thèse d'aborder la problématique de la classification non supervisée (*clustering*) de données multidimensionnelles. L'approche présentée repose sur l'idée de la détection des modes de la fonction de densité de probabilité sous-jacente aux données. Cette détection est effectuée en s'appuyant sur les processus ponctuels marqués. Les modes sont alors considérés comme étant des régions de l'espace à forte concentration de données. L'hypothèse faite est que l'utilisation des processus ponctuels marqués permet de localiser ces régions dans l'espace. Nous construisons donc un processus "objets" en définissant sa fonction de densité de probabilité par rapport à un processus de référence. Les méthodes de Monte Carlo par Chaines de Markov (en particulier l'algorithme de Metropolis Hasting Green) sont utilisées afin d'obtenir des configurations selon la loi du processus. Par ailleurs, la détection des modes n'est que la première étape dans notre méthode de classification automatique, la deuxième étape est un problème de classification supervisée et a fait l'objet de notre deuxième contribution. Ainsi avons-nous proposé une méthode en se basant sur les concepts de la théorie des champs de Markov qui nous permettront de prendre en considération les informations contextuelles des observations à analyser. En outre, cette méthode repose sur une stratégie d'affectation consistant à classer les observations non prototypes une par une en commençant par celle qui est la plus proche aux prototypes et qui en devienne une immédiatement. Le cadre applicatif de cette thèse a concerné la classification des images de textures couleurs ainsi que la reconnaissance d'objets. Lors de la première application, nous utilisons l'erreur de prédiction linéaire pour quantifier les attributs de l'image. Au cours de la deuxième application, une nouvelle méthode a été proposée en construisant une base de connaissances et en utilisant le coefficient de Bathacharyya comme mesure de similarité entre les attributs des différentes régions de l'image segmentée et de cette base.

Abstract

In this thesis, a contribution for the clustering problem is presented. The adopted approach is based on the idea of detection of modes of probability density function underlying the data. This detection is performed based on Marked Point Processes (MPP). Modes are then considered as space areas of high concentration of data. The assumption made is that the use of marked point processes allows locating these regions in the space. We build a process object by defining its probability density function with respect to a reference process. Monte Carlo Markov Chain methods (especially Metropolis Hastings Green algorithm) are used to sample from the process law. This mode detection is only the first step in our automatic classification method, the second step which is the subject of the second contribution of this thesis, is a supervised classification problem. Thus, we have proposed a method based on the concepts of Markov random fields theory that allows us to take contextual information of analysed observation into account. In addition, this method is also based on an assignation strategy aiming to classify non prototype observations one by one starting with the closest one to the prototypes which becomes immediately one of them. The application framework of this thesis concerned the classification of textured colour images and object recognition. In the first application, we quantify image features using linear prediction error. For the second application, a new method was proposed that consists in building a knowledge database and using the Bathacharyya coefficient to measure similarities between attributes of different regions belonging to both the database and the segmented image.

Table des matières

Table des matières **ix**

Table des figures **xii**

Liste des tableaux **xv**

Nomenclature **xv**

Introduction Générale **1**

Contexte et problématique 1
Domaines d'application 4
Contributions de la thèse 4
Détection de modes 4
Classification à partir des modes détectés 5
Extraction d'objets dans les images orthophotoplans 5
Organisation de la thèse 6

Chap 1 Classification : Approches et Méthodes **8**

1.1 Introduction 8
1.2 Le problème de la classification 8
1.3 Classification non supervisée ou clustering 10
1.3.1 Approche hiérarchique 11
1.3.2 Approche géométrique 13
1.3.3 Approche probabiliste 22
1.3.4 Approche par grilles 27
1.3.5 Approche *bio*-inspirée 29
1.3.6 Approche basée sur la densité ou les régions modales 34
1.4 Classification supervisée 39
1.4.1 Approche statistique 42

1.4.2 Approche logique 45
1.4.3 Approche connexioniste 46
1.4.4 Approche géométrique 48
1.5 Conclusion 49

Chap 2 Processus Ponctuels : Formalisme et Applications 51
2.1 Introduction 51
2.2 Processus ponctuels 51
2.2.1 Fondements des processus objets 52
2.2.2 Processus ponctuel de Poisson 54
2.2.3 Construction d'un processus ponctuel 55
2.2.4 Processus ponctuel de Markov 56
2.3 Simulation des processes ponctuels 57
2.3.1 Algorithmes de Monté Carlo par Chaînes de Markov 59
2.3.2 Optimisation par l'algorithme du recuit simulé 63
2.4 Diagnostic de la convergence des méthodes de Monte Carlo par Chaînes de Markov 64
2.4.1 Période d'échauffement et longueur de la chaîne 64
2.4.2 Tests de convergence 66
2.5 Applications précédentes des processus objets 67
2.5.1 Processus ponctuels et les phénomènes stochastiques 67
2.5.2 Processus ponctuels et l'analyse d'images 69
2.6 Conclusion 72

Chap 3 Clustering par Processus Objets 74
3.1 Introduction 74
3.2 Détection des modes par processus objets 74
3.2.1 Hypothèses de l'étude 75
3.2.2 Modélisation des interactions entre objets 77
3.2.3 Modèle d'attache aux données 79
3.2.4 Simulation du processus construit 82
3.2.5 Résultats expérimentaux 88
3.3 Classification à partir des prototypes 96
3.3.1 Notations 97
3.3.2 Modélisation markovienne 98
3.3.3 Algorithme proposé 101

3.3.4 Ajustement des paramètres 102
3.3.5 Résultats expérimentaux 105
3.4 Conclusion 108

Chap 4 Applications en Imagerie **111**
4.1 Introduction 111
4.2 Notions en analyse d'images couleur texturées 112
4.2.1 Processus de formation d'image 112
4.2.2 Description d'une image 116
4.2.3 Traitement d'image 119
4.3 Application à la segmentation d'images couleur texturées 122
4.3.1 Description de matériels 122
4.3.2 Segmentation d'images texturées par classification de pixels . . . 124
4.4 Reconnaissance d'objets par mesure de similarité entre régions à partir des images aériennes 127
4.4.1 Algorithme développé pour la reconnaissance d'objets 128
4.4.2 Application pour la détection de toitures dans les images orthophotoplans 135
4.5 Conclusion 147

Conclusions et Perspectives **151**

Bibliographie **155**

Activités Scientifiques liées à cette Thèse **170**

Table des figures

1 Taxonomie des différentes disciplines du domaine de l'analyse de données 3

1.1 Caractérisations globales des méthodes de classification. 10
1.2 Principales approches de la classification non supervisée. 11
1.3 Exemples de dendrogrammes de classification hiérarchique. 12
1.4 Schéma illustrant une tesselation de Voronoi en utilisant 20 points et deux types de distances ; de gauche à droite : distance euclidienne et distance de Manhattan. 15
1.5 Un exemple de classification automatique par une méthode de fourmis. A gauche, les données initiales ; A droite, les quatre classes émergentes (extrait de [152]). 30
1.6 Structure d'un neurone artificiel. Le neurone calcule la somme de ses entrées puis cette valeur passe à travers la fonction d'activation pour produire sa sortie. 31
1.7 Un exemple d'architectures des cartes auto-organisatrices. 32
1.8 Recherche des modes par la méthode de mean-shift. 38
1.9 Schéma d'un perceptron multi-couche. 47
1.10 Illustration de l'hyperplan séparateur d'un ensemble de donnée par la méthode SVM. 49

2.1 Exemple de distribution cible et l'approximation obtenue par histogrammes de l'échantillonnage par MCMC (extrait de [5]). 58
2.2 Détermination du *burn-in* et diagnostic de convergence par le test Geweke (Extraite de [134]). 68

3.1 Illustration d'un objet (cercle) et de son ombre. 82
3.2 Résultats de la détection des modes issus de l'application de la méthode proposée sur un ensemble de données gaussiennes. 90

3.3 Exemple d'application de la méthode proposée sur un type de données de structure non-linéairement séparable. 92
3.4 Résultats de la détection des modes par la méthode développée sur un ensemble de données présentant un degré de chevauchement. 94
3.5 Application de la méthode proposée sur deux structures différentes de données dans un espace en trois dimensions. 95
3.6 Les trois classes de l'exemple d'illustration et les sous-ensembles associés $\Pi_k, k \in \{1, 2, 3\}$ et Γ 98
3.7 Système de voisinage et cliques associés. 99
3.8 La classification itérative des deux distributions artificiellement générées. 103
3.9 Résultat de la classification des observations de l'exemple 1 106
3.10 Résultats de la classification de l'exemple 2 107
3.11 Résultats de la classification pour des données tridimensionnelles non gaussiennes 109

4.1 Différentes composantes formant un système de vision artificielle. . . . 114
4.2 Exemple d'histogramme de couleur quantifiée sur 4 bins pour chaque composante. 117
4.3 Images de textures couleur utilisées pour tester l'algorithme proposé. . . 124
4.4 Taux d'erreur de clustering par rapport aux espaces couleur utilisé. . . . 126
4.5 Taux d'erreur relatifs aux méthodes de comparaisons ainsi qu'à notre méthode. 126
4.6 Segmentation de l'image 1 (Fig. 4.3(a)) dans IHLS (top), L*a*b* (bottom). 127
4.7 Segmentation de l'image 8 (Fig. 4.3(h)) dans l'espace IHLS 127
4.8 Exemple de bases de connaissances construites pour ce travail. De haut en bas : la base de connaissance $\mathbf{B_{back}}$ des objets du fond d'image (végétation, route, forêt, etc) et la base de connaissances $\mathbf{B_{obj}}$ des toits des bâtiments (des toits rouges et non rouges). 129
4.9 Organigramme de la méthode proposée pour détection d'objets d'intérêt. 130
4.10 Exemple de résultat de segmentation utilisant la méthode SRM. De gauche à droite : image origine et résultat de segmentation SRM. . . . 131
4.11 Extraction automatique de plusieurs toits de bâtiments de l'ensemble des images traitées. De haut en bas : les images originales, les résultats de segmentation par SRM et l'extraction de toits correspondants. . . . 140
4.12 Exemples de certains éléments de toitures qui n'ont pas été extraits (ellipses jaunes) et de certaines fausses détections (ellipses vert). 141

4.13 Le graphe ROC de la méthode proposée appliquée sur les images de test. 143
4.14 Comparaison entre KNN, SVM et la méthode proposée. Première ligne : image de la vérité terrain et l'image des résultats de la détection des toits par la méthode proposée ; deuxième ligne : résultats de la détection par la méthode KNN selon différents paramètres (voir le tableau 4.3) ; dernière ligne : résultats de la détection par la méthode SVM suivant différents paramètres (voir le tableau 4.3). 146
4.15 Représentation graphique de la table 4.4. 147
4.16 Le graphe ROC comparant les variantes de la méthode KNN et la méthode proposée. 148
4.17 Le graphe ROC comparant les variantes de la méthode SVM et la méthode proposée. 149

Liste des tableaux

3.1 Résultats statistiques de l'exemple de données gaussiennes. 89
3.2 Paramètres statistiques des deux distributions bidimensionnelles à partir desquelles ont été tirées les observations de l'exemple du jeu de données non-linéairement séparables. 91
3.3 Résultats statistiques pour le premier exemple (see Fig. 3.6 and Fig. 3.9) 106

4.1 Textures utilisées pour construire les bases de connaissances d'objets d'intérêt et du fond. 137
4.2 Résultats de l'évaluation de la qualité d'extraction des toitures. 142
4.3 Les valeurs des paramètres expérimentaux liés aux méthodes (KNN et SVM) utilisées pour la comparaison. 144
4.4 Évaluation de la qualité de détection obtenue pour les méthodes KNN et SVM (plusieurs variantes selon différents paramètres) et la méthode proposée. 147

Introduction Générale

Contexte et problématique

Il n'est pas un hasard que l'époque actuel de l'histoire de l'humanité se baptise par l'ère de l'information. L'information est en effet devenue la force motrice du progrès dans tous les domaines de l'action humaine. En économie, en politique, en sciences, en industrie , et même dans l'éducation et la culture, tout aujourd'hui s'articule sur l'information et les connaissances. La popularité du dicton "Information, knowledge, is power " en témoigne. Celui qui sait comment obtenir des données, comment les gérer, les traiter les analyser et les utiliser pour prendre des décisions en sa faveur est celui qui en profite le mieux.

C'est pour cette raison que le recours aux nouvelles technologies de l'information et de la communication et aux systèmes intelligents est devenu une nécessité incontournable pour les décideurs des organisations, les entreprises, les industries et l'ingénierie, la médecine, voire pour les particuliers. Ces nouvelles technologies sont inéluctables pour enlever les enjeux du progrès et du développement et pour savoir vivre dans ce monde qui cherche inlassablement, au moyen de la mondialisation à réduire davantage le temps et l'espace.

Le but ultime de ces moyens est, en un mot, l'industrie de l'information. Une industrie qui se présente sous forme d'un processus et pose beaucoup de défis et de difficultés.

Ce processus que l'information suit, va de l'acquisition et la collecte des données pures jusqu'à leur exploitation, tout en passant par le choix d'un espace convenable pour les représenter, par le développement de structures de stockage et par la recherche et l'extraction des connaissances qu'elles contiennent. A l'issue de ce processus, ces données vont être employées pour élaborer des applications relevant de l'intelligence artificielle, pour faire des prévisions et bien gérer les risques et pour l'aide dans la prise des décisions rationnelles et fiables.

Quant aux défis et difficultés, ils résident dans le fait que de nos jours la quantité de

données acquises dans différents domaines sont caractérisées par leur colossal volume. Ce phénomène est connu sous la terminologie de *données massives* ou *mégadonnées* , son équivalent en anglais est le terme *Big data* [105]. Ce terme désigne alors des ensembles de données qui deviennent tellement volumineux qu'ils en deviennent difficiles à travailler avec des outils classiques de gestion de base de données ou de gestion de l'information en général.

Plusieurs disciplines rentrent dans ce contexte essayant de *redéfinir* les opérations de traitement de l'information mentionnées ci dessus. Parmi ces disciplines, on trouve l'intelligence artificielle, l'analyse de données et le traitement de signal.

Dans cette thèse, notre travail porte essentiellement sur le volet de l'analyse de données, plus précisément, il s'agit de la question de la classification ou *clustering* en anglais.

Par donnée nous désignons toute information élémentaire recueillie pour décrire un phénomène, un sujet ou un objet. Les données se présentent initialement sous la forme d'un tableau rectangulaire, dont les lignes correspondent à des individus ou unités statistiques (les objets) et les colonnes à des variables appelées caractères ou caractéristiques (les informations). Les variables peuvent être de l'un de ces types : quantitatives ordinales (jugement humain, température..), quantitatives mesurables (poids d'un individu, salaire..), qualitatives ordinales(classe d'âge, rang..) ou bien qualitatives nominales (sexe, situation matrimoniale..). Selon le type des variables, plusieurs tableaux peuvent être identifiés, tels que les tableaux de mesures, les tableaux de contingence, les tableaux de fréquence, tableaux de Burt, tableaux de proximités..etc.

Étant inutilisables sous leurs forme brutes, ces tableaux de données devraient être traités afin d'en extraire la connaissance utile que nous pouvons interpréter et utiliser. Un exemple de tableaux de données et de leurs utilisations dans la pratique, n'est pas évident mais complet et instructif pour illustrer ce propos, est celui, pris des données images, de la conception et de la construction d'un système intelligent qui soit capable de décider, à partir d'un ensemble donné comportant des images, si l'une d'elles contient un objet bien précis parmi plusieurs autres, le localiser et puis l'extraire.

La terminologie *analyse de données* désigne un ensemble de méthodes statistiques multidimensionnelles et descriptives. En effet, la statistique classique permet d'analyser les variables une par une. Il fallait tracer autant d'histogrammes et de graphiques que de variables. L'analyse de données vient pour simplifier l'étude de ces données de grandes dimensions et remplacer alors les graphiques multiples par un seul, et aussi pour donner une vision globale de l'ensemble des résultats. En bref, l'objectif est de permettre une visualisation des données dans le meilleur espace réduit et leur regroupement dans tout

l'espace.

Une partie des techniques de l'analyse de données visent à faire ressortir les relations qui existent entre les données (plus précisément entre les variables des données) afin d'en tirer les informations statistiques qui les décrivent d'une manière plus pertinente ; c'est une procédure de réduction des dimensions ; ce sont les méthodes de l'analyse factorielle (AF).

Une autre partie de ces techniques sont développées dans l'objectif d'organiser les données dans des groupes homogènes ; elle s'intéresse aux individus plutôt qu'aux variables ; il s'agit des méthodes de classification (*clustering*).

Dans chaque partie , l'objectif est de réduire un ensemble complexe et de grande dimensions à ces éléments essentiels afin d'en dégager la structure sous-jacente. Le schéma 1 illustre ces distinctions.

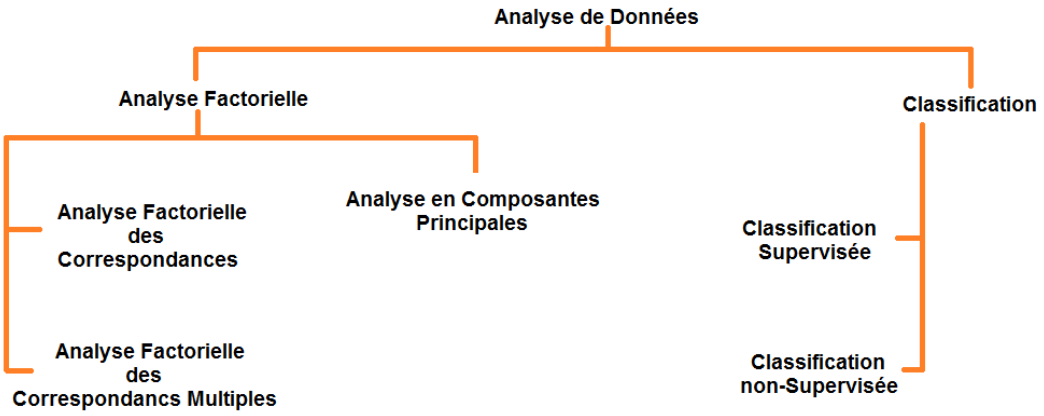


FIGURE 1 – Taxonomie des différentes disciplines du domaine de l'analyse de données

L'analyse factorielle regroupe trois méthodes : l'analyse en composantes principales (ACP), l'analyse factorielle des correspondances (AFC) et l'analyse factorielles des correspondances multiples (AFCM).

Dans le cadre de l'ACP, qui est la méthode la plus employée et particulièrement convenable pour les données dont les variables sont quantitatives et a priori corrélées entre elles, l'analyste cherche à approcher un système de points dans un espace par un sous-espace linéaire de dimensions inférieures. On cherche alors le sous espace donnant la meilleure visualisation du nuage de points des données.

L'analyse factorielle des correspondances (AFC), appelée aussi analyse des corres-

pondances binaires, est elle adaptée aux tableaux de contingence issues d'un croisement de variables qualitatives.

L'analyse factorielle des correspondances multiples (AFCM) est une génaralisation de la premiere qui ne permet que le croisement de deux variables à la fois. Elle est plus adaptées aux tableaux à variables qualitatives et de grandes dimensions.

L'autre type regroupe les méthodes de *classification supervisée*, appelée aussi *analyse discriminante* en statistique, et celle de *la classification non-supervisée* (automatique). Ce type de méthodes est complémentaire avec le précédent.

Une partie des problématiques abordées au cours de ce mémoire est une contribution à ce deuxième type des méthodes de l'analyse de données ; une méthode pour répondre à la question de la classification de données multidimensionnelles.

Domaines d'application

Les techniques de l'analyse de données sont actuellement une partie d'un domaine plus général, très prometteur et très recherché comme nous l'avons mentionné qui est celui de *Big Data* et de *Data mining* ; de ce fait, l'analyse de données est de plus en plus demandée dans plusieurs activités. En marketing, ces techniques sont très sollicitées pour analyser les relations clientèles, leurs profils et leurs tendances. En génétique pour analyser les expressions des molécules ADN. En météorologie, la grande masse des données recueillies nécessite l'utilisation des méthodes de visualisation et de prédiction pour prévoir les changements climatiques. Autrement dit, ces techniques s'utilisent dans tous les domaines qui nécessitent d'examiner et d'analyser des données massives : le domaine scientifique (économie, linguistique, écologie..), le domaine industriel (assurances, banques) et le domaine des nouvelles technologies (opérateurs de télécommunications, réseaux sociaux, moteurs de recherche, sites e-commerce..).

Contributions de la thèse

Dans ce contexte et en relation avec la problématique abordée ci-dessus, les travaux effectués dans cette thèse s'étalent sur trois volets. Le premier concerne la détection des modes à partir d'un jeu de données dont aucune connaissance préalable n'est disponible. Le deuxième volet porte sur une méthode d'affectation développée dans le cadre de l'apprentissage supervisé. Quant au dernier, il est consacré au développement d'une méthode pour la reconnaissance d'objets d'intérêt à partir des images.

Détection de modes

La première contribution dans cette thèse est le développement d'une nouvelle méthode pour la classification automatique des données multidimensionnelles. Dans cette méthode nous utilisons une approche par régions modales dont la stratégie est de détecter dans un premier temps les observations se localisant sur les régions de l'espace de représentation des données dont la densité est la plus élevée. C'est la détection des modes. Pour ce faire, nous utilisons le cadre stochastique des processus ponctuels marqués et les techniques de simulation de Monte Carlo par Chaînes de Markov.

Les observations appartenant aux modes détectés, appelées *prototypes* seront utilisées par la suite pour classifier le reste des données. Cette deuxième phase est le deuxième travail dans cette thèse, il concerne la classification à partir des modes.

Classification à partir des modes détectés

L'opération de détection des modes n'est que la première étape dans le processus de la classification automatique, car elle représente une phase d'apprentissage durant laquelle l'objectif est de délimiter les régions modales. Cependant, à la fin, il y aura encore un ensemble d'observations qui ne sont affectées à aucune classe. Donc, pour classifier l'ensemble des données, une phase de classification supervisée est nécessaire.

Certainement, cette classification peut être faite par l'une des méthodes classiques d'affectation. Néanmoins, nous avons introduit une nouvelle méthode de classification basée sur l'utilisation des champs de Markov. L'objectif est de prendre en considération les informations contextuelles. A l'issue de ces deux étapes, toutes les données brutes seront classées dans l'une des classes détectées.

Extraction d'objets d'intérêt dans les images orthophotoplans

La troisième contribution dans cette thèse est le développement d'une méthode pour la reconnaissance d'objets dans les images orthophotoplans. Cette méthode se base sur l'utilisation des mesures de similarité entre régions. L'objectif de la méthode est qu'elle soit automatique. Pour ce faire, deux bases de connaissances sont construites. La première est formée des textures représentant les objets d'intérêt et la deuxième est formée des textures représentant le fond de l'image.

Les mesures de similarité sont alors calculées entre les régions de l'images en considération d'une part, et les bases de connaissances de l'autre part. Selon les valeurs de similarité, les différentes région de l'image sont étiquetées. Les régions adjacentes appartenant à la même classe sont fusionnées laissant ainsi apparaitre les objets d'intérêt

recherchés.

Organisation de la thèse

Ce rapport de thèse est donc organisé en une introduction générale, quatre chapitres et une conclusion.

Le chapitre 1 est consacré à une récapitulation sur l'état de l'art du sujet. Nous y exposons les principales approches et méthodes proposées dans la littérature pour résoudre le problème de l'apprentissage supervisé et non supervisé. Vu que le domaine est très fécond au sens de la multiplicité des techniques développées, nous présentons les approches les plus connues à partir desquelles de nombreuses méthodes ont été développées. L'objectif est d'aborder le principe de chaque approche et puis de l'illustrer par une ou deux de ses méthodes.

Dans le chapitre 2, nous présentons les fondements mathématiques des processus ponctuels marqués, le cadre stochastique dans lequel notre contribution pour la détection des modes s'inscrit. Une partie de ce chapitre est consacrée aux méthodes de simulation de ces processus. Nous y abordons les algorithmes de Monte Carlo par Chaînes de Markov et l'algorithme de recuit similé qui permet de trouver une solution optimale dans un ensemble de solutions initiales. Ce chapitre sera terminé en évoquant certaines applications précédentes de ces processus qui inspirent de proche notre contribution. Ces applications concernent en particulier l'analyse d'images et la détection d'objets d'intérêt.

La méthode que nous proposons pour contribuer à la classification automatique sera abordée dans une première partie du chapitre 3. Nous présentons notre modélisation pour la notion de *classe* dans le cadre des processus ponctuels marqués. Nous construisons un processus qui prend en considération les interactions entre les observations dans l'espace attributs. La simulation du processus construit nécessite l'adaptation des méthodes de Monte Carlo par Chaînes de Markov en définissant des noyaux de propositions compatibles avec le problème traité.

La méthode de détection des modes proposée sera testée sur plusieurs ensembles de données de différentes structures. L'objectif est de l'étudier dans différents contextes. Ces ensembles de données sont pris aussi dans des espaces multidimensionnels.

Dans la deuxième section du chapitre 3, nous présentons la méthode proposée pour faire de la classification supervisée. Les notions de cliques et de potentiels que nous trouvons dans le cadre théorique des champs de Markov et qui nous permettront de prendre en considération les informations contextuelles des données seront abordées. Également

nous testons cette méthode sur un ensemble très varié de données multidimensionnelles.

Le dernier chapitre 4 est consacré à deux applications effectuées sur des images couleurs. Après avoir commencé par une brève introduction aux notions relevant du domaine de traitement d'images, une application dédiée à la segmentation d'image couleur texturées y est ensuite présentée. La deuxième application 4.4 est ensuite abordée et elle porte sur la reconnaissance d'objets d'intérêt à partir des images couleur. Cette application est par conséquent l'occasion de présenter la troisième contribution de cette thèse relative à la reconnaissance d'objets.

Chapitre 1

Classification : Approches et Méthodes

1.1 Introduction

Dans ce chapitre, nous abordons la problématique de la classification. Nous commençons par clarifier un peu la notion elle même et quelques concepts qui l'accompagnent. Notre attention se concentrera ensuite sur les approches et méthodes qui ont été développées dans le domaine. Nous ne prétendons pas que cette revue soit exhaustive vu la diversité des méthodes existant. Cependant, nous présenterons les grandes approches régissant les deux catégories majeures de la classification, en l'occurrence la classification automatique et la classification supervisée. Pour chaque approche, nous décrirons dans un premier temps son cadre théorique ou son principe de base ; dans un deuxième temps, nous présenterons une ou deux de ses méthodes. Ceci revient dans la plupart des cas à la notoriété de la méthode aux yeux des spécialistes du domaine de la classification ou au fait qu'elle touche de très près notre travail dans cette thèse.

1.2 Le problème de la classification

La classification est le fait de rassembler des données (observations) dans des groupes homogènes appelés souvent classes (parfois clusters) suivant un critère (fonction objective, fonction de coût) ou une mesure de similarité [77]. Plusieurs facteurs rentrent en jeu donnant naissance aux nombreuses approches et méthodes que nous trouvons dans la littérature et qui tentent d'apporter solution à ce problème. Parmi ces facteurs, nous pouvons citer à ce propos, la représentation des données (ce que sont les objets

à classer), l'espace de représentation (extraction/sélection d'attributs pertinents), la modélisation du problème (ce que sont les classes : approche heuristique/théorique) et la mesure de similarité employée (distance, fonction..).

Par ailleurs, la classification comme outils d'analyse de données se subdivise en deux types selon si la procédure s'effectue en disposant de connaissance a priori outre les données analysées elles-même ou non. Dans le premier cas, la classification est dite supervisée (apprentissage par professeur) : construire une fonction de décision pour affecter des points à une partition connue puis choisir, interpréter et évaluer une règle de classement. Dans l'autre cas, elle est dite non supervisée ou automatique : trouver une partition dans un jeu de données et choisir une partition (nombre de classes compris) et l'interpréter.

Une description plus détaillée sera présentée dans les sections dédiées (cf. sections 1.3 et 1.4).

Nous pourrions parfois rencontrer le terme classification semi-supervisée : s'aider de points non étiquetées à une classe pour construire une règle de décision d'affectation à des classes connues.

Une méthode de classification est aussi caractérisée par paramétrique ou non paramétrique. Dans le premier cas, la méthode dispose d'un ensemble de variables ou de valeurs appelées vecteur paramètres. Ces paramètres sont fixés initialement par l'analyste ou c'est la méthode elle même qui propose une phase pour les estimer conjointement avec la procédure de classification. Dans le cas inverse, la procédure de classification n'a pas besoin de telles variables vu qu'elles sont calculées directement à partir des données brutes et du mécanisme de classification adopté(calcul des histogrammes par exemple).

Également une méthode de classification est dite générative si elle modélise le système complet qui a généré les données et les classes sous-jacentes . Si la procédure de classification s'intéresse uniquement aux classes et propose directement une partition des données et non plus à tout le système, on dit que la méthode est discriminative.

La figure 1.1 montre schématiquement cette distinction entre les méthodes de classification. Il est à noter que cette caractérisation est indépendante des approches que nous allons voir par la suite. Elle comprend les méthodes supervisées et non supervisées. Nous pouvons par exemple dire qu'une méthode est supervisée, non-paramétrique et générative.

Une autre notion aussi importante dans la conception de la plupart des systèmes d'apprentissage est celle de la stabilité. En effet, il est très fréquent qu'un individu passe d'une classe à une autre classe différente en se déplaçant d'une itération à une

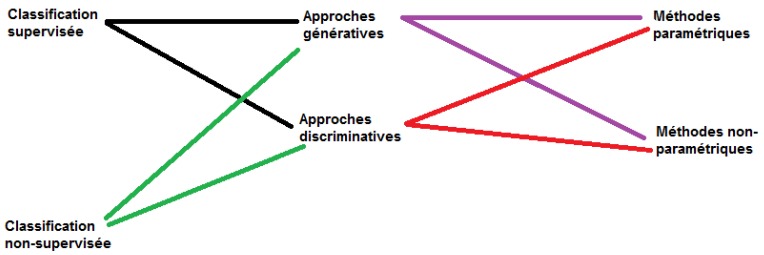


FIGURE 1.1 – Caractérisations globales des méthodes de classification.

autre. Un système est donc dit stable si à partir d'un certain rang d'itération, aucun individu ne change de classe. Cette stabilité devrait être vérifiée suivant le principe de la méthode et le critère qu'elle projette d'optimiser.

Nous présentons dans la section suivante les principales approches de la classification non supervisée.

1.3 Classification non supervisée ou clustering

En supposant dans cette section que les données dont on dispose sont représentées par un ensemble de n éléments que nous appelons observations (patterns) $y = (y_1, ..., y_n)$ prises d'un espace E de d dimensions qui est en général donné par $\mathbb{R}^d$ [1]. Cet espace est nommé espace d'attributs (feature or attributes space). De ce fait, chaque observation $y_i \in y$ est plutôt un vecteur dont les composantes sont notées $y_i = (y_{i,1}, ..., y_{i,d})$. Par conséquent, les données présentées pour classification sont dans la plupart des cas représentées sous forme d'une matrice de dimension $n \times d$.

L'objectif de la classification automatique est de prédire les étiquettes ou labels $x = (x_1, ..., x_n)$ des observations $y = (y_1, ..., y_n)$ à partir de la connaissance des valeurs prises par les $n \times d$ variables explicatives (attributs) $y_{i,j}, i, j \in [1..n] \times [1..d]$.

Deux grandes difficultés sont à surmonter dans ce type de classification. D'abord, nous ne connaissons pas préalablement le nombre k de classes. Deuxièmement, nous ne disposons d'aucun jeu d'apprentissage (training set) nous indiquant les caractéristiques prévues de chaque classe.

1. Étant donné que la classification est utilisée dans plusieurs disciplines, la terminologie utilisée pour décrire les deux éléments des données brutes peut varier largement. Les attributs peuvent ainsi être appelés caractéristiques, variables indépendantes, ou variables explicatives. Les observations peuvent être appelées des instances, des cas, des lignes ou des enregistrements.

Nous pourrions regrouper les différentes méthodes proposées dans la littérature dans les grandes approches suivantes (Figure 1.2) :

- Approche hiérarchique
- Approche géométrique
- Approche probabiliste
- Approche par grille
- Approche *bio*-inspirée
- Approche basée sur la densité

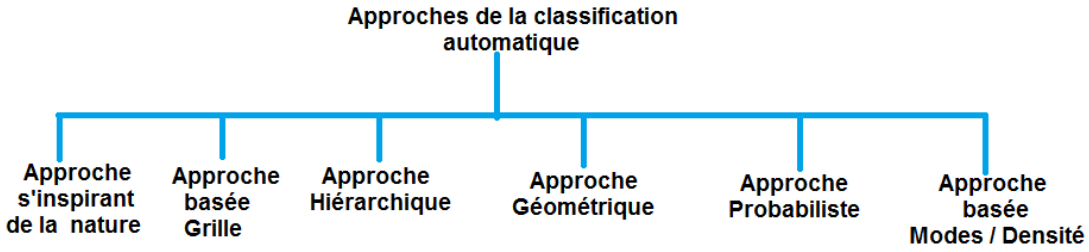


FIGURE 1.2 – Principales approches de la classification non supervisée.

Cette division n'est pas idéale puisque le lecteur pourrait bel et bien rencontrer des méthodes hybrides qui font parties d'au moins deux approches.

Nous détaillons chaque approche en exposant son principe et ses méthodes les plus connues.

1.3.1 Approche hiérarchique

L'approche de classification hiérarchique consiste à regrouper, respectivement diviser, l'ensemble des données pour les présenter sous forme d'un arbre ascendant, respectivement descendant. Dans le premier cas , on parle de classification ascendante hiérarchique (CAH) et dans le deuxième cas , nous parlons de classification descendante hiérarchique (CDH) [77; 82].

- Classification ascendante hiérarchique :

En considérant chaque objet (observation) comme une classe distincte, la CAH, appelée aussi méthode *agglomérative* (approche bottom-up), procède à chaque itération par fusionner deux classes (éventuellement deux objets) à la base d'une mesure de similarité. Ce processus de fusionnement peut se faire selon de nombreuses règles d'agrégation dont les plus connues sont : saut minimum (single linkage), diamètre (complete linkage) et moyenne.

- Classification descendante hiérarchique :

  Au départ, tous les objets forment une seule classe, et le but de la méthode de la classification descendante hiérarchique est de diviser cette classe en deux nouvelles classes en se basant sur un critère donné. A la prochaine itération, cette procédure de division est réitérée en choisissant une autre classe. La classification est donc nommée aussi divisive ou haut-bas (top-bottom).

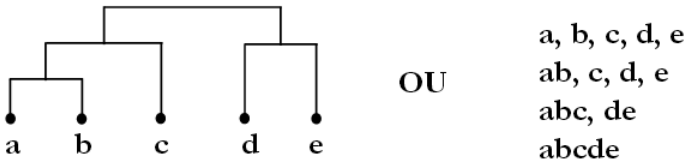


FIGURE 1.3 – Exemples de dendrogrammes de classification hiérarchique.

Le critère sur la base duquel la fusion ou la division s'effectuent est éventuellement à redéfinir pour chaque itération de l'algorithme. Comme le lecteur peut le constater, cette méthode ne tranche pas directement sur le nombre de classes mais elle donne tout un ensemble de partitions possibles emboitées (dendrogramme)de l'ensemble brut des données. C'est à l'utilisateur (l'analyste) de désigner à quel niveau de coupure la procédure de classification devrait s'arrêter et par la suite le nombre de classes désirées. Cela nécessite donc que l'on définisse au préalable une distance entre les classes : c'est ce qu'on appelle le choix d'une stratégie d'agrégation.

La plupart des algorithmes hiérarchique sont des variantes de deux algorithmes, à savoir lien simple (single-link, [142]) et le lien complet (complete-link, [85]). La différence entre ces deux algorithmes réside dans la façon suivant laquelle chacun définit la mesure de similarité entre classes. Alors que le premier considère que cette distance est calculée

comme étant le minimum des distances entre les paires d'objets appartenant aux deux classes, le deuxième en prend la distance maximale. Cependant pour trancher lesquelles des classes vont être agrégées, les deux algorithmes prennent celles qui ont la similarité minimale.

Malgré l'utilisation fréquente des algorithmes hiérarchiques, leur complexité proportionnelle au carré du nombre total des observations les rend non convenables pour des données à grandes dimensions.

1.3.2 Approche géométrique

A la différence des méthodes de l'approche hiérarchique qui produisent progressivement des partitions et laissent le choix de la partition optimale à l'utilisateur, les méthodes relevant de l'approche géométrique fournissent directement une et une seule partition des données en optimisant une fonction objective.

Le principe de cette approche, qui s'appelle aussi approche de partitionnement, consiste d'abord, à se fixer préalablement le nombre k a priori de classes désirées et de désigner ensuite k observations d'une manière aléatoire parmi les données. Ces observations choisies seront les centres des classes initiales. Ensuite, les distances entre les points et ces centres vont être calculées et chaque point est affecté au centre le plus proche au sens d'une distance (souvent euclidienne). Les centres seront mis à jour suivant une procédure (moyenne , barycentre, medoide..). Après, on réitère la procédure d'affectation de chaque observation à l'un des nouveaux centres. Ces étapes sont répétées jusqu'à la convergence de l'algorithme , c'est à dire jusqu'à ce qu'aucun centre ne change de positionnement.

Le type de distance utilisée pour calculer la proximité entre les observations et la procédure de mise à jour des centre des classes sont à l'origine des différentes méthodes et algorithmes relevant de cette approche. Les algorithmes les plus connus se basant sur un principe géométriques sont *k-means*, *k-medoides*, *nués dynamiques* et *fuzzy c-means*.

Tous ces algorithme sont des méthodes linéaires (cf. section 1.3.2.1). Ils produisent généralement des classes sphériques vu qu'ils emploient des séparateurs linéaires (hyperplans) dans l'espace de représentation des données pour délimiter les classes. Ils sont donc inutilisables lorsque les données présentent des structures non linéairement séparables. Pour remédier à cet inconvénient, des variantes de ces méthodes ont été développées dans la littérature visant à trouver des séparateurs non linéaires. Il s'agit des méthodes à noyaux et les méthode spectrales (cf. section 1.3.2.2).

L'approche à noyaux consiste à transformer les observations de leur espace original

vers un autre espace de dimensions plus grandes appelé espace attribut. L'avantage de cette transformation réside dans le fait que trouver un séparateur linéaire dans ce nouvel espace revient d'une manière équivalente à trouver un séparateur non linéaire dans l'espace d'origine. Pour l'approche spectrale, elle se base sur la théorie des graphes et représente le problème de clustering sous forme d'une coupe d'un graphe où une fonction objective devrait être optimisée. Ainsi pour résoudre le problème de clustering, on fait appel à la théorie des graphes spectraux et notamment les vecteurs et valeurs propres de la matrice laplacienne.

Dans ce qui suit, nous allons présenter une description de ces méthodes en se basant sur la notion de cellule de Voronoi et en commençant par les méthodes linéaires puis nous aborderons leurs variantes non linéaires [1].

Soit $X = \{x_1, ..., x_n\}$ un ensemble de points tels que $x_i \in \mathbb{R}^d$. Le codebook ou ensemble de centroides V est défini comme étant l'ensemble $V = \{v_1, ..., v_c\}$, typiquement avec $c \ll n$. Chaque élément $v_i \in \mathbb{R}^d$ est appelé codevecteur ou centroide ou prototoype. La région de Voronoi R_i du codevecteur v_i est l'ensemble de vecteurs dans $\mathbb{R}^d$ pour lesquels v_i est le plus proche vecteur :

$$R_i = \{z \in R^d | i = \arg\min_j \|z - v_j\|^2\}. \tag{1.1}$$

L'ensemble de Voronoi π_i du codevecteur v_i est donné par le sous-ensemble de X pour lequel le codevecteur v_i est le plus proche vecteur. Autrement dit, c'est l'ensemble de vecteurs appartenant à R^i. Nous l'exprimons par :

$$\pi_i = \{x \in X | i = \arg\min_j \|x - v_j\|^2\} \tag{1.2}$$

Une partition de $\mathbb{R}^d$ formée par toutes les régions de *Voronoi* est appelée un *tissu (tesselation)* ou *tissu de Direchlet.*

1.3.2.1 Méthodes linéaires

Centre mobiles ou *k-means* La plus connue de ces méthodes est celle des centres mobiles, due principalement à Forgy ([49]). Son principe est le suivant : si l'on veut k classes, on choisit k points dans l'espace des individus ; on affecte ensuite chacun des n individus celui de ces k points qui lui est le plus proche (au sens de la distance d choisie au départ) ; les k points de départ sont remplacés par les k (ou moins, si un

1. Dans cette section, nous allons adopter la présentation suivie par [46] dans son article sur un review à propos des méthodes à noyaux et celles spectrales.

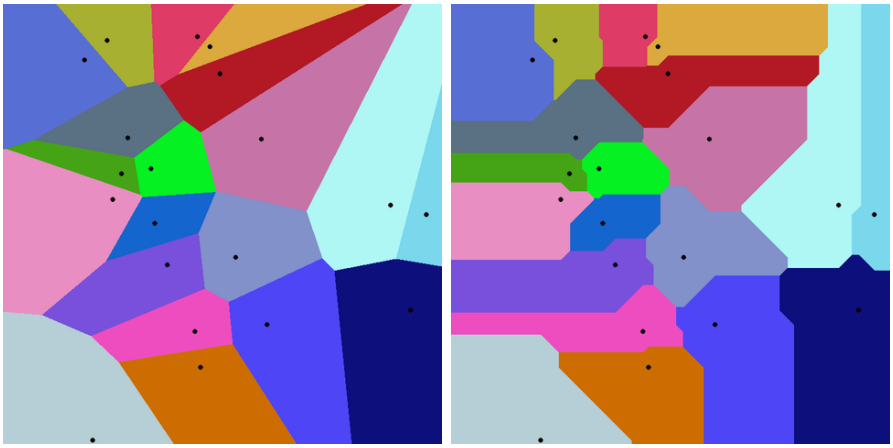


FIGURE 1.4 – Schéma illustrant une tesselation de Voronoi en utilisant 20 points et deux types de distances ; de gauche à droite : distance euclidienne et distance de Manhattan.

point n'a attiré aucun autre) barycentres des individus affectés à chacun ; puis on réitère l'affectation, jusqu'à la convergence ([103]).

Un algorithme très simple pour la construction d'un tissu de Voronoi de l'espace de représentation des données est proposé par Lloyd en 1957 (il a été publié après dans [99]). Cet algorithme est connu sous le nom de *batch k-means*.

En commençant par l'ensemble fini de données X , l'algorithme déplace les k codevecteurs vers la moyenne arithmétique de leurs ensembles de Voronoi $\{\pi_i\}_{i=1,\ldots,k}$.

Théoriquement une condition nécessaire pour un codebook afin de minimiser l'erreur empirique de quantification (empirical quantization error) :

$$E(X) = \frac{1}{2N} \sum_{i=1}^{k} \sum_{x \in \pi_i} \|x - v_i\|^2 \tag{1.3}$$

est que le codevecteur v_i remplisse la condition des centroides ([58]). Dans le cas d'un ensemble fini de données et de la distance euclidienne, la condition des centroides se réduit à :

$$v_i = \frac{1}{|\pi_i|} \sum_{x \in \pi_i} x. \tag{1.4}$$

Le mot batch veut dire qu'à chaque itération l'algorithme considère tout l'ensemble de données pour faire la mise à jour des codevecteurs. Lorsque la dimension des données est élevée, la complexité de la procédure du *batch* devient trop élevée aussi, ce qui a

fait que certaines améliorations y ont été apportées [1].

Nués dynamiques : Les nuées dynamiques sont une généralisation du principe du k-moyennes, pour laquelle chaque partition est représentée par un noyau pouvant être plus complexe qu'une moyenne. Elle a été mise au point par Diday dans [40].

Fuzzy clustering (soft clustering) Avant de présenter la deuxième méthode qui appartient à l'approche géométrique qui est celle de *fuzzy c-means*, laissons-nous préciser ce que nous entendons par *fuzzy clusterung* (soft clustering).

Le concept de *hard and fuzzy clustering* a été introduit par Bezdek ([11]) pour étendre la notion de degré d'appartenance d'un objets à l'une des classes (*membership*). L'idée revient au fait qu'un point n'appartient pas toujours à une et une seule classe, mais éventuellement son appartenance à plusieurs classes est aussi possible. Dans de nombreux cas, il y a une nécessité de décrire des points où leurs appartenances est partagée entre plusieurs classes. Ce degré d'appartenance est mesuré par une probabilité comme nous allons voir en détail.

Définition 1 (Fuzzy c-partition space) *Soit A_{cn} l'espace vectoriel formé des matrice réelles de dimension cn définies sur R. Etant donné X, A_{cn} et $c \in N$ tel que $2 \leq c < n$, l'espace Fuzzy c-partition de X est l'ensemble :*

$$M_{fc} = \{U \in A_{cn} | u_{ih} \in [0,1] \, \forall i, h; \sum_{i=1}^{c} u_{ih} = 1 \, \forall h; \, 0 < \sum_{h=1}^{n} u_{ih} < n \, \forall i\} \tag{1.5}$$

La matrice U est la matrice d'appartenance puisque chaque élément u_{ih} représente le soft appartenance du h^{th} point à la i^{th} classe. La définition de M_{fc}, signifie que la somme des degrés d'appartenance d'un point à toutes les classes est égale à un et qu'une classe ne peut en aucun cas ni être vide ni contenir tous les points. Cette notion est en fait une généralisation de la notion de *hard c-partition* ([11]).

Le principe de ce type de classification se base sur la technique du *multiplicateur de Lagrange*. plus particulièrement, on vise à minimiser une fonction de distances intra-classes avec des pondérations sur le degré d'appartenance d'un pattern à toutes les classes.

Puisque les fonctions objectives à minimiser de ces méthodes impliquent le degré d'appartenance et l'ensemble codevecteur, l'optimisation s'effectue par *la méthode itéra-*

1. La méthode *isodata* est une variante de la méthode des centres mobiles.

tive de Picard ([11]) où chaque itération est composée de deux étapes. Dans la première, un sous ensemble de variables (degré d'appartenance) est gardé inchangé et l'optimisation est effectuée sur le reste des variables (codevecteurs). Lors de la deuxième étape, on permute le rôle des sous ensembles de variables : inchangées et changées. L'algorithme d'optimisation est stoppé lorsque les variables changent d'une manière inférieure à un seuil fixé.

Fuzzy c-means L'algorithme de fuzzy c-means ([11]) identifie les classes comme des fuzzy ensembles. il minimise la fonction :

$$J(U,V) = \sum_{h=1}^{n}\sum_{i=1}^{c}(u_{ih})^m \|x_h - v_i\|^2 \tag{1.6}$$

par rapport à la matrice de degré d'appartenance U et le codebook V avec les pondérations probabilistique

$$\sum_{i=1}^{c}(u_{ih}) = 1 \,\forall i = 1,..,n. \tag{1.7}$$

le paramètre m contrôle la fuzziness du degré d'appartenance et souvent, il se fixe à deux. Pour des valeurs trop élevées, les degrés d'appartenance seront mis égales. Alors que pour des valeurs proche de un, on retrouve l'algorithme de k-means.

La minimisation de l'équation précédente est effectuée en utilisant la fonction de Lagrange.

Possibilistic clustering En fait, cette notion est beaucoup plus similaire à celle de fuzzy clustering, elle la régularise tout simplement. Par cette notion, nous entendons une relaxation des degrés d'appartenance d'un objet à l'une des classes cachées dans les données brutes. Dans cet esprit, les outliers d'un ensemble de données vont avoir un degré d'appartenance bas. En revanche, dans le cas d'un ensemble de données avec des classes enchevêtrées, un objet aurait un degré d'appartenance trop élevé à plus qu'une seule classe.

Donc à l'instar de la terminologie fuzzy c-means, sa version ainsi régularisée est baptisée *possibilistic c-means*. Deux formulations de cette dernière existent. Elles sont similaires, nous nous contentons donc d'en rapporter une.

Cette méthode *possibilistic c-means* propose donc de minimiser la fonction objective suivante qui prend en considération la matrice U de degré d'appartenance et le codebook

$V = \{v_1, ..., v_c\}$:

$$J(U,V) = \sum_{h=1}^{n}\sum_{i=1}^{c}(u_{ih})^m \|x_h - v_i\|^2 + \sum_{i=1}^{c}\eta_i \sum_{h=1}^{n}(1-u_{ih})^m \tag{1.8}$$

Le paramètre η_i est un coefficient régularisant et il est en relation avec la largeur des classes . Les auteurs [92] proposent de l'estimer par :

$$\eta_i = \gamma \frac{\sum_{h=1}^{n}(u_{ih})^m \|x_h - v_i\|^2}{\sum_{h=1}^{n}(u_{ih})^m} \tag{1.9}$$

qui est une moyenne pondérée de la distance intra-classes de la $i^{ème}$ classe. γ est une constante qui est typiquement fixée à un.

le reste est le même que la méthode fuzzy c-means.

1.3.2.2 Méthodes non-linéaires

Approche à noyaux Ces approches sont capables de produire des hyperplans non linéaires de séparation entre les classes. Les méthodes à noyaux sont dans la plupart des cas des extensions des méthodes classiques telles que k-means, fuzzy c-means, les cartes auto-adaptatives.. et bien d'autres.

L'introduction des noyaux dans ces méthodes classiques veut dire représenter les données brutes dans un nouvel espace d'attributs de sorte que la définition d'un hyperplan linéaire dans ce nouvel espace implique la définition d'un hyperplan non-linéaire dans l'espace initial.

Dans le domaine de l'apprentissage machine, l'utilisation des fonctions à noyau ([112]) a été introduite par Aizerman et al ([3]). En 1995, Cortes et Vapnik ont introduit la méthode des *Séparateurs à Vaste Marge* (Support Vector Machines, SVM) ([31]) qui a fonctionné mieux que d'autres algorithmes de classification par rapport à plusieurs problèmes [1]. Le succès des SVM a permis d'étendre l'utilisation des noyaux à d'autres algorithmes d'apprentissage.

Avant d'aborder un exemple de ces méthodes, nous donnons la définition d'un *noyau de Mercer*.

Mercer kernels Nous rappelons la notion du *noyau de Mercer* et, pour simplifier, nous prenons des vecteurs dans $\mathbb{R}^d$ au lieu de $\mathbb{C}^d$ ([135]).

Définition 2 (Noyau de Mercer) *Soit* $X = \{x_1, ..., x_n\}$ *un ensemble non vide tel*

1. Nous allons voir cette méthode dans la section de la classification supervisée. (cf. section 1.4.4).

que $x_i \in \mathbb{R}^d$. Une fonction $K : X \times X \to \mathbb{R}$ est appelée un noyau positif fini (positive definite kernel) ou noyau de Mercer si et seulement si K est symétrique (i.e. $K(x_i, x_j) = K(x_j, x_i)$. Et on a les équations suivantes :

$$\sum_{i=1}^{n}\sum_{j=1}^{n} c_i c_j K(x_i, x_j) \geq 0 \, \forall n \geq 2 \tag{1.10}$$

où $c_r \in \mathbb{R} \, \forall r = 1, .., n$.

Chaque noyau de Mercer peut s'exprimer comme :

$$K(x_i, x_j) = \phi(x_i).\phi(x_j) \tag{1.11}$$

où : $\phi : X \to F$ est une application de l'espace d'entrées X vers un espace d'attributs de dimensions plus élevées F. Un avantage dans les applications est qu'il est possible de calculer la distance euclidienne dans F sans pour autant connaître explicitement ϕ. Ceci s'effectue grâce à un outil appelé *astuce de la distance noyau* (*distance kernel trick*, [116]) :

$$\begin{aligned} \|\phi(x_i) - \phi(x_j)\|^2 &= (\phi(x_i) - \phi(x_j)).(\phi(x_i) - \phi(x_j)) \\ &= \phi(x_i).\phi(x_i) + \phi(x_j).\phi(x_j) - 2\phi(x_i).\phi(x_j) \\ &= K(x_i.x_i) + K(x_j.x_j) - 2K(x_i.x_j) \end{aligned} \tag{1.12}$$

Avec cette astuce, le calcul de la distance des vecteurs dans l'espace d'attributs est tout simplement une fonction des vecteurs d'entrée (input vectors). Afin de simplifier la notation, nous introduisons la matrice appelée *matrice de Gram* dont les éléments k_{ij} sont les produits scalaires $\phi(x_i).\phi(x_j)$. Ainsi l'équation 1.12 peut s'écrire :

$$\|\phi(x_i) - \phi(x_j)\|^2 = k_{ii} + k_{jj} - 2k_{ij} \tag{1.13}$$

Des exemples de la fonction de Mercer sont donnés par :

1. Linéaire :

$$K^1(x_i, x_j) = x_i.x_j \tag{1.14}$$

2. Polynomiale de degré p :

$$K^p(x_i, x_j) = (1 + x_i.x_j)^p, p \in \mathbb{N}. \tag{1.15}$$

3. Gaussienne :

$$K^g(x_i, x_j) = exp(-\frac{\|x_i - x_j\|^2}{2\sigma^2}), \sigma \in \mathbb{R} \tag{1.16}$$

Kernel K-means Étant donné un ensemble de données X , nous transformons nos données dans un espace attributs F, en utilisant une application non linéaire ϕ et nous considérons k centres dans l'espace attributs ($v_i^\phi \in F$ avec $i = 1, ..., k$) ([61]). Nous appelons l'ensemble $V^\phi = (v_1^\phi, ..., v_k^\phi)$ le codebook de l'espace attributs puisque dans notre représentation les centroides dans l'espace attributs jouent le même rôle que les codevecteurs dans l'espace origine. Par analogie avec les codevecteurs de l'espace origine, nous définissons pour chaque centre v_{i^ϕ} sa région de Voronoi et l'ensemble de Veronoi dans l'espace attributs. La région de Voronoi dans l'espace attributs R_i^ϕ du centre v_i^ϕ est l'ensemble de tous les vecteur dans F pour lequel v_i^ϕ est le vecteur le plus proche.

$$R_i^\phi = \{x^\phi \in F | i = \arg\min_j \|x^\phi - v_j^\phi\|\} \tag{1.17}$$

L'ensemble de Voronoi dans l'espace attributs π_i^ϕ pour le centre v_i^ϕ est l'ensemble des vecteurs x dans X tel que v_i^ϕ est le vecteur le plus proche de leurs images $\phi(x)$ dans l'espace attributs.

$$\pi_i^\phi = \{x \in X | i = \arg\min_j \|\phi(x) - v_j^\phi\|\} \tag{1.18}$$

l'ensemble des régions de Voronoi dans l'espace attributs définit un tissu de Voronoi de l'espace attributs.

Approche par graphe Graph clustering, qu'il ne faut pas confondre avec clustering of graphs, est une approche de classification automatique qui modélise les classes sous forme de graphe dont les nœuds sont les objet à classer. Bien que les deux sont en étroite relation, la différence c'est que le clustering of graphs vise à trouver une partition d'un ensemble de graphes et donc les regrouper dans des classes en se basant sur leur similarité. C'est une classification dont les objets sont des graphes.

Selon [138], la classification par graphes classifie les sommets d'un graphe en se basant sur la structure des arêtes du graphes. Le résultat de la classification devrait avoir la propriété que dans chaque classe, les sommets sont fortement connectés et que seulement quelques arêtes séparent les classes.

Les méthodes de classification par graphes se basent sur la théorie des graphes pour

formaliser le concept de classe ou tout simplement utilisent la structure des graphes pour représenter les données tout en développant une procédure heuristique pour les manipuler. [138] présente un review de ces méthodes en se focalisant sur les plus connues.

Parmi ces méthodes celles considérant la classification comme un problème de coupe de graphe et utilisant les concepts de l'analyse spectrale sont les plus connue. Nous explicitons davantage le principe de ces techniques. En considérant un ensemble de données $X = \{x_1, .., x_n\}$ à classer, nous pouvons construire un graphe $G(V, A)$ ayant un ensemble de nœuds $V = \{v_1, ..., v_n\}$ qui correspondent aux n objets et des cotes définis à travers une matrice d'adjacence (d'affinité) A. Cette matrice d'un graphe pondéré est donnée par la matrice dont les éléments a_{ij} sont données par les poids des cotes reliant le nœud i et j. Étant donné que G est un graphe non orienté, la propriété $a_{ij} = a_{ji}$ est toujours tenue. L'adjacence entre deux nœuds peut être définie comme suivant :

$$a_{ij} = \begin{cases} h(x_i, x_j) & \text{if} \quad i \neq j, \\ 0 & \text{otherwise.} \end{cases} \tag{1.19}$$

La fonction h mesure la similarité entre objets et elle est typiquement une gaussienne :

$$h(x_i, x_j) = exp(\frac{-d(x_i, x_j)}{2\sigma^2}) \tag{1.20}$$

avec d mesure la dissimilarité entre les objets et contrôle la rapidité de décroissance de h. Ce choix particulier a la propriété que A a tout simplement quelques termes différents de zéros.

La matrice D des degrés est la matrice diagonale dont les éléments sont les degrés des nœuds de G.

$$d_{ij} = \sum_{j=1}^{n} a_{ij} \tag{1.21}$$

Dans ce cadre, le problème de la classification, comme nous l'avons cité, se voit comme un problème de coupe de graphe [29] où on veut séparer un ensemble de nœuds $S \subset V$ de l'ensemble complémentaire $\overline{S} = V \setminus S$. Le problème de coupe de graphe peut se formuler de plusieurs façon selon le choix de la fonction à optimiser. Une de ces fonctions est donnée par la coupe [29] :

$$cut(S, \overline{S}) = \sum_{v_i \in S, v_j \in \overline{S}} a_{ij} \tag{1.22}$$

Il est facile de vérifier que la minimisation de cette fonction objective favorise les

partitionnements contenant des nœuds isolés. Pour atteindre un meilleur équilibre entre le cardinal de S et $\overline{S}$, il a été proposé d'optimiser la coupe normalisée ([141]) :

$$Ncut(S,\overline{S}) = cut(S,\overline{S})(\frac{1}{assoc(S,V)} + \frac{1}{assoc(\overline{S},V)}) \tag{1.23}$$

où l'association $assoc(S,V)$ est également connu comme étant le volume de S :

$$assoc(S,V) = \sum_{v_i \in S, v_j \in \overline{S}} a_{ij} \equiv vol(S) = \sum_{v_i \in S} d_{ii} \tag{1.24}$$

Il existe d'autres définitions de fonctions à optimiser telles que la conductance, l'association normalisée, le rapport de coupe (ratio cut) [39; 141].

L'optimisation de ces fonctions objectives a une complexité très élevée [46], et pour cette raison il a été proposé de les relaxer en utilisant des concepts spectraux relevant de l'analyse des graphes.

Cette relaxation peut être formulée par l'introduction de la matrice de Laplace [29] : $L = D - A$ qui peut être considérée comme un opérateur linéaire sur le graphe G.

Des exemples d'algorithmes de classification spectrale peuvent être trouvés dans [46; 118; 133]. Des résultats prometteurs de l'application de la théorie des graphes spectrale pour la classification de données peuvent être trouvé dans le domaine de la vision par ordinateur ([101; 141]) et dans la détection des groupes dans les structures de protéines ([81]).

Pour des lectures plus détaillés sur les méthodes spectrales, le lecteur pourrait consulter [46; 117].

Il est à noter que certains auteurs ont constaté que ces deux dernières approches se basent sur les mêmes fondements mathématiques [102; 118].

1.3.3 Approche probabiliste

L'approche probabiliste, plus récente comme outil de résolution de la problématique de classification, se base sur des modèles statistiques qui formalisent l'idée de classe et de données. Elle considère que l'ensemble de données $x_1, ..., x_n$ est une réalisation d'une variable aléatoire X et que également les classes à prédire $y_1, ..., y_n$ sont issues de la réalisation d'une autre variable aléatoire Y à valeurs dans l'ensemble $\{1, ..., k\}$. Nous introduisons également le vecteur aléatoire $S \in \{0,1\}^k$ tel que si $Y = j$ alors $S = (0, ..., 0, 1, 0, ..., 0)$ où le $j^{ème}$ terme vaut 1, et ce pour $j = 1, ..., k$.

Dire que la $i^{ème}$ observation x_i appartient à la $j^{ème}$ classe C_j est équivalent, dans ce contexte, à dire que x_i est une réalisation de X conditionnellement à $Y = j$.

Une autre notion à ne pas manquer dans ce cadre probabiliste est celui des données complètes. Par ce terme, nous entendons que le jeu de données en considération pour l'analyse est représentée par le couple $z = (x, y)$, c'est à dire on connait non seulement les observations x_i mais aussi leurs classes y_i ($z_i = (x_i, y_i)$).

Ceci est en principe dans le cas de la classification supervisée que nous verrons dans la section 1.4. Cependant, dans le cas non supervisée, nous pouvons aussi supposer, a priori , que nous connaissons les labels y_i des classes des observations, juste pour pouvoir concevoir un modèle. Dans ce cas, ces données seront appelées données manquantes. C'est à partir de ce modèle que les vraies classes vont être "estimées" finalement (i.e modèle de mélange 1.3.3.2).

En somme, le résultat de la procédure de classification selon cette approche s'interprète donc statistiquement et non plus d'une manière déterministe. Cela veut dire que la partition trouvée à la fin de la procédure de classification n'est pas unique et qu'elle a été obtenue avec une certaine probabilité. La question est alors comment choisir parmi un ensemble de partitions possibles de l'ensemble de données, celle qui a la probabilité maximale.

C'est ce que nous abordons dans les paragraphes suivants.

1.3.3.1 Maximum a posteriori

La méthode connue dans ce cadre probabiliste pour trouver cette partition maximale est connu sous le nom de règle de Bayes que nous notons δ^*. Cette méthode consiste à affecter une observation à la classe maximisant la probabilité a posteriori :

$$\delta^*(x) = \underset{j=1,\ldots,k}{\arg\max}\, P(Y = j|X = x) \tag{1.25}$$

Nous nous trouvons donc devant deux visions à considérer. D'abord, il y a les méthodes qui proposent de modéliser directement la probabilité conditionnelle $P(Y|X)$ donnant naissance à une conception discriminatoire. Ces méthodes sont non-paramétriques, elles ne supposent aucune distribution préalable du système. Citons, parmi les méthodes les plus utilisées : les méthodes d'estimation non-paramétriques de la densité, connues également sous le nom de méthodes à noyaux (ou fenêtres de parzen) ([41; 79]) et les méthodes utilisant les k-plus proches voisins ([47; 109]) [1].

La deuxième vision modélise dans un premier temps la distribution jointe $P(X, Y)$

1. Nous reviendrons à ces méthodes dans la section consacrée aux méthodes de classification supervisée 1.4.1.

et en déduit ensuite la règle de classification en utilisant la formule de Bayes :

$$P(Y|X) = \frac{P(Y)P(X|Y)}{P(X)} \propto P(Y)P(X|Y) \tag{1.26}$$

L'estimation par Maximum A Posteriori (MAP) permet alors, d'injecter dans les calculs les connaissances a priori concernant la distribution des donnée dans le système. Il est trés utilisé en modélisation probabiliste que ce soit pour l'estimation des paramètres (parameter estimation)

$$\hat{\theta} = \arg\max_{\theta} P(\theta|x) \tag{1.27}$$

ou pour la prédiction des données (data prediction)

$$\hat{y} = \arg\max_{y} P(y|x) \tag{1.28}$$

Dans ce qui suit, nous allons détailler la méthode intéressante et la plus utilisée dans ce contexte de la modélisation probabiliste qui est celle de modèle de mélange.

1.3.3.2 Modèle de mélange de lois

Le modèle de mélange (mixture model) se veut comme un moyen de modéliser des données complexes que les lois usuelles ne peuvent pas faire. Il est à la base de toute une famille de modélisation probabiliste du problème de la classification connu sous le nom de model-based clustering [50; 111].

D'une manière générale, le principe de cette modélisation considère que la population des données brutes $x_1, ..., x_n$ est un échantillon de n réalisations indépendantes du vecteur aléatoire X à valeur dans $\mathbb{R}^p$ dont la distribution est caractérisée par une densité de probabilité $f(x)$ formée d'un nombre déterminé ou à déterminer de composantes. Chaque composante est à son tour modélisée au travers d'une densité de probabilité conditionnelle $f_j(x)$. Donc , on aura la formulation suivante de cette distribution :

$$f(x) = \sum_{j=1}^{k} \pi_j f_j(x) \tag{1.29}$$

où k est le nombre de classes, que nous connaissons dans le cas de la classification supervisée 1.4, et nous l'ignorons dans le cas de la classification automatique. f_j est la densité de la distribution de X conditionnellement à $Y = j$ (de la $j^{ème}$ composante du mélange) et les π_j sont les proportions du mélange ($\pi_j \in [0, 1]$ et $\sum_{j=1}^{k} \pi_j = 1$).

De plus, on suppose généralement , pour rester dans la vision générique, que les densités f_j des classes appartiennent à une famille de lois paramétrées que nous pouvons donc noter $f_j(.) = f(., \theta_j)$.

Nous pouvons écrire donc le mélange de lois sous la forme :

$$f(x) = \sum_{j=1}^{k} \pi_j f(x, \theta_j) \tag{1.30}$$

où $\theta = \{\pi_1, .., \pi_k, \theta_1, .., \theta_k\}$ représente l'ensemble des paramètres du modèle.

En principe, les lois des composantes sont quelconques, cependant le modèle le plus utilisé est celui du mélange gaussien. Les densités $f(., \theta_j)$ des composantes de ce mélange sont des distributions d'une loi normale $N(\mu_j, \Sigma_j)$, de vecteur paramètres $\theta_j = \{\mu_j, \Sigma_j\}$.

Ainsi, nous remarquons que la maximisation de la probabilité a posteriori se ramène à une estimation des maximisation de la vraisemblance des données. Les approches traditionnelles pour estimer les paramètres d'un mélange visent à obtenir une estimation par la méthode du maximum de vraisemblance (MV) [78]. Aussi une autre méthode est la méthode par Espérance-Maximisation (Expectation-Maximzation) qui est une généralisation de l'algorithme du maximum de vraisemblance ([33]) pour les problèmes de données incomplètes.

Nous présentons dans les paragraphes qui suivent ces deux méthodes.

1.3.3.3 Maximum de vraisemblance

Nous introduisons la notion de log-vraisemblance que nous notons $L(y; \theta)$, ou dans la mesure où il ne court pas d'ambiguïté $L(y)$. La log-vraisemblance est la quantité calculée à partir des données complètes et qui s'écrit dans le cadre du modèle de mélange de lois :

$$log(L(\theta)) = \sum_{j=1}^{n} log(\sum_{i=1}^{k} \pi_i f(x_j, \theta_i)) \tag{1.31}$$

lorsque l'on connait les étiquettes de chaque observation, c'est à dire dans le cas supervisé, la log-vraisemblance peut s'écrire :

$$log(L(\theta)) = \sum_{i=1}^{n} \sum_{j=1}^{k} s_{ij} log(\pi_j f(x_i, \theta_j)). \tag{1.32}$$

avec $s_{ij} = 1$ si le $i^{ème}$ observation appartient à la $j^{ème}$ classe , et $s_{ij} = 0$ sinon.

La méthode du maximum de vraisemblance propose alors d'estimer le vecteur paramètre du modèle par θ^{MV} où :

$$\theta^{MV} = \arg\max_{\theta} log(L(\theta)), \tag{1.33}$$

Néanmoins, les inférences sur le modèle ne peuvent pas se faire directement à partir de la maximisation de la vraisemblance tant que nous ne connaissons pas les étiquettes des observations x [1]. En effet le nombre exponentiel de solutions à explorer fait que cette maximisation est malheureusement non abordable même dans le cas d'un ensemble de données limité [17].

Ceci dit, la méthode utilisée dans ce contexte pour calculer les inférences du modèle est l'algorithme nommé Expectation-maximization. Nous le détaillons dans le paragraphe suivant.

1.3.3.4 Expectation-Maximization

L'algorithme Expectation Maximisation n'est pas dédié exclusivement au modèle de mélange de lois. C'est un algorithme général d'estimation des paramètres. Il a été développé par [33]. Il consiste à maximiser itérativement l'espérance conditionnelle de la log-vraisemblance complète :

$$E[L(y;\theta|x,\theta^*] = \sum_{i=1}^{n}\sum_{j=1}^{k} t_{ij} log(\pi_j f(x_i;\theta_j)) \tag{1.34}$$

où $t_{ij} = E[S_j|X = x_i, \theta^*]$ l'espérance du $j^{ème}$ élément du vecteur aléatoire S conditionnellement à $X = x_i$ et à θ^*.

La procédure de maximisation est la suivante. Partant d'une solution initiale θ^0, on effectue deux étapes. La première étape, notée E-step, permet de calculer l'espérance de la log-vraisemblance complète $E[L(y;\theta|x,\theta^q]$ conditionnellement à la valeur courante θ^q du vecteur paramètre. La deuxième étape, M-step, consiste à maximiser $E[L(y;\theta|x,\theta^q]$ par rapport à θ pour obtenir une nouvelle valeur du vecteur paramètre.

L'algorithme consiste donc simplement à construire une suite $(\theta^{(q)})_{q\geq 1}$ qui vérifie :

$$\theta^{(q+1)} = \arg\max_{\theta} E[L(y;\theta)|x,\theta^q] \tag{1.35}$$

L'un des avantages du présent algorithme est que la suite ainsi générée fait croître la vraisemblance $L(x;\theta)$ et converge vers un maximum local de la vraisemblance sous

1. Nous reviendrons à cette méthode dans la section 1.4 relatif à la classification supervisée.

certaines conditions de régularité [110].

En pratique, l'algorithme à l'étape E et à l'itération q, se réduit au calcul de l'espérance de S_j sachant la valeur du paramètre estimé à l'étape précédente et que $X = x_i$:

$$t_{ij}^{(q)} = E[S_j|X = x_i, \widehat{\theta}^{(q-1)}] = P(Y = j|X = x_i, \widehat{\theta}^{(q-1)}) \tag{1.36}$$

puisque S_j est à valeurs dans $\{0, 1\}$.

Utilisant la formule du théorème de Bayes, nous pouvons finalement formuler les probabilités a posteriori t_{ij} de la façon suivante :

$$t_{ij}^{(q)} = \frac{\widehat{\pi}_j^{(q-1)} f(x_i, \widehat{\theta}_j^{(q-1)})}{\sum_{l=1}^{k} \widehat{\pi}_l^{(q-1)} f(x_i, \widehat{\theta}_l^{(q-1)})} \tag{1.37}$$

Arrivant maintenant à l'étape M : l'algorithme met à jour les paramètres du modèle en maximisant , à l'itération q, l'espérance de la log vraisemblance complète conditionnellement aux $t_{ij}^{(q)}$.

Les proportions du mélange sont obtenues simplement par la relation :

$$\widehat{\pi}_j^{(q)} = \frac{n_{ij}^{(q)}}{n} \tag{1.38}$$

où $n_j^{(q)} = \sum_{i=1}^{n} t_{ij}^{(q)}$.

Les estimateurs des paramètres $\theta_i, ...,_n$ sont calculés ensuite par résolution des équations retenues de vraisemblance du modèle.

En appliquant ceci au cas particulier du mélange gaussien, les paramètres seront estimés alors par :

$$\widehat{\pi}_j^{(q)} = \frac{n_{ij}^{(q)}}{n} \tag{1.39}$$

$$\mu_j^{(q)} = \frac{1}{n_j^{(q)}} \sum_{i=1}^{n} t_{ij}^{(q)} x_i \tag{1.40}$$

$$\Sigma_j^{(q)} = \frac{1}{n_j^{(q)}} \sum_{i=1}^{n} t_{ij}^{(q)} (x_i - \mu_j^{(q)})(x_i - \mu_j^{(q)})^t. \tag{1.41}$$

1.3.4 Approche par grilles

Une autre distinctive approche du problème de la classification est donnée par l'utilisation de la notion de grille. Le mécanisme des méthodes de cette approche est de

diviser l'espace des données en plusieurs cellules formant ainsi une structure de grille. Les cellules sont en général des hypercubes dans l'espace (par exemple, des rectangles dans le cas de données bidimensionnelles).

L'idée est d'utiliser des statistiques sur la distribution des points dans chaque cellule pour classer les cellules de la grille plutôt que les données elles même. Ces statistiques vont etre par exemple la moyenne des données, la variance, la densité (nombre de point dans la cellule par rapport au volume..).

Le schéma général de ce type d'algorithmes peut se présenter selon les étapes suivantes :

1. Diviser l'espace en cellules ;
2. Supprimer les cellules de basse densité (selon un seuil préfixé) ;
3. Combiner les cellules adjacentes ayant une forte densité pour former des composantes connexes.
4. étiqueter les points suivant la cellule à laquelle ils appartiennent.

Ces algorithmes ont généralement un temps de traitement rapide puisqu'ils parcourent l'ensemble de données une seule fois, et ceci pour calculer les statistiques pour chaque cellule. Les performances de classification de ce fait dépendent totalement de la taille de la grille qui est généralement beaucoup moins que celle l'ensemble des objets à analyser [97].

Les algorithmes les plus connus dans la littératures sont STING [161],WaveCluster [140] et CLIQUE [2].

Dans ce qui suit nous exposons les démarches de l'algorithme STING.

STING est une technique de classification automatique basée sur une grille dans laquelle l'espace de données est divisé en cellules rectangulaires (à l'aide de latitude et longitude) et emploie une structure hiérarchique.

Cette structure hiérarchique se construit en commençant par le niveau premier ou tout l'espace est considéré comme une seule cellule appelée parent . Ensuite nous divisons cette cellule en quatre sous-cellules appelées fils pour former le deuxième niveau de la hiérarchie. Nous réitérons cette opération pour chacune des cellules fils jusqu'à ce qu'un critère d'arrêt soit rempli tel que le nombre minimale dans la cellule est inférieur à un seuil préfixé..

Les informations statistiques concernant les observations et les attributs sont calculés à l'intérieur de chaque cellule et stockés. Ces statistiques concernent le nombre

d'observations dans chaque cellule, leur moyenne, les deux valeurs extrêmes maximum et minimum des attributs dans la cellule et aussi le type de la distribution (loi de probabilité).

Il est à noter que les statistiques d'une cellule parent peuvent se déduire de ses cellules fils. De ce constat, les informations des bas-niveaux sont les premiers à calculer et enregistrer.

1.3.5 Approche *bio*-inspirée

Un nombre considérable d'algorithmes de classification de données s'inspirent de la nature. En effet, nous pouvons observer, dans la nature, des phénomènes physiques, biologique ou même ontologique(en relation avec ce qui est humain : sociologie, psychologie, pédagogie, communication...) qui produisent ou émanent naturellement des phénomènes d'apprentissage, de classification ou de regroupement autour d'un objet précis. Typiquement, la vie sociale et collective de certaines espèces vivantes peut inspirer certains auteurs pour concevoir des systèmes de classification et d'apprentissage.

Dans ce qui suit, nous présentons certaines méthodes se basant sur ces constats.

Méthodes s'inspirant des essaims Une branche de ces techniques sont les méthodes désignées par méthodes de classification basée essaims (swarm-based clustering, SBC). Effectivement, la vie sociale des insectes tels que les fourmis, les abeilles ou les termites nécessite un travail collective pour résoudre des problèmes de leurs vies . L'incroyable et simples capacités et habiletés sociales de ces insectes constituent une source d'inspiration pour plusieurs auteurs.

Du fait, Des entomologistes ont observé que les fourmis sont capables de construire des nids complexes, d'effectuer le tri du couvain et de former des cimetières pour les fourmis mortes. Les systèmes développés en essayant d'emprunter de ces constats sont formés de simples et autonomes agents. Ils sont pourtant robustes et sont capables de s'adapter aux changements environnementaux, et généralement présentent un effet de regroupement sophistiquée [15]. Ces méthodes ont la possibilité de classifier un ensemble de données sans connaissance préalable du nombre de classes.

Le premier de ces travaux à partir duquel les autres étaient étendus, revient aux auteurs [34] qui ont observé les fourmis et essayèrent de les imiter en concevant un système robotique capable de classer et regrouper des objets physiques suivant leur similarité. Ils s'étaient basés sur deux opérations "ramasser" et "jeter" un objet tout en se déplaçant suivant des directions aléatoires. Le ramasse et le jet sont effectués dans

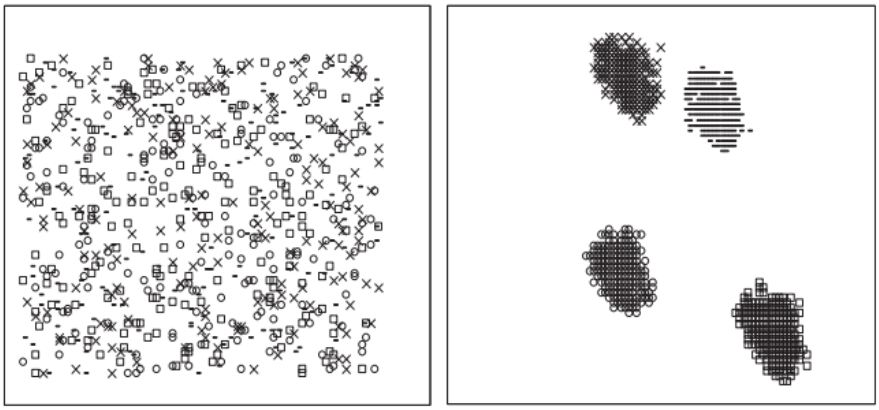


FIGURE 1.5 – Un exemple de classification automatique par une méthode de fourmis. A gauche, les données initiales ; A droite, les quatre classes émergentes (extrait de [152]).

le cas où l'objet en considération est différent, respectivement ressemblant, aux autres objets compris dans un voisinage prédéfini.

[100] étendirent ce modèle aux domaines de l'analyse de données numériques. Ils définissent une fonction de similarité entre une observation et les autres observations appartenant à son voisinage. Ensuite ils définissent deux probabilités p_{pick} et p_{drop} relatives aux opérations de ramasser et de jeter se basant sur cette mesure de similarité précédemment définie.

Réseaux de neurones La deuxième méthode qui s'inspire vaguement de la nature, plus précisément de la biologie, est donnée par la méthode des réseaux de neurones artificiels (Artificial Neural Networks (ANN)). Les auteurs de [68] ont fait une analogie entre un système de classification et un réseau de neurones dans le système nerveux humain.

Pour approcher cette analogie, un neurone biologique est une cellule qui se compose en :

- des points de connexions avec les autres neurones, appelés *les synapses*.
- les entrées du neurone, appelées *les dentrites*.
- la sortie vers d'autres neurones, appelée *axone*.
- le noyau dont le rôle est d'activer la sortie du neurone en fonction des *dentrites*.

Ainsi, le *neurone formel*, qui constitue l'élément fondamental d'un réseau de neurone artificiel est un modèle qui se caractérise par un état interne, $s \in S$, des signaux d'entrée $(x_1, .., x_p)$ et une *fonction d'activation* f. On a donc :

$$s = h(x_1, .., x_p) = f(\alpha_0 + \sum_{j=1}^{p} \alpha_j x_j)$$

α_0 est appelé le biais du neurone. La fonction d'activation est appliquée sur une combinaison linéaire des variables d'entrée. En plus de la structure du réseau et le niveau de complexité, un réseau de neurone dépend aussi de la nature de la fonction d'activation. Celle ci peut être par exemple une fonction linéaire, sigmoïde, un seuil ou stochastique.

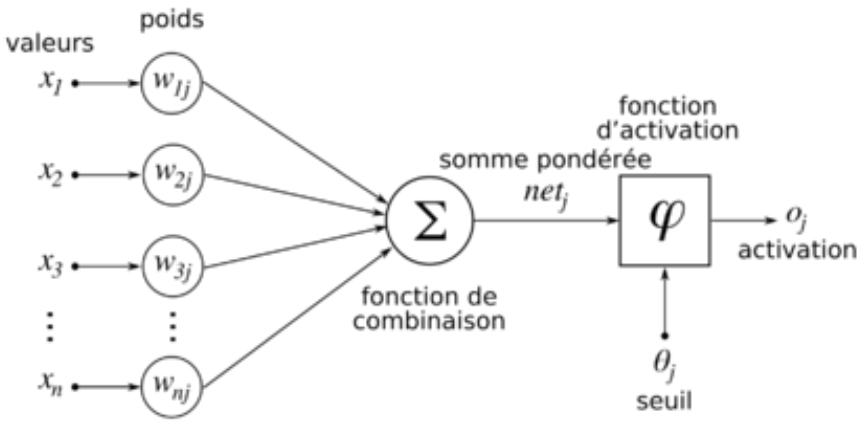


FIGURE 1.6 – Structure d'un neurone artificiel. Le neurone calcule la somme de ses entrées puis cette valeur passe à travers la fonction d'activation pour produire sa sortie.

Les méthodes basées sur les réseaux de neurones sont applicables autant pour des taches de la classification automatique que pour la classification supervisée [75; 139]. Nous reviendrons pour cette dernière dans la section 1.4 concernée par ce type de classification [1].

Les méthodes des réseaux de neurones destinées au contexte non supervisé sont très nombreuses : la méthode de l'apprentissage compétitive (*winner–take–all* [76]), la mémoire bidirectionnelle associative et adaptative, la quantification des vecteurs d'apprentissage de Kohonen (*Kohonen's learning vector quantization* (LVQ)), le modèle

1. Les réseaux de neurones peuvent aussi être employés comme outils puissant pour faire de la sélection d'attributs (*feature selection*)ou pour traiter des problèmes d'optimisation. en sélection d'une manière adéquate les poids des neurones. Cependant, ceci déborde de notre cadre.

de la théorie des résonances adaptatives (*adaptive resonance theory models* [25]).. pour que nous n'en citions que celles-ci. Cependant les plus fréquemment rencontrées dans la littérature restent les cartes auto adaptatives (*self-organizing map* (SOM) [89]).

Ces méthodes sont basées sur une structure en une seule couche. Les observations sont présentées aux nœuds d'entrée et regroupées par les nœuds de sortie du réseau. Les poids entre ces nœuds, d'entrée et de sortie, sont mis à jours pendant certaines itérations jusqu'à la satisfaction d'un critère d'arrêt. C'est cette procédure de mise à jour des poids d'interconnexions qui permet la faculté d'apprentissage au réseau.

Cartes auto-adaptatives (SOM) Les cartes auto-adaptatives (*Self-Organizing Maps*) forment une classe des méthodes de réseaux de neurones artificiels [1]. Le réseau de telles méthodes est représenté sous une structure d'une grille souvent uni- ou bidimensionnelle ([90]). Chaque nœud de cette grille est un neurone qui est référencé par un vecteur appelé vecteur référent pris de l'ensemble des données à analyser et qui en représente un groupe ou une classe. Ces cartes permettent alors une visualisation en dimensions multiples de données croisées. Autrement dit, la méthode est efficace pour donner une *quantification vectorielle* (discrétisation) de l'espace.

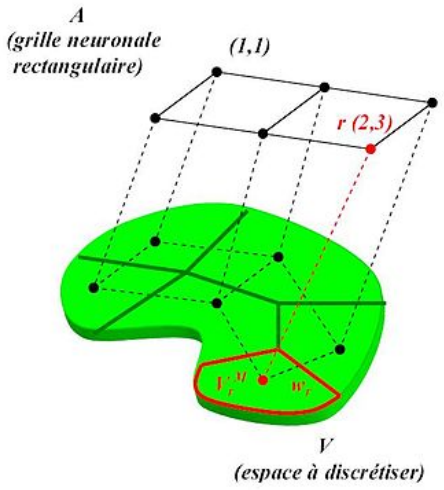


FIGURE 1.7 – Un exemple d'architectures des cartes auto-organisatrices.

Le principe de construction de cette grille représentative des classes sous-jacentes aux données est quasi similaire à la méthode des k-means. En effet, après une initiali-

1. Il est à noter qu'une formulation de ces méthodes par un tissu de Voronoi peut aussi etre adaptée.

sation aléatoire des valeurs des neurones du réseau (nœuds de la grille), les instances sont présentées une à une à la carte. Le neurone qui répond le mieux au *stimulus* (le vainqueur) est modifié de valeur pour qu'il réponde encore mieux à un autre *stimulus* qui aurait des attributs similaires. C'est ce qui est semblable au k-means. Cependant, dans le cas des cartes auto-adaptatives, tous les autres neurones voisins du vainqueur vont être aussi modifiés de valeurs avec un facteur multiplicatif du gain inférieur à un. Cette procédure itérative est arrêtée une fois les neurones ne bougent plus ou très peu.

Méthodes évolutionnistes [1]

Ces méthodes se sont inspirées de certains phénomènes biologiques concernant l'évolution naturelles des espèces vivants. Elles utilisent les opérations de la *théorie de l'évolution* et une population de *solutions* pour obtenir une partition globale et optimale des données. En effet, les éléments de base des organismes vivants sont les cellules et chaque cellule a un noyau qui contient un ensemble d'entités appelées *chromosomes*. Ces derniers sont des *chaines*, des *séquences d'ADN*. L caractère de cette chaîne est appelé *gène*. Ce sont donc les opérations issues de la *théorie de l'évolution* qui permet de comprendre et de modéliser l'évolution de l'ensemble de ces *chromosomes* (*génome*).

Par analogie, certains auteurs ont développé une modélisation du problème de la classification automatique en se basant sur cette théorie. Les clusters qu'ils souhaitaient obtenir ont été considérés comme des *chromosomes*. A partir d'une population générée aléatoirement, de nouvelles populations sont générées d'une manière itérative en appliquant les opérateurs génétiques d'évolution. Ces derniers sont donnés souvent par la *sélection*, la *combinaison* et la *mutation*. Une fonction objective à optimiser est employée pour évaluer la probabilité qu'un chromosome donné subsiste dans la prochaine génération.

La méthode la plus connue est celle baptisée *Algorithme Génétique* (*Genetic Algorithms*, GAs) et est développée par ([62; 70]). Les solutions utilisées dans ce type de méthodes sont des *séquences*, i.e des chaines de caractères binaires (*bits*).

Autres inspirations similaires :

Recuit Simulé : L'algorithme du recuit simulé (de l'anglicisme *Simulated Annealing algorithm*) est aussi inspiré d'un phénomène physique appartenant à la métallurgie. En effet, les physiciens utilisent pour certains métaux, la température pour les chauffer jusqu'à ce qu'ils deviennent liquides. Ainsi ils peuvent par la suite leur donner la forme

1. Pour plus de détails, le lecteur pourrait se rapporter à l'article de [77].

qu'ils veulent grâce à une procédure de refroidissement. Toutefois, ces physiciens ont remarqué que si la température n'est pas bien gérée, la procédure de refroidissement peut ne pas ramener le métal à l'état souhaité. Ainsi, ils ont proposé une méthode permettant d'éviter les états non désirés et aboutir à un état optimal. Il s'agit de la méthode de recuit. Cette méthode sera transposée vers le domaine d'optimisation afin de trouver les extrema globaux d'une fonction.

Donc étant donnée une fonction objective à optimiser, le recuit simulé ([86]) est désigné pour s'affranchir des optima locaux et converger ainsi vers l'optimum global. Ceci est du au fait qu'il accepte avec une probabilité une nouvelle solution pour l'itération suivante même si cette solution dégrade la fonction objective. Cette probabilité d'acceptation est contrôlée par un paramètre appelé *température* (par analogie avec le recuit des métaux), et est typiquement donnée en terme d'une température initiale T_0 et une température finale T_f. Cet algorithme est appliqué au clustering et il est discuté dans [87].

Nous reviendrons à cet algorithme dans le chapitre 2.3.2 où nous l'avons rapporté pour étudier l'optimisation des méthodes de simulation des Chaines de Markov par Monte Carlo (MCMC).

Toujours dans les inspirations, nous pouvons citer également la méthode dite *Super para-magnetic clustering* qui s'inspire des propriétés physiques d'un modèle ferromagnétique hétérogène [14]. Une autre méthode a aussi été proposée, elle se baptise classification par *passage des messages* (*affinity propagation clustering*) [52].

1.3.6 Approche basée sur la densité ou les régions modales

Les méthodes que nous avons regroupées dans cette approche utilisent la notion de densité ou de mode afin de trouver une *meilleure* partition de l'ensemble de données en classes homogènes. Nous avons commencé par présenter un premier type de ces méthodes qui manipulent cette notion de densité d'une manière presque algorithmique, puis nous abordons le deuxième type de ces méthodes qui, quant à elles, spécifient beaucoup plus la notion de densité pour la remplacer par une autre notion qui est celle de *région modale* ou tout simplement *mode*.

1.3.6.1 Méthodes basées sur la densité

Les notions de densité et de fonction de densité sont aussi utilisées par certaines méthodes pour retrouver les classes dans un jeux de données. A la différence des méthodes de l'approche par grilles, cette approche utilise la notion de densité dans l'espace entier

en définissant un voisinage pour chaque objet. Les classes sont modélisées comme étant des régions de densité élevée qui sont séparées par des régions de faible densité. Cette notion de densité est ici prise dans le sens du nombre d'objets par rapport au volume délimité par le voisinage défini.

Dans la littérature, nous identifions deux catégories de méthodes qui s'inscrivent toutes les deux dans cette approche de classification par densité. Il s'agit des méthodes se basant sur la *connexité de la densité*, et des méthodes se basant sur une *fonction de densité*.

Plusieurs méthodes et techniques ont été développées dans ce contexte, nous nous contentons d'aborder une pour chaque catégorie. Nous avons choisi la méthode DBSCAN [44] pour la première, et la méthode DENCLUE [69] pour la deuxième.

DBSCAN La procédure de classification par l'algorithme DBSCAN commence par la fixation de deux paramètres globaux, à savoir ϵ et $minPnt$. Le premier paramètre contrôle la valeur du rayon qui va servir pour déterminer le voisinage dans lequel la densité va être calculée. Le deuxième est un paramètre qui indique le nombre minimal d'objets qu'un objet déterminé devrait avoir dans son voisinage.

Une région de l'espace est censée représenter une classe si un voisinage contient plus d'objets que $minPnt$. La classe est alors entièrement identifiée en combinant les voisinages connexes qui remplissent cette caractéristique.

DENCLUE (DENsity CLUstEring) Cet algorithme est en fait une généralisation du précédent. En effet, au lieu de considérer la densité sur un voisinage de chaque objet, on définit une fonction de densité associée à l'ensemble de données à analyser. Les classes vont donc être identifiées en cherchant les zones de l'espace où cette fonction de densité est supérieure à un seuil préfixé.

La fonction de densité est elle-même définie comme étant la somme d'une *fonction d'influence* que l'algorithme définit pour chaque objet. Cette fonction d'influence mesure l'impacte de l'objet sur ses voisins. Typiquement, la fonction d'influence est symétrique (i.e. elle est la même dans toutes les directions) et sa valeur (contribution) est inversement proportionnelle à la distance du point au centre.

La procédure de classification par l'algorithme DENCLUE s'effectue en deux étapes. D'abord une étape de pré-traitement suivie d'une étape de classification. Dans l'étape de pré-traitement, une grille pour les données est construite en divisant l'hyperrectangle global en petits hyperrectangles avec une longueur d'arêtes égale à σ^2. Seules les cellules occupées par des objets sont numérotées par des clefs à partir de l'origine de l'espace

et gardées dans un arbre de recherche pour en faciliter l'accès.

Dans l'étape de classification, la fonction d'influence est calculée pour chaque objet x, tout en ne considérant que les points encerclés par les cellules voisines de x. L'algorithme rejette les classes associées à un attracteur de densité dont la valeur est inférieure à ξ. Par contre, il fusionne celles qui peuvent être joints par un chemin formé de points et qui ont une densité plus grande que ξ.

1.3.6.2 Méthodes basées sur la détection des modes

Le diviseur commun entre les méthodes que nous allons aborder maintenant est la notion de *régions modales*, appelées aussi *modes*. Les objets ou observations situées dans ces régions sont désignés par *observations prototypes* ou tout simplement *prototypes*.

A l'inverse des méthodes à densité citées dans la section précédente qui désignent par *densité* le sens simple qui est celui du nombre d'objets par rapport à un volume, les méthodes à base *détection des modes* la considèrent dans le sens d'une *fonction de densité de probabilité* (fdp) sous-jacente à la distribution de données. Les modes ou les régions modales sont, à force de cette considération, des zones de l'espace où cette fonction de densité de probabilité est maximale.

La stratégie de ces méthodes consiste alors à réduire la tache de la classification à la détection des modes de la fonction de densité de probabilité. En effet, ce sont ces modes qui représentent les classes via les observations prototypes. Autrement dit, une fois ces prototypes sont détectés, la classification finale de l'ensemble des données sera ramenée à un problème de classification supervisée (cf. section 1.4).

Il est à noter que le fait que ces méthodes emploient la fonction de densité de probabilité sous-jacente à la distribution de données ne signifie pas qu'elles adoptent toutes les outils du cadre probabiliste. Bien au contraire, la multitude de ces méthodes émane de la multitude des angles de vision que chaque méthode adopte pour la notion de mode.

En outre de la notion commune à ces méthodes que celle de la détection des modes, elles se partagent également le trait qu'elles sont des adaptations des méthodes développées pour des objectifs relevant du domaine de la segmentation d'images, exception faite de la méthode *mean-shift*.

Les paragraphes suivants détaillent un ensemble de ces méthodes.

Mean-shift : L'algorithme *mean-shift* est une technique de classification non-supervisée non-paramétrique. Cet algorithme dispose d'un cadre théorique solide et il est capable de détecter automatiquement les classes sans en connaître préalablement le nombre. Il

a également un autre avantage, c'est qu'il s'adapte à plusieurs structures de données, notamment les données non-linéairement séparables.

Le principe de cet algorithme est le suivant. Compte tenu de n points (observations) $x_i, i = 1, .., n$ pris sur un espace de dimension $\mathbb{R}^d$, l'estimation de la fonction de densité de probabilité par un noyau multivarié $K(x)$ et de pas de fenêtre h est donnée par :

$$f(x) = \frac{1}{nh^d} \sum_{i=1}^{n} K(\frac{x - x_i}{h}) \tag{1.42}$$

Pour les noyaux sphériques, il suffit de définir le profil du noyau $k(x)$ satisfaisant :

$$K(x) = c_{k,d} k(\|x\|^2) \tag{1.43}$$

où $c_{k,d}$ est une constante de normalisation qui permet à l'intégrale de $K(x)$ d'être égale à l'unité. Les modes de la fonction de densité de probabilité sont identifiés en cherchant les solutions pour lesquelles son gradient s'annule, c'est à dire $\bigtriangledown f(x) = 0$.

Le gradient de l'estimateur de la densité est donnée par :

$$\begin{aligned} \bigtriangledown f(x) &= \frac{2c_{k,d}}{nh^{d+2}} \sum_{i=1}^{n} (x_i - x) g(\|\frac{x - x_i)}{h}\|^2) \\ &= \frac{2c_{k,d}}{nh^{d+2}} \left[\sum_{i=1}^{n} g(\|\frac{x - x_i)}{h}\|^2)\right] \left[\frac{\sum_{i=1}^{n} x_i g(\|\frac{x - x_i)}{h}\|^2)}{\sum_{i=1}^{n} g(\|\frac{x - x_i)}{h}\|^2)} - x\right] \end{aligned} \tag{1.44}$$

où $g(x) = -k'(s)$.

Le premier terme est proportionnelle à l'estimation de la densité calculée à x avec le noyau :

$G(x) = c_{g,d} g(\|x\|^2)$. Le second terme

$$m_h(x) = \frac{\sum_{i=1}^{n} x_i g(\|\frac{x - x_i)}{h}\|^2)}{\sum_{i=1}^{n} g(\|\frac{x - x_i)}{h}\|^2)} - x \tag{1.45}$$

est le mean-shift.

Le vecteur de mean-shift pointe toujours vers la direction où la fonction de densité de probabilité est maximale. La procédure de mean-shift réitère à chaque itération les deux opérations suivantes :

- calcul du vecteur mean-shift $m_h(x^t)$

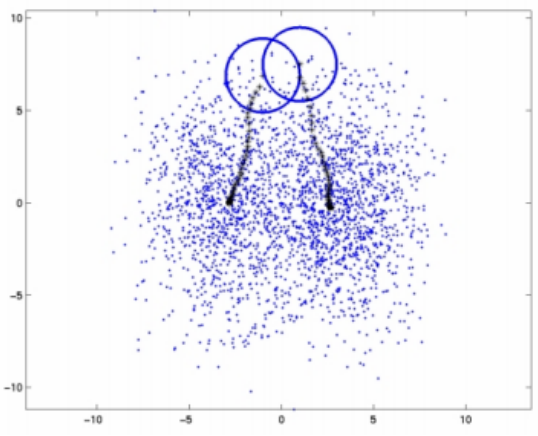


FIGURE 1.8 – Recherche des modes par la méthode de mean-shift.

- translation de la fenêtre à $x^{t+1} = x^t + m_h(x^t)$

Cette procédure converge donc vers les régions modales de la fonction de densité de probabilité.

L'ensemble formé des points convergeant vers le même mode définit le bassin d'attraction de ce mode. Ceux qui sont dans le même bassin d'attraction sont affectés à la même classe.

Nous renvoyons le lecteur à [30] pour plus de détail sur cette méthode.

Méthodes étendant les techniques d'images Tout en cherchant à développer de nouvelles méthodes de classification automatique, certains auteurs ([16]), ont constaté une analogie entre les opérations de segmentation d'images et les taches relevant de la classification d'un ensemble de données. En effet, les fondements de la segmentation dans un contexte purement d'imagerie, peuvent être résumés dans les éléments suivants :

- Chaque pixel de l'image doit appartenir à une région,
- Les régions doivent être composées de pixels adjacents,
- Les pixels constituant une région doivent partager une propriété de similitude.

Ces trois éléments peuvent être facilement étendus pour le contexte général de la classification automatique de données multidimensionnelles. En effet, chaque point de l'ensemble de données à analyser doit être affecté à une classe, et les classes doivent être

composées de points voisins. Et ensuite, les points attribués à la même classe doivent partager certaines propriétés de similitude.

Cette analogie a largement inspiré ces auteurs. Ils considèrent ainsi que les opérations effectuées sur la fonction de densité, contrôlant la distribution des intensités des niveaux de gris dans les problèmes de segmentation d'images, peuvent aisément être importées et appliquées pour la fonction de densité de probabilité sous-jacente à la distribution de données à analyser. Il est à assumer dans ce cadre l'hypothèse selon laquelle chaque mode de cette fonction de densité de probabilité (fdp), que nous cherchons à estimer, représente une classe cachée dans les données.

L'idée d'utiliser une estimation de la fdp pour la recherche des modes remonte aux années 60 ([121]). Dans sa version la plus simple, on utilisait juste une technique de seuillage comme dans le cas de la binarisation d'une image [162].

Des modélisation beaucoup plus sophistiquées ont été ensuite proposées. Une de ces méthodes a formulé la notion de mode comme étant des maxima locaux de la fdp. Afin d'identifier ces modes, les auteurs utilisent des opérateurs différentielles [54] ou des réseaux de neurones à apprentissage compétitif [166]. Ces méthodes sont toutefois très sensibles aux irrégularités locales de la distribution des données. Afin de remédier à cet inconvénient, un autre modèle a ensuite été mis en place sur la base de l'analyse des propriétés de convexité de la fdp [160]. Selon ce point de vue, les modes sont des zones concaves de cette fonction et les outils mathématiques utilisés sont des opérateurs d'intégration.

Dans l'article [136], les auteurs ont abordé cette question de la détection des modes en utilisant les concepts des *statistiques d'ordre*. Il ont appliqué des transformations adaptatives à la fdp et ont ensuite utilisé l'opérateur gradient pour en extraire les modes.

L'adaptation des techniques utilisées dans le domaine de l'analyse d'images a été poursuivie en utilisant les méthodes basées sur la relaxation [154], la détection de contours [155] et la morphologie mathématique [123]. Plus récemment, une modélisation des modes de la fdp par les champs de Markov a été effectuée [115].

1.4 Classification supervisée

Les systèmes doués d'une intelligence artificielle se basent sur l'utilisation des techniques d'apprentissage machine pour pouvoir se doter de la faculté de prise de décisions adaptables aux différentes situations envisagées. L'apprentissage machine quant à lui est effectué essentiellement grâce à un ensemble de méthodes, de techniques et d'algo-

rithmes faisant parties de la discipline du *data mining*. Au cœur de ce dernier figurent les différentes méthodes du champ l'analyse de données, comprenant à son tour les méthodes de classification supervisée et non supervisée.

Nous parlons dans cette section du deuxième aspect de la classification dans le domaine de l'analyse de données, c'est celui de la classification supervisée. Comme nous l'avons cité dans l'introduction de ce chapitre, cet aspect concerne le cas où l'analyste ou le système intelligent dispose des connaissances a priori sur une partie des données. Cette partie est souvent connue sous le nom de *training set*. L'objectif n'est plus de rechercher les classes ou de regrouper les données dans des classes, mais plutôt d'apprendre à partir des données dont les classes sont connues, comment affecter de nouvelles données non encore étiquetées à l'une des classes. Autrement dit, comment construire une règle d'affectation ou un modèle de prédiction en se basant sur un échantillon de données étiquetées. Avant de valider cette règle d'affectation sur de nouvelles données, la méthode doit être testée sur une autre partie de données appelée *test set*.

Plus formellement et en restant encore dans un cadre général, le principal objectif d'une méthode de classification supervisée est de prédire la valeur d'une variable aléatoire $Y \in \mathrm{Y}$ en se basant sur un ensemble de variables explicatives $X \in \mathrm{X}$ [1]. Autrement dit, les méthodes de la classification supervisée cherchent à construire une fonction de prédiction $f : \mathrm{X} \to \mathrm{Y}; X \to f(X)$ choisie parmi un ensemble de toutes les fonctions possibles de prédiction $\mathrm{F} = \{f : \mathrm{X} \to \mathrm{Y}\}$. Pour construire une "bonne" fonction de prédiction nous devrions définir un critère de performance en l'occurrence une *fonction de perte* (*loss function*) que nous notons L. Cette fonction dépend de f, de X et de Y. Plus cette fonction est minimale, plus le classifieur est meilleur. En supposant l'existence d'un et d'un seul optimal classifier $f^* \in \mathrm{F}$, nous pouvons l'écrire :

$$f^* = \arg\min_{f \in \mathrm{F}} L(f, X, Y) \tag{1.46}$$

Cela est en théorie, cependant, en pratique cette optimisation n'est pas envisageable vu que l'ensemble F sur lequel nous voulons optimiser n'est pas définissable explicitement, il est de dimension inconnue. C'est pour cela que nous nous contentons d'optimiser sur une classe de fonctions de prédiction $C \subset \mathrm{F}$, par exemple la classe des modèles linéaires. Nous obtenons donc un optimal différent de f^* et que nous notons f^{**}. A son tour, cette restriction n'est pas non plus saisissable en pratique, tout ce que nous pouvons faire c'est de chercher cette fonction optimale sur un échantillon, de

1. Il est à noter que dans le cas où l'ensemble Y est discret, nous parlons de classification. Si par contre, Y est continu, souvent $\mathrm{Y} \subset \mathbb{R}$, nous parlons plutôt de *regression*.

fonctions de prédiction, de taille n. Cet échantillon, que l'on choisit empiriquement, est noté C ($C \subset F$).

La formule précédente devient :

$$\widehat{f_n} = \arg\min_{f \in C} \widehat{L}_n(f, X, Y) \tag{1.47}$$

$\widehat{f_n}$ étant la version empirique de la fonction optimale de prédiction f^{**}.

Nous sommes donc là en présence de deux types de perte de performance : le premier est relatif au choix de la restriction de l'ensemble F à l'une de ses classes ; ce type est dénommé *erreur de modélisation* (*modeling error* or *approximation error*). Le second type de perte concerne le choix de l'échantillon dans la classe, et c'est une perte d'ordre statistique que l'on appelle *erreur d'estimation*.

La littérature est, tout comme dans le cas de la classification non supervisée, tellement riche en matière de méthodes, techniques et algorithmes qui essayent tous de résoudre cette problématique de la classification supervisée. En conséquence, nous ne pourrons pas les rapporter d'une manière exhaustive dans ce mémoire. Nous continuons donc sur la même façon que dans la section précédente en présentant seulement les principales approches et familles de ces méthodes, tout en veillant à illustrer chaque approche par une description d'une ou de deux techniques qui soient parfois les plus connues auprès des chercheurs faisant partie de la communauté de l'analyse de données ou bien qui soient les plus récentes.

D'après notre lecture dans la littérature, nous pouvons subdiviser les différentes méthodes de classification supervisée selon les approches suivantes : l'approche statistique, l'approche logique, l'approche connexioniste et l'approche géométrique [1].

Avant de détailler dans la suite le principe de ces différentes approches ainsi que certaines de leurs méthodes, il est pertinent de noter qu'une méthode de classification supervisée, *classifieur* peut aussi appartenir à l'une des deux catégories : classifieurs à apprentissage paresseux (*lazy learning*) et classifieurs à apprentissage désireux (*eager learning*).

Un classifier du premier type ne fait que sauvegarder les données du training set et attend jusqu'à l'arrivée d'une nouvelle instance à classer pour qu'il commence à construire la règle d'affectation. Par conséquent, ces classifieurs prennent beaucoup moins de temps dans la phase d'apprentissage que dans la phase d'affectation. Il a

1. Nous rappelons que cette distinction entre les différentes approches ne permet pas de trancher sur l'appartenance d'une méthode forcément à une seule catégorie. Bien au contraire, il se peut qu'une méthode ait des traits relevant de plusieurs approches.

quand même l'avantage de pouvoir apprendre continuellement.

A l'inverse, un classifieur du deuxième type utilise les données d'apprentissage et construit un modèle général et une fonction d'affectation explicite avant qu'il soit sollicité. Il essaye de générer des règles et les généraliser le plus possible à tout l'espace de données. De ce fait, il prend beaucoup plus de temps dans la phase d'apprentissage que dans celle d'affectation d'un nouvel individu.

1.4.1 Approche statistique

Les méthodes de cette catégorie sont issues d'un développement statistique de la formulation précédente du problème de l'apprentissage supervisé. Elles constituent un domaine de recherche très proche aux autres méthodes de l'analyse de données telles que l'Analyse en Composantes Principales (ACP) et l'Analyse Factorielle (cf. Contexte et problématique). Nous parlons particulièrement des méthodes de l'analyse discriminante, la méthode de Bayes naïve.

L'approche statistique cherche à donner les condition suffisantes et nécessaires, sur la classe C des fonctions de prédiction, qui puissent garantir la convergence de l'estimateur $\widehat{f}_n$ vers la fonction optimale de prédiction f^*.

Dans le contexte de la classification (c'est à dire $Y \in [y_1, .., y_J]$), la fonction de perte est donnée par l'erreur de fausse classification, que nous exprimons ainsi :

$$L(f) = P[Y \neq f(X)]$$

Le modèle optimal est donc fourni par la règle d'affectation bayésienne :

$$f^*(X) = \arg\max_{j=[1,..,J]} P[Y = y_j | X = x]$$

Dans ce qui suit, nous considérons que la variable Y à prédire prend ses valeurs dans $[y_1, .., y_J]$ et elle est gouvernée par une loi de probabilité $\pi_1, .., \pi_J$ qui représentent les probabilités *a priori* des classes ou des groupes. Nous supposons aussi que les variables explicatives suivent une loi de densité conditionnelle : $f_j(X) = P[X|Y]$.

Par le théorème de Bayes, la probabilité conditionnelle *a posteriori* s'écrit :

$$P[Y|X] = \frac{P(Y).P[X|Y]}{P(X)}$$

Avec le principe des probabilités totales, $P(X) = \sum_{j=1..J} P(Y = y_j).P[X|Y = y_j]$ et elle ne dépend pas de la variable j, donc la règle de Bayes pour la classification

supervisée devient :

$$f^*(X) = \underset{j=[1,..,J]}{\arg\max}\, \pi_j . f_j(X) \tag{1.48}$$

Cette maximisation a donné lieu à deux types de solutions, la règle bayésienne paramétrique et celle non paramétrique.

1.4.1.1 Analyse discriminante paramétrique

Dans cette modélisation, la loi a posteriori des observations X connaissant la classe y_j est donnée par une loi normale $\mathsf{N}(\mu_j, \Sigma_j)$, sa densité est donc :

$$f_j(X) = \frac{1}{\sqrt{2\pi}(det(\Sigma_j))^{1/2}} exp[-\frac{1}{2}(x-\mu_j)'\Sigma_j^{-1}(x-\mu_j)]$$

Si l'on ne suppose pas d'hypothèse supplémentaire sur la loi de x, les matrices μ_j sont dépendantes de j , et nous nous retrouvons alors face à un critère d'affectation *quadratique* en x. Il est nécessaire d'estimer les paramètres de la loi. Ainsi, nous estimons μ_j par la moyenne arithmétique m_j de x et la matrice de covariance Σ_j par la covariance empirique S_j^* telle que :

$S_j^* = \frac{1}{n_j - 1} \sum_{i=1..n_j} (x_i - m_j)(x_i - m_j)'$. Ce cas là est connu dans la littérature par *l'hétéroscédasticité*.

Dans le cas où nous supposons que les classes ont des matrices de covariance égales, lequel cas est connu par *l'homoscédasticité*, le critère d'affectation est alors linéaire en x, et on a :

$$S_j^* = \frac{1}{n - J} \sum_{j=1..J} \sum_{i=1..n_j} (x_i - m_j)(x_i - m_j)' \tag{1.50}$$

1.4.1.2 Analyse discriminante non paramétrique

Le recours aux modèles non paramétriques permet entre autres de se débarrasser des contraintes et des hypothèses nécessaires pour les modèles paramétriques. De telles hypothèses et conditions sont d'autant plus difficiles à remplir que la tailles de l'échantillon est plus réduite. Ceci ne veut en aucun cas dire que la population étudiée ne dispose pas de paramètres. Tout au contraire, il se peut qu'il en y ait une infinité, et l'un des objectifs des méthodes non paramétriques est d'éviter pour une raison ou pour une autre de se soucier de ces paramètres. Au lieu de chercher à estimer une fonction

$y = f(x)$ ou une densité de probabilité par une loi connue dont on devrait estimer aussi les paramètres, on propose de l'approcher par une fonction $\widehat{f}$.

Dans le domaine de l'analyse discriminante, ces méthodes nous permettent d'estimer directement les probabilités *a posteriori* $f_j(x)$. Deux méthodes sont à considérer dans ce contexte :

Méthodes à noyau : Nous avons déjà vu dans la section précédente sur les approches de la classification automatique l'une des méthodes qui se basent sur l'utilisation de la notion de noyau. Il s'agit de la méthode *mean-shift* (cf. paragraphe 1.3.6.2). La notion de noyau utilisé dans l'analyse discriminante est presque la même sauf qu'ici on impose à ce noyau qu'il soit une densité de probabilité.

Les méthodes à noyaux sont donc utilisées pour trouver des estimations $\widehat{f_j}(x)$ non paramétriques des fonctions de densité de probabilité $f_j(x)$. Ces dernières sont multi-dimensionnelles (définies dans $\mathbb{R}^d$), donc de même le noyau K doit aussi l'être. On a dans ce cas :

$$\widehat{f_j}(x) = \frac{1}{n_j h^d} \sum_{j \in [1,..,J]} K(\frac{x - x_i}{h})$$

Méthodes des k plus proches voisins : La méthode des k plus proches voisins est la méthode la plus intuitive et simple parmi les autres méthodes de l'apprentissage supervisé, elle constitue un bon exemple de la famille des méthodes paresseuses.

Cette méthode consiste à :

- se fixer un entier k ($k \in 1, .., n$).
- calculer les distances $d(x, x_i)$ entre l'observation à classer et toutes les autres.
- retenir les k observations pour lesquelles ces distances sont les plus petites.
- compter le nombre de fois $k_1, .., k_j, .., k_J$ que chaque classe $y_1, .., y_j, .., y_J$ apparait entre ces k observations.
- estimer les densités conditionnelles de chaque classe par :

  $$\widehat{f_j}(x) = \frac{k_j}{kV_k(x)}$$

  avec $V_k(x)$ est le volume de l'ellipsoïde contenant les k observations.

1.4.2 Approche logique

Les méthodes de cette approche sont construites en effectuant une série de tests logiques (ou symboliques) sur les variables explicatives pour arriver à une décision concernant l'appartenance d'une instance à l'une des modalités ou classes de la variable à prédire. Les méthodes les plus connues dans ce contexte sont celles des arbres de décision et les méthodes basées sur les ensembles des règles [91] qui en sont des extensions.

1.4.2.1 Arbres de décision

Les arbres de décision sont une famille de classifieurs qui modélisent une hiérarchie de tests sur les valeurs présentées par les différents attributs. Ils s'emploient autant pour la classification où la variable à prédire est discrète que pour la régression dont la dite variable est continue. Bien qu'elles soient connues initialement sous la terminologie partitionnement récursif, ces méthodes sont depuis les améliorations considérables de Breiman et al. (1984) reconsidérées sous l'acronyme de CART pour désigner *Classification and Regression Tree* ([22]).

La principale idée derrière le développement des CART repose sur un partitionnement récursif de l'espace des attributs en un nombre donné d'hyperplans parallèles aux axes (rectangles dans le cas de la régression ou des ensembles de valeurs dans le cas de la classification).

Chaque nœud dans un arbre de décision représente un attribut d'un objets à classifier, et chaque branche représente la valeur que peut prendre ce nœud. Utilisant un classifieur en arbre de décision veut dire que chaque nouvelle instance devra passer tout d'abord par le nœud racine et remonte suivant l'une des branches de l'arbre en vérifiant à chaque fois un test relatif aux valeurs prises par chaque nœud. Cette remontée finira une fois arrivée à l'une des feuilles qui représentent les classes ou modalités de la variable cible que l'on cherche à prédire.

L'attribut qui sépare au "mieux" l'ensemble des données d'apprentissage serait le nœud racine de l'arbre. Autrement dit, il s'agit d'évaluer localement quel attribut apporte "le plus d'information" (ou encore qui est "le plus corrélé") au résultat à prédire. Pour quantifier cette qualification de "meilleure séparation", un critère à optimiser devrait être défini. Dans ce sens, de nombreuses méthodes ont été proposées que ce soit pour le cas de la régression ou pour le cas de la classification. Les plus fréquemment rencontrées dans la littérature sont le critère de minimisation de la variance pour le premier cas ; le taux d'erreur, l'indice de Gini ([22]) et l'entropie (Gain d'information,

[72]) pour le deuxième.

L'algorithme de base pour les méthodes des arbres de décision est un algorithme qui construit des arbres de décision selon le principe *diviser pour régner* de manière itérative et de haut en bas.

La version implémentée de cet algorithme et qui est la plus connue dans la littérature est la C4.5 ([128]). C'est une amélioration d'ID3 (du même auteur [127]), notamment en terme de simplification de l'implémentation.

1.4.2.2 Méthodes d'ensembles

Si l'un des avantages notables des arbres de décision est qu'ils permettent, grâce à leur présentation sous une forme graphique simple, une intuitive interprétation et une efficace aide à la décision, ils soufrent en revanche de certains inconvénients. En effet, en plus que la phase d'apprentissage de ce type de classifieurs est relativement lente, ils fournissent des règles de décision relativement instables. Ceci est du au fait que leur construction reste dépendante d'une part du choix du nœud racine, et d'autre part des éventuelles modifications au niveau de l'ensemble d'apprentissage.

Ceci dit, nous avons assisté au développement de certaines méthodes essayant de pallier à ces inconvénients. Ces méthodes sont dénommées *méthodes d'ensembles* (*ensemble classifiers* or *aggregation techniques*). Ces techniques se basent sur l'idée de combiner linéairement un ensemble de modèles (classifieurs) de sorte que le modèle obtenu soit plus stable que chaque modèle considéré à part.

Parmi ces méthodes, nous citons la méthode de *bagging* ([19; 20]). Aussi la méthode dite de *boosting* ([51; 168]) qui consiste à appliquer la méthode de discrimination de manière répétée sur l'échantillon d'apprentissage en donnant de plus en plus d'importance aux points mal classés. Une autre méthode est celle de *foret aléatoire* (*random forest* [21]).

1.4.3 Approche connexioniste

Cette approche regroupe les méthodes qui se basent sur les réseaux de neurones. Nous avons déjà vu dans la section précédente dédiée aux méthodes de la classification non-supervisée, précisément l'approche s'inspirant de la nature 1.3, que les réseaux de neurones artificiels peuvent être utilisés aussi pour des problèmes de la classification supervisée que pour celle non-supervisée. Dans ce contexte, nous avons suffisamment présenté le principe qui régit ces méthodes et nous avons aussi décrit leur architecture. Nous nous contentons dans ce paragraphe d'aborder directement une méthode

principale de l'utilisation de ces réseau pour la classification supervisée. Il s'agit de la méthode du *perceptron multi-couches*, bien qu'il en existe d'autres telles que ,ADALINE (*ADAptive LInear NEuron*), *Adaptive Heuristic Critic* (AHC), WINNOW ([98])..

1.4.3.1 Perceptron multi-couches

Pour faire de la classification supervisée avec une structure de réseau de neurones, nous lui présentons à la fois le vecteur attributs d'un objet ainsi que la valeur de sa classe (déjà nous nous disposons d'un ensemble de données d'apprentissage) et nous le forçons à converger vers cet état final précis. Le réseau doit alors reconfigurer ses poids d'interconnexions, les recalculer afin que la sortie qu'il donne corresponde bien à la sortie désirée.

Au cours des premières recherches menées sur les réseaux de neurones artificiels, la première structure élémentaire développée et qui était capable d'apprendre par expérience, était le *perceptron*([131]). Un *perceptron* est un neurone mono-couche linéaire. Il est constitué d'une seule couche qui fait la sommation des poids et produit une sortie binaire. La règle de mise à jour des poids w_i, du $i^{ème}$ nœud, utilisée est la règle dite de *Widrow-Hoff* : si la sortie y_i est égale à la valeur désirée d_i, le poids de connexion n'est pas modifié, sinon il sera modifié en fonction de la valeur de la donnée entrée :

$$w_i = w_i + \alpha.(d_i - y_i)$$

avec α une constante positive.

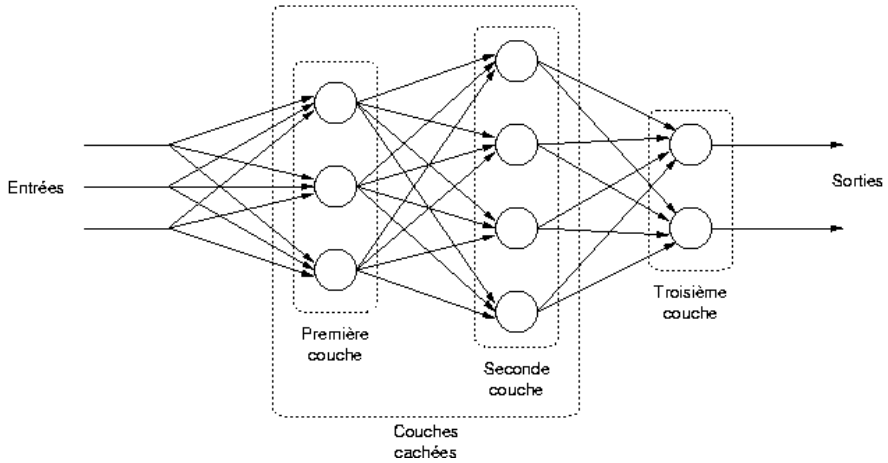


FIGURE 1.9 – Schéma d'un perceptron multi-couche.

Afin de remédier à l'inconvénient que ce modèle de base ne traitant que des données linéairement séparables, une amélioration a été faite en introduisant le *perceptron multi-couche* consistant en deux couches d'entrée et de sortie séparées par une ou plusieurs couches intermédiaires dites *couches cachées*. Pour que le perceptron multi-couche puisse traiter des données non linéaires, il faudrait qu'au moins les neurones d'une couche aient une fonction d'activation non-linéaire. Le cas échéant, le modèle se réduit à un perceptron mono-couche [1].

1.4.4 Approche géométrique

Nous avons déjà rencontré ce type d'approche dans la section précédente quand nous abordions les méthodes de la classification automatique. De même, dans le domaine de la classification supervisée, cette approche existe bel et bien, même si la seule méthode ou plutôt la famille de méthodes qui la représentent, à notre connaissance, est celle basée sur la notion de *Séparateurs à Vaste Marge* (SVM). Ces méthodes visent à construire,dans l'espace, un hyperplan qui sépare au mieux un ensemble de données en deux classes.

Nous décrivons avec plus de détails cette procédure géométrique.

1.4.4.1 Séparateurs à Vaste Marge

Les Séparateurs à Vaste Marge (ou *Machines à Vecteurs de Support*, traduit de l'anglais *Support Vector Machines* (SVM)), sont un ensemble de techniques de l'apprentissage supervisé dont l'objectif est de construire une fonction de prédiction. Elles sont une généralisation des classifieurs linéaires. Elles sont inventées par [31; 159] et disposent d'un fondement théorique solide.

Le principe de base des SVM repose sur l'utilisation de la notion de *marge optimale*. Il s'agit de la maximisation de la distance qui séparent l'hyperplan séparateur des observations (données) qui en sont les plus proches. De telles observations sont appelées les *vecteurs supports*. Le problème se ramène donc à trouver cet hyperplan optimal, à partir d'un ensemble d'apprentissage.

Formellement, étant donnée un ensemble $\{(x_i, y_i) \in (\mathbb{R}^p, \{-1, 1\}); i = 1..n\}$ un ensemble de vecteurs d'apprentissage formé de n observations étiquetées. Un séparateur à vaste marge linéaire SVM est un classifieur linéaire h qui affecte une observation x à la classe 1 si $h(x) \geq 0$ et à la classe -1 sinon ; Il est de la forme :

1. Il existe une famille de méthodes d'apprentissage machine qui s'était initialement basée sur la même structure que les réseaux de neurones avant qu'elle sera généralisée à toute structure composée de plusieurs couches. Il s'agit des méthodes dites *Deep learning* [9].

$$h(x) = sgn(w^T x + b)$$

avec $w \in \mathbb{R}^p$ et $b \in \mathbb{R}$ sont donnés par la résolution du problème quadratique suivant :

$$\begin{array}{ll} min_{w,b} & \frac{1}{2}\|w\|^2 \\ avec & y_i(w^T x_i + b) \geq 1 \quad i = 1, .., n \end{array} \tag{1.51}$$

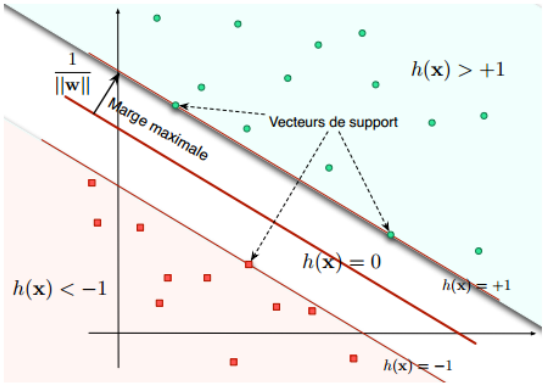


FIGURE 1.10 – Illustration de l'hyperplan séparateur d'un ensemble de donnée par la méthode SVM.

Les méthodes SVM se généralisent aux données non linéairement séparables en introduisant la notion de *fonction noyau*. Nous avons déjà vu la définition et l'utilisation de ces fonctions dans la section précédente pour traiter les problèmes non-linéaires (1.3.2.2). L'introduction de ces fonctions aux méthodes SVM ne pose aucune difficulté particulière. En utilisant la même procédure que dans le cas sans noyau, nous aboutissons à construire un séparateur linéaire dans le nouvel espace défini par la transformation noyau, et qui sera un séparateur non linéaire dans l'espace de départ.

1.5 Conclusion

Au cours de ce premier chapitre, nous avons passé en revue les grandes familles des méthodes et techniques qui ont abordé la problématique de la classification de données multidimensionnelles.

Comme nous l'avons vu, la disponibilité ou non des connaissances préalables sur

l'ensemble de données à analyser, est un facteur déterminant dans la différentiation du problème de la classification en deux branches majeurs : la classification supervisée et la classification non-supervisée. Bien que cette dernière semble être la plus difficile à approcher vu l'absence des informations sur les classes, leur nombre, leurs individus ou même sur tout cela, les deux divisions envisagent pour autant les mêmes défis et sont sujets aux même critères d'évaluation et de validation.

Nous avons essayé de regrouper les méthodes de chaque branche selon les différentes approches ou inspirations. Cette méthodologie nous a été dictée par la difficulté, voire l'impossibilité de rapporter toutes les méthodes proposées dans la littérature d'une part ; et d'autre part, par notre objectif de nous faciliter le repérage des méthodes que nous avons rencontrées tout au long de notre recherche à développer notre propre contributions au sujet. Dresser un état de l'art sur le sujet ainsi présenté nous a permis également d'approprier une source d'inspiration pour de prochains travaux.

Nous allons conclure ce chapitre en mettant l'accent sur les points suivants :

- D'abord, la littérature du domaine de la classification de données multidimensionnelles est très riche en méthodes, ce qui démontre à nouveau l'importance que lui a été accordée dans les communautés des chercheurs dans de nombreux domaines tels que l'analyse de données, l'intelligence artificielle et le traitement d'images.

- Aucune des méthodes développées n'est parfaite, chacune se réjouit de cotés positifs, comme elle soufre de cotés négatifs. D'où la nécessité des études comparatives entre les différentes méthodes afin d'en dégager les situations où chaque méthodes excelle.

- Ces comparaisons constituent la base de départ vers la recherche de nouvelles méthodes. Cette recherche prend la forme de l'amélioration d'une méthode existante, la combinaison entre plusieurs méthodes ou l'introduction d'une nouvelle approche en s'inspirant des avancées dans d'autres domaines scientifiques ou de phénomènes naturels.

Après ce chapitre sur l'état de l'art des méthodes de la classification de données et avant que nous passions à présenter notre contribution sur la détection des modes, nous introduirons et décrirons dans le chapitre suivant, les outils sur lesquels cette contribution sera basée. Il s'agit des *Processus Ponctuels Marqués* (PPM).

Chapitre 2

Processus Ponctuels : Formalisme et Applications

2.1 Introduction

Il s'agit dans ce chapitre de présenter une vue sur les fondements théoriques des processus ponctuels. Nous commençons par une section 2.2 où nous présentons les définitions de ces processus ainsi que leurs extensions ; nous y abordons aussi comment construire ou définir un nouveau processus, notamment en se référant à un processus ponctuel de Poisson ; nous présentons ensuite la notion des processus de Markov. La deuxième section 2.3 de ce chapitre, sera consacrée aux méthodes de simulation des processus ponctuels, en l'occurrence les techniques de Monte Carlo par Chaînes de Markov. Les questions relatives à la convergence de ces méthodes seront discutées dans la section 2.4 vu leur importance. La dernière section 2.5 sera dédiée à une vue sur les applications récentes de ces outils stochastiques dans les contextes de reconnaissance de formes et dans l'analyse des classes.

Ce cheminement est largement inspiré du livre de M. N. M. Van Lieshout sur les processus de Markov et leurs applications [157].

2.2 Processus ponctuels

Nous allons présenter dans cette section le formalisme mathématique des processus ponctuels. Nous commençons par donner les définitions et les propriétés de ces processus ; nous enchaînons par la construction d'un processus par une densité de probabilité ; et nous terminons par introduire la propriété de dépendance entre les points

d'un processus.

2.2.1 Fondements des processus objets

Les processus ponctuels sont des entités stochastiques qui servent à modéliser et à étudier la répartition aléatoire de points dans un espace donné. Ils disposent d'un cadre théorique très solide relevant de la théorie des probabilités et des mesures ainsi que de la géométrie stochastique. De par la vocation du présent document, un mémoire, nous nous contentons de présenter les principaux passages de la théorie des processus stochastiques, notamment les définitions et les propriétés qui nous intéressent pour le développement de la suite de cette thèse.

2.2.1.1 Définition d'un processus ponctuel

Soit (χ, d) un espace métrique, χ définit un espace muni d'une métrique d ; dans le cas général, nous travaillons dans $(\mathbb{R}^p, \|.\|_2)$ où $\|.\|_2$ est la distance euclidienne.

Pour pouvoir définir formellement un processus ponctuel, nous devons nous placer dans un espace probabilisé $(\Omega, \mathcal{A}, \mathbb{P})$, formé d'un ensemble Ω et d'une tribu ou $\sigma-$algèbre $\mathcal{A}$ contenant les sous ensemble de Ω. $\mathbb{P}$ est une mesure de probabilité sur $\mathcal{A}$.

Commençons par la notion de configuration.

Définition 3 (Configuration) *On appelle configuration tout ensemble de points $x = (x_1, .., x_n)$, où $x_n \in (\chi, d)$, sont des points issus d'une expérience aléatoire. On dit qu'une configuration est localement finie si elle possède au plus un nombre fini de points dans n'importe quel borélien borné A de (χ, d), on note N^{lf} la famille des configurations localement finies.*

Une configuration localement finie x ne possède donc pas de points d'accumulation : $\forall a \in \chi, \forall r > 0; a \cap B(a; r)$ est de cardinal fini. La définition du processus ponctuel s'exprime alors comme il suit :

Définition 4 (processus ponctuel) *On appelle processus ponctuel sur (χ, d) une application qui, à une réalisation possible du hasard $\omega \in (\Omega, \mathcal{A}, \mathbb{P})$ associe une configuration localement finie $x \in N^{lf}$ dans χ dont le nombre de points $N(A)$, dans un borélien borné A de χ, vérifie : $\omega \to N(A)(\omega) = N(A)$ est une variable aléatoire discrète finie.*

On peut aussi définir un processus ponctuel comme une variable aléatoire à valeur dans l'espace mesurable des configurations localement finie $(N^{lf}; \mathcal{N}^{lf})$, où $\mathcal{N}^{lf}$ est la plus petite tribu qui rend, pour tout borélien A, l'application $x \to N_x(A)$ mesurable.

Définition 5 (Loi d'un processus ponctuel) *On appelle loi du processus ponctuel la mesure de probabilité π induite sur $\mathbb{N}^{lf}$.*

Cette courte partie définit un analogue des fonctions de répartitions pour les processus ponctuels.

Définition 6 (Fidis) *. Soit X un processus ponctuel sur (χ, d). La famille des lois marginales de dimension finie, appelé Fidis (fidis pour Finite Dimensional Distributions) est l'ensemble des lois marginales $(N(A_1); ..; N(A_m))$, quelque soit le vecteur de taille finie m considéré $(A_1; ..; A_m)$, où $A_1; ..; A_m$ désignent des boréliens bornés quelconques de χ .*

L'analogie se fait au travers du théorème suivant :

Théorème 1 *La loi d'un processus ponctuel sur un espace métrique complet séparable est entièrement déterminée par la donnée de la famille des fidis.*

2.2.1.2 Processus ponctuels marqués

L'un des avantages des processus ponctuels réside dans la possibilité de les utiliser pour modéliser des objets complexes plutôt que de simples points. En effet, à partir d'un ensemble de points pris d'un espace, nous pouvons associer à chaque point un ensemble de caractéristiques ou propriétés appelées *marques*. Ces marques ainsi introduites permettent de définir un processus ponctuel marqué ou processus objet, c'est à dire dont les éléments sont des objets. D'où leurs utilisations dans la modélisation de formes géométriques de base telles que des lignes, des rectangles, des cercles.. etc ; ou également des objets beaucoup plus complexes tels que des réseaux, des cylindres, des polygones..etc. Nous reviendrons à cela dans la section des applications.

Mathématiquement, un processus ponctuel marqué est défini comme suivant :

Définition 7 (Processus ponctuels marqués) *Soit (χ, d) et $(K, d^{'})$ deux espaces métriques complets séparables. Un processus ponctuel marqué à positions dans χ et à marques dans K est un processus ponctuel sur l'espace produit $\chi \times K$ de sorte que le processus ponctuel non marqué soit bien défini.*

2.2.1.3 Processus ponctuels simples

Notons maintenant, N_s^{lf} l'ensemble des configurations x localement finies dont les points sont simples : $N_x(\{x\}) \in \{0, 1\}$ pour tout $x \in \chi$.

Définition 8 *Un processus ponctuel est dit simple s'il prend ses valeurs dans N_s^{lf} presque surement.*

2.2.1.4 Processus ponctuels finis

En plus d'être souvent simples, la plupart des configurations sont aussi finies. Cette aspect provient généralement de l'aspect borné de la fenêtre d'observation mais peut aussi provenir du fait que le processus ne génère qu'un nombre fini de points.

Définition 9 (processus ponctuel fini) *On appelle processus ponctuel fini un processus dont les configurations sont finies.*

Les processus ponctuels finis se modélisent bien par la donnée de :

- une loi de probabilité discrète $(p_n)_{n\in N}$ qui régit le nombre de points.
- une famille de densité de probabilité $j_n(x_1;..;x_n), n \in N$, symétriques en ses variables, qui régit la position des points sur χ^n .

2.2.2 Processus ponctuel de Poisson

Un processus ponctuel est, comme nous l'avons mentionné, un mécanisme aléatoire décrivant la distribution d'un motif de points dans l'espace. Le processus ponctuel le plus fondamental est celui dit *processus de Poisson.* Il modélise une distribution de points totalement indépendants les uns des autres.

Définition 10 (Processus ponctuel de Poisson homogène) *Un processus ponctuel X sur $\chi = \mathbb{R}^d$ est appelé processus ponctuel de Poisson homogène d'intensité $\lambda > 0$ si :*

- $\forall A$ *borélien borné,* $N(A)$ *suit une loi de Poisson de paramètre* $\lambda.\mu(A)$
- $\forall A_1,..,A_k$ *boréliens bornés disjoints,* $N(A_1);..;N(A_k)$ *sont indépendantes*

Définition 11 (Processus ponctuel de Poisson inhomogène) *On appelle processus ponctuel de Poisson inhomogène un processus ponctuel vérifiant :*

- $\forall A$ *borélien borné,* $N(A)$ *suit une loi de Poisson de paramètre* $\nu(A)$ *où* $\nu(A) = \int_A \lambda d\mu$ *et* λ *est une fonction positive, Lebesgue mesurable sur* χ
- $\forall A_1,..,A_k$ *boréliens bornés disjoints,* $N(A_1);..;N(A_k)$ *sont indépendantes*

On peut continuer la généralisation précédente en considérant des paramètres νsans densité. Donnons la définition d'un processus ponctuel de Poisson :

Définition 12 (Processus ponctuel de Poisson) *Soit $\nu(.)$ une mesure de Borel sur un espace métrique complet séparable (χ, d) vérifiant $\nu(\chi) > 0$ et $\nu(\chi) < \infty$ pour tout borélien borné A de χ (une telle mesure est dite localement finie). Un processus ponctuel sur χ est un processus ponctuel de Poisson de mesure d'intensité $\nu(.)$ si*

- *Pour tout borélien borné A, $N(A)$ suit une loi de Poisson de paramètre $\nu(A)$*
- *Pour tout ensemble $A_1, .., A_k$ de boréliens bornés disjoints, $N(A_1); ..; N(A_k)$ sont indépendantes*

2.2.3 Construction d'un processus ponctuel

Dans la littérature, plusieurs façons de construire de nouveaux procédés de points ont été discutés [84] , mais en général ils sont définis à partir d'une densité de probabilité par rapport à un processus de Poisson de référence [37]. En effet, de nombreux modèles réalistes de processus ponctuels peuvent être élaborés par la spécification d'une densité de probabilité par rapport à la mesure de référence.

On notera dans ce qui suit $\pi(.)$ la loi d'un processus ponctuel de Poisson sur χ dont la mesure intensité ν est finie, sans masse ponctuelle.

Définition 13 (Densité de probabilité d'un processus ponctuel fini) .

Soit $p : N^f \to \mathbb{R}^+$ une application mesurable positive vérifiant :

$$\int_{N^f} p(x) d\pi(x) = 1 \tag{2.1}$$

Alors p est une densité de probabilité et définit un processus ponctuel X sur χ. Le processus de Poisson dominant - on dit aussi de référence - étant fini et simple, il en est de même pour le processus X. Par ailleurs les éléments qui caractérisent X $(p_n)_{n\in N}$ et $j_n(x_1, ..., x_n)$, sont donnés en fonction de p par :

$$p_n = \frac{e^{(-\nu(\chi))}}{n!} \int_\chi ... \int_\chi p(\{x_1, ..., x_n\}) d\nu(x_1)...d\nu(x_n) \tag{2.2}$$

$$j_n(x_1, .., x_n) = \frac{p(\{x_1; ...; x_n\})}{\int_\chi ... \int_\chi p(\{x_1, ..., x_n\}) d\nu(x_1)...d\nu(x_n)} \tag{2.3}$$

2.2.4 Processus ponctuel de Markov

Si les interactions entre les points sont définies, en se basant sur une relation de voisinage, le processus de point devient un processus ponctuel de Markov. Cette notion a été introduite dans [130] par une généralisation du processus de Strauss.

Définition 14 (voisin) *Soit $\sim$ une relation binaire symétrique et réflexive sur $(\chi; d)$ - dite relation de voisinage. Soit u et v deux points de χ. On dit que u et v sont voisins si $u \sim v$.*

Définition 15 (voisinage) *Soit A un ensemble χ. On définit le voisinage de A par*

$$\partial(A) = \{x \in \chi : x \sim a, a \in A\} \tag{2.4}$$

En particulier, le voisinage de $\{a\}$ contient tous les voisins de a.

2.2.4.1 Densité conditionnelle de Papangelou

Définition 16 (Densité conditionnelle de Papangelou) *. La densité conditionnelle de Papangelou notée $\lambda(.,.)$ est définie de manière heuristique par l'équation suivante :*

$$\lambda(x; x)dx = \mathbb{P}(N(dx) = 1 | X \cap (dx)^c = x \cap (dx)^c) \tag{2.5}$$

Cette équation s'interprète comme la probabilité qu'a le processus X d'envoyer un point dans une région dx autour d'un point $x \in \chi$ qui respecte aussi la configuration déjà existante à l'extérieur de dx.

Théorème 2 (admis) *Soit X un processus ponctuel de densité $p(.)$ par rapport à un processus de Poisson de mesure d'intensité ν finie, sans masse ponctuelle. Alors X a une densité conditionnelle de Papangelou*

$$\lambda(u; x) = \frac{p(x \cup u)}{p(x)} \tag{2.6}$$

pour tout $u \in x \in N^f$.

Définition 17 (Processus ponctuel de Markov) *. Soit $(\chi; d)$ un espace métrique complet séparable, $\nu(.)$ une mesure de Borel sans masse ponctuelle, $\pi_\nu(.)$ la loi d'un processus de Poisson de mesure d'intensité $\nu(.)$. Soit X un processus ponctuel sur χ de densité $p(.)$ par rapport à $\pi_\nu(.)$. On dit que X est un processus ponctuel de Markov*

par rapport à une relation de voisinage $\sim$ si pour toute configuration $x \in N^{lf}$ telle que $p(x) > 0$:

- $p(y) > 0$ *pour tout* $y \subseteq x$
- $\forall u \in x, \dfrac{p(x \cup u)}{p(x)}$ *ne dépend que de* u *et* $\partial(\{u\}) = \{u' \in x : u \sim u'\}$

Théorème 3 *Soit X un processus ponctuel de Markov de densité $p(.)$ et soit A un borélien de $(\chi; d)$ espace métrique complet séparable. Alors la loi conditionnel de $X \cup A$ sachant $X \cup A^c$ ne dépend que de X restreint au voisinage $\partial(A) \cup A^c$.*

On propose maintenant une autre formulation de la propriété de Markov pour les processus ponctuels en introduisant la notion de clique :

Définition 18 (Clique) *. Soit $\sim$ une relation de voisinage sur χ. Une configuration x est appelée clique si tous les points de x sont voisins les uns des autres : $\forall u, v \in x, u \sim v$. Par convention, $\varnothing$ sera aussi une clique.*

Théorème 4 (Hammersley-Cliford) *. Une densité de processus ponctuel $p : N^{lf} \to [0; +\infty[$ est markovienne i.e. définit un processus ponctuel de Markov par rapport à la relation de voisinage $\sim$ si et seulement si il existe une fonction $\phi : N^{lf} \to [0; +\infty[$ mesurable telle que*

$$p(x) = \prod_{clique\ y \subseteq x} \phi(y) \tag{2.7}$$

pour tout $x \in N^f$.

On peut remplacer le produit sur les cliques par un produit sur toutes les sous-configurations $y \subseteq x$ en prenant $\phi(y) = 1$ lorsque y n'est pas une clique.

Les propriétés abordées ci-dessus concernant la markovienneté des processus ponctuels peuvent se généraliser aux processus ponctuels marqués sans aucune difficulté.

2.3 Simulation des processes ponctuels

Les méthodes de Monte Carlo par Chaînes de Markov (MCMC) sont des techniques populaires utilisées pour générer des échantillons de distributions aléatoires d'intérêt de lois compliquées quelconques. Ces lois se présentent souvent sous la forme de fonctions de dimensions trop élevées. Elles sont nommées ainsi puisqu'elles utilisent les valeurs

de la réalisation antérieure pour générer aléatoirement les valeurs de la réalisation postérieure en construisant ainsi une chaîne qui a la propriété de Markov.

Toutes ces méthodes trouvent leurs racines dans l'algorithme de Metropolis [113; 114]. Ces physiciens ont essayé d'approcher une intégrale complexe en l'exprimant sous forme de l'espérance d'une certaine distribution et puis estimer cette espérance en tirant des réalisation de cette distribution (cf. figure 2.1).

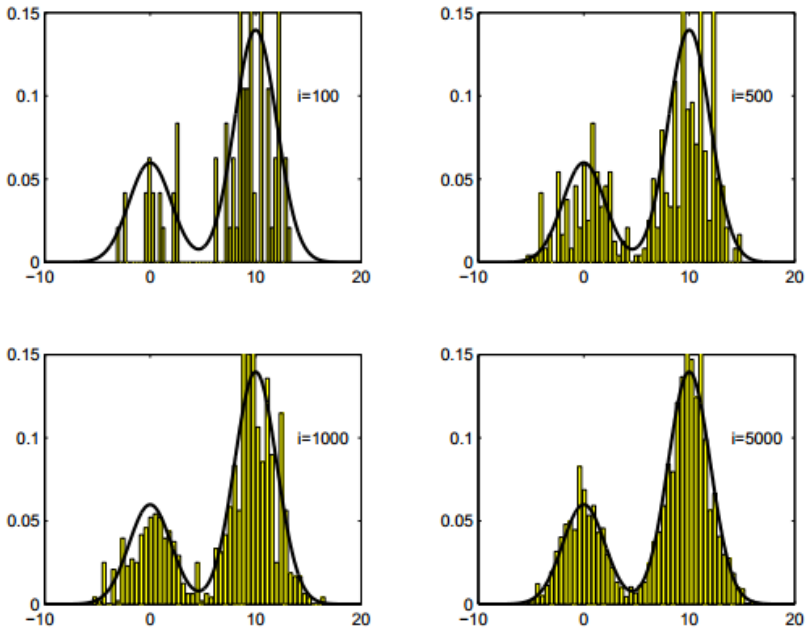


FIGURE 2.1 – Exemple de distribution cible et l'approximation obtenue par histogrammes de l'échantillonnage par MCMC (extrait de [5]).

Après avoir rappelé la technique de Monte Carlo et le formalisme des chaînes de Markov, nous présenterons les trois algorithmes qui sont Metropolis-Hasting [67], Metropolis-Hasting Green [60; 64] et l'échantillonneur de Gibbs [56].

Cette section sera conclue par une présentation de la méthode de recuit simulé [86] qui est souvent utilisée avec ces méthodes pour des problèmes d'optimisation.

2.3.1 Algorithmes de Monté Carlo par Chaînes de Markov

2.3.1.1 Monte Carlo

Comme nous l'avons cité dans l'introduction de cette section, l'approche originale de Monte Carlo était une méthode développée par des physiciens pour la génération de nombres aléatoires dans le but de calculer des intégrales compliquées. Supposons que nous voulions calculer une intégrale de la sorte :

$$\int_b^a h(x)dx \tag{2.8}$$

Si l'on peut décomposer h(x) comme le produit d'une fonction f(x) et une fonction de densité de probabilité p(x) définie sur l'intervalle [a,b], alors :

$$\int_b^a h(x)dx = \int_b^a f(x)p(x)dx = E_{p(x)}[f(x)] \tag{2.9}$$

de telle sorte que l'intégrale peut être exprimée comme une espérance de f(x) sous la densité p(x). Ainsi, si nous tirons un grand nombre : $x_1, ..x_n$ d'une variable aléatoire de densité p(x), alors

$$\int_b^a h(x)dx = E_{p(x)}[f(x)] \simeq \frac{1}{n}\sum_{i=1}^{n} f(x_i) \tag{2.10}$$

C'est ce qu'on appelle l'intégration par Monte Carlo.

2.3.1.2 Chaînes de Markov

Soit E un espace discret, i.e. E est un espace au plus dénombrable muni de la topologie discrète, où tous les points de E sont isolés, et donc de la tribu $\mathcal{E} = \mathcal{P}(E)$.

Définition 19 (Matrice stochastique) *Une matrice $P = (P(x, y), x, y \in E)$ est dite matrice stochastique si ses coefficients sont positifs et la somme sur une ligne des coefficients est égale à 1 :*

$$P(x, y) \geq 0 \text{ et } \sum_{z \in E} P(x, z) = 1, \text{ pour tout } x, y \in E. \tag{2.11}$$

Définition 20 (chaîne de Markov) *Soit P une matrice stochastique sur E. Une suite de variables aléatoires $(X_n, n \in N)$ à valeurs dans E est appelée chaîne de Markov de*

matrice de transition P si pour tout $n \in N, x \in E$, on a

$$\mathbb{P}(X_{n+1} = x|X_n, ..., X_0) = \mathbb{P}(X_{n+1} = x|X_n) = P(X_n, x) \tag{2.12}$$

On dit que la chaîne de Markov est issue de μ_0 si la loi de X_0 est μ_0.

Propriété 1 *La loi d'une chaîne de Markov $(X_n, n \in N)$ est entièrement caractérisée par sa matrice de transition, P, et la loi de X_0, μ_0. De plus, on a, pour tout $n \in \mathbb{N}^*$ et $x_0, .., x_n \in E$:*

$$\mathbb{P}(X_0, .., X_n) = \mu_0(x_0) \prod_{k=1}^{n} P(x_{k-1}, x_k) \tag{2.13}$$

Le passage au cas d'espaces d'états continus se fait en généralisant la notion de matrice de transition $P(x_i, x_j) = p_{ij}$ pour qu'elle devienne noyau de transition $P(x, A)$ vérifiant :

$$\int P(x, A)dx = 1 \text{ pour tout } x \in \Omega \text{ et } A \in \mathcal{B}. \tag{2.14}$$

Dans toute la suite, nous allons travailler sur des espaces d'états continus sauf si mentionné le contraire.

Soit X_t une chaîne de Markov sur l'espace mesuré $(\Omega, \mathcal{B}, \mu)$ avec $P(x, A), x \in \Omega, A \in \mathcal{B}$ comme noyau de transition.

Propriété 2 (Invariance) *une loi π est invariante pour la chaîne de Markov si :*

$$\pi(A) = \int P(x, A)\pi(dx) \forall A \in \mathcal{B} \tag{2.15}$$

Propriété 3 (Réversibilité) *la chaîne est réversible pour π si le noyau de transition vérifie :*

$$\int_A P(x, B)\pi(dx) = \int_B P(x, A)\pi(dx) \forall A, B \in \mathcal{B} \tag{2.16}$$

Propriété 4 (Irréductibilité) *une chaîne de Markov est dite irréductible si $\forall x \in \Omega, \forall A \in \mathcal{B}$,*

$$\pi(A) > 0 \Rightarrow \exists t, P^t(x, A) > 0 \tag{2.17}$$

Propriété 5 (apériodicité) *Une chaîne de Markov est apériodique s'il n'existe pas d'ensemble $A_0, .., A_{d-1} \in \mathcal{B}$ avec $d \geqslant 2$ tels que :*

$$P(x, A_{i+1[d]}) = 1 \forall x \in A_i, i = 0..d - 1 \tag{2.18}$$

Une condition suffisante mais non nécessaire pour qu'une chaîne soit apériodique est

$$\forall x \in \Omega, P(x, x) > 0 \tag{2.19}$$

Dans ce cas la chaîne est dite fortement apériodique. Il suffit dans le cas discret de vérifier cela pour un seul $x \in \Omega$.

Propriété 6 (Récurrence au sens de Harris) *Cette condition est nécessaire pour assurer la convergence de la chaîne pour toute configuration initiale. Il assure que* $\forall x \in \Omega$ *et* $\forall A \in \mathcal{B}$ *:*

$$\pi(A) > 0 \Rightarrow P(\exists t, X_t \in A | X_0 = x) = 1 \tag{2.20}$$

2.3.1.3 Algorithme de naissance et de mort

Deux taux à définir ici : un taux de naissance $B(x, .)$ et un taux de mort $d(x, u)$. Il faudrait ensuite définir les taux globaux associés :

$$B(x) = B(x, U) \tag{2.21}$$

$$D(x) = \sum_{u_i \in x} d(x, u_i) \tag{2.22}$$

Les étapes de l'algorithme sont ensuite décrites comme suivant :

1. A partir d'une étape t donnée, on choisit entre deux modification de la configuration actuelle x_t, une naissance d'un objet avec une probabilité $B(x_t)/B(x_t)+D(x_t)$ ou une mort d'un objet avec une probabilité $D(x_t)/B(x_t) + D(x_t)$.

2. Si la modification est une naissance , on choisit alors un objet u de l'espace selon la loi $B(x_t, .))/B(x_t)$, et on pose $x_{t+1} = x_t \cup \{u\}$

3. Si la modification est une mort , on choisit alors un objet v de la configuration selon la loi $d(x_t, v))/D(x_t)$, et on pose $x_{t+1} = x_t \setminus \{v\}$.

Dans le cas général, la mesure de naissance $B(x_t, .)$ s'exprime sous forme d'une densité par rapport à la mesure μ, notée $b(x, .)$. Il est à signaler qu'il y a une grande liberté dans la spécification de ces densités dont l'objectif est d'accélérer la convergence de l'algorithme vers l'état stationnaire. Pour plus de détails , le lecteur pourra se référencer à [74].

2.3.1.4 L'algorithme de Metropolis-Hasting

Cet algorithme rajoute une densité auxiliaire de transition $q(x, .)$ mais qui devrait être calculable et simulable facilement. Les étapes de l'algorithme sont de la sorte :

1. A partir d'une étape t donnée, on crée une nouvelle configuration y selon la densité $q(x_t, .)$,
2. on calcule ensuite le rapport :

$$R = \frac{f(y)q(y, x_t)}{f(x_t)q(x_t, y)} \tag{2.23}$$

3. avec une probabilité $\alpha = min\{1, R\}$, accepter la configuration $x_{t+1} = y$, sinon, on reste sur $x_{t+1} = x_t$.

2.3.1.5 L'algorithme de Metropolis-Hasting Green

Cet algorithme est une version plus améliorée du précédent, dans le fait qu'il laisse définir un noyau de transition pas forcément défini à partir d'une densité par rapport à la mesure μ de l'espace Ω.

Les changements introduits intéressent la densité f qui est remplacée par la mesure non normalisé π sur l'espace des réalisations Ω. Ensuite, la densité auxiliaire est remplacée par le noyau $Q(x, A), x \in \omega, A \in \mathcal{B}$.

Un autre changement est la supposition de l'existence d'une mesure symétrique ξ sur $\omega \times \Omega$ dominant $\pi(dx)Q(x, dy)$ de sorte que la dérivée de Radon-Nikodym existe :

$$h(x, y) = \frac{\pi(dx)Q(x, dy)}{\xi(dx, dy)} \tag{2.24}$$

qui remplace $f(x)q(x, y)$ dans la version précédente. Ainsi le taux d'acceptation se change à

$$R = \frac{h(y, x)}{h(x, y)} \tag{2.25}$$

Les trois étapes de l'algorithme sont alors :

1. simuler une nouvelle configuration $y \sim Q(x, .)$
2. calculer le rapport $R = h(y, x)/h(x, y)$
3. accepter la nouvelle configuration y selon la probabilité $\alpha = min\{1, R\}$.

2.3.1.6 L'échantillonneur de Gibbs

L'échantillonneur de Gibbs [56] est un cas particulier de l'algorithme de Metropolis Hasting. Cet algorithme accepte tous les états générés. Il les génère à partir de la distribution stationnaire elle même et non pas en utilisant une densité auxiliaire sur l'espace des états comme dans le cas de Metropolis Hasting. Donc, à partir d'une configuration donnée $X = \{x_1, .., x_n\}$:

1. choisir un objet à modifier de x_j de X
2. générer une nouvelle valeur x'_j selon $\pi(x_1, .., x_{j-1}, ., x_{j+1}, ..x_n)$
3. recommencer avec la nouvelle configuration $X = \{x_1, .., x_{j-1}, x'_j, x_{j+1}, ..x_n\}$

2.3.2 Optimisation par l'algorithme du recuit simulé

L'algorithme du *recuit simulé* (simulated annealing) est un algorithme d'optimisation qui permet de se franchir les optimum locaux d'une fonction et obtenir ainsi une convergence vers un optimal global [1]. Son principe est similaire aux autres algorithmes MCMC, néanmoins, il introduit la notion de température qui permet d'échantillonner à partir de la distribution $p_i(x)^{\frac{1}{T_i}}$ au lieu de $p(x)$. Cet algorithme fait que, au début de l'échantillonnage de l'espace, il accepte des états avec certaines probabilités, même s'ils dégradent la fonction à maximiser.

Le paramètre de température T_i est une fonction décroissante qui tend vers 0 à l'infini. Le choix de cette fonction appelée schéma de refroidissement est fortement lié aux applications. Typiquement il est possible de choisir une décroissance géométrique de la température :

$$T_{i+1} = cT_i \tag{2.26}$$

où c est une constante inférieure à 1. Un autre choix assez utilisé est le suivant

$$T_{i+1} = \frac{T_0}{1 + log(T_i)} \tag{2.27}$$

avec T_0 la température initiale.

1. Nous avons déjà vu cette technique dans le chapitre précédent en rapportant les techniques de classification de données s'inspirant des phénomènes naturels, voir la section 1.3.5.

2.4 Diagnostic de la convergence des méthodes de Monte Carlo par Chaînes de Markov

Même si la convergence, vers la loi invariante, des chaînes de Markov construites lors de la simulation des distributions est assurée théoriquement dès que les propriétés mentionnées dans la section 2.3.1.2 aient été réalisées, savoir quand cette convergence ait lieu reste loin d'être une tache facile. En effet, ceci aurait permis de définir aisément un critère d'arrêt lorsque la loi stationnaire est atteinte. Or ceci n'est pas le cas, puisque plusieurs facteurs rentre dans ce problème de convergence. Déjà les valeurs de l'initialisation de la chaîne affecte cette convergence ; aussi la définition des noyaux de proposition non adéquats peut influencer à son tour la convergence et sa rapidité. De ce fait, plusieurs études ont été menées pour essayer de trouver des solutions à ce problème. D'où le développement des méthodes de diagnostics de la convergence des méthodes de Monte Carlo par chaîne de Markov.

Parmi les points critiques dans cette question de la convergence des chaînes de Markov, deux sont souvent discutés dans la littérature concernant le diagnostique de la convergence. Le premier est ce que l'on appelle *période d'échauffement* et le deuxième est celui de la longueur de la chaîne.

Dans ce qui suit, nous allons aborder ces deux points plus la présentation de deux méthodes pour tester la convergence bien qu'il en existe plusieurs dans la littérature. Pour des lectures plus détaillées sur le sujet, nous renvoyons aux article de [43].

2.4.1 Période d'échauffement et longueur de la chaîne

L'une des points clés pour une bonne utilisation des méthodes de Monte Carlo par Chaînes de Markov est de savoir déterminer le nombre de réalisations ou d'itérations à écarter avant que nous obtenions des réalisations convenables, c'est-à-dire provenant de la loi stationnaire de la chaîne ; cela est appelé *la période d'échauffement* (burn-in period). La longueur totale de la chaîne est aussi important à savoir. Nous présentons ici la méthode de Raftery and Lewis ([129]) qui permet de les estimer.

2.4.1.1 Raftery and Lewis

La méthode de Raftery and Lewis (1992) est destiné à détecter à la fois la convergence vers la distribution stationnaire et à fournir un moyen de délimitation de la variance des estimateurs des quantiles des fonctions de paramètres. Supposons que nous voulons estimer à partir de la distribution a posteriori la valeur de u donnée par

$Pr(\theta(X) \leqslant u) = q$ pour certains q fixé sous la contrainte que $Pr(\theta(X) \leqslant \pm\hat{u})$ est avec la précision $\pm r$ de la valeur réelle avec la probabilité p. La méthode de Raftery et Lewis donne d'abord la longueur de la période de burn-in p_0 et la seconde le nombre d'itérations, $n - n_0$, nécessaire pour estimer la probabilité ci-dessus avec précision. Définir $Z_t = I(\theta(X^t) \leq u)$ où I est la fonction indicatrice.

La séquence Z_t n'est pas une chaîne de Markov et la méthode de Raftery et Lewis détermine la taille du sous échantillon s de telle sorte que Z_{sr} est une chaîne de Markov.

L'idée sous-jacente à cette approche est de réduire le problème de diagnostic de la convergence à l'analyse d'une chaîne de Markov à deux états. Des études pratiques montrent que les valeurs de s obtenues par cette méthode sont généralement très faibles (voir, par exemple [32]). Il est donc possible de travailler avec la séquence complète Z_t sans l'aide de l'approximation de Markov. Si Z_t est traitée comme une chaîne de Markov homogène de matrice de transition :

$$P = \begin{pmatrix} 1-\alpha & \alpha \\ \beta & 1-\beta \end{pmatrix} \tag{2.28}$$

la longueur de la période de rodage peut être déduite à partir de

$$|P(Z_{n_0} = i|Z_0 = j) - P(Z_\infty = i)| \leq \epsilon \tag{2.29}$$

pour $i, j = 0, 1$, où Z_∞ désigne la distribution stationnaire de la chaîne. En utilisant les résultats standards relatifs aux chaînes de Markov à deux états, Raftery et Lewis ont trouvé que

$$n_0 = log((\alpha + \beta)\epsilon/max(\alpha, \beta))/log(|1 - \alpha - \beta|). \tag{2.30}$$

Si la taille de la chaîne nn_0 est suffisamment grande, la quantité $\delta_{n-n_0} = [1/(n - n_0)] \sum_{t=n_0}^{n} Z_t$ converge vers $\dfrac{\alpha}{(\alpha+\beta)}$.

En utilisant une approximation normale de δ_{n-n_0} , la taille de l'échantillon permettant d'obtenir la convergence est :

$$n - n_0 = \frac{\alpha\beta(2-\alpha-\beta)}{q'^2(\alpha+\beta)^3}\Phi^{-1}(\frac{\epsilon^{'}+1}{2}) \tag{2.31}$$

sous la contrainte $Pr(|\delta_{n-n_0} - \alpha/(\alpha + \beta)| < q') \geq \epsilon^{'}$.

2.4.2 Tests de convergence

2.4.2.1 Gelman and Rubin

La méthode de Gelman et Rubin [55] suppose que m chaînes ont été simulées indépendamment. Si la convergence a été obtenue, l'inférence est réalisée en calculant la moyenne et la variance de l'échantillon de chaque chaîne. Dans le cas de la convergence, les m inférences obtenues devraient être similaires. L'idée est d'estimer σ^2, la variance de θ et calculer le rapport de cet estimateur, noté $\hat{V}$ et σ^2. L'estimateur $\hat{V}$ est construit en mélangeant mn réalisations tirées de toutes les chaînes et considère à la fois la variance entre-chaînes et intra-chaînes. Nous utilisons la notation suivante : $\theta_i^t = \theta(x_i^t)$ est la fonction de l'intérêt évalué à l'observation de t^{th} de la chaîne i et on a :

$$\bar{\theta}_i = \frac{1}{n}\sum_{t=n_0+1}^{2n}\theta_i^t; \bar{\theta} = \frac{1}{m}\sum_{i=1}^{m}\bar{\theta}_i; s_i^2 = \frac{1}{n-n_0-1}\sum_{t=n_0+1}^{2n}(\theta_i^t - \bar{\theta}_i) \tag{2.32}$$

La variance B/n entre les moyennes m des chaînes peut être calculée comme étant :

$$\frac{B}{n} = \frac{1}{m-1}\sum_{i=1}^{m}(\bar{\theta}_i - \bar{\theta}) \tag{2.33}$$

La moyenne des variances entre m-séquence dans s_i^2 est donnée par :

$$W = \frac{1}{m}\sum_{i=1}^{m}s_i^2 \tag{2.34}$$

L'estimateur $\hat{V}$ est alors défini par

$$\hat{V} = \frac{n-1}{n}W + (1+\frac{1}{m})\frac{B}{n}. \tag{2.35}$$

Nous calculons le rapport

$$\hat{R}_c = (\frac{d+1}{d+3})\frac{\hat{V}}{W}, \tag{2.36}$$

avec $(d+3)/(d+1)$

est un terme de correction, d est la valeur estimée du degré de liberté d'une distribution de Student [24] et peut être estimée par la méthode des moments. Gelman et Rubin (1992) font l'interprétation suivante de $\hat{R}_c$. Pour les grandes $\hat{R}_c$ il y a deux possibilités :

1. la variance $\hat{V}$ peut être diminuée par plus de simulations,

2. W peut être augmentée par plus de simulations.

Dans l'autre cas, si $\hat{R}_c$ est proche de 1, nous concluons que chacune des m chaînes a convergé.

2.4.2.2 Geweke

L'idée de Geweke [59] est basée sur un test de l'égalité des mesures de localisation pour deux séquences. Supposons que l'objectif est d'estimer la valeur espérée de certains fonctionnelle $\theta(X)$. Après la période d'échauffement, n_0, nous considérons deux séquences de $\theta^t = \theta(X^{t+n_0})$, $\{\theta^t, t = 1, .., n_A\}$ et $\{\theta^t, t = n^*, .., n\}$ où $1 < n_A < n^* < n$. Si la chaîne a convergé, les mesures de l'emplacement de ces deux séquences doivent être égales. Sur la base de cet remarque, Geweke propose le test suivant : Si la séquence $\{\theta_t\}$ est stationnaire, alors :

$$Z_n = \frac{(\bar{\theta}_A - \bar{\theta}_B)}{\sqrt{(1/n_A)\hat{S}_0^A(0) + (1/n_B)\hat{S}_0^B(0)}} \longrightarrow^d \mathcal{N}(0,1), \tag{2.37}$$

avec $n_B = n - n^* + 1$ et $\bar{\theta}_A = (1/n_A)\sum_{t=1}^{n_A}\theta^t$, $\bar{\theta}_B = (1/n_B)\sum_{t=n^*}^{n_B}\theta^t$; $\hat{S}_0^A(0)$ et $\hat{S}_0^B(0)$ sont des estimateurs de θ sur respectivement $\theta^t, t = 1, .., n_A$ et $\theta^t, t = n^*, .., n$, Geweke propose $n_A = n/10$ and $n^* = n/2$. Selon Brooks et Roberts ([23]), ce choix est arbitraire.

La figure 2.2 montre une simulation d'un paramètre X et une première fenêtre de deux tailles différentes A et A' qui sont testées avec le diagnostic Geweke contre la seconde fenêtre B. On voit ici que la fenêtre A' est choisie trop grande et la moyenne des valeurs dans cette fenêtre sera trop proche de la moyenne des valeurs dans la fenêtre B, donc le Geweke diagnostic ne découvre pas la période du *burn-in*. Toutefois, la fenêtre A est de taille appropriée pour la détecter.

2.5 Applications précédentes des processus objets

2.5.1 Processus ponctuels et les phénomènes stochastiques

Les processus ponctuels marqués sont présent dans presque tous les domaines où les phénomènes à étudier présentent des aspects de l'aléatoire. Ce sont de puissants outils pour modéliser, interpréter et prédire le comportement de tels phénomènes. Originellement, les processus ponctuels sont introduits en physique pour modéliser le comportement des systèmes de particules d'un fluide ou d'un gaz. Les points sont les positions

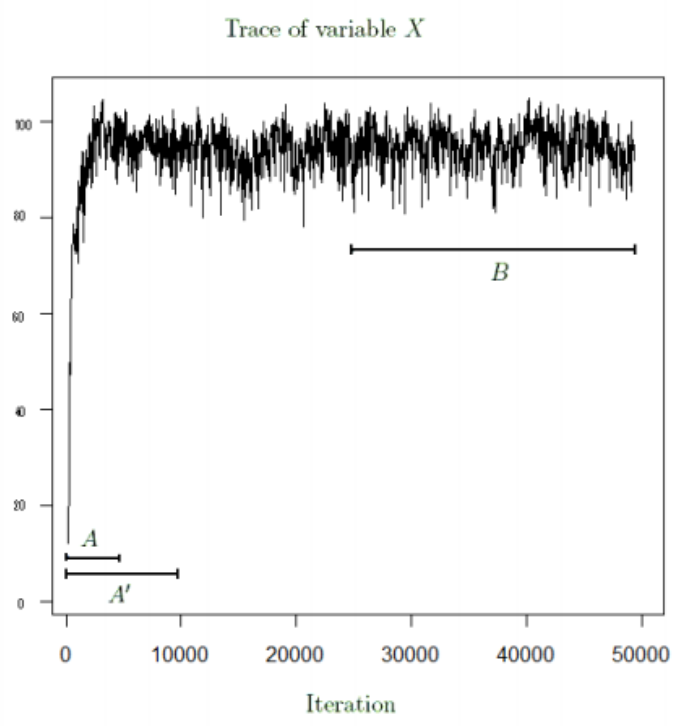


FIGURE 2.2 – Détermination du *burn-in* et diagnostic de convergence par le test Geweke (Extraite de [134]).

des particules et les marques sont leurs caractéristiques telles que la vitesse, la charge, etc. Ces marques peuvent aussi être une fonction continue représentant la trajectoire des particules en mécanique stochastique [57].

Ils sont ensuite employés dans les études de la géométrie stochastique. Ils sont utilisés comme des blocs de base pour construire des modèles d'ensemble aléatoire d'autant plus complexes comme le *modèle booléen*) ou des exemples instructifs sous forme de simples ensembles aléatoires [7; 83; 157].

Les champs d'application des processus ponctuels succèdent. Nous les trouverons en astronomy [106], dans le domaine des télécommunications spatialement distribuées [6], dans la foresterie [148], l'ingénierie des matériaux [153], etc.

Plus récemment, un nouveau processus ponctuel a vu le jour, il s'agit de *determinantal and permanental point processes*, qui a une relation avec la théorie des matrices aléatoires [108].

Cependant, les applications qui nous intéressent beaucoup plus dans ce travail de thèse, ce sont plutôt les applications des processus ponctuels marqués dans les domaines

de l'imagerie. C'est ce que nous aborderons dans les paragraphes suivants.

2.5.2 Processus ponctuels et l'analyse d'images

2.5.2.1 Modèles linéaires

Les premiers travaux menés comme application des processus ponctuels marqués en analyse d'images sont les travaux destinés à extraire dans les images des objets ayant la forme géométrique d'une ligne. [147] présentaient d'une manière générale un modèle pour l'extraction de réseaux ou de courbes. Mathématiquement, chaque objet est représenté par :

$$C = \{(cos\alpha, sin\alpha, p) \in \mathbb{R}^3, p \in \mathbb{R}, \alpha \in (0, \pi]\} \tag{2.38}$$

avec α l'angle partant de l'axe horizontal des abscisses et p la distance entre la ligne et l'origine de l'espace.

Un modèle linéique préalable a été élaboré même s'il a été présenté comme modèle ponctuel pour l'extraction du ponctuel [38]. Cette considération revient au fait que les objets utilisé pour définir le processus associé étaient des segments de courtes longueurs de quelques millimètres. Le modèle concerne la détection des lésions focales cérébrales indiquant un vieillissement pathologique à partir des images par résonance magnétique (IRM). La configuration d'objets a été donnée par :

$$X = \{\overrightarrow{s}(1) = (x(1), \overrightarrow{l}(1)), .., \overrightarrow{s}(n) = (x(n), \overrightarrow{l}(n))\} \tag{2.39}$$

avec $\overrightarrow{s}(i)$ le $i^{éme}$ segment , $x(i)$ son point central et $\overrightarrow{l}(i)$ est un vecteur de $\mathbb{R}^3$.

La densité du processus marqué est donnée par :

$$f(X) = \beta^n \prod_{k \in <l,n>} q_1(\| \overrightarrow{l}(k) \|) \prod_{k,k' \in <l,n>; k \sim k'} q_2(\overrightarrow{s}(k), \overrightarrow{s}(k')), \tag{2.40}$$

où $\sim$ définit une relation de voisinage. β est un paramètre de densité qui contrôle le nombre d'objets dans la configuration relativement aux réponses de l'attache aux données. Le terme $q_1(\| \overrightarrow{l}(k) \|)$ est un a priori sur la longueur des segments. Le terme $q_2(\overrightarrow{s}(k), \overrightarrow{s}(k'))$ définit des interactions entre segments voisins telles que les intersections, l'attraction et la répulsion. Pour aboutir à de telles configurations, les noyaux de transition proposés sont : Naissance/mort, division/fusion, translation, retournement et mouvement d'un point terminal.

Candy model : Pratiquement, c'est le *Candy model* et ses variantes qui sont les plus utilisés pour extraire les réseaux de lignes dans les images aériennes en deux dimensions. Ce modèle est introduit par [144] et il est basé sur des segments. Ces derniers sont caractérisés par la position de leur point central, leur longueur, leur orientation et leur épaisseur. La densité est liée au nombre n de segments de la configuration.

Un réseau linéique est selon ce point de vue un ensemble de segments ayant une relation de connexité. Deux segments sont dits connectés si la distance entre leurs extrémités est inférieure à une constante fixée. La simulation, par l'algorithme de Metropolis-Hasting-Green, de la chaîne de Markov est construite à partir des modifications se basant sur la naissance/mort, fusion/division ou translation d'un segment de telle sorte que le processus favorise les configurations approximant la vraisemblance du réseau.

Les études et les améliorations sur ce modèle se succèdent. Lacoste [93], introduit par exemple le modèle des lignes brisées. La particularité de ce modèle est le fait qu'il utilise comme objet de base non plus des segments tout court mais toute une ligne formée de plusieurs segments connectés. Chaque ligne est donc décrit par son origine (le point marqué), son épaisseur, le nombre de segments qui la forment avec leurs longueurs et les angles précisant leurs directions.

Le Candy model est également utilisé pour la détection de failles séismiques [84]. Ce modèle est également étendu pour modéliser non seulement des lignes mais plutôt des courbes. Les objets utilisés pour décrire le processus étaient donc des courbes paramétrées connues sous le nom de courbes de *Bézier*. Le modèle est appelé modèle parabolique.

Ce modèle a été étendu pour modéliser des objet cette fois-ci dans trois dimensions. Il s'agit d'une application sur la détection des filaments [146]. Il a été baptisé *Bisous model*.

2.5.2.2 Modèles surfaciques

Extraction des arbres : Une méthode permettant de prendre en compte des informations sur la géométrie et la radiométrie des houppiers ainsi que sur la répartition spatiale de l'ensemble des arbres.

Perrin [65] a élaboré donc un modèle pour la détection des arbres (houppiers) à partir des images aériennes. Les objets employés sont initialement des cercles . Chaque cercle est alors représenté par la position de son centre dans l'espace des points et par

son rayon dans l'espace des marques,

$$\mathcal{U} = \mathcal{P} \times \mathcal{K} = [0, X_M] \times [0, Y_M] \times [R_m, R_M] \tag{2.41}$$

Le terme de l'a priori qui nous intéresse dans ce document est donnée par la densité suivante :

$$h(x) = exp(\gamma_a \mathcal{A}(x)) \tag{2.42}$$

où $\mathcal{A}(x) = \sum_{u \in x} \lambda_2(\cap_x(u))$ définit l'aire totale des ensembles d'intersection de chaque objet dans la configuration x. L'auteur définit des noyaux de propositions adaptées à la loi objective pour accélérer la convergence de la chaîne de Markov construite. Ainsi naissance / mort, division /fusion , translation et dilatation d'un objet.

Extraction de bâtiments : [107] a présenté un modèle de processus rectangulaire pour extraire les bâtiments des zones urbaines à partir des images aériennes. Plus précisément, il voulait reconstruire les bâtiments en 3D à partir de leurs trace au sol en prenant en considération les données provenant d'un modèle numérique d'élévation (MNE, Digital Elevation Model DEM). Les objets du processus, étant des rectangles, ont été définis leurs point central, la largeur , la longueur et l'orientation. Un objet est donc un élément d'un espace noté M tel que :

$$M = [0, X_{max}] \times [0, Y_{max}] \times [L_{min}, L_{max}] \times [l_{min}, l_{max}] \times [\frac{-\pi}{2}, \frac{\pi}{2}]. \tag{2.43}$$

Le terme de l'a priori associé à la densité du modèle est donné par :

$$h(x) = exp(-V_{inter} \times s(x)) \tag{2.44}$$

où $s(x)$ est le nombre de paires de rectangles de la configuration s'intersectant et V_{inter} une constante réelle strictement positive et par conséquent le terme est répulsif et pénalise les intersections.

Détection des puits de pétrole : Une autre application des processus objet est celle développée dans [96]. Les auteurs ont présenté une méthode pour la détection de puits de pétrole dans les zones maritimes à partir des images construites par la technique des capteurs microonde *Synthetic Aperture Radar* (SAR).

Le modèle des objets utilisés est un rectangle décrit par les mêmes caractéristique

que celui pour la détection de bâtiments. Aussi , il n'y avait pas de nouveautés concernant les noyaux de proposition ni la construction de la chaîne de Markov.

Détection de regroupements d'objets : C'est une application qui nous a beaucoup inspiré dans le développement de notre contribution à la classification de données par les processus objet. Elle a été présentée par Stoica et al [145]. Les auteurs ont appliqué les processus ponctuels pour identifier dans l'espace, des régions qui ont les mêmes caractéristiques statistiques. Ces régions seront utilisées pour identifier les classes (*cluster pattern*) sous-jacentes aux données. Cette application rentre dans le domaine de la biométrie et a été destinée plus précisément à la localisation, dans le terrain de l'étude, des fermes où se sont répandues certaines pathologies.

Le modèle des objets utilisés est donné par des disques. Chaque disque est représenté par un centre et un rayon. La densité du processus comporte un terme d'interaction se basant aussi sur un potentiel d'interaction par aire et un modèle de Strauss.

2.5.2.3 Segmentation par régions

Une étude récente [4] a développé un nouveau processus objets destiné à prendre en considération d'autres types d'interactions. Le modèle construit a été dénommé *Grouping/Degrouping Point Process* (GDPP). Ce modèle est appliqué avec succès pour la segmentation d'image basée régions. A la différence des méthodes de segmentation d'image basées sur la fusion des régions, cette méthode a permis l'incorporation de certaines propriétés topologiques dans la procédure de segmentation.

L'idée est d'utiliser un processus ponctuel pour sélectionner des régions à partir d'une partition initiale de l'image. Cette partition initiale peut être obtenue par une sur-segmentation de l'image, suivant un critère, par l'une des méthodes classiques de segmentation. Le processus ponctuel proposé divise l'ensemble des régions en deux catégories : les *régions objets* et les *régions de fond*.

Les propriétés topologiques prises en considération par ce nouveau processus sont par exemple le positionnement de points, la présence de points isolés, la surface d'intersection entre deux régions données et le nombre de trous qu'elles forment.

2.6 Conclusion

Ce chapitre nous a été l'occasion de présenter les fondements mathématiques des processus ponctuels marqués. Ceci est important pour le développement de la suite de ce mémoire. En effet, avant de présenter notre modélisation de la problématique de la

classification des données multidimensionnelles par le biais des processus ponctuels, il a fallu tout d'abord bien les décrire et comprendre. Ainsi, nous avons abordé la plupart des aspects de ces outils stochastiques : les définitions, les propriétés fondamentales, comment les construire et comment les simuler.

La question de la simulation est un point clé et délicat qui démontre la difficulté et la complexité de travailler avec de tels concepts stochastiques. Ceci revient en sa grande partie à la façon de construire de nouveaux processus qui, le plus souvent, s'effectue au moyen d'une fonction de densité de probabilité que l'on définit par rapport à un processus de référence. Les espaces dans lesquels nous cherchons à optimiser cette densité ou dans lesquels nous voulons générer des réalisations du processus suivant sa densité sont souvent de dimensions trop élevées.

C'est ici que les méthodes de Monte Carlo par Chaînes de Markov interviennent, notamment l'algorithme de Metropolis-Hasting-Green qui a été montré qu'il est beaucoup plus adapté aux processus ponctuels que celui de Naissance et de Mort ou celui de Metropolis-Hasting.

Néanmoins, une autre difficulté vient de s'ajouter à ce niveau, c'est celle de savoir quand est ce que les chaînes simulées convergent. La réponse est à chercher alors dans les méthodes développées dans cet objectif, à savoir les tests de diagnostics de la convergence des méthodes MCMC. Il est à souligner ici qu'il existe une possibilité de se passer des méthodes de diagnostics mentionnées en faisant recours à une méthode de simulation récemment développée et qui dispose d'un critère d'arrêt explicite. Cette méthode est dénommée *simulation exacte* (de l'anglais *exact sampling* ou encore *perfect simulation*) [124].

Nous avons terminé ce chapitre en présentant quelques exemples d'applications récemment introduites dans la littérature des processus ponctuels marqués. Ces applications ont couvert plusieurs domaines, mais un intérêt particulier à été accordé à celles relatives à l'analyse d'image, un domaine d'inspiration pour cette thèse.

Le chapitre suivant sera maintenant consacré à la présentation de notre contribution à la classification de données multidimensionnelles par les processus objets de Markov.

Chapitre 3

Clustering par Processus Objets

3.1 Introduction

Le présent chapitre aborde la première contribution de cette thèse qui consiste à développer une nouvelle méthode pour classifier automatiquement des données multidimensionnelles. Cette méthode est fondée sur les processus ponctuels marqués introduits dans le chapitre précédent 2.

Nous allons présenter, dans une première partie, notre modélisation de la procédure de la classification en employant les concepts de ces processus stochastiques. Nous appliquerons la méthode développée sur des ensembles de données synthétiques de structures différentes.

Dans la deuxième partie de ce chapitre, nous proposerons une méthode pour effectuer la classification supervisée, en considérant comme données d'apprentissage un ensemble d'observations prototypes obtenues à l'issue de la phase de la détection des modes précédente. Cette méthode sera encore testée sur des bases de données multidimensionnelles.

3.2 Détection des modes par processus objets

Dans cette section, nous allons essayer de répondre à la question de la modélisation du problème de la détection des modes par le moyen des processus ponctuels marqués. Pour ce faire, nous avancerons tout d'abord les hypothèses sur lesquelles se base cette modélisation.

Comme nous l'avons mentionné dans le chapitre 2, la méthode la plus générale pour construire un nouveau processus ponctuel marqué est la définition d'une densité

de probabilité par rapport à un processus ponctuel de Poisson ; cette densité permet de modéliser les interactions entre les points du processus, ce qui permet d'injecter les connaissances *a priori* dont on dispose. En outre, nous pouvons y incorporer également un terme qui contrôle les positions des points dans l'espace, c'est le terme *attache aux données*. Nous verrons comment définir de tels termes et par conséquent une telle fonction de densité du processus.

Une fois le processus construit, nous présenterons l'algorithme général de la détection des modes par les processus ponctuels marqués ; certainement, nous devrions dans cet objectif, adopter l'une des méthodes de Monte Carlo par Chaines de Markov présentées au cours du chapitre précédent 2 pour la simulation des processus et l'adapter à notre cas particulier d'étude en introduisant des noyaux de proposition convenables.

A la fin de cette section, des résultats expérimentaux seront présentés pour montrer l'efficacité de la méthode proposée.

3.2.1 Hypothèses de l'étude

L'extension des processus ponctuels marqués/objets à la classification non-supervisée de données de grandes dimensions s'effectue en adoptant deux principes sur lesquels se base souvent la procédure de détection d'objets dans un contexte d'images :

1. Le premier suppose que les objets complexes de l'image peuvent être reconstruits par des objets de base auxquels nous associons des relations *d'interactions* ;
2. Le deuxième principe postule que les objets sont distribués sur l'image selon un processus ponctuel marqué.

A ces deux principes, nous rajouterons les hypothèses suivantes qui vont nous permettre de faire l'analogie entre la détection d'objets dans le cadre de l'analyse d'images et la détection des classes dans le cadre de l'analyse de données en général :

1. Les *modes*, qui sont d'un point de vue analytique des maxima locaux de la fonction de densité de probabilité sous-jacente à la distribution des données dans l'espace, sont géométriquement des régions à forte concentration de points dans l'espace de représentation.
2. Une classe est représentée par un ensemble de points prototypes :

   En effet, les observations prototypes sont les observations qui se trouvent dans les régions modales de l'espace. Chaque classe est donc formée par des observations

prototypes et des observations non prototypes. Ces dernières sont une minorité par rapport aux prototypes.

3. Les modes sont assimilés aux objets d'intérêt que l'on cherche à détecter dans le cas de l'analyse d'image.

4. Détecter les modes est la première étape pour déterminer l'ensemble des individus d'une classe. La deuxième étape est la classification supervisée à partir d'un ensemble d'apprentissage qui sera donné par les observations prototypes.

Dans cette optique, la question qui se pose maintenant est celle de la forme géométrique que prennent les modes. La réponse à cette question est importante pour pouvoir définir des objets du coté du processus ponctuel marqué. Cette question nous invite à observer la structure des données dans l'espace des attributs. Or les données, dont on dispose généralement, n'ont pas de structures spatiales uniques. Elles sont des distributions aléatoires pouvant prendre n'importe quelle forme dans l'espace. De ce fait, il semble difficile de supposer une forme géométrique préalable pour ces données.

Néanmoins et grâce au premier principe que nous avons cité plus haut, nous pourrions concevoir que les modes sont en fin de compte des objets (au sens géométrique) complexes qui s'intersectent, c'est-à-dire qu'ils sont des composantes connexes (au sens topologique) formées par un ensemble d'objets.

Le problème se ramène donc au choix d'un objet (une forme géométrique) qui soit en mesure de prendre en considérations ces contraintes.

Par conséquent, nous avons choisi des hypersphères comme objets du processus. En plus que ce choix remplit les contraintes précédentes, il s'est justifié aussi par le fait que la seule marque que nous aurons à leur associer est le rayon ; ce qui signifie un gain en terme de complexité de l'algorithme que nous allons proposer.

Tout au long de ce chapitre, nous allons noter $hs = (c, r)$ pour désigner l'hypersphère hs centré en c et de rayon r ; par ailleurs , nous allons garder la notation x pour dire configuration d'objets.

Nous définissons à présent la fonction de densité du processus objet qui aura, comme réalisation, une configuration d'objets qui puissent se localiser au sein de ces modes. Cette densité est comme nous l'avons vu au chapitre précédent, peut s'exprimer sous la forme d'une fonction d'énergie :

$$p(x|\theta) \propto exp[-(U_d(x|\theta) + U_i(x|\theta))] \tag{3.1}$$

avec $U_d(x|\theta)$ et $U_i(x|\theta)$) sont respectivement le terme d'attache aux données et le terme d'interactions ; θ est le vecteur paramètre du modèle.

Nous expliciterons ces termes dans ce qui suit.

3.2.2 Modélisation des interactions entre objets

Définition des différentes interactions entre objets du processus : Dans le but de modéliser les interactions a priori entre les objets (hypersphères) du processus, nous spécifions les contraintes auxquelles ces objets sont soumis.

1. D'abord, nous voulons que le nombre de composantes connexes formées à partir des hypersphères reflète aux mieux le nombre de modes ;
2. Ensuite, nous voulons que les hypersphères s'intersectent, mais sans qu'elles se recouvrent ;
3. Enfin, nous voulons que le nombre moyen d'hypersphères générées par le processus soit raisonnable.

Ces contraintes vont être formulées au travers de fonctions d'interactions ; En adoptant son équivalence sous l'expression de Gibbs, elles peuvent être exprimées avec des potentiels d'énergie ; l'énergie d'interactions notées $U_i(x|\theta)$, peut alors s'exprimer de la façon suivante :

$$U_i(x|\theta) = n(x)log\beta + U_i^2(x|\theta) + U_i^n(x|\theta) \tag{3.2}$$

1. $n(x)log\beta$ est l'énergie d'un processus de Poisson de paramètre d'intensité β et $n(x)$ le nombre d'objets dans la configuration du processus, de sorte qu'il contrôle le nombre moyen des objets dans une configuration.
2. $U_i^n(x|\theta)$ est l'énergie d'un processus connu appelé processus de composantes connexes [28] et est donnée par :

$$U_i^n(x|\theta) = |c(x)|log\gamma \tag{3.3}$$

avec $c(x)$ l'ensemble de toutes les composantes connexes trouvées dans la configuration x et $|c(x)|$ son cardinal. Si $\gamma < 1$, la création de nouvelles classes est favorisée. Si par contre $\gamma > 1$, elle sera pénalisée.

3. $U_i^2(x|\theta)$ est l'énergie d'un processus par paires (*pairewise interaction process*. Elle est donnée par :

$$U_i^2(x|\theta) = \sum_{(x_i,x_j)\in x} \varphi(x_i, x_j) \tag{3.4}$$

$\varphi(x_i, x_j)$ est un potentiel d'interaction d'ordre deux.

$\varphi(x_i, x_j)$ sera défini en fonction de la distance entre les centres des objets. Nous avons défini et testé deux modèles pour ces interactions avant que nous définissions le troisième qui en apportent des améliorations.

Modèle 1 : Nous voulons dans cette modélisation de l'a priori, pénaliser les objets qui se chevauchent, favoriser ceux qui s'intersectent, mais aussi pénaliser ceux qui sont séparés d'une distance inférieure à un seuil donné. Nous formulons ces contraintes par les relations de voisinage suivantes :

- Relation de recouvrement : $x_i \sim_{rec} x_j$ si $d(c_i, c_j) < |r_i - r_j|$
- Relation d'intersection : $x_i \sim_{int} x_j$ si $|r_i - r_j| \leq d(c_i, c_j) \leq (r_i + r_j)$
- Relation de séparation : $x_i \sim_{sep} x_j$ si $d(c_i, c_j) > (r_i + r_j)$

où c_i et r_i (resp c_j et r_j) sont le centre et le rayon de l'hypersphère hs_i (resp hs_j) ; $d(,)$ est la distance euclidienne .

De ces définitions, nous exprimons alors l'énergie d'interactions par :

$$U_i^2(x|\theta) = \sum_{(x_i\sim_{rec}x_j)} \varphi_{rec}(x_i, x_j) + \sum_{(x_i\sim_{int}x_j)} \varphi_{int}(x_i, x_j) + \sum_{(x_i\sim_{sep}x_j)} \varphi_{sep}(x_i, x_j) \tag{3.5}$$

et chaque potentiel est défini par :

$$\begin{cases} \varphi_{rec}(hs_i, hs_j) &= log(\kappa) \\ \varphi_{int}(hs_i, hs_j) &= log(\varrho) \\ \varphi_{sep}(hs_i, hs_j) &= log(\eta) \end{cases} \tag{3.6}$$

Les quantités κ, ϱ et η sont des hyper-paramètres (ici des constantes strictement positives) qui spécifient les différents potentiels définis.

En somme, la densité de ce modèle sera de la forme :

$$p_i(x) \propto \beta^n \gamma^{-|c|} \kappa^{n_{rec}} \varrho^{n_{int}} \eta^{n_{sep}} \tag{3.7}$$

Modèle 2 : Ce modèle est pareil au précédent, sauf qu'on utilise des potentiels variables plutôt que des constantes :

$$\varphi_{rec}(hs_i, hs_j) = \frac{d(c_i, c_j)}{|r_i - r_j|} \tag{3.8}$$

$$\varphi_{int}(hs_i, hs_j) = \frac{d(c_i, c_j)}{(r_i + r_j)} \tag{3.9}$$

$$\varphi_{sep}(hs_i, hs_j) = \frac{d(c_i, c_j)}{2 \times r_{max} + (r_i + r_j)} \tag{3.10}$$

Une autre possibilité aurait été de pondérer chaque relation par une valeur, positive dans le premier et le dernier cas et négative dans le deuxième. Dans ce cas, nous aurions des potentiels de la sorte :

$$\varphi_{rec}(hs_i, hs_j) = log(\kappa) \times \frac{d(c_i, c_j)}{|r_i - r_j|} \tag{3.11}$$

$$\varphi_{int}(hs_i, hs_j) = -log(\varrho) \times \frac{d(c_i, c_j)}{(r_i + r_j)} \tag{3.12}$$

$$\varphi_{sep}(hs_i, hs_j) = log(\eta) \times \frac{d(c_i, c_j)}{2 \times r_{max} + (r_i + r_j)} \tag{3.13}$$

Modèle 3 : Ce modèle est une amélioration des premiers :

$$\varphi(hs_i, hs_j) = \begin{cases} 1 - \dfrac{d(c_i, c_j)}{max(r_i, r_j)} & \text{si} \quad d(c_i, c_j) \leq max(r_i, r_j) \\ 0 & \text{sinon.} \end{cases} \tag{3.14}$$

avec c_i and r_i (resp c_j and r_j) sont le centre et le rayon de l'hypersphère hs_i (resp hs_j). $d(,)$ est la distance euclidienne .

3.2.3 Modèle d'attache aux données

Le terme d'attache aux données contrôle la localisation des objets dans les régions de l'espace où nous avons la forme ou l'objet à détecter. Dans la littérature de l'ana-

lyse d'image par les processus objets, les modèles d'attache aux données développés exploitent les différences des niveaux radio métriques entre les pixels appartenant aux objets d'intérêt et ceux appartenant au fond de l'image. Nous nous sommes basés sur cette idée pour concevoir notre modèle d'attache aux données dans le contexte de la classification.

Nous allons utiliser un terme d'attache aux données du type *modèle détecteur* [48]. Dans ce modèle, l'énergie associée à une configuration d'objets x est obtenue comme la somme des énergies locales associées à chaque objet x_i de cette configuration :

$$U_d(x) = -\sum_{x_i \in x} U_d(x_i) \tag{3.15}$$

Modèle 1 : Pour notre cas d'étude, nous considérons qu'une hypersphère $hs(c, r)$ est bien positionnée si, tout d'abord, le nombre de points qu'il recouvre est supérieur à un seuil n_{min} donné ; Ce paramètre est déterminant dans la qualité de détection des modes. Deuxièmement, si la densité des points (observations) par rapport au volume de l'hypersphère hs n'est pas inférieure à une valeur d_0 préfixée.

$$U_d(hs_i|\theta) = \begin{cases} v - n & \text{si} \quad n > n_{min} \quad \text{ou} \quad n/\rho(hs) > d_0 \\ -v & \text{sinon.} \end{cases} \tag{3.16}$$

avec n et $\rho(hs)$ sont respectivement le nombre de points recouverts par l'hypersphère hs et le volume qu'il occupe. La valeur v est une valeur réelle positives fixée qui détermine la contribution de chaque objet de la configuration .

Modèle 2 : Dans ce modèle, nous déterminons pour chaque hypersphère de la configuration une autre hypersphère ayant le même centre mais un rayon un peu plus élevé (dans ce travail, nous avons pris le trois demi du rayon initial, voir la figure 3.1). Nous appellerons le volume entre les deux objets une ombre. Nous dirons alors qu'une hypersphère est bien positionnée si, d'une part, le nombre de points n_{in} qu'elle recouvre est supérieur à un seuil n_{max} donné ; et d'autre part, si la densité de points d_{in} à l'intérieur de l'objet est beaucoup plus grande que celle d_{out} calculée dans son ombre.

$$U_d(hs_i|\theta) = \begin{cases} (n_{in} - n_{out}) & \text{si} \quad n_{in} \geq n_{max} \quad \text{et} \quad d_{in} \geq d_{out} \\ u_{max} & \text{sinon.} \end{cases} \tag{3.17}$$

où u_{max} est une valeur réelle positive fixée et n_{out} le nombre d'observations dans l'ombre de l'hypersphère hs_i.

Modèle 3 : Dans ce troisième modèle, nous présentons une autre façon de calculer le terme d'attache aux données. Nous préciserons toujours deux conditions à remplir pour qu'un objet $hs(c_i, r_i)$ soit bien placé :

1. Le nombre n_i d'observations qu'il couvre doit être supérieur à un seuil N_{max} qui se calcule de la façon suivante :

   Au début de la procédure du calcul de la contribution de chaque objet, la valeur de N_{max} est réglée sur le nombre d'observations recouvert par le premier objet généré. Ensuite, il sera mis à jour lors du calcul de la contribution de l'objet suivant. Ainsi, pour un objet donnée $hs(c_i, r_i)$, N_{max} sera égale à :

   $$N_{max} = max\{n_j, j < i\} \tag{3.18}$$

   Toutefois, nous avons remarqué que cette condition pénalise fortement la majorité des objets créés surtout lorsque l'on est devant une structure de données dont les classes présentent des densités trop variées. Par conséquent, nous avons pensé à introduire un paramètre que nous appelons *sensibilité* et dont l'objectif est de diminuer l'effet pénalisant. Nous désignons ce paramètre par s.

   Cette première condition s'exprime alors comme suit :

   $$\mathbb{1}_{'nbrObs'} = \mathbb{1}\left[n_i \geq s \times N_{max}\right] \tag{3.19}$$

2. La densité des observations de l'intérieur de l'objet vers l'extérieur est spatialement décroissante :

   En effet, les modes des classes (selon la vision qu'ils sont des maxima locaux) ont la particularité qu'en se déplaçant de leurs centres vers les marges, la densité des observations diminue et inversement. Nous voulons alors exploiter ce constat pour déterminer les objets qui se localisent sur ces régions modales.

   Pour remplir cette condition, nous divisons chaque objet (hypersphère) en trois sections, et nous aurons alors trois sous-hypersphères. Les densités à l'intérieur de chacune de ces dernières vont ensuite être calculées et les objets pour lesquels ces trois densités décroissent seront considérés comme avoir rempli la deuxième condition de bien positionnement. Nous pouvons reformuler cette condition comme suit :

   $$\mathbb{1}\{'density'\} = \mathbb{1}\left[dens_1 \geq dens_2\right] \times \mathbb{1}\left[dens_2 \geq dens_3\right] \tag{3.20}$$

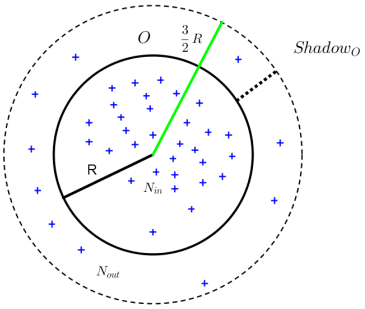


FIGURE 3.1 – Illustration d'un objet (cercle) et de son ombre.

Afin de construire le terme d'attache aux données de ce modèle, nous comptons le nombre d'objets bien positionnés et celui des objets mal positionnés que l'on note respectivement n_w et n_{nw}. Ces quantités vont être utilisées avec deux hyper-paramètres δ et ζ constituant ainsi le potentiel de l'énergie d'attache aux données.

En somme, l'apport d'énergie d'une configuration donnée est définie selon les conditions 1 et 2, comme étant :

$$U_d(x|\theta) = n_w \times log(\delta) - n_{nw} \times log(\zeta) \tag{3.21}$$

Nous verrons comment ajuster ces paramètres dans la section 3.2.4.3.

Il est à noter ici, avant de passer à la procédure de simulation, que ce sont les derniers modèles de chaque termes (terme d'interaction et d'attache aux données) que nous retenons pour le reste de ce mémoire.

3.2.4 Simulation du processus construit

3.2.4.1 Définition des noyaux de transition

Notre modélisation par les processus ponctuels marqués de la problématique de la classification de données multidimensionnelles est maintenant spécifiée par la fonction de densité $p(x|\theta)$ que l'on a construite. Nous cherchons alors à obtenir des configurations d'objets contrôlés par la loi π de ce processus. La solution est d'utiliser les dynamiques de Metropolis Hastings. Plus particulièrement, nous employons l'algorithme de Metropolis Hastings Green, qui est également appelé Reversible Jump Markov Chain Monte Carlo (RJMCMC) [1]. Le principal avantage de cette technique, est qu'il permet de se déplacer entre les sous espaces dont la dimension varie d'un état de la chaîne de Markov

1. Nous avons déjà décrit cet algorithme dans le chapitre précédent 2.

à un autre. Ceci nous convient bien vu que le nombre d'objets dans les configurations engendrées varient incessamment.

Notre objectif dans les paragraphes qui suivent est d'adapter cet algorithme d'échantillonnage à notre cas d'étude qui est celui de la détection des modes. Pour ce faire, nous devrions définir des transitions ou noyaux de proposition plus adéquates avec ce problème. En effet, un choix judicieux de ces noyaux de proposition permet d'accélérer la convergence de la chaine et donc de réduire le temps de calcul.

Cela signifie que tout en tenant compte de l'état s de la chaîne de Markov, nous devrions choisir au hasard un mouvement m afin d'apporter la chaîne dans le nouvel état $s^{'}$ et cela se fait avec une densité de probabilité $Q_m(s, s^{'})$.

Chaque mouvement défini doit permettre le mouvement inverse afin de garantir la condition de réversibilité nécessaire pour la convergence de la chaîne. Un autre point avant de présenter ces mouvements, est la façon de sélectionner l'un d'eux à une itération donnée. Il y a une grande liberté à ce propos, mais nous préférons donner avantage à certains mouvements toujours dans le but d'accélérer la convergence en pratique.

Comme nous avons déjà vu au chapitre précédent, le passage d'un état à un autre est accepté selon une certaine probabilité $\alpha = min\{1, R_m\}$, avec R_m le rapport de Green donné par :

$$R_m = \frac{\pi(s^{'})Q_m(s^{'}, s)}{\pi(s)Q_m(s, s^{'})} \tag{3.22}$$

avec $\pi(.)$ la distribution du processus ponctuel de densité $p(.)$ par rapport à une mesure de référence.

Les types de mouvements conçus dans le présent document comprennent :

- Naissance / Mort uniforme d'une hypersphère.
- Augmentation / réduction du rayon d'une hypersphère.
- Déplacement d'une hypersphère.

Chaque mouvement doit avoir son inverse, ceci permet d'assurer la condition de réversibilité de la chaine.

- Naissance d'une hypersphère : avec une probabilité P_n, nous tirons aléatoirement et uniformément un point de l'espace et une valeur pour le rayon dans $[r_{min}, r_{max}]$.

$$Q_n(s, s^{'}) = P_n \times \frac{1}{\nu(U)} \times \frac{1}{r_{max} - r_{min}} \tag{3.23}$$

$$Q_m(s^{'}, s) = P_m \times \frac{1}{n+1} \tag{3.24}$$

$$R_n = \frac{P_m}{P_n} \times \frac{r_{max} - r_{min}}{n+1} \times \nu(U) \times \frac{p(x \cup \{\xi\})}{p(x)} \tag{3.25}$$

- Mort d'une hypersphère : C'est le mouvement opposé du précédent, il s'agit de la suppression d'une hypersphère tirée aléatoirement et uniformément de la configuration actuelle avec une probabilité P_m.

$$R_d = \frac{P_n}{P_m} \times \frac{n}{r_{max} - r_{min}} \times \frac{1}{\nu(U)} \times \frac{p(x/\{\xi\})}{p(x)} \tag{3.26}$$

- Augmentation du rayon d'une hypersphère : soit P_{ag} et P_{rd} la probabilité de prolonger respectivement de réduire le rayon d'une hypersphère ; uniformément, nous choisissons un objet et nous lui modifions le rayon de telle sorte que :

$$Q_{ag}(s, s^{'}) = P_{ag} \times \frac{1}{n} \times \frac{1}{r_{max} - r} \tag{3.27}$$

$$Q_{rd}(s^{'}, s) = P_{rd} \times \frac{1}{n} \times \frac{1}{r - r_{min}} \tag{3.28}$$

$$R_{ag} = \frac{P_{rd}}{P_{ag}} \times \frac{r_{max} - r}{r - r_{min}} \times \frac{p(x')}{p(x)} \tag{3.29}$$

- Réduction du rayon d'une hypersphère : uniformément, nous choisissons un objet et nous lui modifions le rayon de telle sorte que :

$$R_{rd} = \frac{P_{ag}}{P_{rd}} \times \frac{r - r_{min}}{r_{max} - r} \times \frac{p(x')}{p(x)} \tag{3.30}$$

- Déplacement d'une hypersphère : avec une probabilité P_d, nous choisissons de déplacer une hypersphère de la configuration en lui changeant la position de son centre dans un voisinage ; ce mouvement s'effectue dans le même sous-espace, donc le rapport de Green devient un rapport de Hasting :

$$R_d = \frac{p(x')}{p(x)} \tag{3.31}$$

3.2.4.2 Simulation du modèle a priori

A présent, nous avons défini les différents noyaux de proposition nécessaire pour l'échantillonnage du processus du modèle proposé. Nous décrirons sa procédure dans ce qui suit.

Nous commençons par simuler tout d'abord le modèle de l'a priori, c'est-à-dire sans tenir en considération le terme d'attache aux données. Nous prenons dans ce modèle, le terme d'intensité plus le terme du processus par composantes connexes et le troisième modèle (modèle 3) d'interaction par paires d'objets. Ainsi, aurons-nous le modèle suivant :

$$p(x|\theta) \propto \beta^n \gamma^{-|c|} \prod_{x_i,x_j} \phi(x_i, x_j) \tag{3.32}$$

La procédure de simulation s'effectue en faisant tourner l'échantillonneur Metropolis Hasting Green pour un nombre fixé d'itérations (e.g 300 itérations) ; ces itérations sont intégrées dans une autre boucle d'itérations relative au schéma du recuit simulé. Nous ajustons ce dernier à 1000 itérations.

Par ailleurs, nous avons vu au chapitre précédent que savoir quand est ce que la chaine a convergé n'est pas si évident pour qu'on puisse fixer le nombre maximal d'itérations en amont ; nous faisons alors recours à la méthode Geweke dédiée au diagnostic de convergence 2.4.2. Cette technique nous permettra de définir un critère d'arrêt et donc stopper la simulation une fois que le critère est vérifié. A la fin de la simulation, nous afficherons graphiquement les courbes de certaines statistiques appelées *statistiques suffisantes* pour s'assurer visuellement de la convergence ; pour cela, nous retenons les valeurs de ces statistiques toutes les 100 itérations afin de réduire la corrélation entre les échantillons et nous traçons leurs moyennes cumulées. En ce qui concerne la période d'échauffement de la simulation, nous écarterons les 30% premières configurations.

Nous initialisons la simulation par une configuration de 30 objets. Pour le schéma de diminution de la température, nous avons opté pour :

$$T_i = \frac{T_0}{1 + log(i)} \tag{3.33}$$

Il est à rappeler ici que la température T dans le schéma du recuit simulé, affecte chaque fonction de potentiel sauf celle de l'intensité :

$$p_T(x|\theta) \propto \beta^n exp(\frac{-(U_i^2(x|\theta) + U_i^n(x|\theta))}{T}) \tag{3.34}$$

3.2.4.3 Ajustement des paramètres

Les paramètres d'un modèle sont des valeurs numériques qui ne sont pas des variables à entrer et qui ne sont pas calculées explicitement par le modèle. L'estimation de ces paramètres est une tache très importante et est d'un impact majeur sur l'efficacité de n'importe qu'elle méthode paramétrique de classification. Bien qu'il existe des méthodes dans la littérature, comme celle de l'estimation par espérance-maximisation (EM) que nous avons présentée précédemment, cette question est toujours un champ de recherche notamment dans le contexte des processus ponctuels marqués.

Les méthodes utilisant les processus ponctuels marqués dont nous en avons cité quelques unes dans le chapitre antérieur, ont ajusté leurs paramètres soit manuellement, c'est-à-dire par essais et erreurs, soit elles les estiment conjointement avec la configuration des objets recherchées en utilisant la méthode espérance-maximisation [38; 65; 147].

Dans ce travail de thèse, nous ne sommes pas pour l'instant arrivés à les estimer, par conséquent nous nous contentons de les fixer empiriquement par essais-erreurs.

Nous avons trois types de paramètres : ceux du terme d'interaction, ceux du terme d'attache aux données et ceux que l'on appelle *méta-paramètres*, à savoir : r_{min}, r_{max} et s.

Le vecteur paramètre de notre modèle proposé est donné par :

$$\theta = \{\gamma, \delta, \zeta, s, r_{min}, r_{max}\}$$

Paramètre γ : Le premier paramètre à régler est celui qui pénalise le nombre de composantes connexes dans la configuration x. En effet, pour $\gamma > 1$, le processus tend à pénaliser les configurations avec plusieurs composantes connexes, ce qui va ne garder que les composantes connexes dont l'attache aux données est très forte. Dans nos expériences, nous avons fixé ce paramètre sur la valeur $\gamma = 30$.

Paramètres δ et ζ : ce sont les hyper-paramètres associés au potentiel du terme d'attache aux données. Ils contrôlent les positions des objets de la configuration pour qu'ils se placent sur les régions modales. Dans nos expériences, nous avons pris $\delta = 15$. Pour le paramètre d'intensité des objets mal positionnés ζ, nous l'avons fixé sur $\zeta = 1000$ pour donner une faible contribution de ce type d'objets à l'énergie du modèle.

Paramètre s : Cet hyper-paramètre, *sensibilité*, est introduit au niveau du calcul de la contribution de chaque objet à l'énergie d'attache aux données. Il permet de diminuer

l'impact de N_{max}. En effet, les données analysées sont souvent formées de classes de densités différentes, N_{max} est une contrainte très forte qui croit au fil des itérations, ce qui ne permet au processus de ne garder que les objets dont le nombre de points qu'ils recouvrent lui est supérieur. En conséquence, le processus aura tendance à pénaliser les configurations à plusieurs classes. De ce fait, le paramètre s va diminuer l'effet de N_{max}. Dans les expériences menées dans le cadre de cette thèse, s prend des valeurs dans $\{0.1, 0.25, 0.5, 0.75, 1\}$ afin de s'adapter aux différentes structures de données.

Paramètres r_{min} et r_{max} : Pour le paramètre r_{min}, il est fixé comme étant la distance minimale entre toutes les paires d'observations. On a :

$$r_{min} = min\{d(x_i, x_j), \forall x_i, x_j \in x\} \tag{3.35}$$

Il aurait été possible de ne pas définir ce paramètre et laisser le processus générer des objets avec des rayons aléatoires quelconques ; cependant, le préciser est une façon d'accélérer la convergence de la chaine en évitant de générer trop d'objets qui vont être refusés par la suite.

Quant au paramètre r_{max}, on l'estime en utilisant l'étendu des projections des données sur chaque dimension pondéré par 0.04 (ou par 0.03 pour des donnée très enchevêtrées).

Ce qui peut être formulé de la façon suivante :

$$r_{max} = \sqrt{\sum_{j=1}^{p} (0.04 \times (M_j - m_j))^2} \tag{3.36}$$

avec $M_j = max_i\{x_{ij}\}$ et $m_j = min_i\{x_{ij}\}$, p le nombre d'attributs décrivant les données (la dimension de l'espace attribut) , x_{ij} la valeur du $i^{éme}$ individu pour le $j^{éme}$ attribut.

3.2.4.4 Algorithme de détection des modes par processus objets

Après avoir présenté les différents aspects de la méthode proposée pour classifier des données multidimensionnelles par les processus objets, nous récapitulerons cela sous la forme algorithmique suivante :

Algorithm 1 Algorithme de détection des modes par processus objets

Require: $\beta, \gamma, \delta, \zeta, N_{it}, r_{min}, r_{max}$
Ensure: Π_k
1: Initialiser par $x^{(0)}$
2: **for** $i = 0$ **to** Nit **do**
3: Choisir un mouvement : m_n, m_m, m_{pr}, m_{rd}, m_d
4: Simuler une nouvelle configuration $x^{(i)}$ selon le mouvement m et le noyau $Q_m(.,.)$
5: Calculer le rapport de Green R_m associé à chaque mouvement m .
6: Avec une probabilité $\alpha = min(1, R_m)$, accepter cette configuration
7: **end for**
8: Maintenir la configuration optimale $x^{(f)}$
9: Extraire les modes Π_k

3.2.5 Résultats expérimentaux

Dans le domaine de l'apprentissage machine, chaque méthode de classification doit être testée et évaluée afin d'en mesurer ses performances et pouvoir la comparer avec d'autres méthodes compétitives. Une telle évaluation devrait reposer sur des critères objectifs d'évaluation. Bien que plusieurs critères aient été utilisés dans la littérature, le critère qui minimise l'erreur de généralisation reste le plus fréquemment employé. Ce critère indique la capacité de généralisation d'une méthode. Il mesure les écart entre les valeurs de sortie, celles qui sont fournies par le modèle, et les vraies valeurs de l'ensemble de données entrées.

Pratiquement, cet erreur est calculé par l'utilisation de la méthode dite *validation croisée* (*cross-validation*) et qui se base sur les matrices de confusion.

Pour ce faire, nous devons avoir un ensemble de données sur lesquelles nous disposons de connaissances explicites. Ces dernières concernent particulièrement le nombre de classes, leurs structures et notamment l'étiquette de chaque individu.

Dans cette section, nous présenterons quelques résultats expérimentaux qui montrent la méthode proposée pour la classification de données en utilisant les processus ponctuels marqués. Nous avons essayé de travailler sur un ensemble de données de différentes structures allant du plus simple au plus complexe.

Pour classifier les autre points non prototypes, nous allons employer la méthode des k-plus proche voisins en fixant k sur 1.

TABLE 3.1 – Résultats statistiques de l'exemple de données gaussiennes.

	Mean-Shift			k-means			Méthode proposée		
Taux d'erreur	10%			2.75%			2.25%		
Matrice de confusion	399	1	0	387	3	10	388	2	10
	53	341	6	4	389	7	4	394	2
	60	0	340	6	3	391	6	3	391

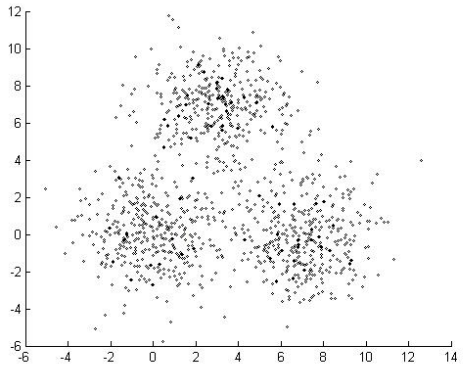
3.2.5.1 Données gaussiennes

Le premier exemple sur lequel nous appliquerons notre méthode est constitué d'un ensemble de données générées artificiellement suivant trois distributions de lois gaussiennes $N(\mu_i, \sigma_i^2)$. Chaque distribution a donc deux paramètres : une moyenne μ_i et une variance σ_i^2. La distribution de l'ensemble est par conséquent composée de trois classes distinctes.

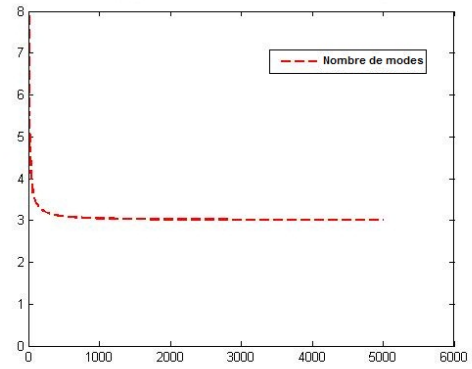
Contrairement à certaines méthodes (le k-means par exemple), notre méthode a réussi tout d'abord à découvrir automatiquement le nombre de classes cachées dans les données brutes ; aussi, une comparaison en terme des taux d'erreur avec la méthode des k-moyennes et la méthode *mean-shift*, a abouti à une valeur minimale en faveur de notre méthode (cf. tableau 3.1).

La figure 3.2 montre les résultats obtenus en appliquant notre méthode sur ce premier ensemble de données synthétiques. Sur la figure 3.2.(a), l'ensemble des données brutes est présenté ; la configuration finale du processus objet est montrée sur la figure 3.2.(c). Les objets ici sont donnés par des cercles vu que nous travaillons sur un espace bidimensionnel. Nous voyons que les objets sont bien positionnés sur les trois régions modales de l'espace en formant des composantes connexes. La figure 3.2.(b) confirme ce résultat en montrant la stabilité de la courbe qui représente l'évolution du nombre de modes détectés (classes) sur la valeur trois. L'extraction des composantes a permis par la suite la détection des modes sous-jacents aux données. Les observations appartenant à chaque modes sont enfin extraites en donnant les points prototypes 3.2(e).

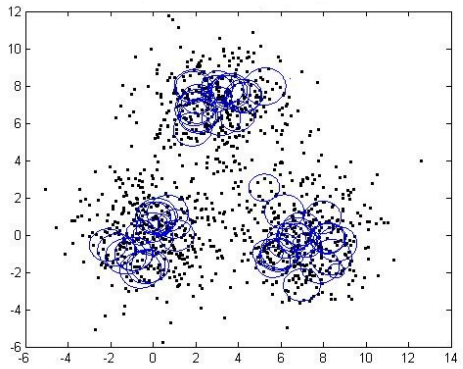
En ce qui concerne les statistiques suffisantes, nous voyons sur la figure 3.2.(d) les courbes correspondant aux nombre d'objets totale dans la configuration (60 objets), les effectifs d'objets bien positionnés (59 objets) et mal positionnés (un seul objets), ainsi que la courbe représentant l'effectif des paires d'objets qui se stabilise sur la valeur 100.



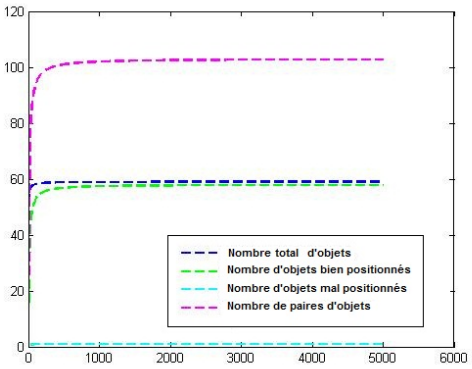
(a) Données brutes



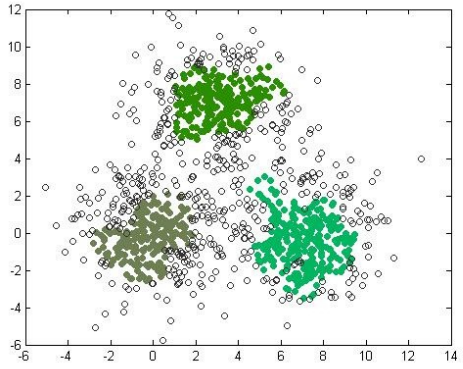
(b) Evolution du nombre de modes détectés



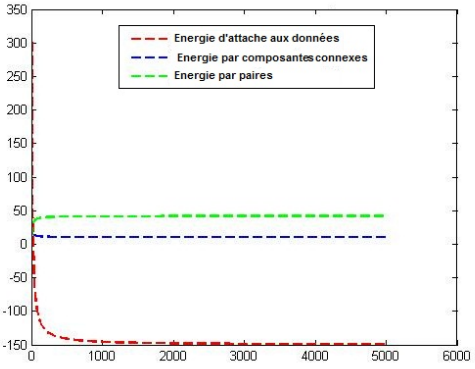
(c) Configuration obtenue par le processus objet



(d) Evolution des statistiques suffisantes



(e) Modes détectés



(f) Evolution des énergies du processus

FIGURE 3.2 – Résultats de la détection des modes issus de l'application de la méthode proposée sur un ensemble de données gaussiennes.

3.2.5.2 Données non linéairement séparables

Pour tester la méthode proposée sur un ensemble de données avec une structure de données un peu plus difficile, nous proposons de l'appliquer sur un ensemble de données qui ne sont ni sphériques ni linéairement séparables. Il s'agit d'un ensemble de données formé de deux classes synthétiques qui sont engendrées suivant une distribution non usuelle. Chaque classe à une forme géométrique d'un croissant comme le montre la figure 3.3.(a). Les caractéristiques A_1 et A_2 décrivant les observations de ces deux distributions sont données par :

$$A_1 = A'_1 cos\Psi + B_1$$

$$A_2 = A'_2 cos\Psi + B_2$$

où Ψ est une variable aléatoire normale de moyenne m et de variance s, et $B_{i=1,2}$ est une variable aléatoire normale de moyenne $\mu_{i=1,2}$ et de variance $\sigma_{i=1,2}$ (Table 3.2).

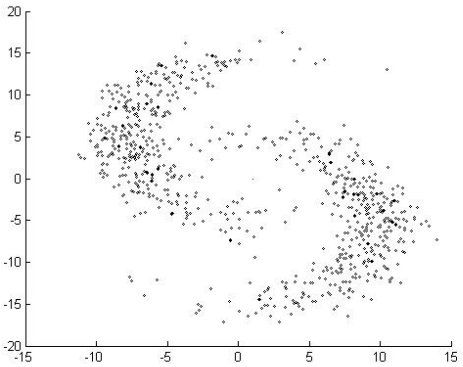
Nous pouvons voir sur la figure 3.3.(c) la configuration optimale du processus objet ainsi que les deux régions modales détectées. La figure 3.3.(e) montre ensuite les observations prototypes et non prototypes issues de l'extraction des deux composantes connexes détectées.

Une comparaisons avec les résultats obtenus par la méthode classique k-means appliquée sur cet ensemble de données, démontre bien l'efficacité de notre méthode. D'abord, il faudra fixer le nombre de classes pour le k-means (k=2) ; ensuite, cette méthode tend à séparer l'espace attribut en deux parties suivant les barycentres des éventuelles classes. Cela conduit à une classification erronée d'un nombre considérable d'observations, notamment celles appartenant aux extrémités des deux croissants.

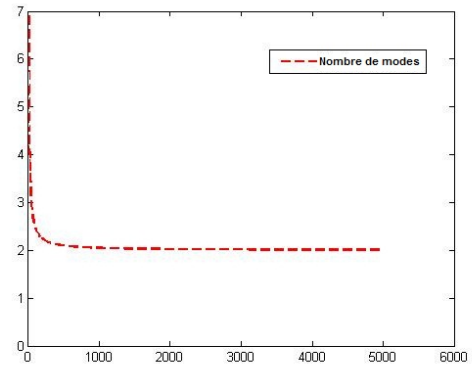
En revanche, notre méthode vient de localiser automatiquement les modes des deux classes cachées dans les données tout en réduisant fortement le nombre d'observations mal classées.

TABLE 3.2 – Paramètres statistiques des deux distributions bidimensionnelles à partir desquelles ont été tirées les observations de l'exemple du jeu de données non-linéairement séparables.

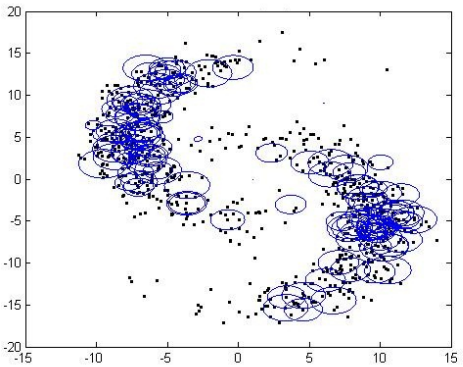
	Ψ	B_1	B_2	A'_1	A'_2
Population 1	m=0 s= 45	$\mu_1 = 0\, \sigma_1 = 2$	$\mu_2 = 5\, \sigma_2 = 2$	10	10
Population 2	m=180 s= 45	$\mu_1 = 2\, \sigma_1 = 2$	$\mu_2 = 5\, \sigma_2 = 2$	10	10



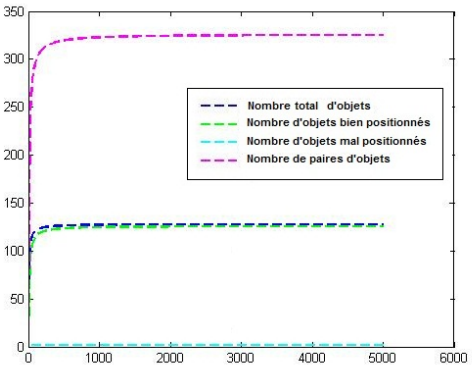
(a) Données brutes non-linéairement séparables



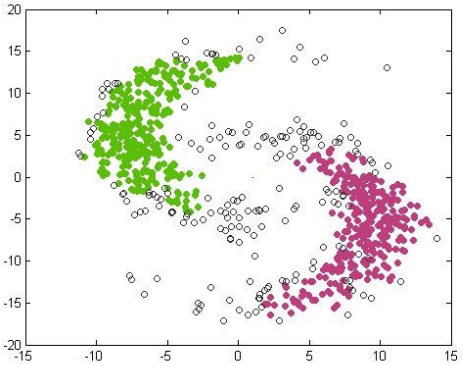
(b) Évolution du nombre de modes détectés



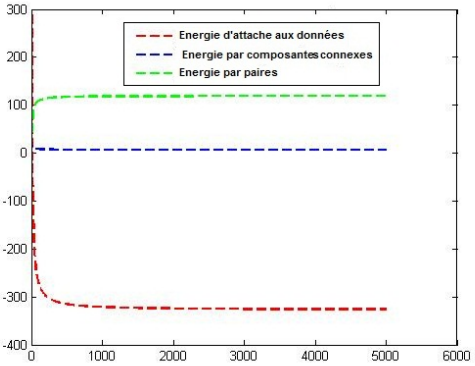
(c) Configuration d'objets obtenue par le processus



(d) Évolution des statistiques suffisantes



(e) Modes extraits



(f) Évolution des énergies du processus

FIGURE 3.3 – Exemple d'application de la méthode proposée sur un type de données de structure non-linéairement séparable.

3.2.5.3 Données enchevêtrées

Nous augmentons la difficulté du contexte d'évaluation de notre méthode en proposant dans ce paragraphe de l'évaluer dans le contexte de données enchevêtrées, l'un des défis majeurs et souvent envisagés dans la problématique de la classification. Les données enchevêtrées se caractérisent par le fait que les classes ne sont pas très isolées les unes des autres, mais au contraire elles sont beaucoup plus proches de sorte que certaines observations semblent appartenir non pas à une seule classe, mais plutôt à plusieurs à la fois.

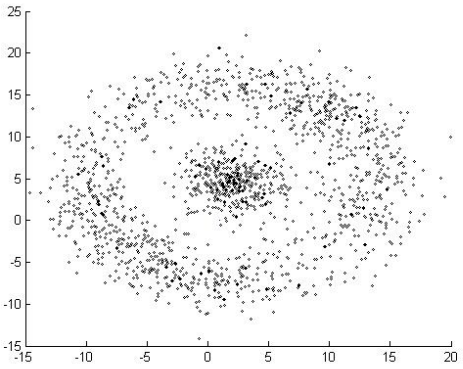
L'ensemble de données présenté est constitué de trois classes distribuées dans un espace bidimensionnel. L'une des classes a une structure sphérique et elle est générée suivant une distribution de loi normale centrée réduite ; les deux autres sont d'une forme de croissant (arc de cercle). Les statistiques des trois classes sont données dans la table et elles ont été fixées de telle façon que la structure finale de l'ensemble garde la propriété de chevauchement. La propriété de la non-séparabilité linéaire a été aussi gardée.

La figure 3.4, nous montre comme précédemment les données brutes, la configuration optimale du processus et les modes détectés. Il est à noter à propos de cet exemple que la valeur du seuil de sensibilité s affecte le nombre de modes et la qualité de la détection en terme d'observations prototypes. Pour $s = 0.5$ le seul mode détecté est celui relatif à la classe gaussienne (au centre) ; pour $s = 0.25$ les trois modes sont détectés mais le nombre d'observation prototypes n'est pas optimal. Pour une valeur de $s = 0.15$, les trois modes sont bien localisés et l'effectif des observations prototypes augmente.

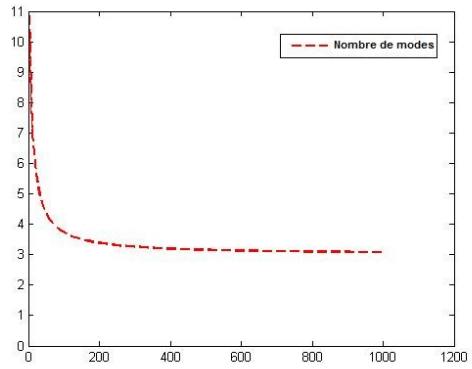
3.2.5.4 Données tridimensionnelles

La généralisation de notre méthode à des espaces attribut de plus que deux dimensions ne pose aucun problème particulier. En effet, cette méthode se base d'une part sur les processus ponctuels marqués qui peuvent être définis sur des espace de dimensions infinies et d'autre part sur une modélisation des classes en termes de modes. La seule contrainte est l'augmentation de la complexité algorithmique.

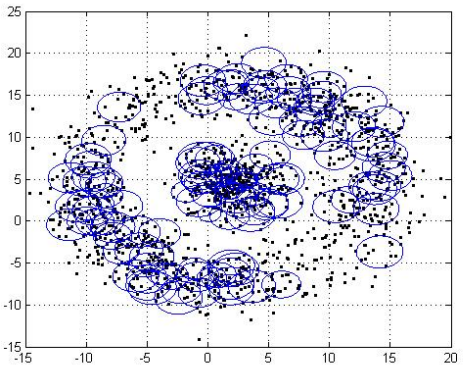
Nous proposons alors d'appliquer la méthode proposée sur deux exemples tirés de deux ensembles de données générées artificiellement dans un espace tri-dimensionnel. Le premier ensemble est formé d'une simple distribution de trois classes générées selon une loi normale (Figure 3.5(a)). Le deuxième ensemble est, par contre, formé d'une distribution de deux classes de structure anneau chacune. De plus, nous avons fait que ces deux anneaux s'intersectent afin que l'on rajoute la propriété de la non-séparabilité



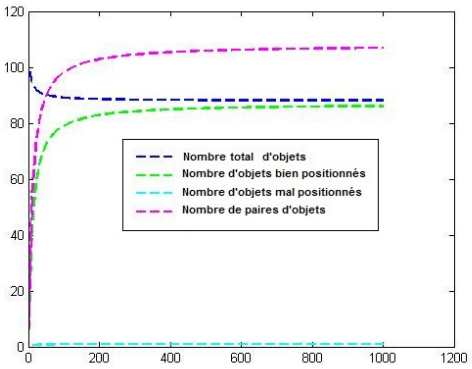
(a) Données brutes enchevauchées



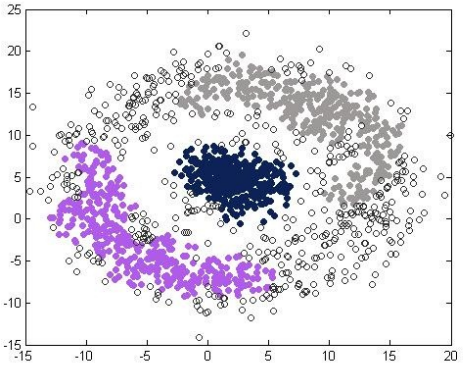
(b) Évolution du nombre de modes détectés



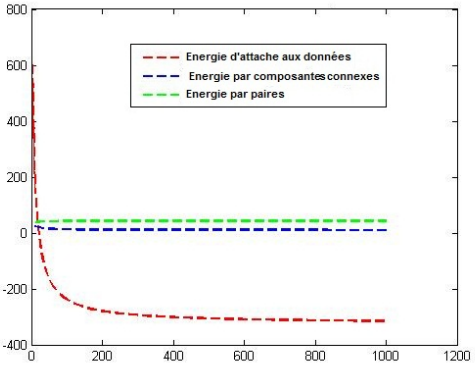
(c) Configuration d'objets obtenus par le processus



(d) Évolution des statistiques suffisantes

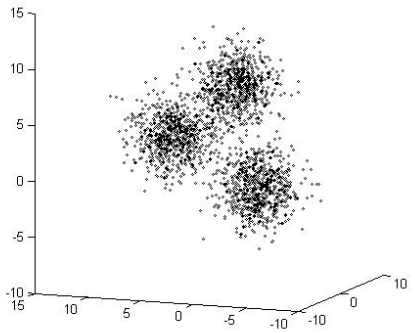


(e) Modes extraits

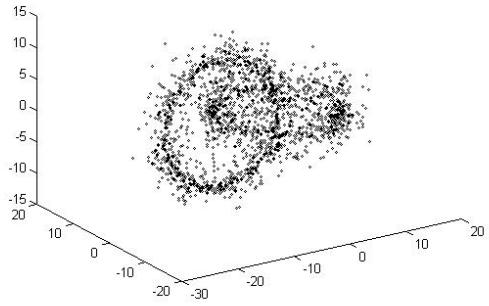


(f) Évolution des énergies du processus

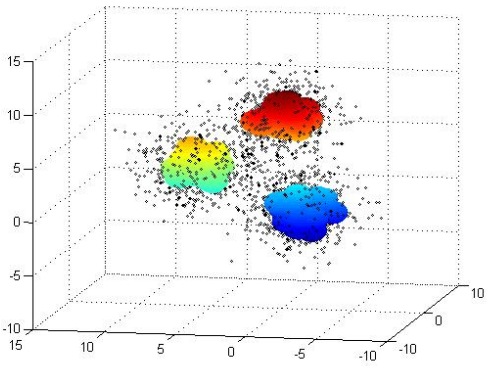
FIGURE 3.4 – Résultats de la détection des modes par la méthode développée sur un ensemble de données présentant un degré de chevauchement.



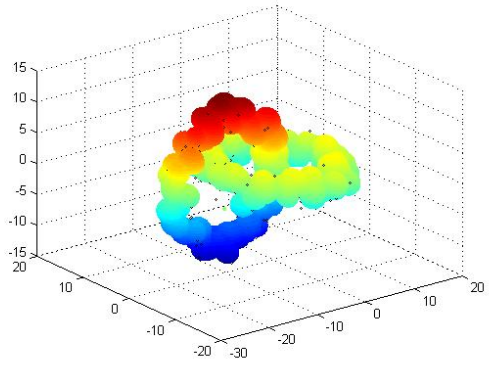
(a) Données brutes gaussiennes tridimensionnelles



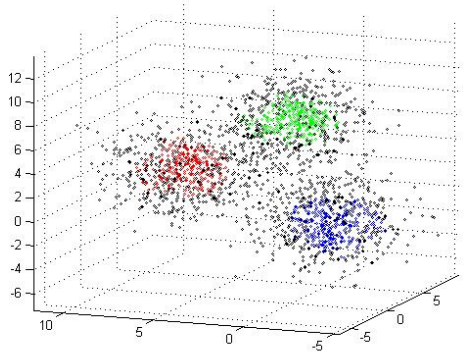
(b) Données brutes en structure d'anneaux interconnectés



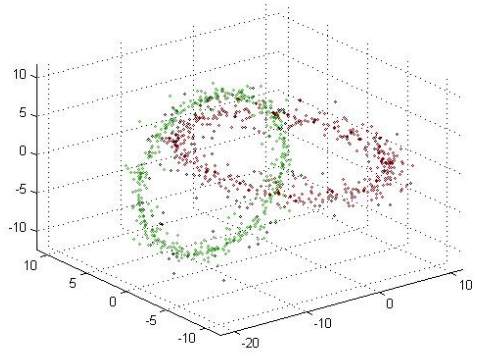
(c) Réalisation du processus objet sur les données gaussiennes



(d) Réalisation obtenue sur les données anneaux



(e) Modes extraits des données gaussiennes



(f) modes extrait ayant la même structure que les données brutes

FIGURE 3.5 – Application de la méthode proposée sur deux structures différentes de données dans un espace en trois dimensions.

linéaire.

Les données de cet exemple présentent aussi un niveau de chevauchement remarquable (Figure 3.5(b)).

La figure 3.5 montre les modes détectés dans les deux exemples. Les sphères (objets du processus marqué) des deux configurations optimales obtenues par la simulation du processus objet sont bien localisées sur les régions modales. L'extraction par la suite des composantes connexes a permet d'identifier les observations prototypes et non-prototypes comme l'illustrent les figures 3.5(e) et 3.5(f). Les résultats sont encourageants pour les deux exemples.

L'objectif de la méthode proposée jusqu'à maintenant dans ce mémoire était précisément la détection des modes de la fonction de densité de probabilité (approximation de la fdp) sous-jacente à un ensemble de données multidimensionnelles. Or, comme nous avons déjà mentionné, ceci n'est que la première étape de notre procédure de classification automatique. Par ailleurs, la deuxième étape émane du problème de la classification supervisée et peut s'effectuer par l'une des méthodes classiques une fois les modes détectés. Toutefois, nous avons développé une nouvelle méthode qui effectue le travail en se basant sur les champs de Markov. C'est ce que nous allons présenter dans le reste de ce chapitre.

3.3 Classification à partir des prototypes

Dans cette section, nous présenterons la deuxième contribution de cette thèse. Il s'agit d'une méthode de classification supervisée. Comme nous l'avons évoqué dans la section 1.4 du premier chapitre, les méthodes de cette catégorie de classification nécessite un ensemble d'apprentissage formé de données complètes. Cela signifie que nous devons disposer d'un ensemble d'individus ainsi que leurs étiquettes de classes. Cet ensemble peut être soit fourni sous forme de données brutes que l'on connait par expertise du domaine, soit un ensemble de prototypes issus d'une première phase d'une méthode de classification automatique. La méthode que nous venons de proposer dans la partie précédente de ce chapitre est un exemple de ce dernier cas : la classification à partir des prototypes.

Pour une raison d'homogénéité, nous continuons cette deuxième partie en utilisant la même terminologie que celle de la partie précédente, c'est-à-dire avec l'emploi des termes de modes, observations prototypes et observations non prototypes.

Lors d'une méthode d'apprentissage supervisé, chaque observation non prototype, c'est-à-dire qui n'appartient à aucune région modale, doit être affectée à l'un des modes

détectés en utilisant uniquement l'information qui peut être extraite de l'ensemble des prototypes ([42]).

Cependant, nous avons constaté que les procédures proposées dans la littérature ont tendance à générer des classes sphériques, même lorsque les formes des modes ou les structures de la distribution des prototypes sont loin de l'être.

Dans ce qui suit, nous introduisons une méthode qui remédie à cet inconvénient. Cette méthode se base sur les concepts des champs de Markov qui prend en compte les informations contextuelles attachées aux données ainsi que la forme complexe de leur distribution. Nous nous sommes inspirés des méthodes développées dans le contexte de l'analyse d'images que nous avons citées dans le chapitre 1.3.6.2, en particulier les travaux de [115; 137].

Avant de commencer à décrire la procédure proposée dans la section 3.3.2, nous fixons quelques notations que nous utiliserons dans la suite. Puis la section 3.3.3 présente l'algorithme de la méthode ainsi que le réglage des différents paramètres. Quelques résultats expérimentaux vont être présentés dans la section 3.3.5, en utilisant des données artificielles.

3.3.1 Notations

Soit $\Pi_k = (P_k^1, \ldots, P_k^n, \ldots, P_k^{N_k})$ le mode (l'ensemble des N_k prototypes) de la classe Δ_k, $k = 1, 2, \ldots, K$ où K est le nombre de classes. Chaque prototype de classe Δ_k est défini par ses D coordonnées $P_k^n = [p_{k,1}^n, \ldots, p_{k,d}^n, \ldots, p_{k,D}^n]^T$ dans $\mathbb{R}^D$ l'espace de données, où l'exposant T indique l'opérateur transposé.

L'ensemble des observations non classées est noté $\Gamma = (Q_1, \ldots, Q_i, \ldots, Q_I)$ avec $Q_i = [q_{i,1}, \ldots, q_{i,d}, \ldots, q_{i,D}]^T$.

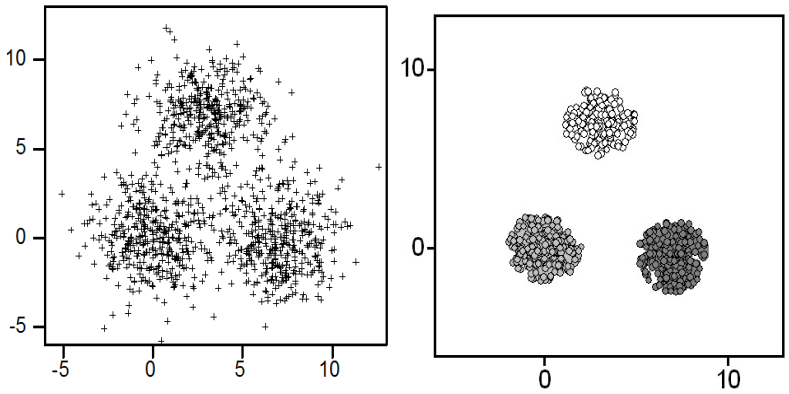
L'ensemble des observations, y compris les prototypes et les observations non prototypes qui restent à classer est noté Σ, tel que :

$$\Sigma = \Pi_1 \cup \Pi_2 \cup \cdots \cup \Pi_K \cup \Gamma = \{O_s, s = 1, \ldots, S\}$$

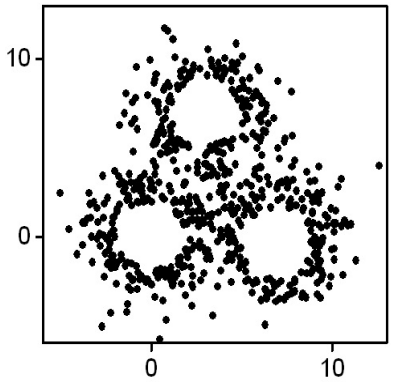
et il est composé de S éléments avec $S = N_1 + N_2 + \cdots + N_k + \cdots + N_K + I$. Une observation O_s peut donc appartenir à Γ ou à l'un des ensembles Π_k.

Figure 3.6(a) présente un ensemble de données générées artificiellement, composé de trois distributions bidimensionnelles gaussiennes. Figure 3.6(b) représente les trois ensembles de prototypes Π_1, $_2$ et Π_3, ce sont les observations de l'ensemble des données de la figure 3.6(a) qui se trouvent au sein des trois modes détectés par l'algorithme de Markov proposée dans ([56]). Le reste des observations de l'ensemble Γ sont présentées

dans la figure 3.6(c).



(a) Ensemble de données brutes Σ (b) Ensembles prototypes Π_1, Π_2 et Π_3



(c) Ensemble des observations non classées Γ

FIGURE 3.6 – Les trois classes de l'exemple d'illustration et les sous-ensembles associés $\Pi_k, k \in \{1, 2, 3\}$ et Γ

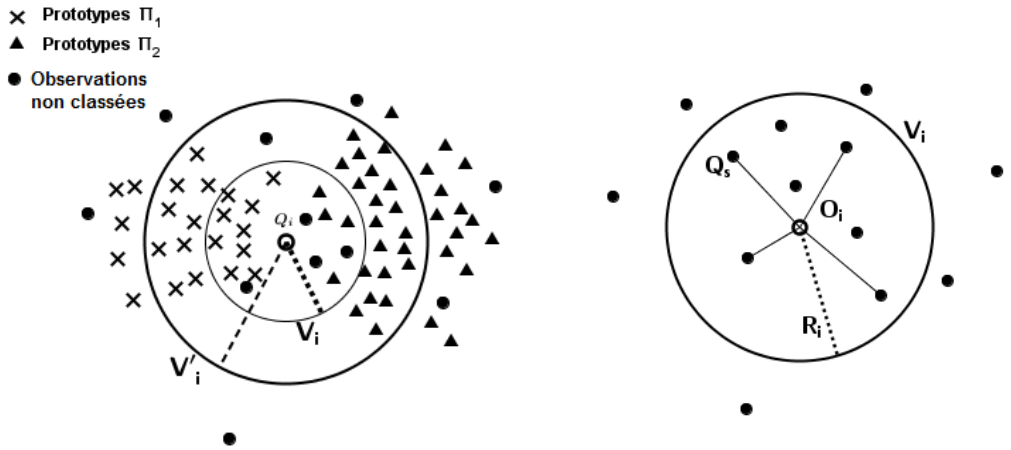
3.3.2 Modélisation markovienne

Pour classifier les observations non prototypes, considérons un champ de Markov $X = (X_1, \ldots, X_s, \ldots, X_S)$, où X_s est la variable aléatoire dont l'espace des réalisations est l'ensemble de labels associés à la s^{th} observation de Σ, notée O_s. Une réalisation de X_s indique la classe associée à O_s. Au début de la procédure de classification, $X = x^0 = (x_1^0, \ldots, x_s^0, \ldots, x_S^0)$ et $x_s^0 \in \Omega = \{0, 1, 2, \ldots, K\}$ où $x_s^0 \in \Omega \setminus \{0\}$ pour les prototypes et $x_s^0 = 0$ pour les observations non classées. Ce champ est itérativement mis à jour

à chaque itération de sorte que dans la t^{th} itération, $X = x^t = (x_1^t, \ldots, x_s^t, \ldots, x_S^t)$. Pendant ce processus itératif, les cardinaux des ensembles des observations prototypes Π_k, $k = 1, \ldots, K$ augmentent tandis que le cardinal de l'ensemble Γ diminue. L'objectif est d'avoir $\Gamma = \emptyset$ à la fin de la procédure.

Rappelons qu'un champ de Markov est défini par un ensemble de variables X, une mesure de probabilité $P(X)$, un ensemble de réalisations possibles $\Omega_X = \Omega^S$ et un système de voisinage V qui garantit les propriétés du champ de Markov X à travers les concepts de cliques et de fonctions de potentiels ([10]).

En nous inspirant de la procédure de régularisation déterministe proposée dans ([36]), notre objectif est d'affecter chaque observation non classée à l'une des K classes définies par leurs prototypes.



(a) L'impact du rayon R (b) Cliques d'ordre deux

FIGURE 3.7 – Système de voisinage et cliques associés.

Pour chaque observation non prototype Q_i, nous définissons dans l'espace de données $\mathbb{R}^D$ un voisinage hypersphérique de centre cette observation même et de rayon R. Pratiquement, la procédure consiste à considérer une hyper-sphère V_i centrée sur chaque observation Q_i de Γ (cf. Fig. 3.7). Les observations qui tombent dans V_i peuvent être soit des prototypes soit des non prototypes. Notre objectif est maintenant d'affecter chaque observation non prototype Q_i à l'une des K classes en prenant en compte toute l'information apportée par les observations entourées par le voisinage V_i.

Nous supposons que le champ X associé à ce système de voisinage et dont les réalisations x appartiennent à l'espace Ω_X, est un champ de Markov satisfaisant les deux conditions suivantes ([10]) :

1. $P(X = x) > 0$, autrement dit chaque réalisation de X est possible.

2. $P(X_i = k|X_s = x_s, s = 1, 2, \ldots, S, s \neq i, O_s \in \Sigma) =$
   $P(X_i = k|X_s = x_s, s = 1, 2, \ldots, S, s \neq i, O_s \in V_i)$.

Selon le théorème de Hammersley-Clifford ([10]) le champ X est également un champ de Gibbs et la probabilité conditionnelle de Gibbs pour chaque observation O_i, $i = 1, 2, \ldots, S$, sachant les étiquettes de ses voisins, peut être estimée comme dans ([56]) par :

$$\begin{aligned} P(X_i = k|X_s = x_s, s = 1, 2, \ldots, S, s \neq i, O_s \in V_i) \\ = \frac{\exp(-U(X_i = k|V_i)}{\sum_{j\in[1,K]} \exp(-U(X_i = j|V_i)} \end{aligned} \tag{3.37}$$

où $U(X_i = k|V_i)$ est l'énergie de Gibbs qui est définie par l'introduction de la notion de clique et fonction de potentiel ([56]).

Une clique c est un ensemble d'observations associé à un système de voisinage défini et qui est composé de paires d'observations. Le cardinal de c est appelé l'ordre de la clique. Le nombre de cliques est proportionnel à la taille du système de voisinage. Nous utilisons dans ce travail, des cliques d'ordre deux (composé de deux observations) pour modéliser les interactions entre les paires d'observations.

Considérons une variable X_i d'étiquette k, l'énergie de Gibbs correspondante s'exprime :

$$U(X_i = k|V_i) = \sum_{c\in C_i} \varphi_c(X_i = k, X_s = k')) \tag{3.38}$$

où C_i représente l'ensemble de toutes les cliques d'ordre deux c qui se trouvent dans le voisinage V_i et qui contiennent O_i. Dans d'autres termes, $c \in C_i$ quand $O_i \in c$ et donc, si $O_s \in c$, $d(O_i, O_s) \leq R$ (voir Fig. 3.7(b) où $Q_i = O_i$). $d(O_i, O_s)$ peut être la distance euclidienne entre les observations O_i and O_s ou une autre distance qui reflète mieux la forme géométrique des observations dans l'espace.

En règle générale, le choix des fonctions potentielles φ_c dépend de la nature des configurations que nous voulons favoriser dans le processus de mise à jour itérative du champ aléatoire. Dans un problème de classification, la fonction potentiel doit encourager l'affectation des observations voisines aux mêmes classes. Ainsi, nous proposons une fonction potentielle φ_c sur la base du modèle décrit dans ([36]), qui prend en compte les étiquettes des observations relevant de V_i et les distances entre ces observations.

Pour chaque clique c composée de l'observation centrale O_i et d'une de ses voisines

O_s, le potentiel peut être défini comme étant :

$$\varphi_c(x_i, x_s) = \frac{\gamma_{i,s}}{1 + d(O_i, O_s)^2} \tag{3.39}$$

où :

$$\gamma_{i,s} = \begin{cases} \beta & \text{if } x_i = x_s \\ -\beta, & \text{if } x_i \neq x_s \end{cases} \tag{3.40}$$

Le réglage du paramètre β sera abordé dans les paragraphes suivants.

En somme, l'énergie de Gibbs peut être écrite comme suit :

$$U(X_i = k|V_i) = \sum_{s, O_s \in V_i} \varphi_c(X_i = k, X_s = k') \tag{3.41}$$

La règle d'affectation consiste alors à attribuer chaque observation non prototype à la classe qui maximise la probabilité conditionnelle de Gibbs associée (voir l'équation 3.37).

3.3.3 Algorithme proposé

La modélisation markovienne de la problématique de la classification supervisée que nous venons de présenter ci-dessus n'est pas le seul éléments clé de la méthode proposée. En effet, afin de renforcer l'idée d'utilisation des informations contextuelles qui nous permettent d'éviter d'obtenir des classes sphériques et donc de bien délimiter les structures réelles des classes, nous avons adopté une démarche qui consiste à traiter les observations non prototypes suivant un ordre donné. Cette démarche réordonne ces observations à chaque itération de l'algorithme pour récupérer l'observation le plus proche à l'ensemble des prototypes.

Ainsi, à chaque itération de l'algorithme, les distances entre chaque observation non prototype Q_i et tous les prototypes sont calculées et l'observation la plus proche est le premier à être traitée et intégrée dans l'une des classes Δ_k. Cette observation prendra l'étiquette de la classe qui maximise la probabilité conditionnelle associée Gibbs 3.37. A la fin d'une itération donnée, certaines observations sont affectées à leurs classes et deviennent directement de nouveaux prototypes pour l'itération suivante. Les autres observations restent non affectées avec leurs étiquettes égal à zéro.

Dans le cas où aucune observation n'a été classée pendant une itération donnée, l'algorithme augmente le rayon R du système de voisinage afin d'inclure plus de prototypes à son voisinage lors de la prochaine itération.

L'algorithme itératif s'arrête une fois que le sous-ensemble Γ des observations non prototypes soit vide.

Algorithm 2 Algorithme de Classification

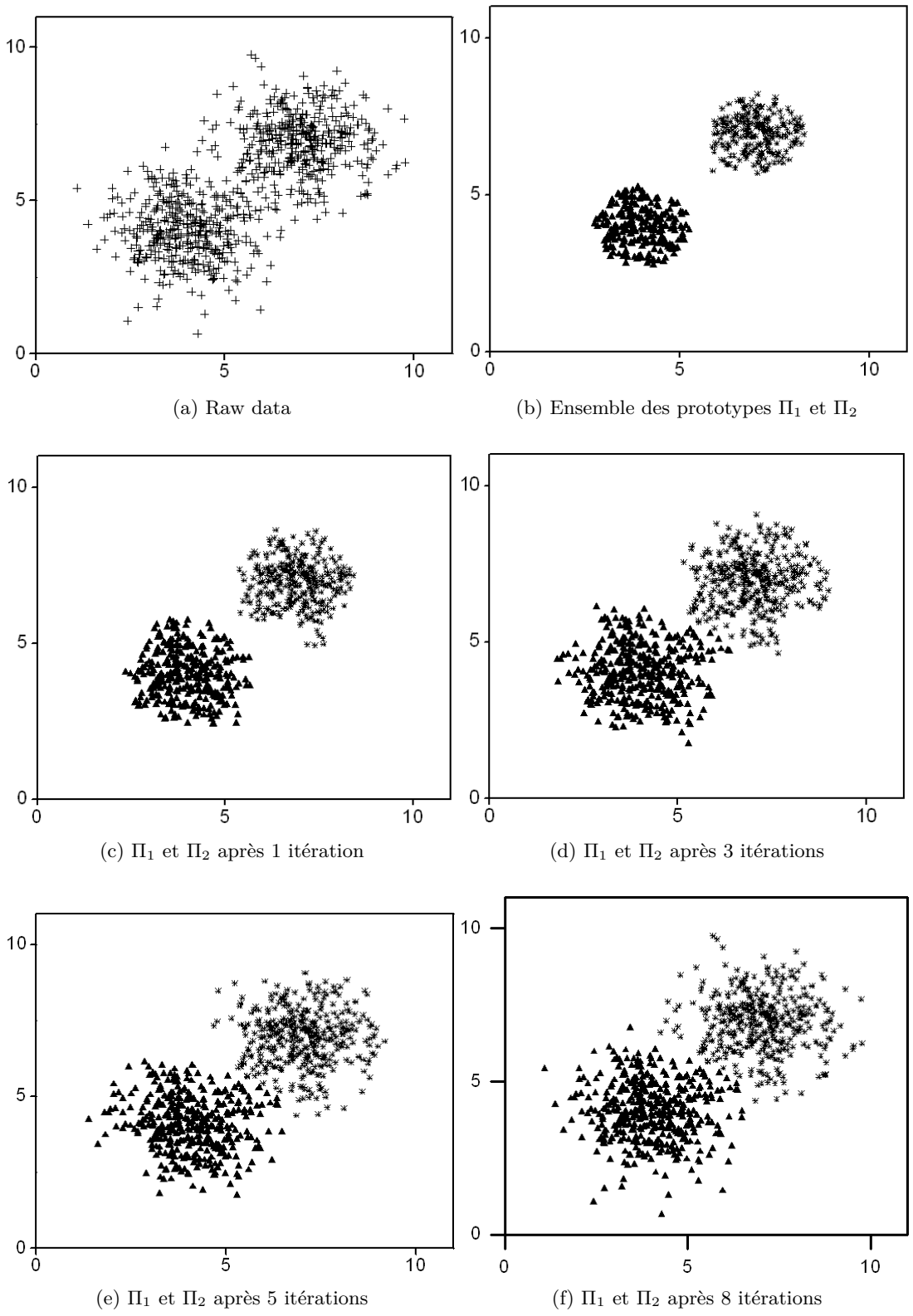
Require: Σ, R^0, X^0, ε
$\Gamma \leftarrow$ L'ensemble des observations non classée.
$K \leftarrow$ Le nombre de classes.
$N \leftarrow$ Le nombre de prototypes
$I \leftarrow Card(\Gamma)$
$R \leftarrow R^0$
while $I > 0$ **do**
  $count \leftarrow 0$
  $Dist(Q, P) \leftarrow$ il contient les distances dans l'ordre croissant entre les observations non classées et tous les prototypes.
  for $q = 1$ to I **do**
    for $k = 1$ to K **do**
      Calculer $P(X_q = k/V_q)$ {(Eq.3.37)}
    end for
    $\widehat{k} \leftarrow \arg\max_k P(X_q = k/V_q)$
    if $\widehat{k} \neq 0$ **then**
      $X_q = \widehat{k}$
      $\Gamma \leftarrow \Gamma \setminus q$
      $\Pi_{\widehat{k}} \leftarrow \Pi_{\widehat{k}} \cup \{q\}$
      $I \leftarrow I - -$
      $count \leftarrow count + +$
    end if
  end for
  $R \leftarrow R + \varepsilon$
end while
return Σ

La figure 3.8(a) montre un échantillon composée de deux classes gaussiennes générée artificiellement. A partir de l'ensemble des prototypes (Fig. 3.8(b)) l'algorithme proposé a classifié progressivement toutes les observations non prototypes Γ dans leurs classes respectives, après huit itérations (Fig. 3.8(c, d, e, f)).

Comme il est indiqué plus haut, la méthode de classification proposée classe d'abord les observations les plus proches des prototypes. Au fur et à mesure les observations les plus éloignées seront intégrées.

3.3.4 Ajustement des paramètres

La méthode présentée a des paramètres qui doivent être réglés ou initiés afin d'aboutir à des résultats satisfaisants. Ces paramètres sont le rayon R qui définit le système de voisinage et β le paramètre d'interaction de la fonction d'énergie.



(a) Raw data

(b) Ensemble des prototypes Π_1 et Π_2

(c) Π_1 et Π_2 après 1 itération

(d) Π_1 et Π_2 après 3 itérations

(e) Π_1 et Π_2 après 5 itérations

(f) Π_1 et Π_2 après 8 itérations

FIGURE 3.8 – La classification itérative des deux distributions artificiellement générées.

3.3.4.1 Ajustement du rayon du système de voisinage

Le rayon R de l'hypersphère qui détermine le voisinage de chaque observation non prototype Q_i joue un rôle très important dans le succès de la procédure de classification supervisée que nous proposons.

L'idée de la procédure est d'examiner en particulier les observations prototypes entourées par le voisinage de chaque observation non prototype. Une valeur faible de R, le voisinage pourrait ne pas contenir suffisamment de prototypes, ce qui ne conduira qu'à une valeur sous-estimée de l'énergie (cf. équation 3.38.). Inversement, une grande valeur de R, conduirait à un excès de prototypes dans le voisinage de l'observation non prototypes, ce qui engendre une sur-estimation de l'énergie. Dans les deux cas, la classification finale peut être loin de la vérité terrain.

Sur la figure 3.7, nous illustrons l'impact du choix de R sur la procédure de la classification. Avec le rayon V_i l'observation concernée Q_i va être affectée à la classe représentée par des étoiles alors que avec le rayon $V_i^{'}$ va être affectée à la deuxième classe représentée par des triangles.

Pour contrôler cet effet du rayon, sa valeur serait choisie de telle sorte que deux modes Π_i et Π_j $(i \neq j)$ de deux différentes classes ne fassent pas partie du même système de voisinage. Il peut alors être calculé en fonction de la variance des observations prototypes au sein de chaque mode et des distances entre les observations prototypes de tous les mode, comme suit :

$$R = 0.1 \times \min(W_{\min}, V_{\min}) \tag{3.42}$$

avec $V_{\min} = \min(\sigma_1^2, \ldots, \sigma_k^2, \ldots, \sigma_K^2)$, et σ_k^2 est la variance entre la classe Δ_k, $k = 1, \ldots, K$. $W_{\min}$ est la distance minimale entre tous les prototypes de différentes classes. Ainsi, pour une itération donnée, un rayon initial est calculé en utilisant la formule précédente. Si quelques observations restent encore non classées à la fin de l'itération, le rayon est augmenté pour la prochaine itération en rajoutant un pas ε à sa valeur précédente ($R^{t+1} = R^t + \varepsilon$). Dans cette étude, ce pas s'est fixé sur $0.05 \times R$ et il a donné des résultats satisfaisants dans les expériences menées.

3.3.4.2 Ajustement du paramètre d'interaction

Le paramètre d'interaction β a été introduit dans l'objectif de favoriser ou pénaliser l'affectation d'observations à la même classe ([10]). Une valeur de β inférieure à 1, fournit des valeurs de faible énergie de Gibbs, c'est-à-dire, des probabilités de valeurs négligeables. Si β est trop élevé, la contribution de la distance entre les observations

(cf. Eq. 3.39) sera trop faible pour obtenir des résultats satisfaisants. Au cours de nos expériences, nous avons constaté qu'une valeur de β dans l'intervalle $[1, 2]$ encourage la formation de classes compactes tout en préservant leur structures spatiales. Pour cela, nous avons choisi de mettre $\beta = 1,5$, ce qui a permis d'obtenir de bons résultats de classification dans nos différentes expériences. Nous en présentons quelques unes dans les paragraphes suivants.

3.3.5 Résultats expérimentaux

Nous présenterons dans cette dernière section de ce chapitre, une évaluation des performances de la méthode de classification proposée. Comme nous avons fait dans la partie expérimentale concernant la méthode de détection des modes, nous utiliserons des échantillons de données artificielles.

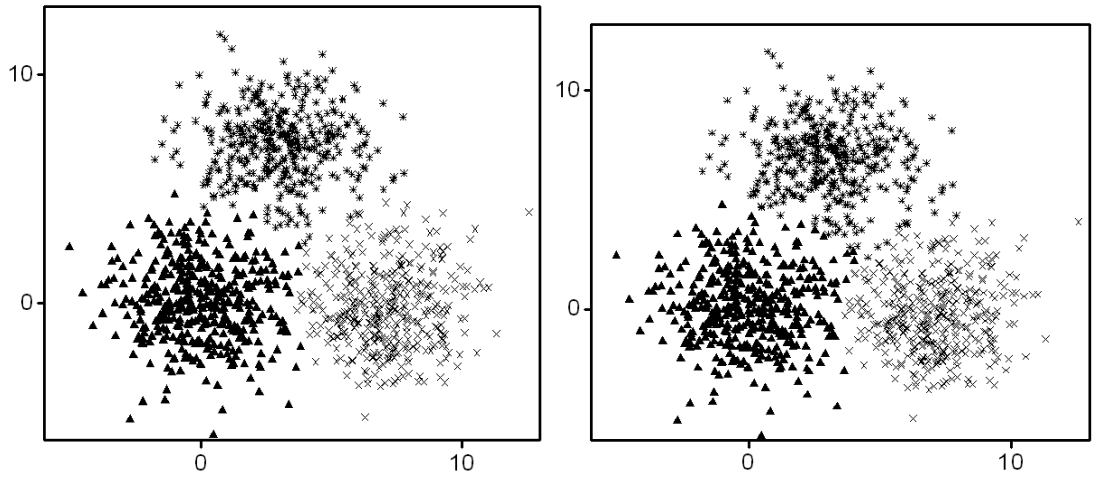
Dans le cas d'échantillons gaussiens, nous comparerons également les résultats à ceux obtenus par la méthode de classification bayésienne optimale ([42]).

3.3.5.1 Distributions gaussiennes

Nous commençons par l'exemple de la figure 3.6 qui est utilisé pour illustrer l'ensemble Σ, les sous-ensembles de prototypes Π_k, $k = 1, 2, 3$, et le sous ensemble des observations non prototypes Γ. Les prototypes disponibles représentent trois classes distribuées selon la loi normale (cf. Fig. 3.6(b)). Nous avons classé les observations appartenant à Γ par la méthode proposée, comme indiqué dans la figure 3.9(a). Ces résultats sont comparés à ceux obtenus avec la méthode des k-plus proche voisins (KNN)(cf. figure 3.9(b)) (cf. section 1.4.1). Les taux d'erreurs relatifs aux deux méthodes de classification (cf. table 3.3) peuvent être comparés au taux d'erreur minimum théorique de Bayes. La différence entre le résultat obtenu par notre méthode et le taux d'erreur théorique minimal de Bayes qui table sur 2,01%, correspond à seulement trois observations mal classées, ce qui est équivalent à environ la moitié du nombre d'observations mal classées obtenu par l'algorithme des k-plus proches voisins.

3.3.5.2 Distributions non linéairement séparables

Nous présentons maintenant un deuxième exemple constitué de deux classes bidimensionnelles avec une structure non linéairement séparables. L'ensemble se compose de deux populations de 400 observations chacune. Deux modes étendus définissent Π_1^1 et Π_2^1 de 320 éléments chacun sont disponibles, qui correspondent assez bien aux structures des deux classes comme le montre la (figure 3.10(a)). Pour évaluer le comportement de



(a) Méthode proposée (b) L'algorithme des k-ppv

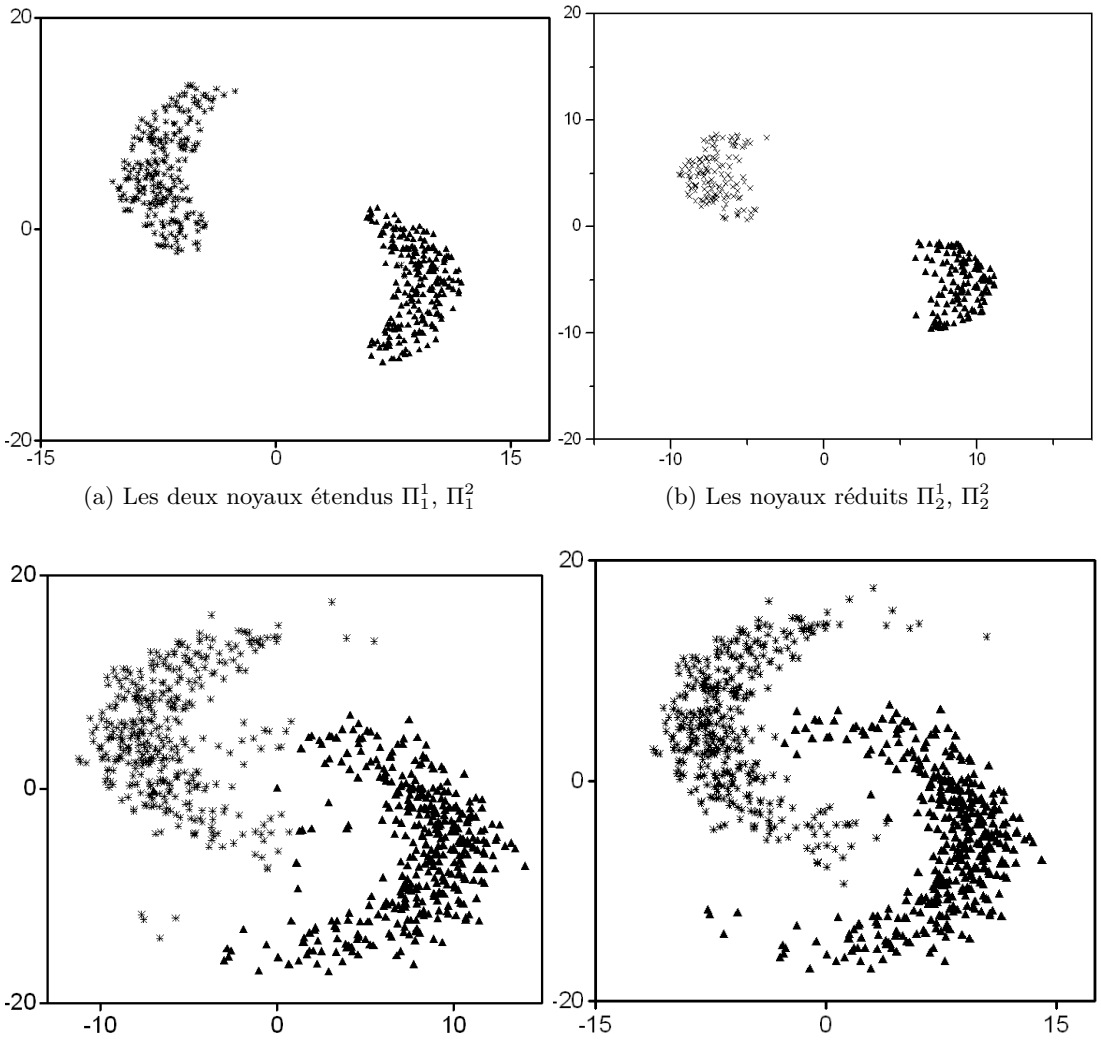
FIGURE 3.9 – Résultat de la classification des observations de l'exemple 1

TABLE 3.3 – Résultats statistiques pour le premier exemple (see Fig. 3.6 and Fig. 3.9)

Méthodes de classification	Taux d'erreurs	Matrices de confusion
Algorithme k-ppv	2.58%	$\begin{pmatrix} 392 & 7 & 7 \\ 2 & 389 & 5 \\ 6 & 4 & 388 \end{pmatrix}$
Méthode Proposée	2.25%	$\begin{pmatrix} 388 & 4 & 6 \\ 2 & 394 & 3 \\ 10 & 2 & 391 \end{pmatrix}$

la méthode proposée dans une situation encore plus difficile, deux autres modes réduits Π_1^2 et Π_2^2 de seulement cette fois-ci 220 prototypes pour chaque classe, sont également considérés (figure 3.10 (b)). Notez que l'objectif est de ne pas révéler la structure finale des classes, c'est à dire que la structure réduites et quasi-globulaire des modes (les prototypes) ne donnent pas une information précise sur les structures spatiales des classes initiales.

Les résultats démontrent la performance importante de notre nouvelle technique de classification supervisée. En effet, pour l'algorithme des k-plus proches voisins avec $k = 1$ et les premières ensembles de prototypes de 320 points, le nombre d'observations mal classées, situées en particulier aux extrémités des deux classes non sphériques, est relativement grand (Fig. 3.10(c)), résultant en un taux d'erreur égale à 3,25%. Ceci est du au fait que la méthode des k-plus proches voisins ne prend pas en compte les formes



(a) Les deux noyaux étendus Π_1^1, Π_1^2

(b) Les noyaux réduits Π_2^1, Π_2^2

(c) Avec l'algorithme des k-ppv

(d) Avec notre méthode et les noyaux réduits

FIGURE 3.10 – Résultats de la classification de l'exemple 2

réelles des classes qui pourraient être apprises de la distribution des prototypes (fig. 3.10(a)).

Au contraire, les résultats obtenus avec la méthode proposée (figure 3.10(d)) sont meilleurs. Le nombre d'observations mal classées s'est considérablement réduit, de sorte que le taux d'erreur de 1,37% ait été baissé à la moitié du taux atteint avec l'algorithme des k-plus proches voisins. L'amélioration est particulièrement perceptible dans les extrémités des deux classes où notre méthode est localement guidée par la distribution des ensembles des prototypes.

Les deux méthodes de classification ont été ensuite testées sur l'ensemble réduit de prototypes qui, affiché sur la figure 3.10(b), ne fournissent pas une information complète sur les formes des deux classes. Même dans ces conditions difficiles, avec l'algorithme proposé, on a obtenu un taux d'erreur égal à 2% qui est inférieure à la moitié de 4,27% qui est la valeur obtenue avec l'algorithme des k-plus proches voisins.

3.3.5.3 Distributions tridimensionnelles

Le dernier exemple d'évaluation de la méthode proposée est un ensemble de données tridimensionnelles. Il contient une première population de données qui a la forme d'anneau, entourée par une autre populations sphérique distribuée selon une loi de Gauss (figure 3.11(a)). En utilisant les prototypes affichés sur la figure 3.11(b) et qui ont été obtenus grâce à l'algorithme de détection des modes proposé dans [115], les résultats de la classification ont abouti à une classification avec un taux d'erreur de 2,36% (cf. figure 3.11.(c)). Ce taux a atteint la valeur 2,69% pour la méthode des k-plus proches voisins.

Les résultats présentés confirment l'efficacité de la méthode de classification proposée face à des données multidimensionnelles non-linéairement séparables et de structures spatiales complexes.

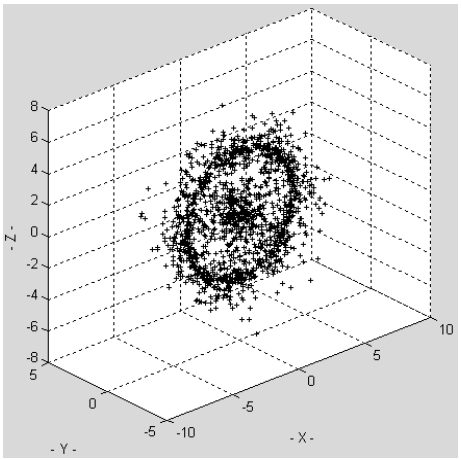
3.4 Conclusion

Dans la première partie de ce chapitre, nous avons présenté notre méthode pour la détection des modes en utilisant les processus ponctuels marqués. Les simulations par les techniques de Monte Carlo par Chaines de Markov permettent d'obtenir des réalisations à partir de la loi du processus stochastique construit. L'objectif était de pouvoir contrôler les objets (hypersphères) du processus pour les pousser à se localiser sur les régions modales de l'espace de représentation des données éventuellement multidimensionnelles.

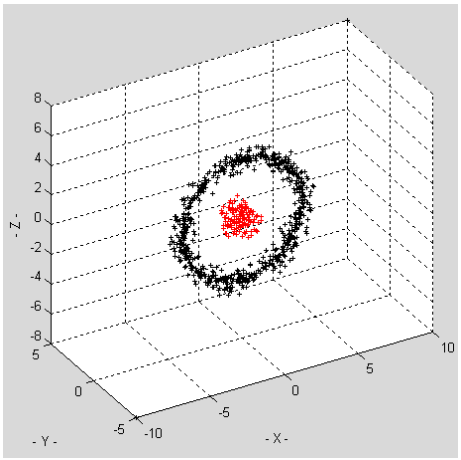
Les résultats obtenus en appliquant cette méthode sur des données synthétiques dans des espaces de deux et de trois dimensions, ont permet de confirmer les conclusions suivantes :

1. Il est possible de deviner, grâce aux processus ponctuels marqués, le nombre de classes cachées dans un ensemble de données brutes ;

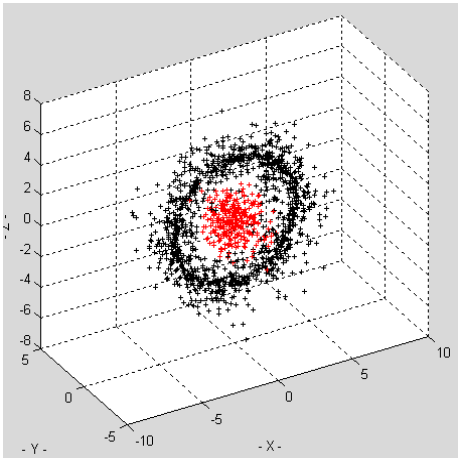
2. Il est possible de détecter, grâce aux processus ponctuels marqués, les modes de la fonction de densité de probabilité sous-jacente aux données ;



(a) L'ensemble des données brutes



(b) Ensemble Π_1 et Π_2



(c) Résultats de la classification

FIGURE 3.11 – Résultats de la classification pour des données tridimensionnelles non gaussiennes

3. L'algorithme de détection des modes par les processus objet (Mode Detection by Object Process, MDOP) a pu classer des données de structures géométriques et spatiales différentes ;

4. L'algorithme de détection des modes par les processus objet a pu classifier des données présentant un degré considérable de chevauchement.

5. L'aspect aléatoire de la méthode proposée fait que les réalisations obtenues du processus donnent parfois de fausses détections, comme la séparation d'une classe réelle en deux classes ou le regroupement de deux classes réelles ou plus dans une

seule.

Dans la deuxième partie, une nouvelle méthode de classification par affectation a été présentée. Cette méthode utilise des outils markoviens pour construire un algorithme approprié pour la classification. Cette dernière est basée sur l'utilisation d'informations contextuelles et les distances entre les observations pour les assigner à leurs pôles les plus probables. Les résultats obtenus confirment la capacité de cet algorithme de classification proposé de classer les données non gaussiennes avec des formes complexes telles que celles extraites de couleur images texturées.

Chapitre 4

Applications en Imagerie

4.1 Introduction

Nous consacrerons le dernier chapitre de ce mémoire à deux applications en imagerie. La première application est dédiée à la segmentation d'images couleur texturées, tandis que la deuxième portera sur la problématique de la reconnaissance d'objets dans les images à hautes résolutions. Ce choix est justifié d'abord par l'intérêt toujours en augmentation que l'on accorde aux problèmes de l'imagerie dans plusieurs domaines scientifiques et technologiques tels que l'imagerie médicale et la vision artificielle ; mais aussi parce que la structure, en *pixels*, des images couleur permet d'avoir une base de données à analyser de grandes dimensions, ce qui représente la principale caractéristique des données modernes.

Nous rappellerons tout d'abord quelques aspects indispensables pour comprendre la formation d'images numériques 4.2.1. Ces aspects concernent notamment la constitution de la couleur, l'acquisition et l'affichage d'une image ainsi que certains problèmes souvent abordés dans le domaine du traitement d'images. Nous présenterons ensuite, dans la section 4.3, la première application sur la segmentation d'images de textures couleur en y montrant l'apport de la méthode proposée dans le chapitre précédent pour faire de la classification à partir des prototypes. Lors de la section 4.4, nous exposerons la deuxième application sur la reconnaissance d'objets dans les images aériennes, ainsi que la nouvelle méthode qui y est associée. Un intérêt particulier sera accordé à la détection de toitures à partir des images orthophotoplans 4.4.2.

4.2 Notions en analyse d'images couleur texturées

4.2.1 Processus de formation d'image

Avant de considérer une *image numérique* et se demander comment décrire son contenu et comment pouvons nous en extraire des connaissances, il faudrait tout d'abord comprendre ce que nous entendons par ce terme et comment l'obtenir à partir d'une scène naturelle.

4.2.1.1 La couleur et ses mesures

La couleur : La notion de la couleur se décompose en trois composantes : la lumière, les matériaux et le système visuel humain. En effet, afin de pouvoir visualiser les objets dans la nature, il est indispensable qu'ils soient éclairés par une lumière. Dans l'obscurité, nous sommes incapables de percevoir les objets. Cette lumière peut être considérée comme un rayonnement électromagnétique, c'est-à-dire des *ondes électromagnétiques* qui se propagent dans l'air via des particules lumineuses appelées *photons*. La lumière n'est pour autant que la partie visible par l'homme de ces rayonnements. Même cette lumière (visible), on ne le perçoit pas directement et complètement, mais c'est juste la partie émise ou réfléchie par les matériaux des objets. L'interaction lumière-matériau pourrait s'approcher par trois relations : La réflexion, l'absorption et la transmission. Les particules du matériau qui sont responsable de ces interactions s'appellent des *pigments*.

En dépit de ces aspects physiques de la notion de la couleur tels que la lumière et la matière des objets, il vient s'ajouter deux autres aspects liés cette fois ci à l'homme. Le premier est un aspect physiologique, c'est à dire lié à la formation organique et biologique du récepteur du stimulus de la couleur qui n'est autre que l'œil. Le deuxième est un aspect psychologique. Il réside dans le fait que notre perception de la couleur dépend de notre connaissance a priori de l'environnement et de l'apprentissage que nous avons reçu des couleur. Notre cerveau interprète et ré-interprète continuellement les informations visuelles qu'il reçoit. Cette subjectivité de la perception de la couleur est souvent exprimée avec des termes qui indiquent des attributs par lesquels nous décrivons notre visualisation des objets couleur. Ces attributs sont : la *luminosité*, la *teinte* et la *saturation*.

Cette subjectivité de la perception de la couleur nous amène à chercher une technique qui puisse la rendre objective. Tel est l'objectif de la *colorimétrie*, la science de la mesure de la couleur.

La colorimétrie : La mesure de la couleur s'appuie sur la théorie trichromatique (principe de la trivariance visuelle) et sur le concept d'un observateur de référence. La théorie trichromatique est une théorie selon laquelle un stimulus de couleur est perçu comme étant la synthèse de trois couleurs initiales appelées *primaires* ou *stimulus de référence*. En théorie, cette synthèse, effectué par les expériences d'égalisation ou d'appariement, permet de reproduire n'importe quelle couleur dans le spectre visible. Toutefois, le nombre de couleurs reproductible en pratique, est beaucoup plus restreint pour des raisons technologiques liées aux dispositifs employés. Deux types de synthèses existent, la synthèse additive et la synthèse soustractive. Dans la littérature, les synthèses les plus utilisées sont la RGB CIE 1931 et la XYZ CIE 1964. Ce dernier constitue depuis son invention la base de la colorimétrie et il existe des formules pour passer d'un système de primaires à l'autre.

Afin de représenter les couleurs dans des espaces géométriques, des chercheurs ont associé à chaque système de primaire des vecteurs unitaires correspondants à ces trois primaires. Ensuite, pour pouvoir les ramener à un espace à deux dimensions, un espace à deux dimensions a été défini à partir de ces nouvelles coordonnées. Cet espace est nommé *diagramme de chromoticité* et noté xyY.

Dans ce diagramme, les spécialistes en colorimétrie se sont confrontés à un problème connu sous le nom de la *non-uniformité* de la perception aux écarts de chromaticité. Cela signifie que si deux couleurs prises dans ce diagramme ont des valeurs de chromaticité très proches, ils ne le sont pas forcément par rapport à notre perception. Tantôt elles vont être distinguées, tantôt elles vont être confondues selon leurs positions dans le diagramme. Pour remédier à cet inconvénient et donc construire un espace de couleurs *perceptuellement uniforme*, plusieurs travaux on été effectués.

Ces travaux ont abouti aux développement de deux espaces colorimétriques qui se base sur l'association de la chrominance et de la luminance. Ces deux espaces sont l'espace L*a*b (ou CIE Lab, CIELAB) et l'espace L*u*v (ou CIE Luv, CIELUV). Ils ont été normalisés par la CIE en 1976.

Les deux espace sont très similaires. L* représente l'axe des luminances tandis que a*b ou u*v représentent la chrominance. Les espace a*b et u*v, à luminance constante, sont formés par une opposition des couleurs rouge et vert pour la première composante, bleu et jaune pour la deuxième.

Pour ces deux espaces, deux grandeurs sont importantes :

- le *chroma (C*)* indiquant la distance de la couleur marquée dans l'espace de l'axe de la luminance L*.

- L'angle de teinte (h) qui décrit l'angle entre le vecteur déterminé par la couleur marquée dans l'espace et l'origine du repère d'une part, et le vecteur directeur de l'axe des abscisses de l'autre part.

Toujours dans le but de développer un système colorimétrique très proche du système d'interprétation visuelle humaine, des chercheurs ont proposé les *systèmes perceptuels*. Ces systèmes représentent la couleur selon des entités subjectives liées à la luminosité, la teinte et la saturation. Il en existe de nombreux systèmes dans la littérature, présentés sous différentes dénominations. D'une manière générale, ils sont divisés en deux grandes catégories : ceux qui s'évaluent à partir des systèmes de Luminance/Chrominances (Les systèmes de coordonnées polaires) et ceux évalués à partir des systèmes de primaires (Modèle triangulaire, modèle de cône hexagonal..). Les plus utilisés sont TSL (HSL en anglais, ou son dérivé IHSL) , TSI (HSI), TSV (HSV)(voir [158]).

4.2.1.2 Images numériques

Obtenir une image numérique, autrement numériser une scène naturelle, est une procédure qui passe par au moins trois opérations : l'acquisition, la numérisation et l'affichage. Qu'entendons-nous alors par ces termes.

Acquisition à l'aide d'une caméra L'opération d'immortaliser une scène naturelle s'appelle *acquisition de l'image* (*imaging*). Pour ce faire, nous devrions disposer d'un système visuel artificiel similaire au système visuel humain. La figure 4.1 montre les différentes composantes de ce système. Les deux composantes les plus importantes sont le système d'acquisition et celui d'affichage.

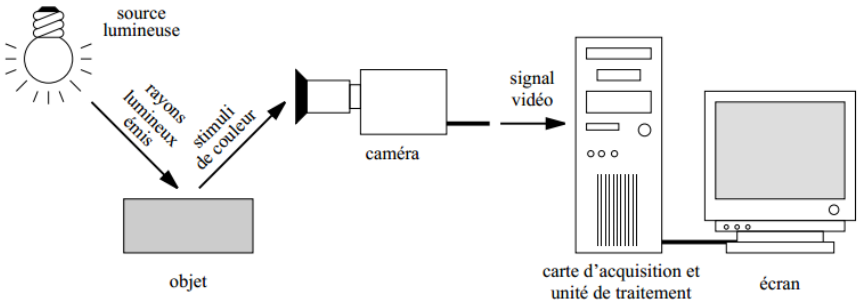


FIGURE 4.1 – Différentes composantes formant un système de vision artificielle.

Le premier se compose d'une caméra vidéo et d'une carte d'acquisition qui peut

être soit intégrée dans la caméra, soit séparée d'elle.

Une caméra est un appareil électronique formé d'un ensemble de *photorécepteurs* dont la fonction est de recevoir les flux lumineux puis les convertir en informations (charges) éléctriques. Ces informations seront transmises au système de numérisation sous forme d'un ou de signaux vidéo. Ces photorécepteurs sont disposés linéairement ou sous forme d'une matrice. D'où la notion d'*image*, qui veut dire un ensemble de cases appelées *pixels* (*picture-element*) correspondant chacune à un photorécepteur.

Numérisation des images couleur Une carte d'acquisition est nécessaire pour pouvoir numériser une image obtenue par une caméra sous forme d'un signal vidéo. Cette numérisation s'effectue grâce à la quantification de l'information couleur. La quantification est la conversion d'un ou plusieurs signaux vidéo en un triplet de valeurs numériques manipulables par l'ordinateur. La numérisation des images couleur consiste donc à associer à chaque pixel de l'image trois valeurs numériques correspondant aux composantes trichromatiques R, G et B de la couleur qu'il représente. D'une manière générale, chaque valeur d'une composante R, G ou B est codée sur 8 bits et peut prendre alors 256. Au total, pour un pixel donné, une couleur est codée sur 24 bits et un ensemble de $2^{24} = 16777216$ couleurs peut être représentées.

Une fois l'information couleur d'un pixel est quantifiée, la valeur obtenue sera stockée dans la mémoire de la carte d'acquisition et peut donc être restaurée pour une éventuelle visualisation.

Affichage des images couleurs Pour visualiser les images couleur, nous devrions généralement disposer d'un écran. Un écran est donc un dispositif électronique capable de restituer par affichage les informations couleur provenant de la mémoire d'une carte d'acquisition (ou d'une carte graphique).

La plus ancienne technologie, datant de 1926, était un écran analogique réalisé par un tub cathodique, appelé *moniteur CRT* (Cathodic Ray Tube). Il s'agit d'un tube à vide constitué d'un filament chauffé, d'électrodes en forme de lentilles trouées qui, soumises à une différence de potentiel (tension), créent un champ électrique accélérant les électrons. La surface interne de l'écran est constituée de luminophores (points lumineux) s'éclairant à l'arrivée des électrons. Des petits cristaux rectangulaires de rouge, de vert et de bleu correspondant aux trois primaires de la couleur, sont juxtaposés horizontalement pour chaque luminophores. Il y a trois canons à électrons, un par couleur, et chaque canon ne peut allumer que les points d'une couleur, un masque (plaque métallique percée de trous : un par point) est disposé dans le tube pour éviter qu'un

canon ne déborde sur l'autre.

Afin d'obtenir toute l'image, la technologie est basée sur la notion de balayage séquentiel développé à l'origine pour transmettre une image à distance. On est alors contraint de transmettre séquentiellement les états de chacun des pixels. Si l'image est mobile (TV), le cycle doit se répéter à un rythme suffisant : on récupère ainsi le signal vidéo. Le balayage est réalisé de la gauche vers la droite et de haut en bas.

Ces écrans ont les avantages d'avoir un angle de vision large et de reproduire fidèlement les couleurs. Cependant, ils sont lourds, volumineux et grands consommateurs d'énergie. Ces inconvénients laissent apparaitre de nouvelles technologies d'affichage vidéo telles que les afficheurs plats : *LCD*, *LED* et *Plasma*.

Ces écrans n'utilisent plus un tube à canon, mais des matrices dont les cellules contiennent des cristaux liquides pour les LCD, du plasma pour l'écran Plasma et des diodes couleurs pour les LED. Ceci a permis non seulement de réduire énormément la taille et l'épaisseur des écrans (design), mais aussi la consommation liée à la haute tension utilisée classiquement avec les technologies à tube cathodique.

4.2.2 Description d'une image

Pour pouvoir traiter une image numérique donnée et en extraire certaines connaissances pour des objectifs de prise de décisions, les analystes doivent définir des descripteurs de l'image et les quantifier. Ces descripteurs se subdivisent en deux catégories, à savoir : descripteurs globaux et descripteurs locaux.

4.2.2.1 Descripteurs globaux

La couleur Une image numérique peut être décrite par les attributs couleur. Ces attributs sont définis sur l'un des espaces de représentation de la couleur que nous venons de décrire dans la section précédente. Plusieurs attributs couleur ont été développés dans la littérature [165]. Parmi les plus utilisés, nous citons : les histogrammes de couleur, les moments de couleur et l'autocorrélogramme de couleur.

L'histogramme couleur représente la distribution globale des couleurs dans l'image. Il consiste à quantifier l'espace de représentation de la couleur en des *bins* et ensuite calculer le nombre de pixels appartenant à chaque *bins* de couleur. Les avantages de cet attribut résident dans sa robustesse aux changement dus à la translation et à la rotation. Cependant, il ne permet pas de prendre en considération l'information spatiale des pixels sur l'image. Ceci peut se traduire par le fait que deux scènes visuellement différentes pourront avoir le même histogramme couleur. Si l'on considère à titre d'exemple,

une image codée dans un espace de primaires RVB classique et que l'on quantifie chaque plan couleur sur 4 *bins*, l'histogramme résultant aura une dimension $4^3 = 64$. Si l'image est initialement codée sur 24 bits (la valeur de chaque plan est dans l'intervalle [0-255]), l'histogramme couleur sur 64 *bins* pourrait être représenté par un cube de cote $4 \times 4 \times 4$.

		Rouge															
		0-63				64-127				128-191				192-255			
		Vert				Vert				Vert				Vert			
		0-63	64-127	128-191	192-255	0-63	64-127	128-191	192-255	0-63	64-127	128-191	192-255	0-63	64-127	128-191	192-255
Bleu	0-63	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	64-127	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	128-191	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	192-255	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

FIGURE 4.2 – Exemple d'histogramme de couleur quantifiée sur 4 bins pour chaque composante.

Les moments de couleur sont des descripteurs simples qui consistent à calculer certaines statistiques sur la distribution de la couleur sur une image. Les statistiques les plus fréquentes dans les systèmes de recherche d'images sont la moyenne, la variance et l'asymétrie.

Quant à l'autocorrélogramme de couleur, il désigne une façon de prendre en considération l'information spatiale lors de la description d'une image par des descripteurs couleur. Plus précisément, c'est la version couleur des matrices de co-occurrence, dont nous allons parler, qui sont calculées sur le plan des niveaux de gris pour caractériser des textures. Ils caractérisent donc la distribution de paires de couleurs sur une image[71]. L'autocorrélogramme de couleur d'une image peut être vu comme un histogramme à trois dimensions. Les deux premières dimensions représentent la paire de couleurs et la troisième est leur distance spatiale.

La texture : Le deuxième attribut utilisé pour décrire le contenu visuel d'une image est donné par les descripteurs de texture. Bien qu'ils se basent sur des calculs bas-niveau, c'est-à-dire qui s'effectuent au niveau des pixels de l'image, ces descripteurs rentrent aussi dans la catégorie des descripteurs globaux.

La définition de ce que nous entendons par texture, n'est pas unique et c'est d'ailleurs l'une des raisons derrière la multiplicité des méthodes développées dans le but de décrire une image en caractérisant ses textures. De chaque définition nait donc une approche les décrivant. Cependant, une texture se caractérise d'une manière générale par la répétition d'un motif ou de certains éléments que certains auteurs appellent

textons. Les chercheurs de la communauté de l'indexation et de la recherche d'images par contenu classent les méthodes de l'extraction des attributs texture en deux grandes approches. L'approche spatiale et l'approche spectrale.

Dans l'approche spatiale, nous pouvons rencontrer trois sous approches : structurelle, statistique et approche par modèle. La sous-approche structurelle consiste à concevoir une texture comme étant formée de *minitextures* (textons) qui constituent une sorte de primitives et de *macrotextures* qui sont des arrangements spatiaux hiérarchiques de ces primitives. Des opérateurs morphologiques et les graphes d'adjacence sont employés pour manipuler les textures structurelles.

La sous-approche statistique pour caractériser des textures est basée sur des statistiques effectuées à bas niveau sur l'image niveaux de gris. Les plus connues de ces statistiques spatiales sont les moments, les attributs de Tamura et les matrices de co-occurrence.

Quant à la sous-approche basée sur les modèles, les propriétés de la texture analysée sont modélisées par les paramètres associés aux modèles stochastiques ou génériques utilisés. Un exemple de ces modèles est donné par les champs aléatoires de Markov (Markov Random Field MRF) et Simultaneous Auto-Regressive (SAR).

L'approche spectrale pour la caractérisation des attributs texture s'est intéressée par l'étude des périodicités et des régularités spatiales en utilisant l'analyse spectrale. Pour ce faire, il faudrait tout d'abord transformer l'image en question vers l'espace fréquentiel en utilisant les transformées de Fourier, les transformés en cosinus discret, les filtres de Gabor, les ondelettes ou plus récemment les *courbelettes* (curvelet). Des chercheurs ont montré que les courbelettes sont beaucoup plus appropriées que les ondelettes ou même les filtres de Gabor, vu qu'elles sont capables de mieux discerner les propriétés curvilignes telles que les lignes et les contours [143].

4.2.2.2 Descripteurs locaux

La forme La forme ou *shape* en anglais et ces caractéristiques sont des attributs visuels importants pour la représentation ou la description du contenu d'images. Ces attributs se basent sur une description soit des contours des formes soit des contours et de leurs contenus à la fois. De ce constat, les méthodes développées visant à extraire les attributs de la forme se subdivise en deux classes essentielles, à savoir les méthodes basées sur les contours et celles basées sur les régions [164].

Parmi les attributs de forme utilisés dans la littérature, nous citons les moments, la surface ou taille (area), *circularity* qui indique le rapport entre la taille et le contour

et *eccentricity* qui mesure le rapport de la longueur de l'axe majeur à celui de l'axe mineur.

Bien que les attributs associés à la forme donnent des résultats encourageant, ils envisagent encore certains difficultés liées en particulier à la nature des formes elles-même dans les images. Ces difficultés proviennent par exemple de la perte d'une dimension des objets de la scène tridimensionnelle une fois projetée sur une image bidimensionnelle ou d'une distorsion de la forme.

4.2.3 Traitement d'image

4.2.3.1 Objectifs des traitements

Dans le domaine de la vision par ordinateur (*computer vision*), les images brutes ainsi produites et décrites ne sont pas en général importantes en soi, mais elles sont plutôt utiles de par les informations que nous pourrions en extraire, et cela pour des applications ultérieures bien précises. De ce fait, une phase de traitement de ces images devient nécessaire. Le traitement d'image (*image processing*) est un terme qui désigne une opération ou un ensemble d'opérations qui sont appliquées à une image ou à une partie d'une image et produisent une nouvelle image dans le but est d'améliorer sa qualité ou de mettre en reliefs certaines de ses composantes.

Étant donné que la qualité que nous souhaitons apporter à une image n'est pas la même ni pour l'analyste ni pour la machine, les techniques de traitement d'images développées dans la littérature sont très variées. Chaque objectif attendu exige une re-formulation de la notion de qualité bien précise. Cependant, nous pouvons regrouper ces différentes techniques d'une manière générale selon le type d'opérateur utilisé [151].

4.2.3.2 Typologie d'opérateurs utilisés en traitement d'image

Opérateurs ponctuels Ils s'appellent aussi opérateurs de modifications pixel à pixel ou opérateurs point à point [1]. Ces opérateurs, qui sont les plus simples à appliquer parmi les différents types des traitements d'images, sont appliqués sur les valeurs individuelles des pixels de l'image traitée. La nouvelle valeur du pixel considéré ne dépend alors que de son ancienne valeur. Parmi les traitements basés sur ce type d'opérateurs, nous citons l'égalisation d'histogramme, changement de la dynamique de l'image et la modification des couleurs.

1. *Point operators.*

Opérateurs locaux Ce type d'opérateurs[1] prennent en considération les valeurs d'un ensemble de pixels qui se situent à un voisinage donné du pixel en question. Les pixels vont alors recevoir de nouvelles valeurs en fonction de leurs pixels avoisinants. Ils peuvent être appliqués comme des filtres pour ajuster le caractère flou de l'image, en accentuer certaines parties ou en supprimer le bruit qui peut l'affecter lors de son acquisition par exemple. Les opérateurs *filtres linéaires* et *filtres non-linéaires* sont très répandus dans ce contexte. Les premiers consistent à appliquer sur le pixel en question une combinaison pondérée des valeurs des pixels qui y sont voisins. Les deuxièmes se basent sur des combinaisons non-linéaires telles que la prise de la valeur médiane de l'ensemble des valeurs des pixels voisins. Autres opérateurs sont à citer aussi dans cette catégorie, notamment les opérateurs utilisant la *morphologie mathématique*, les transformations basées sur les distances entre pixels voisins et les transformations basées sur les composantes connexes.

Opérateurs fréquentiels Ces opérateurs servent à accélérer les opérateurs locaux et les analyser en transformant les informations spatiales données par les valeurs des pixels de l'image vers un autre espace, celui des fréquences. Ils se considèrent comme des opérateurs de pré-traitement. Les transformées de Fourier sont l'élément clé de ces modifications. Parmi ces dernières, nous trouvons : la superposition, la multiplication, la convolution et bien d'autres [151].

Opérateurs de transformations en pyramides et en ondelettes A l'inverse des opérateurs précédents qui reçoivent en argument une image et fournissent une nouvelle image qui a les mêmes dimensions que l'image originale, les opérateurs en pyramides permettent de réduire ou d'augmenter les dimensions d'une image. Cela veut dire que nous pouvons diminuer ou améliorer la résolution de l'image en manipulant le nombre de pixels. Ceci peut nous être utile lorsque nous voulons par exemple interpoler une petite image pour faire correspondre sa résolution à celle de l'imprimante de sortie ou à celle d'un écran d'ordinateur ou alternativement, lorsque nous voulons réduire la taille d'une image afin d'accélérer l'exécution d'un algorithme, pour économiser de l'espace de stockage ou pour gagner en temps de transmission. D'une manière générale, et dans des applications spécifiques, on ignore la résolution même de l'image originale et nous souhaitons donc générer un pyramide d'images, à différentes échelles et à différentes tailles, et étudier chaque cas séparément.

1. *Neighborhood (area-based or local) operators.*

Il existe plusieurs opérateurs en pyramides qui permettent de changer la résolution d'une image donnée. Les plus connus sont la sur-échantillonnage (*upsampling*), le sous-échantillonnage ou décimation (*downsampling or decimation*) et les pyramides multi-résolution. Il est à noter ici que ces opérateurs s'emploient aussi dans le domaine fréquentiel.

Une notion très proche de celle des opérateurs en pyramides dans le sens qu'ils décomposent le signal en plusieurs échelles, est celle des ondelettes [1]. Intuitivement, les ondelettes désignent des fonctions présentant des allures avec de petites oscillations, d'où le nom. Elles s'utilisent en traitement de signal comme des filtres qui localisent le signal dans le domaine spatial et dans le domaine fréquentiel. La décomposition en ondelettes se basent d'une autre part sur le formalisme des transformées de Fourier. Les transformées en ondelettes se localisent dans les deux domaines, spatial et fréquentiel.

Les deux opérateurs, en pyramides et transformations en ondelettes, sont très employés dans les domaines de l'imagerie notamment dans la vision par ordinateur, l'infographie, la compression de données..etc.

Opérateurs d'optimisation globale Ce type d'opérateurs se base sur des critères à optimiser. Ils modélisent préalablement l'objectif finale, souhaité obtenir en traitant l'image, sous forme d'une *fonction de coût* ou d'une *fonction objective*. La tâche suivante est de chercher comment obtenir une solution optimale de cette fonction.

Les opérateurs de ce type peuvent se classer en deux grandes variantes : opérateurs de *régularisation* et opérateurs utilisant le *formalisme bayésien*.

Les applications de ces méthodes comprennent par exemple l'interpolation de surface à partir de point distribués dans l'espace, le débruitage d'images, la restauration des régions disparues et la segmentation d'images en régions.

4.2.3.3 Utilisations des outils de traitement d'images

Les outils de traitement d'images sont utilisés à des fins d'*analyse d'images*. Cette dernière désigne des opérations effectuées sur une image dans le but de produire un ensemble de mesures et d'informations. Cela comprend l'indexation d'images, la recherche d'images, la détection de visages, la reconnaissance de formes (*Pattern recognition*) [2], la détection de caractéristiques (*feature detection*), le traitement d'images médicales, la télédétection (*remote sensing*) et bien d'autres champs scientifiques et industriels. Le traitement d'image est donc une discipline indispensable au domaine de la vision

1. *Wavelets* en anglais.
2. A ne pas confondre avec la reconnaissance d'objets (*Object recognition*).

par ordinateur. Le premier manipule les images sur un niveau inférieur (*low-level*) et a comme but l'amélioration de la qualité de l'image, le développement de méthodes de représentation (segmentation contours/régions), de compression (codage d'images, extensions des fichiers numériques) et aussi de restauration des parties défectueuses ou déformées. Tandis que l'autre les analyse sur un niveau supérieur (*high-level*) pour comprendre leurs contenus.

Pour la segmentation d'images, qui nous concerne de plus dans ce chapitre consacré aux applications, de nombreuses méthodes ont été développées dans la littérature. Ces méthodes peuvent se regrouper en deux approches globales, à savoir : la segmentation par régions et la segmentation par contours. Certaines de ces méthodes sont des applications directes des méthodes de classification de pixels comme la méthode de *mean-shift*, d'autres sont de nature différentes telles que la méthode de segmentation non supervisée JSEG [35], Watershed, Turbopixels [95] ou Fusion Statistique des Régions (SRM, Statistical Region Merging) [120].

Quant aux méthodes développées dans le contexte de la reconnaissance d'objets, et sans distinction des domaines d'applications ni des types d'images utilisées, nous pouvons les regrouper en quatre grandes approches : approches interactives [18; 119; 132], approches semi-automatique [13; 88; 156], approche par composantes [1; 12; 104; 163] et approches avec données externes [66; 94; 167].

4.3 Application à la segmentation d'images couleur texturées

Dans cette section, nous allons présenter une application de notre méthode développée dans cette thèse et qui concerne la classification à partir de prototypes. L'application porte sur la segmentation d'images couleur texturées.

4.3.1 Description de matériels

Nous avons vu que les descripteurs de la texture sont des caractéristiques très importantes dans le domaine du traitement d'image et des domaines qui y sont associés. Ils nous fournissent des informations sur les propriétés spatiales de l'image et sur l'agencement de ces éléments fondamentaux que sont les pixels ([36; 73; 122; 126]).

Afin d'appliquer la méthode proposée dans cette thèse, dans sa partie de classification à partir des prototypes, pour classifier des images couleur texturées, nous devrions considérer les données extraites des espaces de couleurs considérés et du modèle de

textures présentes dans l'image.

Dans le but d'apporter une comparaison des résultats, les analyses présentées dans cette section utilisent trois espaces différents de couleurs. Il s'agit de l'espace de primaires rouge, vert et bleu (RVB), l'espace de luminance-chrominance $L^*a^*b^*$ et l'espace TSL amélioré que nous préférons lui garder sa notation par son sigle anglais IHSL (pour Improved, Hue, Saturation and Luminance, comme nous l'avons déjà vu dans la section sur les espaces colorimétriques).

Nous utilisons également un modèle de prédiction linéaire bidimensionnel (en deux canaux) ([126]). Ce modèle se base sur la construction de deux canaux à partir des informations extraites de l'image à analyser. Le premier est un canal réel qui contient des valeurs de luminance. Le second est un canal complexe contenant les valeurs de chrominance.

Lorsque nous passons de l'espace de primaire RVB à l'espace perceptuel IHLS, le canal de chrominance est calculé de la manière suivante :

$$C = S \times exp^{(j \times \frac{\pi}{180} \times H)}$$

avec H la valeur de la teinte et S la valeur de la saturation d'une couleur donnée. Alors que dans la transition vers le système $L*a*b*$, le canal de chrominance est donné par :

$$C = a^* + j \times b^*$$

avec a^* et b^* sont les deux variables de chrominance ([122]).

Chaque pixel n dans l'image est alors représenté par un vecteur composé d'une valeur de luminance réelle appelée l_n et d'une valeur complexe de chrominance notée c_n telles que :

$$y_n = \begin{bmatrix} l_n \\ c_n \end{bmatrix}$$

avec $l_n \in \mathbb{R}$, $c_n \in \mathbb{C}$, $n = (n_1, n_2) \in \Lambda \subset Z^2$, Λ est une région bidimensionnelle de taille $|\Lambda| = M$.

Soit $Y = (Y_n)_{n \in Z^2}$ un champ aléatoire bidimensionnel avec P canaux (ici $P = 2$ pour les vecteurs de luminance-chrominance et $P = 3$ dans le cas d'espace de couleur RVB).

Un modèle de prédiction linéaire autoregressif $\widehat{Y}$ peut être défini par la variable aléatoire prédictive en n par :

$$\widehat{Y}_n = -\sum_{m \in D} A_m (Y_{n-m} - \mu)$$

L'erreur de prédiction linéaire est donnée par :

$$E_n = (Y_n - \mu) - \widehat{Y}_n$$

D est la région support de prédiction et $m = (m_1, m_2) \in D \subset Z^{*2}$ et $A_m, m \in D$, les matrices de coefficients de dimensions $P \times P$.

L'erreur de prédiction linéaire $E = (E_n)_{n \in Z^2}$ est supposée être un bruit blanc.

Pour des détails supplémentaires concernant ces modèles multicanaux de prédiction linéaire et les supports D qui y sont associés, nous renvoyons au papier [125]. En particulier, causal Quarter Plan (QP), causal Non Symmetric Half Plane (NSHP), Auto-Regressive(AR) et Gauss-Markov Random Field (GMRF).

Dans ce qui suit, nous donnons les résultats obtenus par la segmentation utilisant la méthode de classification proposée.

4.3.2 Segmentation d'images texturées par classification de pixels

Nous avons choisi d'appliquer la méthode développée sur un ensemble test formé d'images contenant des textures couleur réelles ainsi que d'autres générées artificiellement. La figure 4.3 montre les huit images utilisées.

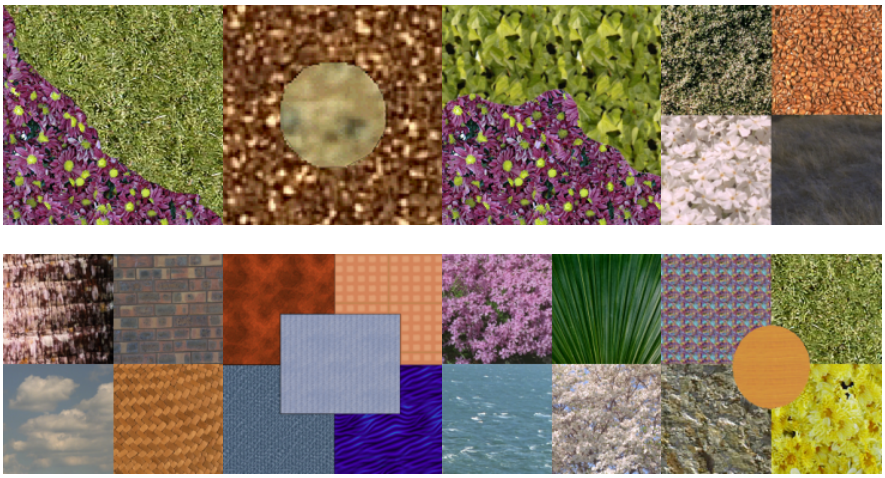


FIGURE 4.3 – Images de textures couleur utilisées pour tester l'algorithme proposé.

Chaque image a une taille de 184×184 et appartient à la base de données utilisée

dans ([73]).

Dans la première étape de cette application, nous utilisons pour chaque image de l'ensemble test, une imagette de taille 32×32 pixels extraits des textures qui la composent. Chaque imagette constituera un échantillon pour la texture de l'image. Pour utiliser le formalisme de la méthode de classification développée, nous calculons l'erreur de prédiction linéaire multi-canal de chacune des imagettes. A partir de ces valeurs nous définissons les observations prototypes : Π_k, $k = 1, \ldots, K$.

Dans la deuxième étape, nous calculons les erreurs de prédiction multi-canal pour la partie restante de l'image analysée, ce qui représentera alors le groupe Γ d'observations non classées.

L'algorithme de classification proposé est appliqué alors à ces données en utilisant un système de voisinage basé sur la distance cosinus. Cette distance (ou mesure) permet de calculer la similarité entre deux vecteurs à n dimensions en déterminant le cosinus de l'angle entre eux. Elle est donnée par :

$$cos(\theta) = \frac{A.B}{||A||.||B||}$$

La valeur $\cos\theta$ est comprise dans l'intervalle $[-1, 1]$. Si $\cos\theta = -1$, cela indiquera que les vecteurs sont opposés ; si $\cos\theta = 0$ les vecteurs sont indépendants ; si $\cos\theta = 1$, les vecteurs sont similaires (colinéaires de coefficient positif). Les valeurs intermédiaires permettent d'évaluer le degré de similarité.

Le but de cette application est, d'une part, de confirmer les résultats concernant le choix de l'espace de couleur approprié, et d'autre part, de montrer la robustesse de l'algorithme de classification proposé, dans le chapitre précédent, dans la classification de données complexes extraites des images couleur texturées.

En ce qui concerne la comparaison en différents espaces colorimétriques, les résultats obtenus sont présentés dans la figure 4.4. Ils montrent les taux d'erreur des modèles de prédiction linéaire en utilisant les trois espaces couleur.

En outre, une comparaison avec d'autres méthodes de classification est faite et la figure 4.5 montre les résultats de classification obtenus en utilisant l'algorithme k-nn, la méthode bayésienne et la méthode proposée.

De plus, des résultats de segmentation sont également donnés pour deux images (figure 4.6 et la figure 4.7) : L'image de la figure 4.3(a) et l'image de la figure 4.3(h). La figure 4.6 montre les résultats obtenus sur l'image de la figure 4.3(a) avec les deux espaces couleur (L*a*b* et IHLS) qui donne les meilleurs résultats dans cette étude.

La figure 4.6 et la figure 4.7 fournissent également une comparaison visuelle numé-

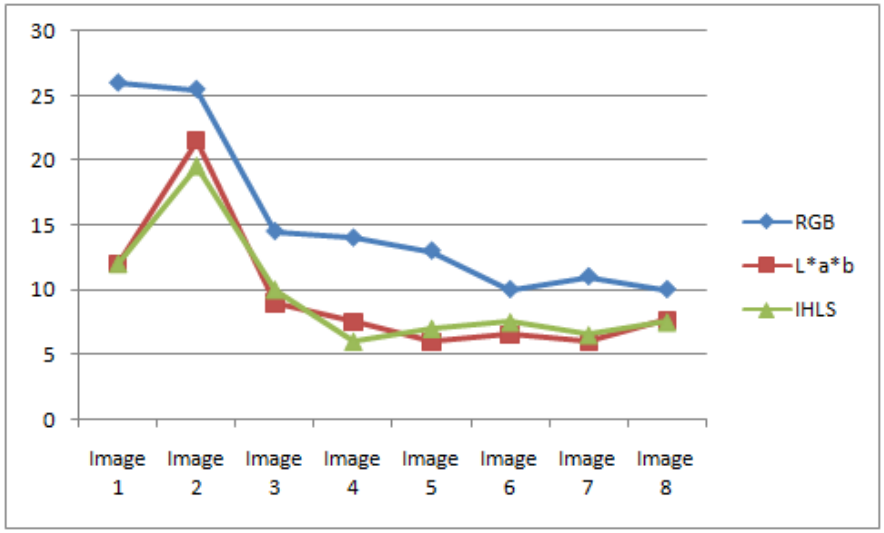


FIGURE 4.4 – Taux d'erreur de clustering par rapport aux espaces couleur utilisé.

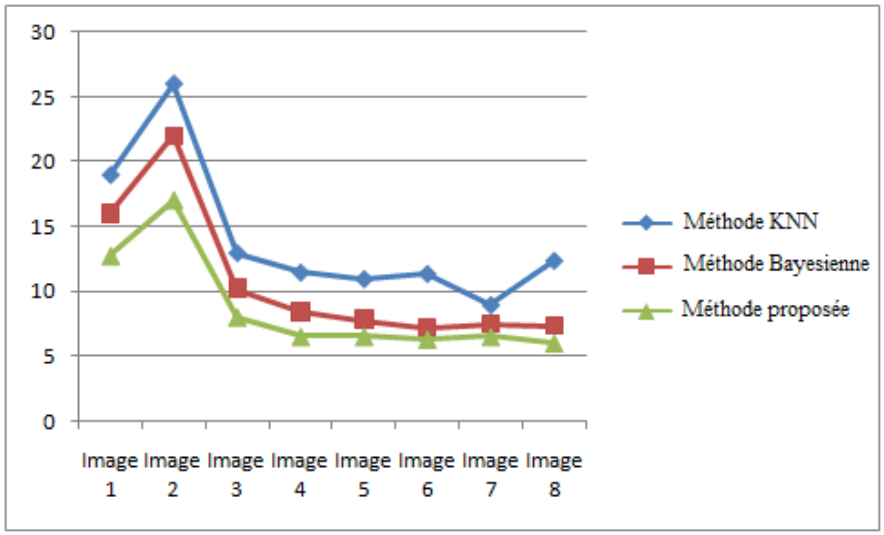
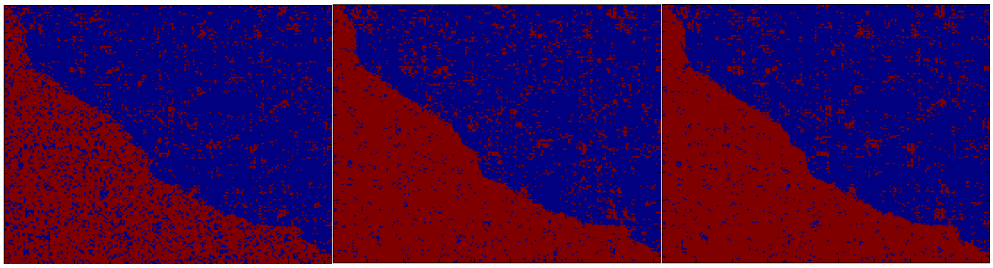
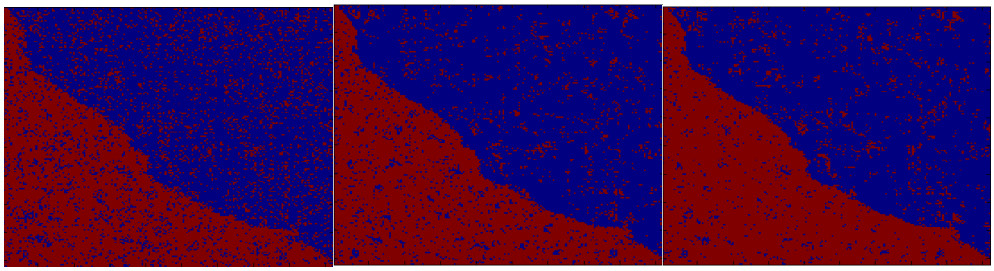


FIGURE 4.5 – Taux d'erreur relatifs aux méthodes de comparaisons ainsi qu'à notre méthode.

rique et entre la méthode proposée et les autres procédés de classification mentionnés ci dessus confirmant l'efficacité de celle que nous avons proposée.

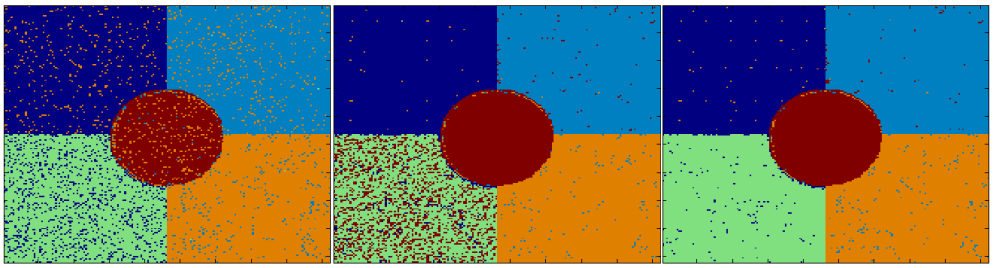


(a) K-nn (13.8%) (b) Bayes (9.3%) (c) Our method (8.1%)



(d) K-nn(14.5%) (e) Bayes (8.9%) (f) Our method (7.4%)

FIGURE 4.6 – Segmentation de l'image 1 (Fig. 4.3(a)) dans IHLS (top), L*a*b* (bottom).



(a) knn method (b) Bayesian method (c) Proposed method

FIGURE 4.7 – Segmentation de l'image 8 (Fig. 4.3(h)) dans l'espace IHLS

4.4 Reconnaissance d'objets par mesure de similarité entre régions à partir des images aériennes

Nous sommes arrivés maintenant à la partie qui concerne le dernier travail dans ce mémoire, il s'agit du développement d'une méthode pour la reconnaissance/ extraction d'objets d'intérêt dans les images à haute résolution. La contribution que nous proposerons est introduite afin de palier à l'inconvénient des méthodes de reconnaissance basées sur l'interaction de l'utilisateur (voir 4.2.3.3).

Nous présenterons dans le paragraphe 4.4.1 les différentes étapes de l'algorithme développé, nous aborderons ensuite une application particulière destinée à la reconnaissance de toitures à partir des images orthophotoplans 4.4.2.

4.4.1 Algorithme développé pour la reconnaissance d'objets

Nous présenterons dans ce paragraphes la technique proposée pour la reconnaissance d'objets d'intérêt à partir des images aériennes.

4.4.1.1 Description générale de l'algorithme

Comme la plupart des méthodes de reconnaissance, la méthode que nous proposons pour faire de la reconnaissance/détection d'objets comporte deux phases principales : hors ligne et en ligne.

Dans la première phase, deux bases de connaissances sont à créer dont l'objectif est principalement d'éviter l'interaction de l'utilisateur. La première base contient des textures représentatives et distinctives prises des objets d'intérêt à extraire. L'autre base est composée de quelques textures prises à partir des objets appartenant au fond de l'image.

Afin d'illustrer ces propos de la construction des bases de connaissances, nous prenons un exemple d'une application de cette méthode pour l'extraction de toits à partir d'images aériennes prises pour des zones urbaines[1]. La première base de connaissances $\mathbf{B_{obj}}$ sera construite par la prise de m textures distinctives des toits des bâtiments tandis que la seconde base $\mathbf{B_{back}}$ sera construite par la prise de n textures distinctives des autres objets tels que la végétation, les routes, la forêt,.. etc. Ces deux bases de connaissances sont donc construites à partir d'un échantillons d'images. La figure 4.8 présente un exemple possible de ces bases de connaissances utilisées dans ce contexte.

Ayant ces deux bases de connaissances $\mathbf{B_{obj}}$ et $\mathbf{B_{back}}$ comme référence, il sera donc possible d'extraire automatiquement les toits des bâtiments à partir de toute image aérienne en confrontant leurs caractéristiques visuelles. Ceci est l'objet du traitement effectué dans la phase en ligne.

Dans cette phase, le processus de reconnaissance/détection d'objets d'intérêt est effectuée. Pour ce faire, nous commençons par une sur-segmentation de l'image originale qui va nous permettre d'avoir une partition de l'image en de nombreuses régions homogènes. C'est ce qu'on appelle une étape de traitement bas-niveau. Dans ce travail,

1. Nous reviendrons à cette application dans la section suivante (cf. 4.4.2)

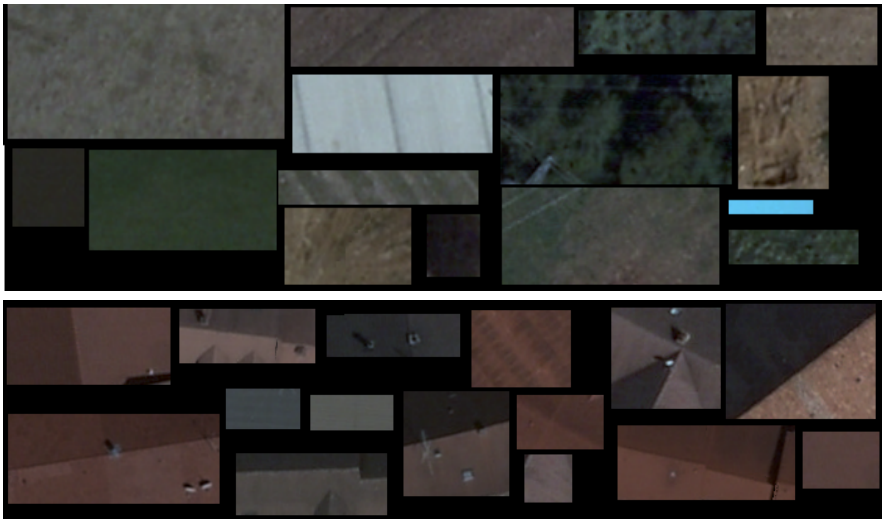


FIGURE 4.8 – Exemple de bases de connaissances construites pour ce travail. De haut en bas : la base de connaissance $\mathbf{B_{back}}$ des objets du fond d'image (végétation, route, forêt, etc) et la base de connaissances $\mathbf{B_{obj}}$ des toits des bâtiments (des toits rouges et non rouges).

nous avons utilisé l'algorithme de fusion statistique des régions SRM (pour *Statistical Region Merging*) [120] comme méthode de segmentation (cf. section 4.4.1.2).

A partir de l'image segmentée, la tâche suivante est une étape de traitement haut-niveau qui consiste à extraire les attributs (*features*) des régions à la fois de l'image segmentée et des bases de connaissances construites. Dans ce travail, nous utilisons les histogrammes de couleur prises dans l'espace colorimétrique de primitives RGB (cf. section 4.4.1.3).

La question qui se pose alors est celle du choix de la mesure de similarité entre les régions. Parmi les mesures développées dans la littérature, nous avons retenu le coefficient de Bhattacharyya pour accomplir cette tâche (cf. section 4.4.1.4).

Une fois la mesure de similarité est évaluée pour toutes les régions, chacune d'entre elles peut être classée comme étant une partie d'un objet d'intérêt ou bien comme étant une partie de l'arrière-plan de l'image (voir la section 4.4.1.5).

Enfin, les contours des objets d'intérêt vont être délimités pour ne garder que les régions marquées par l'étiquette d'objet d'intérêt.

L'organigramme de la figure 4.9 résume cette procédure de détection de toitures des bâtiments.

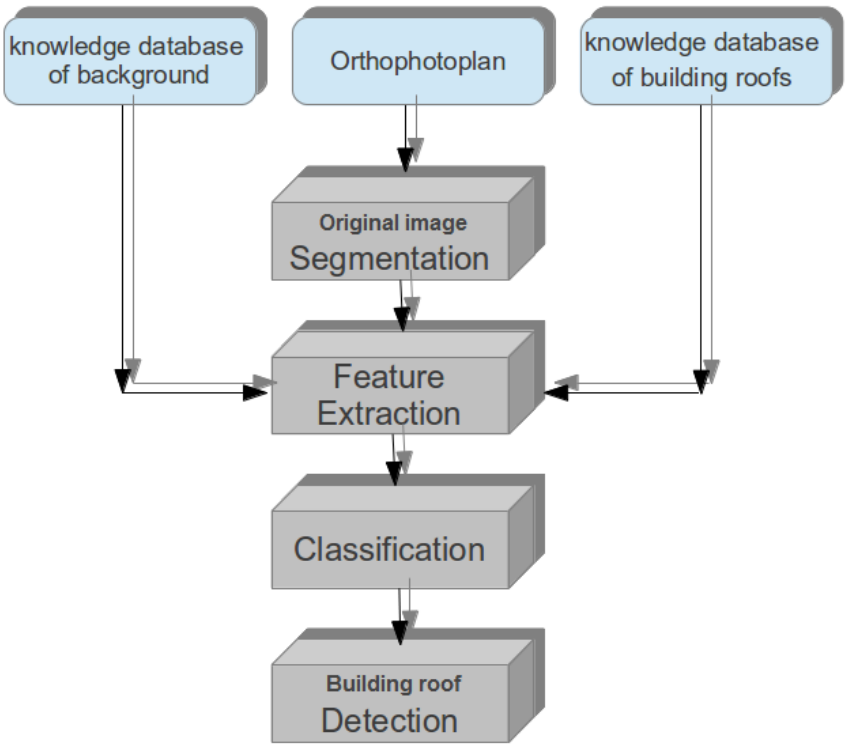


FIGURE 4.9 – Organigramme de la méthode proposée pour détection d'objets d'intérêt.

4.4.1.2 Segmentation initiale par la méthode SRM

L'étape du traitement bas-niveau consiste à sur-segmenter les images test en plusieurs régions en se basant sur des critères d'homogénéité, c'est à dire obtenir une partition des images en régions disposant des mêmes caractéristiques visuelles. Le but de cette segmentation initiale est d'éviter le problème de la sous-segmentation qui confond des régions et donc pousser l'algorithme à extraire correctement toutes les régions importantes dont les frontières coïncident aussi étroitement que possible avec les contours des objets présents dans l'image.

Dans ce travail, nous avons choisi l'algorithme SRM. Parmi les avantages de cet algorithme dans le contexte du traitement des images à haute résolution, nous mentionnons d'abord le fait qu'il utilise une structure de graphe d'adjacence pour représenter l'ensemble des régions (*Region Adjacency Graph*, RAG). Deuxièmement, la méthode SRM permet de définir une hiérarchie de partitions. Troisièmement, elle fonctionne en temps linéaire en utilisant l'algorithme de *tri par paquets* (*bucket sort*) tout en traversant

la structure RAG. En fin, la méthode SRM prend en considération les informations spectrales, les structures spatiales des formes et les changements de l'échelle, en plus d'une robustesse au bruit et à l'occlusion. La figure 4.10 présente un exemple de la segmentation d'une image par l'algorithme SRM.

L'ensemble des régions obtenues lors de la segmentation initiale par SRM, sera noté $\mathbf{M_{SRM}}$.

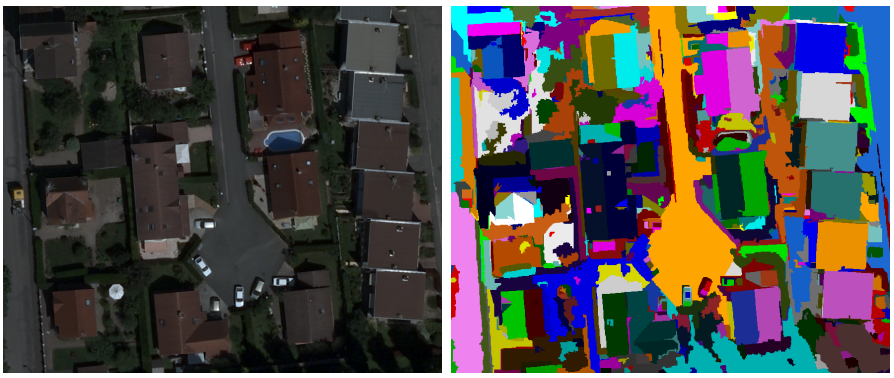


FIGURE 4.10 – Exemple de résultat de segmentation utilisant la méthode SRM. De gauche à droite : image origine et résultat de segmentation SRM.

4.4.1.3 Représentation des régions

Nous disposons maintenant d'une image segmentée obtenue par l'algorithme de SRM. Il est toutefois précoce d'extraire avec précision les contours des objets d'intérêt à partir de cette image. En fait, la seule information dont nous disposons vient de la délimitation des contours des régions et aucune estimation de l'information sur leurs contenus, qui est nécessaire pour le processus de reconnaissance, n'est encore disponible.

Afin de quantifier l'information du contenu visuel des régions, il faudrait passer par la tâche du choix des descripteurs d'abord et puis extraire les attributs correspondants.

Nous avons vu qu'il y a dans la littérature plusieurs descripteurs qui pourraient être considérés pour représenter une région, à savoir la texture, la forme, la taille ou la couleur. Dans le contexte de la détection de toitures à partir des images aériennes, l'information la plus appropriée est la couleur. En effet, les descripteurs de la texture, qui peut être définie comme étant la répétition de motifs locaux suivant des règles d'agencement spatial, sont malheureusement difficile à décrire ; en outre, les mêmes difficultés peuvent être rencontrées en ce qui concerne la forme et les contours. Quant au descripteur de taille, bien qu'elle peut être mesurée simplement en calculant le nombre

de pixels de chaque région, ce descripteur ne permet pas une distinction unique des objets recherchés. La raison est qu'un objet d'intérêt peut avoir des tailles différentes en passant d'une image à une autre voire dans la même image avec plusieurs instances de l'objet d'intérêt ; ou simplement parce que l'objet peut avoir la même taille que d'autres objets appartenant au fond de l'image.

Par conséquent, le choix est tourné vers les descripteurs de la couleur, plus précisément, les histogrammes de la couleur. Ces histogrammes de la couleur des régions sont, pour rappeler, des histogrammes locaux qui reflètent les caractéristiques locales dans l'image. Nous exploitons donc ces descripteurs pour caractériser les régions de l'image segmentée ainsi que celles des deux bases des connaissances construites.

D'abord, nous quantifions uniformément chaque canal de couleur en $l = 16$ classes ; après, l'histogramme de couleur de chaque région est calculée dans l'espace caractéristique de dimensions $l \times l \times l = 4096$ classes. La quantification réduit les informations du contenu des régions et elle est utilisée lorsque l'on veut réduire le temps de traitement. Ces calculs sont effectués dans l'espace colorimétrique des primaires RGB.

4.4.1.4 Mesure de similarité adoptée

Maintenant que nous avons défini et explicité les descripteurs de régions, nous devrions calculer les mesures de similarité que nous noterons ϱ entre les régions de la base et celles de l'image test.

Dans le domaine de la reconnaissance d'objets ou d'une manière générale, dans la plupart des tâches relevant de l'intelligence artificielle, nous sommes souvent invités à faire des correspondances et des comparaisons entre deux grandeurs ou deux entités afin d'en établir une ressemblance. Ceci est connu sous le terme de *recherche de similarités* (*Similarity search*). Il est fortement sollicité dans les tâches de la recherche par contenu des données multimédia telles que les données images, audio ou vidéo [8]. Ceci est requiert à titre d'exemple lors de l'étude des variations temporelles du contenu visuel dans une vidéo, ou lors de la recherche d'images similaires à une image requête.

Cette recherche de similarité s'effectue au travers de la détermination d'une distance ou d'une mesure de similarité. Nous pouvons constater dans la littérature, qu'il y a autant de mesures que les applications et les domaines de recherche. Dans les applications de la recherche d'images par contenu, à titre d'exemple, ces mesures dépendent des critères de la recherche mais également de la représentation des caractéristiques de l'image.

La plupart des mesures de similarité couramment utilisées se définissent dans un

espace vectoriel. Cela veut dire que les caractéristiques des objets sont représentées sous forme de vecteurs multi-dimensionnels et qu'une distance ou une métrique est définie sur cette espace.

Étant donnés deux vecteurs X et Y de mêmes dimensions, une mesure de similarité respectivement une mesure de dissimilarité est une mesure qui quantifie la dépendance respectivement l'indépendance entre les deux vecteurs.

Avec une mesure de similarité, plus la dépendance entre les vecteurs X et Y est grande, plus la valeur de la mesure l'est. A l'inverse, avec une mesure de dissimilarité, plus l'indépendance entre les deux vecteurs X et Y est grande, plus la distance entre eux est grande.

Il est à noter ici qu'une mesure peut être utilisée avec efficacité pour la spécification de similarité entre les objets d'un espace sans pour autant être une métrique même si ceci est souhaitable. Pour plus de détails et d'exemples sur ces mesures, nous renvoyons le lecteur à [63].

Parmi les mesures de similarité les plus communes, nous évoquons la mesure de Minkowski, la mesure d'intersection [150], distance de deuxième type [149], le coefficient de Bhattacharyya [80], et le rapport de log-vraisemblance [53].

Il existe bel et bien d'autres mesures de similarité et d'autres distances, mais nous n'avons évoqué que celles qui peuvent être adaptées à notre application sur la reconnaissance d'objets d'intérêt. Ces mesures peuvent être trouvées dans les références citées ci-dessus.

Parmi ces mesures, nous avons opté pour le coefficient de Bhattacharyya, qui représente le cosinus de l'angle entre les vecteurs unitaires de deux objets donnés.

En utilisant la notation $\varrho(R, Q)$ pour représenter la valeur de la mesure de similarité entre les régions R et Q. Plus ϱ est grande, plus la similarité visuelle entre les régions R et Q l'est aussi. Nous allons obtenir l'expression suivante :

$$\varrho(X, Y) = \sum_{i=1}^{z} \sqrt{Hist_R^i . Hist_Q^i} \tag{4.1}$$

avec $Hist_R^i$ et $Hist_Q^i$ les i^{eme} éléments des vecteurs unitaires $(\sqrt{Hist_R^1}, ..., \sqrt{Hist_R^z})^T$ et $(\sqrt{Hist_Q^1}, ..., \sqrt{Hist_Q^z})^T$ représentant les deux histogrammes normalisés des régions R et Q respectivement. $z = l \times l \times l = 4096$ représente la dimension de l'espace des attributs couleur après une quantification.

Certes, deux histogrammes similaires n'impliquent pas nécessairement que les deux régions correspondantes sont visuellement similaires. Néanmoins, sa combinaison avec

la procédure de classification proposée, c'est à dire la règle de décision de l'appartenance d'une région à une classe que nous allons introduire dans la section suivante 4.4.1.5, a permis d'obtenir des résultats satisfaisants.

Bien que les descripteurs d'histogrammes couleur soient des descripteurs globaux de l'image, nous les utilisons dans ce travail comme des descripteurs locaux en les calculant sur des régions obtenues à partir de l'image segmentée. Il sont robuste donc au bruit et aux petites variations. Étant donné que le coefficient de Bhattacharyya est le produit intérieur de deux vecteurs des histogrammes couleur, ce coefficient est aussi robuste au bruit et aux petites variations. Cette mesure a été utilisée dans [26; 119] pour le développement d'une méthode interactive de reconnaissance d'objets.

4.4.1.5 Étape de classification

Nous chercherons dans cette étape à élaborer une règle de décision qui nous permette d'affecter chaque région de l'ensemble $\mathbf{M_{SRM}}$ à l'une des deux classes : la classe des régions faisant parties des objets d'intérêt et la classe des objets qui font partie du fond de l'image.

Dans cet objectif, nous calculons d'abord les valeurs de la mesure de similarité pour chaque région $R \in \mathbf{M_{SRM}}$ avec toutes les régions des deux bases de connaissances $\mathbf{B_{obj}}$ et $\mathbf{B_{back}}$. Puis, nous enregistrons ces valeurs dans des vecteurs V^R_{obj} et V^R_{back} respectivement. Ces vecteurs vont être ordonnés selon un ordre décroissant. Ensuite, calculons-nous la moyenne sur ces vecteurs en ne prenant que les p respectivement q premières valeurs. En fin, la région R est donc affectée à la classe qui a la valeur moyenne maximale.

Une fois toutes les régions de $\mathbf{M_{SRM}}$ sont classées, nous pouvons directement extraire les objets d'intérêt en fusionnant ceux qui sont adjacents.

Nous résumons cette description de la méthode proposée sous forme de l'algorithme suivant 3.

Il est vrai que la règle de décision est simple, cependant elle est très efficace pour la procédure de classification. Il est à noter que la valeur moyenne des mesures de similarité moy^R_{obj} (respectivement moy^R_{back}) est inversement proportionnelle à la valeur de p (respectivement q) ; ceci veut dire que plus la valeur de p ou de q est élevée, moins la valeur moyenne de similarité l'est.

Dans le cas de grandes valeurs de p et de q, nous obtiendrions une dispersion des valeurs moyennes de similarité et la classification de certaines régions de $\mathbf{M_{SRM}}$ serait incorrecte.

Algorithm 3 Multiple objects extraction algorithm

Require: $I \leftarrow$ input image.
$\mathbf{B_{obj}} \leftarrow$ Knowledge database of objects of interest (building roofs).
$\mathbf{B_{back}} \leftarrow$ Knowledge database of background (vegetation, road, forest, etc)

(over)Segment I into regions through SRM algorithm in order to obtain the set $\mathbf{M_{SRM}}$ of segmented regions.
Calculate the RGB color histogram features for all regions of $\mathbf{M_{SRM}}$ and for those composing the two constructed knowledge databases $\mathbf{B_{obj}}$ and $\mathbf{B_{back}}$.
for each candidate region R $\in \mathbf{M_{SRM}}$ **do**
  Calculate the similarity vector $V_{obj}^R = \{\varrho(R, Q_i); (Q_i)_{i=1..m} \in \mathbf{B_{obj}}\}$ between R and $\mathbf{B_{obj}}$. $\varrho(R, Q_i)$ is the similarity between the region R and the region $Q_i \in \mathbf{B_{obj}}$.
  Calculate the similarity vector $V_{back}^R = \{\varrho(R, Q_j); (Q_j)_{j=1..n} \in \mathbf{B_{back}}\}$ between R and $\mathbf{B_{back}}$. $\varrho(R, Q_j)$ is the similarity between the region R and the region $Q_j \in \mathbf{B_{back}}$.
  Get the order of V_{obj}^R and V_{back}^R by decreasing sorting ;
  Calculate $moy_{obj}^R = \frac{\sum_{i=1}^{q} \varrho(R,Q_i)}{q}$, $q \leq m$, the mean of the q first elements of V_{obj}^R.
  Calculate $moy_{back}^R = \frac{\sum_{j=1}^{p} \varrho(R,Q_j)}{p}$, $p \leq n$, the mean of the p first elements of V_{back}^R.
  if $(moy_{obj}^R \geq moy_{back}^R)$ **then**
    The region R maximizes the similarity with $\mathbf{B_{obj}}$, it is then classified as a part of building roof.
  else
    The region R maximizes the similarity with $\mathbf{B_{back}}$, it is then classified as a part of background.
  end if
end for
The classification process is finished.
return The final segmentation map.

Bien que le calcul des valeurs moyennes moy_{obj}^R et moy_{back}^R soient sensibles aux valeurs trop élevées de p et de q, nous avons constaté empiriquement qu'il y a une plage de valeurs où le résultat de la classification reste stable. Dans les expériences présentées dans ce mémoire, nous avons fixé p et q sur les valeurs 7 et 9 respectivement.

4.4.2 Application pour la détection de toitures dans les images orthophotoplans

Dans cette dernière section, nous présenterons une application de la procédure proposée pour faire la multi détection de toitures des bâtiments à partir des orthophotoplans. Nous calculerons les descripteurs des régions des deux bases de connaissances construites une fois pour toutes et nous les enregistrerons dans un fichier binaire. Cela nous évitera de les calculer plusieurs fois et donc nous gagnerons en temps de calcul. Ainsi, le processus de mesure de similarité est réalisée à l'aide de ce fichier binaire et non plus les deux bases de connaissances.

Nous effectuerons également une évaluation des résultats obtenus tout en les comparant avec d'autre méthodes classiques.

4.4.2.1 Données expérimentales

Nous décrivons dans ce paragraphe les données expérimentales ainsi que les deux bases de connaissances.

Matériel et bases de connaissances : Les données utilisées dans cette application afin d'évaluer les performances de l'algorithme proposé sont des images aériennes à hautes résolutions, précisément des orthophotoplans.

Les images utilisées sont des images qui ont été acquises pour la région de la ville de Belfort, situé au nord-est de la France en 2003, à partir d'un ballon à air chaud. Leur résolution spatiale est de 16 cm/px. Ces images couvrent une zone très large, où apparaissent des objets très divers. Les différences visuelles concernent la couleur, la texture, les occultations, l'appartenance à différentes catégories, la présence des ombres et une certaine variabilité au niveau de la hauteur au terrain.

Les objets auxquels nous nous sommes intéressés pour en faire la reconnaissance font partie de la catégorie *bâtiment* ; Il s'agit plus précisément des toitures de ces bâtiments. Bien que ces objets d'intérêt soient majoritairement distinguables des autres objets du fond de l'image, ils exhibent pour autant des différences entre eux-même, ce qui constituera un niveau supplémentaire de difficulté à surmonter par la méthode proposée. Ces différences entre les instances de l'objet d'intérêt (toitures) intéressent notamment :

- des différences minimes de couleur : rouge et moins rouge.
- des différences au niveau de la contraste/luminosité.
- des problèmes d'occultation.

Étant donné que l'efficacité de la méthode repose en premier temps sur la construction des bases de connaissances, ces différences doivent être prises en considération pendant cette étape et avoir une représentation satisfaisante.

Cette construction est explicitement réalisée, comme nous l'avons déjà mentionné, en extrayant un certain nombre de textures distinctes représentant les toits et d'autres représentant les objets du fond de l'image. Nous avons illustré cela sur la table 4.1.

Sur la deuxième ligne de cette table, nous comptons un total de treize textures relevant des toits. Parmi elles, sept sont rouges et six autres sont gris ou presque noir.

Ainsi, nous serons rassurés que les différences de contraste et de luminosité des objets d'intérêt ont été prises en compte.

Également, la troisième ligne de la même table 4.1 montre les différentes textures prises pour les objets du fond. Celles ci sont au totale de quinze textures dont cinq sont liées à la végétation et le même effectif pour les routes ; quatre textures sont choisies pour le sol et enfin une seule texture pour représenter des eaux de piscines.

TABLE 4.1 – Textures utilisées pour construire les bases de connaissances d'objets d'intérêt et du fond.

	Végétation	Routes	Sol	Piscines	Toits rouges	Toits non-rouges	Total
Object database	-	-	-	-	7	6	13
Background database	5	5	4	1	-	-	15

Images test : L'ensemble des images sur lesquelles la méthode a été appliquée est formé d'images extraites d'une énorme image (tout comme il a été décrit au paragraphe précédent 4.4.2.1). La raison est pour avoir un temps raisonnable pour effectuer les expériences au cours de l'étude. De plus, nous n'avons présenter dans ce rapport de thèse que les résultats apportés par une partie réduite de cet ensemble et qui contient six images.

Lors de l'extraction de ces images, nous avons essayé de respecter les différents niveaux de difficultés que nous avons précédemment évoquées. Ayant ceci à l'esprit, les variables suivantes ont été respectées : le nombre de toits dans la scène, leurs tailles, leurs couleurs et enfin le degré de divergence entre les toits et le fond de l'image.

Les résultats de cette application de la méthode proposée sur quatre de ces six images traitées sont présentés sur la figure 4.11. Nous voyons les images originales sur la première ligne de cette figure, les mêmes images après avoir été segmentées par la méthode SRM sur la deuxième ligne et enfin les résultats de l'extraction des toits de chacune des images sont exposés sur la dernière ligne.

4.4.2.2 Évaluation des performances de la méthode

Afin d'évaluer les résultats obtenus, nous proposons de faire une évaluation qualitative en utilisant ces résultats ainsi qu'une évaluation quantitatives.

S'appuyant sur l'observation visuelle des résultats, nous pouvons affirmer dorénavant que l'approche développée démontre une très bonne performance en termes de reconnaissance des objets d'intérêt. En effet, la majorité des toitures des bâtiments présents dans les images sont détectées avec une précise délimitation de leurs contours.

La méthode a donné de bons résultats de détection même dans un environnement complexe composé de bâtiments avec des toits de couleur différentes rouges et non-rouges, des routes, de la végétation,.. etc. Les images 4.11.a, 4.11.b, 4.11.c et 4.11.d. confirme bien ce constat.

Cependant, nous voyons aussi à partir des résultats expérimentaux montrés sur la figure 4.11 que de fausses alarmes ou de fausses détections ont été générées. Ce qui peut être relié à la forte ressemblance entre la radiométrie des toitures et celle du fond de l'image.

En effet, bien qu'elle ait abouti à la détection automatique et multiple de la majorité de toitures, la méthode a raté une partie des objets d'intérêt, et ce notamment lorsque leur contraste est faible comparativement aux objets du fond.

D'autre part, certaines zones qui font parties de la catégorie *végétation* sont affectées à tort à la classe des objets d'intérêt vu encore leurs caractéristiques radiométriques assez similaires. La figure 4.12 montre certains de ces éléments qui n'ont pas été extraits et d'autre qui le sont à tort. Les premiers sont marqués par des ellipses jaunes tandis que les derniers sont marqués par des ellipses vertes.

En ce qui concerne l'évaluation quantitative, nous utilisons des mesures d'évaluation largement utilisées dans la littérature dans le but de mesurer l'efficacité des méthodes de classification et qui constituent un outil utile et incontournable dans le domaine de la reconnaissance d'objets [136].

Les images orthophotoplans utilisées dans ces expériences comportent un total de 100 bâtiments (toitures). Les contours de ces toitures ont été tout d'abord délimités manuellement. Chaque pixel de ces images est ainsi affecté à l'une des deux classes *object* et *background*. Ceci nous permet de les utiliser comme vérité terrain (*ground truth*) pour le confronter par la suite avec les contours délimités automatiquement.

Les résultats de l'extraction manuelle et ceux de l'extraction automatiques sont ainsi comparés pixel par pixel. Chaque pixel de l'image est ensuite classé dans l'un des quatre ensembles :

1. Vrai positive (True Positive TP) : Les deux méthodes manuelle et automatique affectent les pixels à la classe *object*.

2. Vrai négative (True Negative TN) : Les deux méthodes manuelle et automatique affectent les pixels à la classe *background*.

3. Faux positive (False Positive FP) : La méthode automatique affecte à tort les pixels à la classe *object*.

4. Faux négative (False Negative FN) : La méthode automatique affecte à tort les pixels à la classe *background*.

Nous calculons ensuite le nombre de pixels qui tombent dans chacune des quatre catégories TP, TN, FP, FN et nous en déduisons les mesures suivantes :

$$\begin{aligned} \text{Branching Factor (B.F)} &= \frac{FP}{TP} \\ \text{Miss Factor (M.F)} &= \frac{FN}{TP} \\ \text{Detection Percentage (D.P)} &= 100.\frac{TP}{TP+TN} \\ \text{Quality Percentage (Q.P)} &= 100.\frac{TP}{TP+FP+FN} \end{aligned}$$

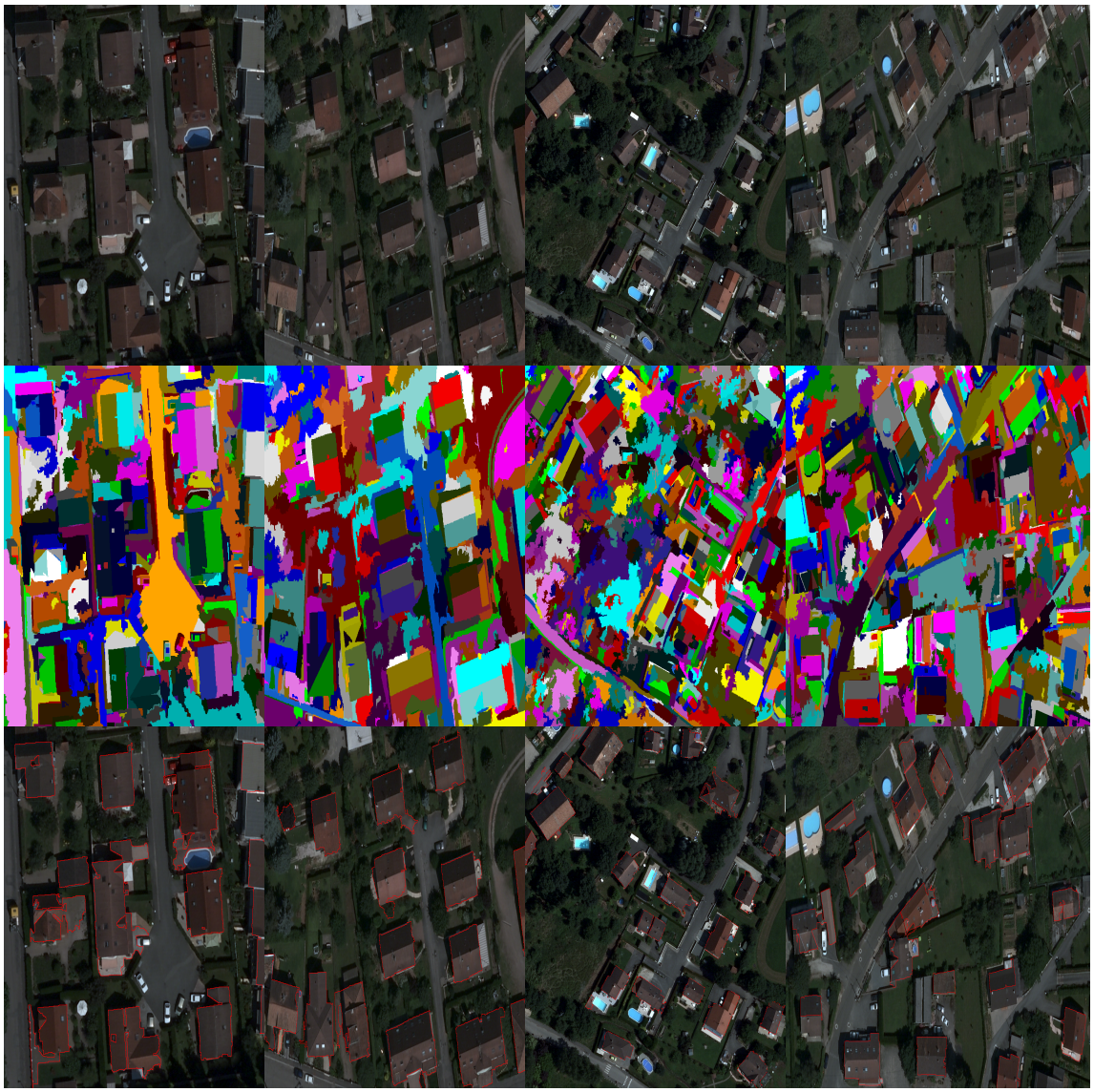
L'interprétation des mesures ci-dessus est la suivante. Le facteur de branchement mesure l'erreur que l'algorithme a commis en étiquetant des pixels de fond comme faisant partie des toitures. Plus la détection est précise, plus sa valeur est proche de zéro. Le *miss factor* mesure l'erreur d'omission où la méthode étiquettes d'une manière incorrecte, des pixels de toitures comme étant des pixels de fond. Le pourcentage de détection indique le pourcentage des pixels de toitures correctement étiquetés par la méthode.

Contrairement à ces métriques de qualité qui sont étroitement liées à la performance de la délimitation des contours des objets d'intérêt par la méthode d'extraction, la mesure relative au pourcentage de la qualité mesure la qualité absolue de l'extraction/reconnaissance et elle est donc la mesure la plus significative. Pour obtenir une qualité à 100%, l'algorithme devrait étiqueter correctement tous les pixels de toitures ($FN = 0$) sans pour autant mal étiqueter aucun pixel de fond ($FP = 0$).

Les résultats de l'évaluation de la qualité de la méthode pour les images tests illustrés dans les figures 4.11 et 4.12, sont montrés sur la table 4.2.

Nous nous intéressons en particulier à la dernière ligne de cette table sur lequel nous avons rapporté les valeurs moyennes obtenues sur toutes les images orthophotoplans utilisées dans ce travail. Ces valeurs confirment les conclusions de l'évaluation qualitative précédente en ce qui concerne la performance de l'approche proposée.

En effet, Ces résultats montrent que l'approche proposée pour la reconnaissance/extraction des toitures a suffisamment réussi avec des valeurs moyennes de D.P et Q.P de 93,91% et 85,30% respectivement. De plus, les valeurs moyennes de chacun du facteur de branchement et du facteur d'omission ont atteint 0,111 et 0,067, respectivement.



a	b	c	d
Image 1	Image 2	Image 3	Image 4

FIGURE 4.11 – Extraction automatique de plusieurs toits de bâtiments de l'ensemble des images traitées. De haut en bas : les images originales, les résultats de segmentation par SRM et l'extraction de toits correspondants.

a. Image 5

b. Image 6

FIGURE 4.12 – Exemples de certains éléments de toitures qui n'ont pas été extraits (ellipses jaunes) et de certaines fausses détections (ellipses vert).

TABLE 4.2 – Résultats de l'évaluation de la qualité d'extraction des toitures.

Images \ measures	B.F	M.F	D.P	Q.P
Fig.4.11.a	0.1745	0.1116	90	77.75
Fig.4.11.b	0.14	0.0289	97.19	85.58
Fig.4.11.c	0.1804	0.0303	97.06	82.59
Fig.4.11.d	0.0871	0.1126	89.87	83.35
Fig.4.12.a	0.0184	0.0789	93.37	92.66
Fig.4.12.b	0.0710	0.0414	96.01	89.88
Avr./100 building roofs	**0.1118**	**0.0673**	**93.91**	**85.30**

Par ailleurs, nous avons traduit ces résultats de comparaison en termes de la *caractéristique de performance* connue fréquemment sous le terme *courbe sensibilité/spécificité* ou encore *courbe ROC* (de l'anglais *Receiver Operating Characteristic*) [45].

Dans le domaine de l'apprentissage machine, les courbes ou graphes ROC sont utilisées comme une technique pour la visualisation et la sélection des classifieurs en fonction de leurs performances. Les graphes ROC se représentent sur un espace à deux dimensions dans lequel le **taux de vrais positifs (TPR)** (aussi appelé *rappel* ou *sensibilité*) est tracée sur l'axe des X et le **taux de faux positifs (FPR)** (aussi appelé *taux de fausses alarmes*) est tracée sur l'axe des Y.

Ces deux taux sont calculés en utilisant les quatre catégories mentionnées ci-dessus, et on a les formules suivantes :

$$TPR = \frac{TP}{TP + FN} \tag{4.2}$$

$$FPR = \frac{FP}{FP + TN} \tag{4.3}$$

Comme règle générale dans le contexte des classifieurs discrets, le point le plus important dans un espace ROC est le coin supérieur gauche, c'est à dire le point de coordonnées (0,1). Il représente une parfaite classification .

Un point dans l'espace ROC est ensuite meilleure qu'un autre si le premier est situé sur le côté nord-ouest du deuxième, autrement dit,le taux TPR est élevé tandis que le taux FPR est faible). Les points situés au dessus de la diagonale divisant l'espace ROC représentent de bons résultats de classification (mieux qu'une classification aléatoire), alors que les points au dessous de cette ligne représentent de mauvais résultats (pire qu'un classifieur aléatoire). En somme, plus la courbe ROC est tirée vers le point haut

à gauche de l'espace ROC, plus la précision des résultats est parfait.

Nous représentant donc les résultats de l'évaluation de la méthode de reconnaissance d'objets que nous avons proposée avec le graphe ROC ; ceci est tracé sur la figure 4.13. S'appuyant sur ce graphe, toutes les mesures sont sur le côté supérieur gauche, indiquant et confirmant ainsi la performance de détection de la méthode appliquée sur l'ensemble des images test.

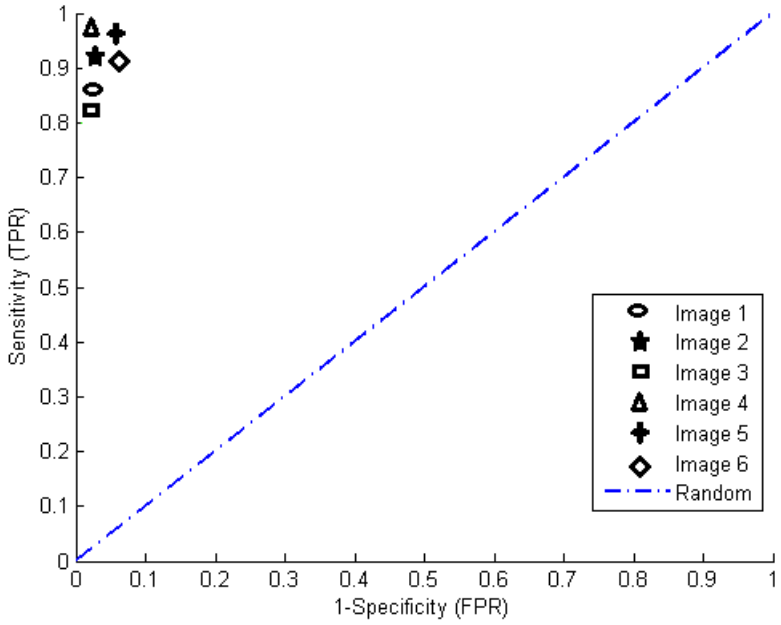


FIGURE 4.13 – Le graphe ROC de la méthode proposée appliquée sur les images de test.

4.4.2.3 Évaluation Comparative

Dans cette section, nous effectuons les résultats d'une évaluation comparative de la performance de la méthode développée dans ce travail.

Considérant que l'approche proposée repose essentiellement sur une technique de classification supervisée, nous proposons dans cette évaluation de la comparer avec deux méthodes de classification supervisée : la méthode des K plus proche voisins (KPP) et les Machines à Vecteurs Support (Support Vector Machine (SVM)).

Nous avons déjà évoqué ces méthodes dans le premier chapitre, nous rappelons tout simplement ici que toutes les deux demandent des paramètres à ajuster préalablement.

Pour la méthode KPP, nous devrions déterminer le paramètre k des voisins et le type de distance. Quant à la méthode SVM, les principaux paramètres sont le type de SVM, le type de fonction de noyau et le degré de la fonction de noyau. Le lecteur pourrait se référencer à [27] pour de plus d'explications sur ces paramètres.

Dans cette comparaison avec les classifieurs de KNN et SVM, l'ensemble d'apprentissage est donné par les histogrammes de couleur des régions des deux bases de connaissances dont les étiquettes sont connues a priori tandis que l'ensemble test est donné par l'ensemble des histogrammes couleur des régions des images test segmentées.

Afin d'obtenir une comparaison significative, chacune de ces méthodes de comparaison devrait être testée avec plusieurs combinaisons de ces paramètres dans le but de trouver un meilleur compromis qui donne le meilleur résultat. Ensuite, chaque méthode est appliquée sur les images test, puis nous calculerons les mesures d'évaluation de la performance précédemment employées.

Le jeu de paramètres que nous avons gardés dans ces expériences de comparaison est listé sur la table 4.3 suivant.

TABLE 4.3 – Les valeurs des paramètres expérimentaux liés aux méthodes (KNN et SVM) utilisées pour la comparaison.

Methods	Parameters	Used Values			
		KNN_0	KNN_1	KNN_2	KNN_3
KNN	K :	1	3	1	3
	Distance :	Euclidian	Euclidian	Cityblock	Cityblock
		Used Values			
		SVM_0	SVM_1	SVM_2	SVM_3
SVM	Type of SVM :	C-SVC	ν-SVC	ν-SVC	ν-SVR
	Type of kernel function :	Radial basis function	Linear	Sigmoid	Polynomial
	Degree in kernel function :	3	4	4	1

La figure 4.14 illustre un exemple des résultats de l'extraction des toitures obtenus avec différentes combinaisons de paramètres indiqués dans le tableau 4.3 pour les deux algorithmes KPP et SVM.

L'analyse visuelle montre que KNN_0 (utilisant la distance euclidienne et k = 1) et SVM_2 (avec $\nu - SVC$ comme type de SVM, sigmoïde pour la fonction du noyau et un degré de fonction de noyau égal à 4) donnent de bons résultats de détection.

En dépit des plusieurs combinaisons des différents paramètres essayés, aucun des deux algorithmes de comparaison ne dépasse la méthode développée en terme de performance. En effet, cette approche a réussi comme nous avons vu à reconnaitre les

toitures à partir des images, c'est à dire que les résultats de l'extraction correspondent beaucoup plus à la vérité terrain ; tandis que les algorithmes KNN et SVM conduisent à de nombreux faux positives sur les zones des routes et des végétations et des faux négatifs sur quelques toitures.

Les résultats des mesures concernant l'évaluation des performances des méthodes de comparaison ont été résumés sur le tableau 4.4. Les mêmes résultats ont été repris sur la figure 4.15 sous une représentation graphique cette fois-ci. Les mesures utilisées pour évaluer la qualité de détection de cette analyse comparative sont les mêmes que celles utilisées dans l'expérience ci-dessus pour évaluer notre méthode séparément (cf. section 4.4.2.2).

Le tableau 4.4 montre les mesures quantitatives calculées sur la base des résultats obtenus pour chaque méthode. Pour KNN, nous notons que le résultat optimal en terme de qualité de détection (Q.D) est obtenu en utilisant le second jeu de paramètres KNN_1 avec une valeur de 47,5%, tandis que pour le SVM, l'indicateur de qualité de détection a atteint une valeur de 70% pour le jeu de paramètre SVM_2. Comparativement à ces méthodes, notre méthode est plus performante en tablant sur 85,6% pour l'indicateur de qualité de détection. L'indicateur du pourcentage de détection (D.P) a atteint quant à lui les valeurs 73,7% et 81,27% respectivement pour KNN et SVM contre 97,19% pour la méthode proposée.

Ces résultats sont davantage renforcés par les valeurs obtenues pour l'indicateur du facteur d'omission (M.F). Cet indicateur a atteint les valeurs 0,36 et 0,23 pour KNN (KNN_0) et SVM (SVM_2) respectivement. Le même indicateur descend à une valeur de 0,02 pour notre méthode, démontrant ainsi sa supériorité par rapport aux autres méthodes en termes de pourcentage de pixels qui ne sont pas classés comme faisant partie des toitures.

Cette amélioration de la qualité de détection, notamment par rapport à la méthode KNN est expliquée par le fait que contrairement à cette dernière, qui classe un élément par vote de majorité de l'ensemble de ses voisins, c'est à dire qu'un élément donné est affecté à la classe la plus fréquente parmi ses k plus proches voisins, la règle de décision de la méthode proposée est basée sur la valeur maximale parmi les deux moyennes arithmétiques prises sur les p et les q premières mesures de similarité de chaque région.

Les figures 4.16 et 4.17 montrent les graphes ROC tracés à la base des résultats obtenus. Ces graphes indiquent une meilleure qualité de la méthode proposée relativement aux variantes des classifieurs KNN et SVM.

Enfin, nous terminons par noter que la méthode nécessite en moyenne 5 secondes pour extraire les objets d'intérêt (toitures des bâtiments dans l'application abordée)

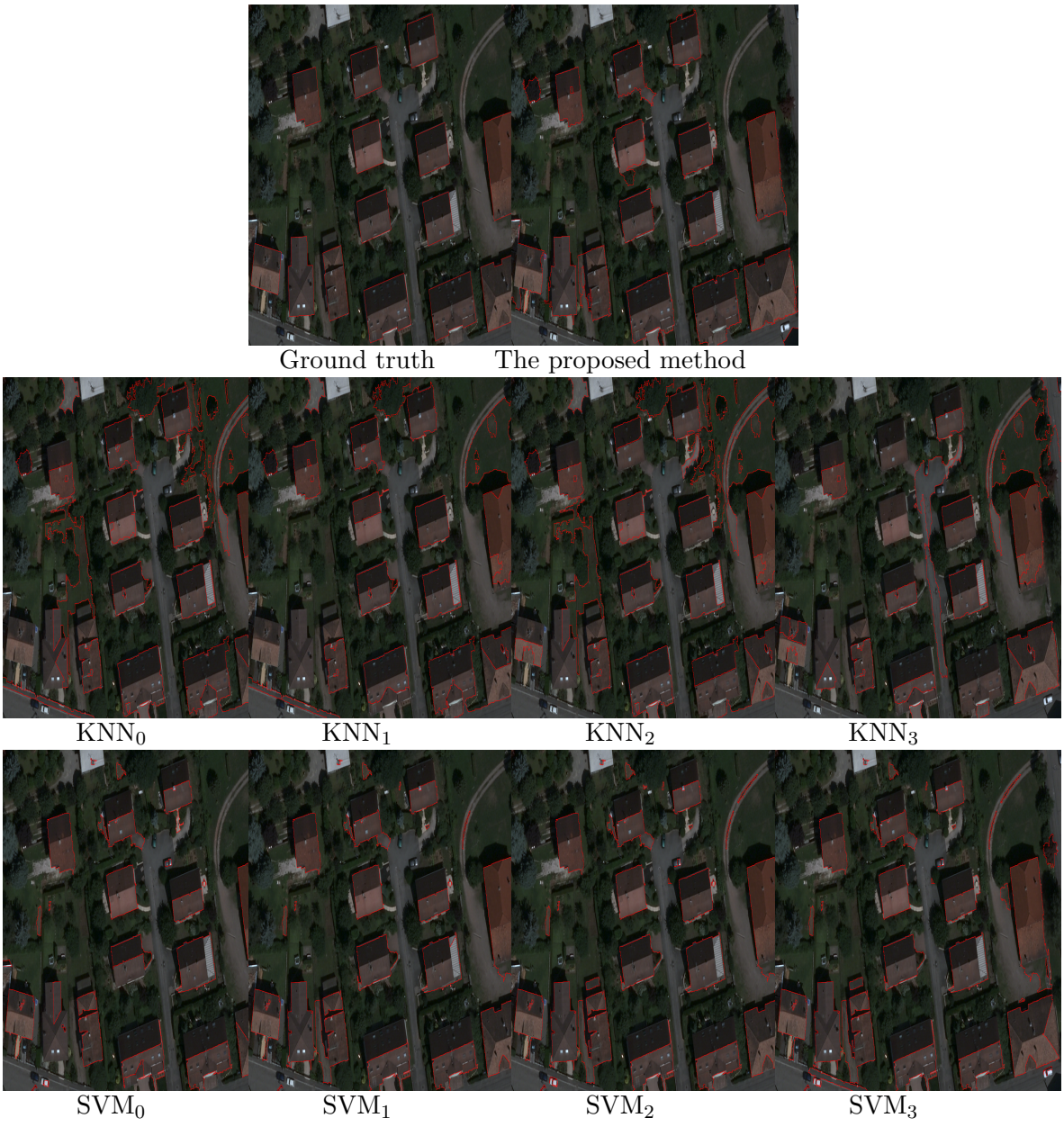


FIGURE 4.14 – Comparaison entre KNN, SVM et la méthode proposée. Première ligne : image de la vérité terrain et l'image des résultats de la détection des toits par la méthode proposée ; deuxième ligne : résultats de la détection par la méthode KNN selon différents paramètres (voir le tableau 4.3) ; dernière ligne : résultats de la détection par la méthode SVM suivant différents paramètres (voir le tableau 4.3).

à partir des images de résolutions de 1500 par 1000 pixels. La machine sur laquelle l'algorithme a été exécuté a un processeur de 2,75 MHz (CPU) et une mémoire de 3 GO (RAM). Notons aussi que cette complexité dépend aussi du nombre de régions obtenues par la segmentation des images test.

TABLE 4.4 – Évaluation de la qualité de détection obtenue pour les méthodes KNN et SVM (plusieurs variantes selon différents paramètres) et la méthode proposée.

(%)	KNN				SVM				Our method
	KNN_0	KNN_1	KNN_2	KNN_3	SVM_0	SVM_1	SVM_2	SVM_3	
Q.P	42.34	47.48	41.63	29.96	58.79	68.53	70.11	66.33	85.59
D.P	73.69	67.71	69.58	30.86	64.12	78.51	81.27	78.54	97.19
B.F	1.00	0.63	0.97	0.09	0.14	0.19	0.20	0.23	0.14
M.F	0.36	0.48	0.44	2.24	0.56	0.27	0.23	0.27	0.02

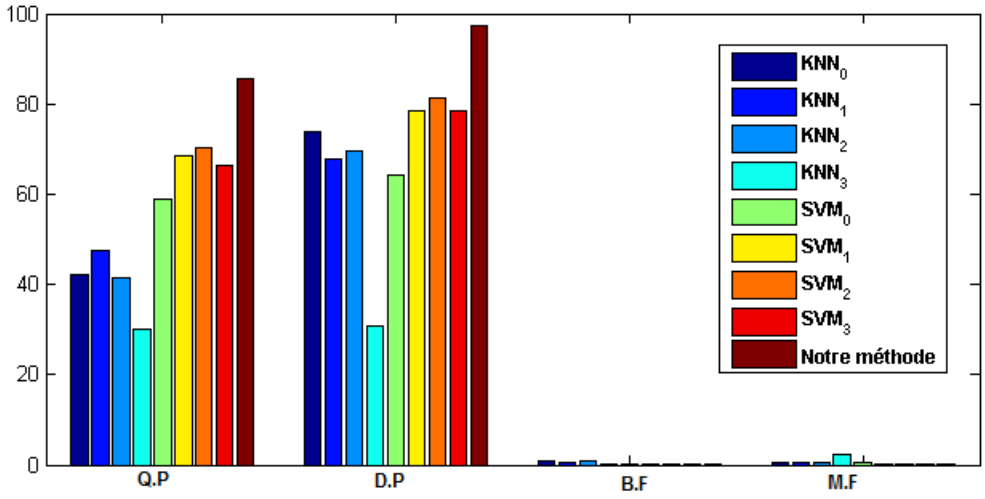


FIGURE 4.15 – Représentation graphique de la table 4.4.

4.5 Conclusion

Ce chapitre a été une occasion d'appliquer les méthodes développées dans ce mémoire pour la classification de données multidimensionnelles. Cette application a concerné des données multidimensionnelles provenant de deux domaines qui deviennent de plus

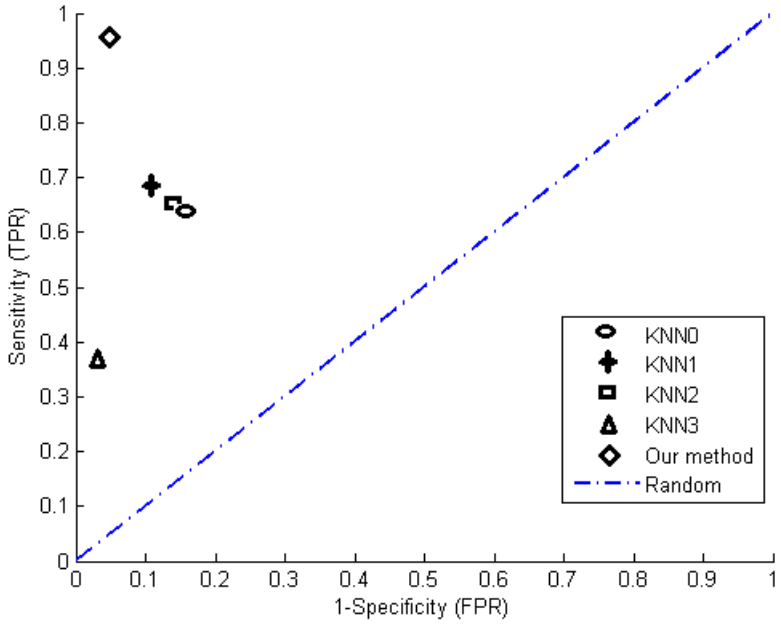


FIGURE 4.16 – Le graphe ROC comparant les variantes de la méthode KNN et la méthode proposée.

en plus indispensables pour la construction des systèmes intelligents, en particulier la vision artificielle. Il s'agit de la segmentation d'images couleur texturées et la reconnaissance d'objets.

Avant d'avoir présenté chaque application et les expériences qui lui ont été associées, il a fallu introduire et préciser les différents outils et notions qui en sont liés.

Les conclusions que nous pouvons retenir de ce qui a été présenté dans ce chapitre sont les suivantes :

- La méthode proposée pour la classification de données multidimensionnelles à partir des prototypes a montré son efficacité pour la segmentation des images de textures couleur.

  En effet, une évaluation dans cette première application a été effectuée en utilisant différents espaces colorimétriques RGB, L*a*b et IHLS. Les résultats ont montré une efficacité supérieure de la méthode proposée en travaillant dans les deux derniers espaces en termes des taux d'erreur comparativement à l'espace RGB.

  Une autre évaluation a été donnée en comparant les résultats de la segmentation des images texturées obtenus par notre méthode d'une part et les méthodes

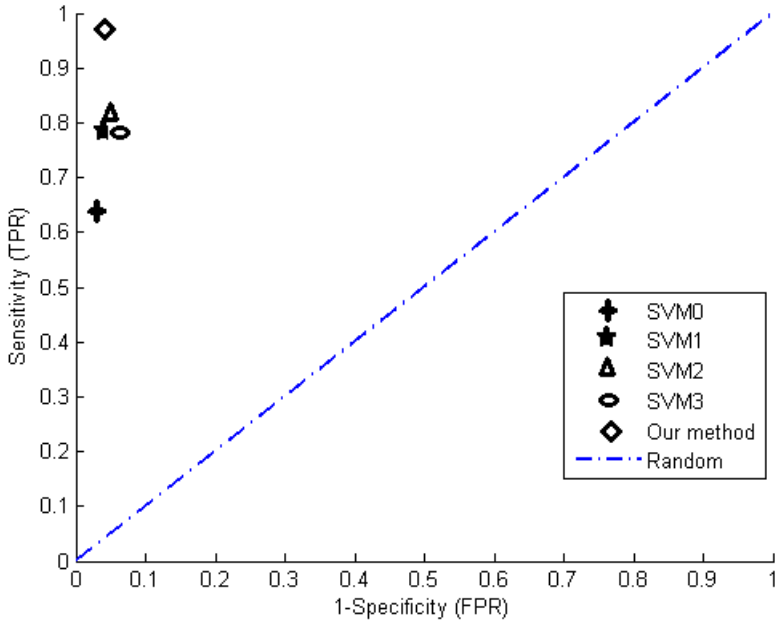


FIGURE 4.17 – Le graphe ROC comparant les variantes de la méthode SVM et la méthode proposée.

classiques de KNN et de Bayes d'autre part. Les taux d'erreur marqués par notre méthodes sur l'ensemble des images testées ont été les meilleures face aux résultats obtenus par les deux autres méthodes.

- La méthode développée pour la reconnaissance d'objets d'intérêt dans les images à haute résolution a été appliquée avec succès pour la détection de toitures des bâtiments à partir des images aériennes.

  La méthode présentée au cours de cette deuxième application a tenté de remédier à l'inconvénient de l'interactivité dont souffrent pas mal de méthodes comme nous l'avons souligné dans le paragraphe sur l'état de l'art sur les approches et méthodes relevant du champ de la reconnaissance d'objets.

  L'idée principale était d'isoler les attributs visuels des objet d'intérêt dans une base de connaissance et d'adopter une règle de décision efficace dans une phase de classification supervisée.

  Les résultats obtenu en appliquant la méthode pour la détection de toitures des bâtiments à partir des orthophotoplans nous ont démontré la robustesse de la méthode, surtout en terme de la détection simultanée de plusieurs instances de

l'objet d'intérêt. Une autre difficulté que la méthode a surmonté avec efficacité a été la diversité visuelle, colorimétrique et géométrique. Les différentes mesures de qualité utilisées (résumées sur les courbes ROC) et les comparaisons effectuées avec les variantes des méthodes KNN et SVM ont confirmé cette efficacité.

Certainement, l'efficacité des méthodes proposées dans ce mémoire n'est pas parfaite. Il reste bel et bien des points à améliorer, c'est ce que nous aborderons dans la section des perspectives qui va suivre.

Conclusions et Perspectives

La problématique de la classification automatique de données de grandes dimensions est un domaine intermédiaire entre l'intelligence artificielle d'une part et l'analyse de données de l'autre part. Comme nous l'avons vu au cours du premier chapitre consacré à l'état de l'art, les approches et méthodes de la classification proposées dans la littérature et pendant des décennies sont d'un nombre énorme. Ce qui témoigne de l'importance accordée à cette question. Parmi les approches reportées, celle selon laquelle la classification non supervisée peut se ramener à la détection des modes avait attiré notre attention et nous nous en sommes énormément inspirés pour développer notre nouvelle méthode.

De ce fait, la première méthode développée au cours des travaux de recherche effectués et présentés dans ce mémoire, avait pour objectif de détecter ces modes. La méthode a été élaborée en utilisant une approche stochastique basée sur les processus ponctuels marqués. Ces derniers sont des outils très puissants pour la modélisation, l'interprétation et la prédiction des comportements des phénomènes aléatoires. Le chapitre 2 a été réservé à ces outils qui disposent d'un cadre théorique solide. La faculté d'incorporer des informations supplémentaires, sous forme de *marques*, dont dispose ces processus leur a permis d'être appliqués dans plusieurs domaines. Les applications récentes dans le domaine de l'analyse d'images notamment la reconnaissance d'objets telles que l'extraction de routes à partir des images satellitaires, l'extraction de bâtiments, la détection des puits de pétroles nous ont également inspiré au cours de ce travail d'élaboration d'une méthode de classification non supervisée.

La construction de notre modèle stochastique pour la classification non supervisée en utilisant les processus ponctuels marqués, et que nous avons présenté au travers du chapitre 3, a pu bénéficier de leur avantage qui réside dans leur construction par une fonction de densité de probabilité par rapport à un processus de référence, le plus souvent de Poisson. Cette densité, qui peut s'exprimer grâce au théorème de Hammersly Clifford sous une forme énergétique, comporte deux termes : un terme d'interaction et un autre d'attache aux données ; le premier terme nous a servi pour incorporer les

relations entre les objets du processus définissant notre modèle tandis que le deuxième terme nous a permis de contrôler la localisation de ces objets sur les zones de l'espace qui contiennent les modes à détecter.

L'échantillonnage à partir du modèle élaboré a été effectué grâce aux méthodes de Monte Carlo par chaînes de Markov, en particulier l'algorithme de Metropolis Hasting Green. Cet algorithme a la propriété de pouvoir simuler des lois de probabilité dans des espaces à dimensions variables, ce qui nous a bien convenu vu que les réalisations du modèle construit sont des configurations qui changent de dimension d'un état de la chaînes de Markov à un autre. La difficulté majeure dans ces simulations était la lenteur dont se déroulait l'exécution du programme. Les résultats expérimentaux, présentés dans la section 3.2.5, ont été encourageants.

Néanmoins, La modélisation que nous avons adoptée pour approcher cette problématique a fait qu'une autre méthode en relation soit aussi développée. En effet, la modélisation présentée s'est reposée comme nous avons évoqué sur l'idée de la classification automatique de données par la détection des modes de la fonction de densité de probabilité sous jacente à la distribution de données brutes. Ceci veut dire que le problème principal va être divisé en deux sous problèmes : le premier concerne la détection des modes et le deuxième concerne la classification supervisée ou bien la classification à partir de prototypes une fois les modes détectés. Une telle méthode a été l'objet de la section 3.3.

Cette méthode a été conçue de sorte que les informations contextuelles soient prises en considération lors de l'affectation d'une nouvelle observation à une classe donnée. Le cadre théorique que nous avons adopté a été les Champs Aléatoires de Markov avec les concepts de système de voisinage et de cliques. Nous avons encore pu passer, grâce à un théorème d'équivalence, à une expression énergétique de Gibbs à travers laquelle nous avons défini les interactions entre les observations prototypes et les observations non prototypes. En outre, nous avons adopté une démarche lors de la sélection de l'observation non prototype à traiter et qui consiste d'abord à réordonner l'ensemble des observations non prototypes dans un ordre croissant suivant leur proximité de l'ensemble des prototypes et ensuite commencer par l'observation non prototype la plus proche. Les résultats des expériences effectuées ont été satisfaisants comparativement aux méthodes KNN et Bayes.

Dans la partie applications, le travail dans le domaine de l'analyse d'images nous a amené à aborder deux sujets d'une grande importance dans le domaine de la vision assistée par ordinateur ou dans le domaine du développement des systèmes de vision artificielle, il s'agit de la segmentation d'images couleur texturées et la reconnaissance

d'objets. L'objectif de la première application était de segmenter une image couleur texturée par la méthode de classification à partir des prototypes développée. Dans ce but, nous avons extrait une imagette (qui va représenter un prototype) de chaque image et nous avons quantifié ses descripteurs colorimétriques en terme d'erreur de prédiction linéaire bidimensionnelle. Cette dernière a été calculée par la suite pour l'ensemble d'images tests. L'application de la méthode de classification à partir des prototypes sur ces données et dans différents espaces colorimétriques a montré de bons résultats.

La deuxième application de classification à partir des prototypes a été menée dans la partie 4.4 et elle a porté cette fois-ci sur la reconnaissance d'objets dans les images aériennes. Dans cette partie, nous avons proposé une méthode qui se base d'une part sur la construction de deux bases de connaissances contenant les informations visuelles en terme d'histogrammes couleur RGB, et d'autre part sur la définition d'une règle de décision basée sur le coefficient de Bathacharyya comme mesure de similarité entre les régions issues d'une segmentation de l'image analysée. Les performances et la qualité de détection de la méthode ont été démontrées par une série d'expériences sur des images test ainsi que sur une application particulière dédiée à la détection de toitures de bâtiments à partir des orthophotoplans.

Les travaux présentés au cours de cette thèse restent ouverts sur plusieurs perspectives. Dans ce sens et au niveau de la méthode de détection des modes par les processus objets, de nombreux points peuvent être améliorés. Ainsi, serait-il judicieux d'abord de faire une étude sur les éventuelles possibilités de l'estimation des paramètres du modèle, à l'instar de certaines méthodes déjà proposées dans la littérature. Également, une amélioration pourrait être introduite en définissant d'autres noyaux de proposition dans la procédure de simulation par les méthodes de Monte Carlo par Chaînes de Markov, soit dans le but d'accélérer davantage la vitesse de convergence de la simulation (qui nécessite aussi une optimisation au niveau de la programmation de la méthode), soit pour aider les objets (les hypersphères) du processus à bien se localiser sur les régions modales de l'espace. Aussi, serait-il intéressant d'étudier la question de l'initialisation du processus par une configuration aléatoire et l'effet de ce choix initial. Cependant, l'amélioration importante serait d'étudier un nouveau modèle de l'énergie d'attache aux données qui prenne en considération les interactions entre les observations comprises à l'intérieure de chaque objet. Cela pourrait contribuer de façon significative à la localisation des objets sur les régions modales.

Quant à la méthode proposée pour faire de la reconnaissance d'objets, il serait intéressant dans le souci d'amélioration, tout d'abord, étudier et évaluer les différents espaces de couleur et/ou invariants colorimétriques. En effet, les caractéristiques de

l'histogramme de couleur sont calculées en utilisant l'espace colorimétrique des primaires RVB, cependant les images orthophotoplan contiennent une certaine hétérogénéité en termes de luminosité, de contraste, des ombres, etc, ce qui constitue une source vive pour les fausses détections. En outre, cette méthode proposée permet une intégration souple de différents descripteurs. Nous proposons alors d'étudier l'effet d'autres attributs des régions, descripteurs, sur la qualité des résultats. On peut citer dans ce contexte les motifs binaires locaux (Local Binary Pattern, LBP) qui constituent un modèle structurel puissant d'analyse de texture. Quant aux paramètres de la méthode, nous envisageons d'estimer analytiquement les valeurs des deux paramètres p et q impliqués dans le calcul de la similarité. Enfin, une application de la méthode sur d'autres types d'images (médicales par exemple) est aussi envisageable pour évaluer la généricité de la méthode proposée.

Bibliographie

[1] H. Abdi and D. Valentin. *La place de l'image dans la cognition*, chapter De la reconnaissance des objets à la reconnaissance des visages, pages 13–41. ÉPURE-Éditions et Presses universitaires de Reims, 1997. 122

[2] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. In *Proceedings of the ACM-SIGMOD Conference on the Management of Data*, volume 27, pages 94–105, 1998. 28

[3] M. A. Aizerman, E. A. Braverman, and L. Rozonoer. Theoretical foundations of the potential function method in pattern recognition learning. In *Automation and Remote Control*, volume 25, pages 821–837, 1964. 18

[4] O. Alata, S. Burg, and A. Dupas. Grouping/degrouping point process, a point process driven by geometrical and topological properties of a partition in regions. *Computer Vision and Image Understanding*, 115, 2011. 72

[5] C. Andrieu, N. D. Freitas, A. Doucet, and M. I. Jordan. An introduction to mcmc for machine learning. *Machine Learning*, 50, Issue 1-2 :5–43, 2003. xii, 58

[6] F. Baccelli, M. Klein, M. Lebourges, and S. Zuyev. Stochastic geometry and architecture of communication networks. *Telecommunication Systems*, 7, Issue 1-3 :209–227, 1997. 68

[7] A. Baddeley. *Stochastic Geometry : Lecture Notes in Mathematics, Volume 1892*, chapter Spatial Point Processes and their Applications, pages 1–73. Springer Berlin Heidelberg, 2007. 68

[8] C. Beecks, M.S. Uysal, and T. Seidl. A comparative study of similarity measures

for content-based multimedia retrieval. In *IEEE International Conference on Multimedia and Expo (ICME), Suntec City*, pages 1552 –1557. IEEE, 2010. 132

[9] Y. Bengio, I. J. Goodfellow, and A. Courville. Deep learning. Book in preparation for MIT Press, 2015. 48

[10] J. Besag. Spatial interaction and the statistical analysis of lattice system. *Journal of the Royal Statistical Society*, B-36 :192–236, 1974. 99, 100, 104

[11] J. C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Kluwer Academic Publishers, Norwell, MA, USA, 1981. 16, 17

[12] I. Biederman. Recognition by components : A theory of human image understanding. *Psychological review*, 94 :115–145, 1987. 122

[13] H. Bischof, W. Schneider, and A. J. Pinz. Multispectral classification of landsat-images using neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, 30 (3) :482–490, 1992. 122

[14] M. Blatt, S. Wiseman, and E. Domany. Superparamagnetic clustering of data. *Physical Review Letters*, 76, Issue 18 :3251–3254, 1996. 34

[15] E. Bonabeau, M. Dorigo, and G. Theraulaz. *Swarm intelligence : from natural to artificial systems*. Oxford University Press, Inc. New York, NY, USA, 1999. 29

[16] C. Botte-Lecocq, K. Hammouche, A. Moussa, J.-G. Postaire, A. Sbihi, and A. Touzani. *Scene Reconstruction, Pose Estimation and Tracking*, chapter Image Processing Techniques for Unsupervised Pattern Classification, pages 467–488. I-Tech Education and Publishing, 2007. 38

[17] C. Bouveyron and C. Brunet-Saumard. Model based clustering of high-dimensional data : A review. *Computational Statistics and Data Analysis*, 71 :52–78, 2014. 26

[18] Y. Boykov and M. P. Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in n-d images. In *IEEE International Conference on Computer Vision, (ICCV)*, volume 1, pages 105–112, 2001. 122

[19] L. Breiman. Bagging predictors. *Machine Learning*, 24, Issue 2 :123–140, 1996. 46

[20] L. Breiman. Heuristic of instability and stabilization in model selection. *The Annals of Statistics*, 24, Issue 6 :2350–2383, 1996. 46

[21] L. Breiman. Random forests. *Machine Learning*, 45, Issue 1 :5–32, 2001. 46

[22] L. Breiman, J. Friedman, C. J. Stone, and R.A. Olshen. *Classification and Regression Trees*. 1984. 45

[23] S. Brooks and G. Roberts. Convergence assessment techniques for markov chain monte carlo. *Statistics and Computing*, 8 :319–335, 1998. 67

[24] S.P. Brooks and A. Gelman. Alternative methods for monitoring convergence of iterative simulations. *Journal of Computational and Graphical Statistics*, 7 :434–455, 1998. 66

[25] G. Carpenter and S. Grossberg. Art3 : Hierarchical search using chemical transmitters in self-organizing pattern recognition architectures. *Neural Networks*, 3 :129–152, 1990. 32

[26] M. Chen, X. Hou, Y.and Zeng, and Y. Lu. A multi-label interactive image segmentation method based on region merging. *IEEE International Conference on Mechatronic Science, Electric Engineering and Computer*, pages 2336–2340, 2011. 134

[27] C. Chih-Chung and L. Chih-Jen. Libsvm : a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology, Software available at http ://www.csie.ntu.edu.tw/ cjlin/libsvm*, 2-27, 2011. 144

[28] Y. C. Chin and A. J. Baddeley. On connected component markov point processes. *Advances in Applied Probability*, 31, Issue 2 :279–282, 1999. 77

[29] F.R.K. Chung. *Spectral Graph Theory*. CBMS Regional Conference Series in Mathematics, vol. 92, 1997. 21, 22

[30] D. Comaniciu and P. Meer. Mean shift : A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 603–619, 2002. 38

[31] C. Cortes and V. Vapnik. Support vector networks. *Machine Learning*, 20, Issue 3 :273–297, 1995. 18, 48

[32] M. K. Cowles and B. P. Carlin. Markov chain monte carlo convergence diagnostics : A comparative review. *Journal of the American Statistical Association*, 91, Issue 434 :883–904, 1996. 65

[33] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39, Issue 1 :1–38, 1977. 25, 26

[34] J. L. Deneubourg, S. Goss, N. Franks, A. Sendova-Franks, C. Detrain, and L. Chrétien. The dynamics of collective sorting robot-like ants and ant-like robots. In *Proceedings of the first international conference on simulation of adaptive behavior on From animals to animats*, pages 356–363. MIT Press Cambridge, MA, USA, 1991. 29

[35] Y. Deng and B. S. Manjunath. Unsupervised segmentation of color-texture regions in images and video. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 23 :800–810, 2000. 122

[36] H. Derin and H. Elliott. Modelling and segmentation of noisy and textured images using gibbs random field. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9, Issue 1 :39–55, 1987. 99, 100, 122

[37] X. Descombes. *Méthodes stochastiques en analyse d'image : des champs de Markov aux processus ponctuels marqués*. Habilitation à Diriger les Recherches, 2004. 55

[38] X. Descombes, F. Kruggel, G. Wollny, and H. J. Gertz. An object based approach for detecting smallbrain lesions : application to virchow-robin spaces. *IEEE Transactions on Medical Imaging*, 23, Issue 2 :246–255, 2004. 69, 86

[39] I. S. Dhillon, Y. Guan, and B. Kulis. A unified view of kernel k-means, spectral clustering and graph partitioning. Technical report, UTCS Technical Report TR-04-25, 2004. 22

[40] E. Diday. Une nouvelle méthode en classification automatique et reconnaissance des formes la méthode des nuées dynamiques. *Revue de Statistique Appliquée*, 19, Issue 2 :19–33, 1971. 16

[41] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. A Wiley Interscience Publication. Wiley, 1973. 23

[42] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. 2nd edition edition, 2001. 97, 105

[43] S. El-Adlouni, A. C. Favre, and B. Bob Comparison of methodologies to assess the convergence of markov chain monte carlo methods. *Computational Statistics and Data Analysis*, 50 :2685–2701, 2006. 64

[44] M. Ester, H. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. *In the 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96)*, pages 226–231, 1996. 35

[45] Tom Fawcett. An introduction to roc analysis. *Pattern Recognition Letters*, 27 :861–874, 2006. 142

[46] M. Filipponea, F. Camastra, F. Masulli, and S. Rovetta. A survey of kernel and spectral methods for clustering. *Pattern Recognition*, 41 :176–190, 2008. 14, 22

[47] E. Fix and J.L. Hodges. An important contribution to nonparametric discriminant analysis and density estimation. *International Statistical Review*, 57, Issue 3 :233–247, 1951. 23

[48] C. Florent, X. Descombes, and J. Zerubia. Parameter estimation for marked point processes. application to object extraction from remote sensing images. In *XXIIe colloque GRETSI (traitement du signal et des images), Dijon (FRA)*. GRETSI, Groupe dEtudes du Traitement du Signal et des Images, 8-11 septembre 2009. 80

[49] E. W. Forgy. Cluster analysis of multivariate data : efficiency versus interpretability of classifications. *Biometrics*, 21 :768–769, 1965. 14

[50] C. Fraley and A.E. Raftery. Model-based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association*, 97, Issue 458, 2002. 24

[51] Y. Freund. Boosting a weak learning algorithm by majority. *Information and Computation*, 121 (2) :256–285, 1995. 46

[52] B. J. Frey and D. Dueck. Clustering by passing messages between data points. *Science*, 315 :972–976, 2007. 34

[53] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, second edition edition, 1990. 133

[54] K. Fukunaga and L.-D. Hosteler. The estimation of the gradient of a density function with applications in pattern recognition. *IEEE Transactions on Information Theory*, 21, Issue 1 :32–40, 1975. 39

[55] A. Gelman and D. B. Rubin. Inference from iterative simulation using multiple sequences. *Statistical Science*, 7, Issue 4 :457–472, 1992. 66

[56] S. Geman and D. Geman. Stochastic relaxation, gibbs distribution and the bayesian restoration of image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6 :721–741, 1984. 58, 63, 97, 100

[57] H. O. Georgii, O. Haggstrom, and C. Maes. The random geometry of equilibrium phases. *Phase transitions and critical phenomena*, 18 :1–142, 2001. 68

[58] A. Gersho and R. M. Gray. *Vector Quantization and Signal Compression*. The Springer International Series in Engineering and Computer Science, 1992. 15

[59] John Geweke. Evaluating the accuracy of sampling-based approaches to the calculation of posterior moments. In *IN BAYESIAN STATISTICS*, pages 169–193. University Press, 1992. 67

[60] C.J. Geyer and J. Moller. Simulation and likelihood inference for spatial point process. *Scandinavian Journal of Statistics*, B, Issue 21 :359–373, 1994. 58

[61] M. Girolami. Mercer kernel based clustering in feature space. *IEEE Transactions on Neural Networks*, 13, Issue 3 :780–784, 2002. 20

[62] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Publishing Co., Inc., Red-wood City, CA, 1989. 33

[63] A. A. Goshtasby. *Image Registration*, chapter Similarity and Dissimilarity Measures, pages 7–68. Advances in Computer Vision and Pattern Recognition, 2012. 133

[64] P. Green. Reversible jump markov chain monte carlo computation and bayesian model determination. *Biometrika*, 57 :97–109, 1995. 58

[65] P. Guillaume. *Etude du couvert forestier par processus ponctuels marqués*. PhD thesis, Mathematics. Ecole Centrale Paris, 2006. 70, 86

[66] N. Halla and C. Brenner. Extraction of building and trees in urban environment. *ISPRS Journal of Photogrammetry and Remote Sensing*, 54 :130–137, 1999. 122

[67] W. K. Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57 :97–109, 1970. 58

[68] J. Hertz, A. Krogh, and R. D. Palmer. *Introduction to the theory of neural computation*. 1991. 30

[69] A. Hinneburg and D. Keim. An efficient approach to clustering in large multimedia databases with noise. In *Proceeding of The International Conference on Knowledge Discovery and Data Mining (KDD98)*, pages 58–65, 1998. 35

[70] J. H. Holland. *Adaption in Natural and Artificial Systems*. 1975. 33

[71] J. Huang, S. Kuamr, M. Mitra, W.-J. Zhu, and R. Zabih. Image indexing using colour correlogram. In *Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition*, pages 762–765, 1997. 117

[72] E. Hunt, J. Marin, and P . Stone. *Experiments in Induction*. 1966. 46

[73] D Ilea and E. P. F Whelan. An adaptive unsupervised segmentation algorithm based on color-texture coherence. *IEEE Transactions on Image Processing*, 17, Issue 10 :1926–1939, 2008. 122, 125

[74] M. Imberty and X. Descombes. Simulation de processus objets : Etude de faisabilité pour une application à la segmentation dimage. Technical report, Institut National de Recherche en Informatique et en Automatique, 2000. 61

[75] A. K. Jain and J. Mao. *Computational Intelligence : Imitating Life*, chapter Neural networks and pattern recognition, pages 194–212. J. M. Zurada and R. J. Marks, and C. J. Robinson, Eds., 1994. 31

[76] A. K. Jain and J. Mao. Artificial neural networks : A tutorial. *IEEE Computer*, 29, Issue 3 :31–44, 1996. 31

[77] A.K. Jain, M.N. Murty, and P.J. Flynn. Data clustering : A review. *ACM Computing Surveys*, Vol. 31, No. 3, 1999. 8, 11, 33

[78] Anil K. Jain and Richard C. Dubes. *Algorithms for clustering data*. Prentice-Hall, Inc. Upper Saddle River, NJ, USA, 1988. 25

[79] Mike James. *Classification algorithms*. Wiley-Interscience New York, NY, USA„ 1985. 23

[80] T. Kailath. The divergence and bhattacharyya distance measures in signal selection. *IEEE Transactions on Communications Technology*, 15 :52–60, 1967. 133

[81] N. Kannan and S. Vishveshwara. Identification of side-chain clusters in protein structures by a graph spectral method. *Journal of Molecular Biology*, 292 :441–464, 1999. 22

[82] L. Kaufman and P. J. Rousseeuw. *Finding Groups in Data : An Introduction to Cluster Analysis*. 2005. 11

[83] D. G. Kendall. A survey of the statistical theory of shape. *Statistical Science*, 4, Issue 2 :87–99, 1989. 68

[84] B. Keresztes. *Segmentation de Zones d'Intérêt par Processus Objets : Application à la Détection des Réseaux de Failles Sismiques*. PhD thesis, Université Bourdaux, Ecole Doctorale des Sciences Physiques et de l'Ingénieur and Université Technique de Cluj-Napoca (ROUMANIE), 2009. 55, 70

[85] B. King. Stepwise clustering procedures. *Journal of the American Statistical Association*, 69 :86–101, 1967. 12

[86] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science, New Series*, 220, No. 4598 :671–680, 1983. 34, 58

[87] R. W. Klein and R. C. Dubes. Experiments in projection and clustering by simulated annealing. *Pattern Recognition*, 22 (2) :213–220, 1989. 34

[88] S. Knerr, L. Personnaz, and G. Dreyfus. *Neurocomputing, Algorithms, Architectures and Applications*, chapter Single-layer learning revisited : a stepwise procedure for building and training a neural network, pages 41–50. 1990. 122

[89] T. Kohonen. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43, Issue 1 :59–69, 1982. 32

[90] T. Kohonen. *Self-Organizing Maps*, volume 30 of *Springer Series in Information Sciences*. Springer-Verlag Berlin Heidelberg, 2001. 32

[91] S. B. Kotsiantis, I. D. Zaharakis, and P. E. Pintelas. Machine learning : a review of classification and combining techniques. *Artificial Intelligence Review*, 26 :159–190, 2006. 45

[92] R. Krishnapuram and J.M. Keller. A possibilistic approach to clustering. *IEEE Transactions on Fuzzy Systems*, 1, Issue 2 :98–110, 1993. 18

[93] C. Lacoste, X. Descombes, and J. Zerubia. Point processes for unsupervised line network extraction in remote sensing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27 :1568–1579, 2005. 70

[94] X. Lafarge, F.and Descombes, J. Zerubia, and M. Pierrot-Deseilligny. Automatic building extraction from dems using an object approach and application to the 3d-city modeling. *ISPRS Journal of Photogrammetry and Remote Sensing*, 63 :365–381, 2008. 122

[95] A. Levinshtein, A. Stere, K. N. Kutulakos, and S. J.and Siddiqi K. Fleet, D. J.and Dickinson. Turbo pixels : Fast super pixels using geometric flows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31 :2290–2297, 2009. 122

[96] Yu Li and Jonathan Li. Oil spill detection from sar intensity imagery using a marked point process. *Remote Sensing of Environment*, 114 :1590–1601, 2010. 71

[97] W. K. Liao, Y. Liu, and A. Choudhary. A grid-based clustering algorithm using adaptive mesh refinement. In *7th Workshop on Mining Scientific and Engineering Datasets*, 2004. 28

[98] N Littlestone and M Warmuth. The weighted majority algorithm. *Information and Computation*, 108, Issue 2 :212–261, 1994. 47

[99] S. P. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28, Issue 2 :129–137, 1982. 15

[100] E. Lumer and B. Faieta. Diversity and adaptation in populations of clustering ants. In *Proceedings of the Third International Conference on Simulation of Adaptive Behavior : From Animals to Animats 3*, volume 1, page 501508. MIT Press, Cambridge., 1994. 30

[101] B. Luo, R.C. Wilson, and E.R. Hancock. Spectral embedding of graphs. *Pattern Recognition*, 36 :2213–2230, 2003. 22

[102] U.V. Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4) :395–416, 2007. 22

[103] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297. University of California Press, 1967. 15

[104] A. C. Berg T. L. Berg J. Malik. Shape matching and object recognition using low distortion correspondences. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1*, pages 26–33. IEEE Computer Society Washington, DC, USA, 2005. 122

[105] J. Manyika, M. Chui, J. Bughin, B. Brown, R. Dobbs, C. Roxburgh, and A-H. Byers. Big data : The next frontier for innovation, competition, and productivity. Technical report, McKinsey Global Institute, 2011. 2

[106] V. J. Martinez and E. Saar. *Statistics of The Galaxy Distribution*. Chapman and Hall, 2001. 68

[107] O. Mathias. *Processus ponctuels marqués pour l'extraction automatique de caricatures de bâtiments à partir de modèles numériques d'elévation*. PhD thesis, Université de Nice-Sophia Antipolis, 2004. 71

[108] P. McCullagh and J. Mller. The permanental process. *Advances in Applied Probability*, 38, Issue 4 :873–888, 2006. 68

[109] G. J. McLachlan. *Discriminant Analysis and Statistical Pattern Recognition*. Wiley, New York, 1992. 23

[110] G.J. McLachlan and T. Krishnan. *The EM Algorithm and Extensions*. 1997. 27

[111] G.J. McLachlan and D. Peel. *Finite Mixture Models*. 2000. 24

[112] J. Mercer. Functions of positive and negative type and their connection with the theory of integral equations. In *Proceedings of the Royal Society of London*, volume 209, pages 415–446, 1909. 18

[113] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. Teller, and H. Teller. Equations of state calculations by fast computing machines. *Journal of Chemical Physics*, 21 :1087–1091, 1953. 58

[114] N. Metropolis and S. Ulam. The monte carlo method. *Journal of the American Statistical Association*, 44 :335–341, 1949. 58

[115] A. Moussa, A. Sbihi, and J. G. Postaire. A markov random field model for mode detection in cluster analysis. *Pattern Recognition Letters*, 29 :1197–1207, 2008. 39, 97, 108

[116] K. R. Muller, S. Mika, G. Ratsch, K. Tsuda, and B. Scholkopf. An introduction to kernel-based learning algorithms. *IEEE Transactions on Neural Networks*, 12, Issue 2 :181–202, 2001. 19

[117] M.C.V. Nascimento and A.C.P.L.F. de Carvalho. Spectral methods for graph clustering-a survey. *European Journal of Operational Research*, 211 :221–231, 2011. 22

[118] A.Y. Ng, M.I. Jordan, and Y. Weiss. On spectral clustering : analysis and an algorithm. *Advances in Neural Information Processing Systems*, 14 :849856, 2001. 22

[119] L. Ninga, J.and Zhanga and D. Z. C. Wub. Interactive image segmentation by maximal similarity based region merging. *Pattern Recognition*, 43, Issue 2 :445–456, 2010. 122, 134

[120] R. Nock and F. Nielsen. Statistical region merging. *The IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26, Issue 11 :1452–1458, 2004. 122, 129

[121] E. Parzen. On estimation of a probability density function and mode. *The Annals of Mathematical Statistics*, 33, Issue 3 :1065–1076, 1962. 39

[122] G. Paschos. Perceptually uniform color spaces for color texture analysis : An empirical evaluation. *IEEE Transactions on Image Processing*, 10, Issue 6 :932–937, 2001. 122, 123

[123] J.-G. Postaire, R.D. Zhang, and C. Botte Lecocq. Cluster analysis by binary morphology. *IEEE Transactions on Pattern Analysis and Machin Intelligence*, 15, Issue 2 :170180, 1993. 39

[124] J. G. Propp and D. B. Wilson. Exact sampling with coupled markov chains and applications to statistical mechanics. *Random Structures and Algorithms*, 9, Issue 1-2 :223–252, 1996. 73

[125] I.-U.-H Qazi, O Alata, A. Burie, J.-C.and Moussa, and C. Fernandez-Maloigne. Choice of a pertinent color space for color texture characterization using parametric spectral analysis. *Pattern Recognition*, 44 :16–31, 2011. 124

[126] I.-U. H. Qazi, O. Alata, J-C. Burie, M. Abadi, A. Moussa, and C. Fernandez-Maloigne. Parametric models of linear prediction error distribution for color texture and satellite image segmentation. *Computer Vision and Image Understanding*, 115, Issue 8 :1245–1262, 2011. 122, 123

[127] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1, Issue 1 :81–106, 1986. 46

[128] J. R. Quinlan. C4.5 : Programs for machine learning. *Machine Learning*, 16, Issue 3 :235–240, 1993. 46

[129] A. E. Raftery and S. Lewis. How many iterations in the gibbs sampler ? In *Bayesian Statistics 4*, pages 763–773. Oxford University Press, 1992. 64

[130] B. D. Ripley and F. P. Kelly. Markov point processes. *Journal of the London Mathematical Society*, s2-15, Issue 1 :188–192, 1977. 56

[131] F. Rosenblatt. The perceptron : a perceiving and recognizing automaton. Technical report, Cornell Aeronautical Laboratory, 1957. 47

[132] C. Rother, V. Kolmogorov, and A. Blake. Grabcut -interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics*, 23, Issue 3 :309–314, 2004. 122

[133] M. Saerens, F. Fouss, L. Yen, and P. Dupont. The principal components analysis of a graph, and its relationships to spectral clustering. *Machine Learning : ECML 2004, Lecture Notes in Computer Science*, 3201 :371–383, 2004. 22

[134] K. Sahlin. *Estimating convergence of Markov chain Monte Carlo simulations.* Mathematical Statistics , Stockholm University, 2011. xii, 68

[135] S. Saitoh. *Theory of Reproducing Kernels and its Applications.* 1988. 18

[136] E. H. Sbai. La classification automatique par les statistiques d'ordre. *Traitement du Signal*, 16, Issue 6 :437–449, 1999. 39, 138

[137] A. Sbihi and J. G. Postaire. *Data to Knowledge : Theoretical and Practical Aspects of Classification, Data Analysis and Knowledge Organization*, chapter Mode Extraction by Multivalue Morphology for Cluster Analysis, pages 212–221. Springer, Berlin, 1995. 97

[138] S. E. Schaeffer. Graph clustering. *Computer Science Review*, 1 :27–64, 2007. 20, 21

[139] I. Sethi and A. K. Jain. *Artificial Neural Networks and Pattern Recognition : Old and New Connections*. Elsevier Science Inc., New York, NY., 1991. 31

[140] G. Sheikholeslami, S. Chatterjee, and A. Zhang. Wavecluster : A multi-resolution clustering approach for very large spatial databases. *In the 24th International Conferenceon Very Large Data Bases (VLDB'98)*, pages 428–439, 1998. 28

[141] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22, Issue 8 :888–905, 2000. 22

[142] P. H. A. Sneath and R. R. Sokal. Numerical taxonomy, the principles and practice of numerical classification. *Systematic Zoology*, 24, Issue 2 :263–268, 1975. 12

[143] J. Starck, E. J. Candes, and D. L. Donoho. The curvelet transform for image denoising. *IEEE Transactions on Image Processing*, 11, Issue 6 :670–684, 2002. 118

[144] R. Stoica, X. Descombes, and J. Zerubia. A gibbs point process for road extraction from remotely sensed images. *International Journal of Computer Vision*, 57, Issue 2 :121–136, 2004. 70

[145] R. S. Stoica, E. Gay, and A. Kretzschmar. Cluster pattern detection in spatial data based on monte carlo inference. *Biometrical Journal*, 49 :505–519, 2007. 72

[146] R. S. Stoica, V. J. Martinez, and E. Saar. A three-dimensional object point process for detection of cosmic filaments. *Journal of Applied Statistics*, 56 :459–477, 2007. 70

[147] D. Stoyan, W. S. Kendall, and J. Mecke. *Stochastic geometry and its applications*. Second edition edition, 1995. 69, 86

[148] D. Stoyan and A. Penttinen. Recent applications of point process methods in forestry statistics. *Statistical Science*, 15 :61–78, 2000. 68

[149] M. Stricker, M.and Orengo. Similarity of color images. *SPIE Storage and Retrieval for Image and Video Databases*, 3 :381–392, 1995. 133

[150] D.H. Swain, M.J.and Ballard. Color indexing. *International Journal of Computer Vision*, 7 :11–32, 2002. 133

[151] R. Szeliski. *Computer Vision : Algorithms and Applications*. Texts in Computer Science. Springer-Verlag London, 2010. 119, 120

[152] S. C. Tan, K. M. Ting, and S. W. Teng. A general stochastic clustering method for automatic cluster discovery. *Pattern Recognition*, 44 :2786–2799, 2011. xii, 30

[153] S. Torquato. Random heterogeneous materials. *Springer-Verlag*, 2002. 68

[154] A. Touzani and J. G. Postaire. Mode detection by relaxation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10, Issue 6 :970–978, 1988. 39

[155] A. Touzani and J. G. Postaire. Clustering by mode boundary detection. *Pattern Recognition Letters*, 9, Issue 1 :1–12, 1989. 39

[156] B. Tso and P. M. Mather. *Classification Methods for Remotely Sensed Data*. 2nd edition edition, 2009. 122

[157] M. N. M. Van Lieshout. *Stochastic Geometry Models in Image Analysis and Spatial Statistics*. 1995. 51, 68

[158] N. Vandenbroucke. *Segmentation d'images couleur par classification de pixels dans des espaces d'attributs colorimétriques adaptés. Application à l'analyse d'images de football*. PhD thesis, Université des Sciences et Technologies de Lille 1. U.F .R. d'Informatique, Électronique, Électrotechnique et Automatique, 2000. 114

[159] V. Vapnik. *The nature of statistical learning theory*. Information Science and Statistics. 2000. 48

[160] C. P. A. Vasseur and J. G. Postaire. A convexity testing method of cluster analysis. *IEEE Transactions on Systems, Man and Cybernetics*, SMC-10 :145–149, 1980. 39

[161] W. Wang, J. Yang, and R. R. Muntz. Sting : A statistica information grid approach to spatial data mining. In *Proceedings of the 23rd International Conference on Very Large Data Bases (VLDB '97)*, pages 186–195, 1997. 28

[162] J. S. Weszka. A survey of threshold selection techniques. *Computer Graphics and Image Processing*, 7 :259–265, 1978. 39

[163] L. Ying-Ho, J.T.L. Anthony, and C. Fu. Object recognition using discriminative parts. *Computer Vision and Image Understanding*, 116 :854–867, 2012. 122

[164] D. Zhang and G. Lu. Review of shape representation and description techniques. *Pattern Recognition*, 37, Issue 1 :1–19, 2004. 118

[165] D. Zhang, Monirul Islam Md., and G. Lu. A review on automatic image annotation techniques. *Pattern Recognition*, 45 :346–362, 2012. 116

[166] G. P Zhang. Neural networks for classification : A survey. *IEEE Transactions on Systems, Man and Cybernetics*, 30, Issue 4 :451–462, 2000. 39

[167] B. Zhao and J. C. Trinder. Integrated approach based automatic building extraction. *International Archives of Photogrammetry and Remote Sensing*, XXXIII, Part B3, 2000. 122

[168] J. Zhu, S. Rosset, H. Zou, and T. Hastie. Multi-class adaboost. Technical report, Standford University, 2006. 46

Activités Scientifiques liées à cette Thèse

Articles publiés ou soumis pour publication

- Auteurs : **Abdellatif El Idrissi**, Olivier Alata, Jack G-Postaire, Abderrahmane Sbihi, Ahmed Moussa.
- Titre : « Classification Using Markov Random Field in Multidimensional Data Analysis ».
- Publié dans : **Wulfenia Journal, Volume. 22, Issue. 3, 2015. (indexé Thomson Reuters, Impact Factor : 0.267)**.

☐ Auteurs : **A. El Idrissi**, Y. El Merabet, Y. Ruichek, C. Meurie , A. Sbihi, A. Moussa

☐ Titre : « A Novel Multiple Objects Recognition Method Based on Region Similarity Measures, Application to Roof Extraction from Orthophotoplans».

☐ Submitted to **The EURASIP Journal on Image and Video Processing**.

- Auteurs : Khadidja Henni, Lynda Zaoui, **Abdellatif El Idrissi**, Olivier Alata, Brigitte Vannier et Ahmed Moussa.
- Titre : « An Unsupervised Clustering Algorithm based on a Marked Point Process for Mode Detection »,
- Submitted to **The Journal of Pattern Recognition**.

Communications internationales

- Auteurs : **El Idrissi Abdellatif** ; El Merabet Youssef ; Ruichek Yassine ; Meurie Cyril ; Sbihi Abderrahmane ; Moussa Ahmed.
- Titre : «Region Similarity Measures for Automatic Detection of Building Roofs from Orthophotoplans »
- 2014 Tenth International Conference on Signal-Image Technology and Internet-Based Systems (SITIS), **IEEE CONFERENCE PUBLICATIONS**
- Page(s) : 125-132, DOI : 10.1109/SITIS.2014.101
- Publication Year : 2014

☐ Auteurs : Khadidja Henni, Lynda Zaoui, **Abdellatif El Idrissi**, Olivier Alata, et Ahmed Moussa.
☐ Titre : «Les Processus Ponctuels Marqués pour l'Extraction des Prototypes des Classes des Distributions Multidimensionnelles»
☐ In the proceeding of 12th African Conference on Research in Computer Science and Applied Mathematics (CARI), pp 101-108. **Saint Louis du Sénégal**
☐ 16-23 October 2014.

- Auteurs : **A. El Idrissi**, Y. Elmerabet, Y. Ruichek, R. Touahni, A. Sbihi et A. Moussa.
- Titre : « Nouvelle Méthode pour la Reconnaissance d'Objets basée sur une Mesure de Similarité entre Régions, Application à la Détection de toitures dans les images orthophotoplans »
- **In the proceeding of 21ème Rencontre de la SFC, CNRST-Rabat, Maroc**.
- 10-12 Septembre, 2014

☐ Auteurs : **Abdellatif El Idrissi**, Abdellatif Ghacham-Amrani, Ahmed Moussa et Olivier Alata.
☐ Titre : « Classification non supervisée à l'aide des Processus Ponctuels Marqués »
☐ **In the proceeding of the TAIMA'2013 conference, Tunisie**, 13-18 May 2013

- Auteurs : A. El Idrissi, A. Moussa, O. Alata , A. Ghacham-Amrani and A. Sbihi.
- Titre : « Marked Point Process for Modes Extraction in Multidimensional Data Analysis »
- **In the abstract proceeding of the ECDA'2013, Luxembourg**, page 33
- 10-12 July 2013

☐ Auteurs : **A. El Idrissi**, A. Ghacham-Amrani, A. Moussa, and O.Alata.
☐ Titre : «Processus Ponctuels Marqués pour l'Extraction des Modes en Analyse de Données Multidimensionnelles».
☐ **Acte de la 3éme édition de la conférence Maghrébine sur l'Extraction et la Gestion des Connaissance (EGC-M), Tunisie**, pp 46-51.
☐ Novembre 2012

Communications nationales

- Auteurs : **A. El Idrissi**, A. Ghacham-Amrani, A. Sbihi, O. Alata et A. Moussa,
- Titre : « Classification non supervisée par les Processus Ponctuels Marqués »
- Acte des Journées Doctorales en Technologies de L'information et de la Communication (JDTIC'13) Kénitra, Maroc.
- 29-30 Novembre 2013

☐ Auteurs : **Abdellatif El Idrissi**, Abdellatif Ghacham-Amrani, Olivier Alata et Ahmed Moussa.
☐ Titre : « Classification Automatique de Données par les Processus Objets de Markov »
☐ Actes de la 15ème Journée de la Recherche Scientifique. Université Abdelmalek Essaadi, Ecole Normale Supérieure. Martil, Maroc.
☐ 20-21 Décembre 2013

Enseignement Cours, Travaux pratiques

1. Enseignement des travaux pratiques (TP) du module Traitement d'Images pour les filières : GSEA2 et Master Informatique et Systèmes Complexes (MISC), Semestre 2 (2013/2014), 48H, à l'Ecole Nationale des Sciences Appliquées de Tanger.
2. Enseignement des travaux dirigés (TD) du module Probabilités Statistiques pour les filières : GSEA, GSTR, GINFO, GINDUS et Master MISC, Semestre 1 (2013/2014), 60H à l'Ecole Nationale des Sciences Appliquées de Tanger.
3. Enseignement du module Processus Markoviens (Cours Travaux de Recherche) au profit des étudiants du Master MISC, 2012/2013 à l'Ecole Nationale des Sciences Appliquées de Tanger.
4. Enseignement des travaux pratiques (TP) du module Traitement d'Images pour les élèves ingénieurs de la filière : GSEA2, 2012/2013 , à l'Ecole Nationale des Sciences Appliquées de Tanger.

Co-encadrement

1. Hossein Taouass, **Abdellatif El Idrissi**, Ahmed Moussa : « Généralisation d'un programme Matlab à des données multidimensionnelles ». Stage d'été, 2011/2012.
2. Khadidja Henni, **Abdellatif El Idrissi**, Ahmed Moussa : « Reprogrammation en langage Java du modèle développé avec applications sur des données génomiques ». Séjours de recherche au Maroc, 2012 et 2013.
3. Abdelilah Nouayti, **Abdellatif El Idrissi**, Ahmed Moussa : « Classification de données Multidimensionnelles basée sur le Modèle de super-paramagnétisme. Projet de fin d'étude, Master En Informatique et Systèmes Complexes, 2ème semestre, 2012/2013.

Reviewing de papiers

Un reviewing de papiers où nous avons appris l'expérience à vérifier les points suivants :

i Originalité du travail.
ii Le coté novateur du papier (clarté de l'idée exposée)
iii Les fautes d'orthographe.
iv La concordance entre la bibliographie et le texte.

v Présence de plagiat.
vi Remplissage du fichier d'évaluation.

Prix

i Prix de meilleure présentation lors de la Journées Doctorales en Technologies de L'information et de la Communication (JDTIC'13) Kénitra, Maroc, 29-30 Novembre 2013.
ii Prix du meilleur papier lors de la 3éme conférence sur l'Extraction et la Gestion des Connaissances-Maghreb Connaissance (EGC-M), Tunisie, Novembre 2012.

Séjours de recherche

Nous avons effectué un séjour de recherche au sein du Laboratoire Systèmes et Transport de l'Université de Technologie de Belfort Montbéliard en France. Ce séjour était l'occasion d'approfondir davantage l'étude de quelques questions du domaine, notamment les méthodes de traitement d'images bas niveau. Le résultat était la soumission d'un abstract au journal SPIE :

- Nombre du papier : 9024-30,
- Titre du papier : An efficient automatic object recognition method based on region similarity : application to roof detection from orthophotoplans.

Séminaire

La participation par une communication orale lors d'un séminaire dans le laboratoire Systèmes et Transport de l'Université de Technologie de Belfort Montbéliard en France :

- Titre : Classification de données par les processus ponctuels marqués (PPM)
- Abdellatif El Idrissi, stagiaire au SeT et doctorant au laboratoire LTI de l'Université Abdelmalek Essaadi à Tanger, Maroc
- 14 Juin 2012 à 14h
- Salle H210, UTBM-SeT.

Membre de comité, de conseil et d'association

i La participation au comité d'organisation de la conférence maghrébine pour l'extraction et la gestion des connaissances dans sa 2ème édition tenue en novembre 2011 à Tanger.

ii Membre de conseil d'établissement, ENSAT, en tant que représentant des doctorants et des étudiants en Master, 2012-2014.

iii Membre de la Société Marocaine de Classification (SMC).