



République Algérienne Démocratique et Populaire
Ministère de l'enseignement supérieur et de la recherche scientifique



UNIVERSITE DES FRERES MENTOURI CONSTANTINE

FACULTE DES SCIENCES DE LA TECHNOLOGIE

Département : ELECTRONIQUE

Laboratoire d'Automatique et de Robotique

Année :
N° d'ordre :
N° de série :

THÈSE

Pour l'obtention du grade de

DOCTEUR EN SCIENCES EN ELECTRONIQUE

Option : **Traitement du signal**

Présentée par

AMEL BOULEMNADJEL EP HADJI

Thème

**La classification dans les sous espaces pour
l'analyse d'images**

Soutenue publiquement le : 25/05/2016

devant le jury composé de :

JURY

N. MANSOURI	Prof	Université des frères Mentouri Constantine	Présidente
F. HACHOUF	Prof	Université des frères Mentouri Constantine	Rapporteur
L. HAMAMI	Prof	Ecole Nationale Polytechnique Alger	Examinatrice
M.C. BATOUCHE	Prof	Université Abdelhamid MEHRI Constantine 2	Examinateur
S. BENIERBAH	M.C.A	Université des frères Mentouri Constantine	Examinateur
S. KHARFOUCHI	M.C.A	Université Rabah Bitat Constantine 3	Invitée

Année Universitaire : 2015-2016

Remerciements

J'adresse mes plus grands remerciements à ma directrice de thèse Prof. **Fella HACHOUF** pour m'avoir accompagnée durant toutes ces années de doctorat, pour le temps qu'elle m'a consacré, pour ses conseils judicieux m'incitant tout le temps à améliorer mon niveau de recherche.

Je remercie vivement Prof. **Noura MANSOURI** pour avoir accepté de présider ce jury et juger ce travail.

Les professeurs **Latifa HAMAMI**, **Mohamed Chaouki BATOUCHE** et **Said BENIERBAH** ont accepté de juger cette thèse; qu'ils trouvent ici l'expression de ma profonde reconnaissance.

Cette thèse ne serait pas complète sans l'aide considérable et les compétences en mathématiques de madame **Soumia KHARFOUCHI**, je tiens à lui exprimer toute ma gratitude.

Cette thèse a été réalisée au sein de l'équipe **Image et vision** du laboratoire Automatique et Robotique. Je remercie tous mes camarades de l'équipe et notamment **Lamia MEZAI**, **Amel HEBBOUL** et **Safia RASELAIN** pour leur précieuse aide.

Enfin, je remercie toutes les personnes que j'ai peut-être oubliées et qui m'ont aidé pour l'accomplissement de cette thèse.

Dédicace

Avec un énorme plaisir, un cœur ouvert et une immense joie que je dédie mon travail, premièrement à mes chers parents qui sont soutenus tout au long de ma vie.

En particulier à mon mari HADJI Billel qui fait l'impossible pour mon aide en me rappelant toujours qu'avec la volonté on peut réaliser nos rêves. Je tiens à lui remercier pour tous les efforts qu'il a fournis pour moi.

Je dédie ce travail aussi à mes chères sœur Meriem et Hadjer et mon frère Houcine pour leur efforts et leur soutient inoubliables

A mon petit ange, mon fils ZEM

A ma tante Djemaa pour tout aide durant les deux stages à Paris.

A mon frère Tallel et sa famille.

Amel

Résumé

La classification automatique des données consiste à diviser un jeu de données en sous-ensembles de données appelés classes pour que tous les individus dans une même classe soient similaires et les individus de classes distinctes soient dissimilaires. La classification dans des données de haute dimensionnalité est extrêmement difficile. Dans le cas de données de grande dimensionnalité, les groupes peuvent être caractérisés uniquement par certains sous-ensembles de dimensions. Ces dimensions pertinentes peuvent être différentes d'un groupe à l'autre. Une nouvelle problématique a donc émergé ; la classification dans les sous espaces (subspace clustering). C'est une extension de la classification traditionnelle qui recherche un ensemble de classes qui peut être défini dans différents sous-espaces. Différents domaines grand public utilisent les techniques de traitement et d'analyse d'images. L'exploitation de ces images dépend fortement de leurs textures. La classification de ces textures joue un rôle important dans l'interprétation de ces images. Dans ce travail, nous avons développé deux approches de la classification des images. La première est une méthode de classification dans les sous espaces, C'est un algorithme itératif basé sur la minimisation d'une fonction objective. Cette nouvelle fonction est développée par l'intégration de la séparabilité et de la compacité des classes dans le quel nous avons introduit leur densité. Une étape d'initialisation est automatisée par un algorithme SVM multi classe. L'apprentissage actif pour l'algorithme SVM est introduit dans l'étape itérative pour accélérer la convergence de la méthode afin d'améliorer la localisation des centres des classes. La deuxième approche proposée est basée sur un nouveau modèle spatial non linéaire qui généralise au cadre spatial le modèle autorégressif à coefficients aléatoires (RCA) standard noté 2D-RCA. L'estimation de ces coefficients est opérée par la méthode (GMM). C'est une méthode supervisée. La classification est peut être effectuée par le calcul d'une distance ou par l'utilisation d'un algorithme de classification supervisée, en l'occurrence l'algorithme des séparateurs à vaste marge. Les méthodes proposées ont été évaluées sur des bases de données synthétiques, des images de synthèse texturées et des images réelles. Les résultats obtenus sont significatifs et prouvent l'efficacité des idées proposées en comparant avec des algorithmes établis dans l'état de l'art.

Mots clés : Classification, sous espace, SVM, Apprentissage actif, 2D-RCA, GMM.

Abstract

Clustering problem consists in partitioning a given data set into groups called clusters, such that the data points in a cluster are more similar to each other than points in different clusters. The clustering in high-dimensional data is extremely difficult. In high dimensional datasets, the clusters can be characterized only by some dimensions subsets. These relevant dimensions can be different from one cluster to another. A new challenging research field has emerged, namely the subspace clustering. It is an extension of traditional clustering that seeks to find clusters in different subspaces within a dataset. Image processing and image analysis tools are widely used in different domains. However, exploiting these images is tightly dependant of their textures. In this work, we have developed two approaches to image classification. The first one is a subspace clustering method. It is an iterative algorithm based on the minimization of an objective function. This function is formed by a separation and compactness terms. The cluster density is also introduced in the compactness term. An initialization step has been improved by a multi class SVM algorithm. An active learning with SVM is incorporated in the classification process to speed the proposed algorithm convergence. It allows enhancing the cluster center location. The second approach is based on a new non linear model which extends the random coefficients autoregressive model (RCA) to a bidimensionally RCA model (2D-RCA). The coefficients are estimated by the generalized moments method (GMM). It is a supervised method.

We have proposed different versions of classification algorithms. The developed approaches have been tested and evaluated on different synthetic datasets and textures and real images. Experimental results have corroborated the effectiveness of the proposed method compared to well-established and state-of-the-art methods.

Mots clés : clustering, subspace, SVM, active learning, 2D-RCA, GMM.

ملخص

التصنيف الذاتي للمعطيات يتركز أساسا على تقسيم هاته المعطيات في مجموعات جزئية تعرف باسم الأقسام حتى تكون جميع العناصر من نفس القسم متشابهة والعناصر من الأقسام الأخرى مختلفة، تصنيف المعطيات ذات الأبعاد الكبيرة يكون صعبا دائما ذلك لان المجموعات قد تعرف بعدد محدود من الأبعاد، هذه الأبعاد قد تكون مختلفة من مجموعة إلى أخرى. ذلك تم طرح إشكالية جديدة وهي تصنيف في الفضاء الجزئي هذه الطريقة هي توسع للطرق القديمة للتصنيف والتي تبحث عن مجموعة من الأقسام التي قد تكون موجودة في مختلف الفضاءات الجزئية.

مجالات واسعة تستعمل تقنيات معالجة وتحليل الصور استغلال الصور يعتمد على طبيعة النسيج تصنيف هاته الأنسجة يلعب دور كبير في فهم الصور في هذا العمل اقترحنا طريقتين لتصنيف الصور الأولى تتمثل في تصنيف المعطيات في الفضاءات الجزئية، هذه الطريقة عبارة عن خوارزم يتركز أساسا على تصغير دالة مكونة من مفهومين هما التفريق والتجميع أين نقوم بإدخال مفهوم الكثافة. اقترحنا طريقة ذاتية للمرحلة الابتدائية عن طريق خوارزم SVM. إدخال التعليم النشط مع الخوارزم SVM في مرحلة التكرار ساهمت في تسريع الطريقة وأيضا في تحسين موقع مراكز الأقسام.

الطريقة الثانية التي قمنا باقتراحها تركز أساسا على نموذج خطي بمعامل عشوائي موحد هو 2D-RCA من اجل استخراج الخصائص المهمة التي تسمح فيما بعد بتصنيف الصور النسيجية. تقديرها الخصائص او العوامل يعتمد على طريقة GMM. هي طريقة مراقبة، قد يعتمد التصنيف عن طريق حساب المسافة او باستعمال خوارزم الموازة مع الخوارزم SVM لقد قمنا بتقييم الطرق المقترحة باستعمال معطيات للتقييم وصور طبيعية وكذلك صور نسيجية. النتائج المحصل عليها تثبت فعالية الطرق المقترحة مقارنة ببعض الطرق المقدمة في الفصل الأول

كلمات البحث: التصنيف، الفضاء الجزئي، SVM، التعليم النشط، 2D-RCA، GMM

Table des matières

Introduction Générale	1
1 Classification dans les sous espaces : État de l'art	4
1.1 Introduction	4
1.2 Le problème de la classification dans les sous espaces	5
1.2.1 Concepts et définitions :	5
1.2.2 Propriétés souhaitées des algorithmes dans les sous espaces	6
1.3 Les algorithmes de base de la classification	7
1.3.1 L'algorithme des k-moyennes	7
1.3.2 L'algorithme des k-médoïdes	7
1.3.3 L'algorithme des C-moyens flous	8
1.3.4 Les cartes de Kohonen	8
1.4 La classification dans les grands espaces	9
1.4.1 Réduction de dimension	10
1.4.2 Sélection des attributs	12
1.4.3 Les modèles parcimonieux	13
1.4.4 bi-classification	13
1.5 Méthodes de classification dans les sous espaces	14

1.5.1	Classification dure dans les sous espaces	14
1.5.2	Classification floue dans les sous espaces	17
1.6	Les critères d'évaluation d'une classification	24
1.6.1	L'information mutuelle normalisée (NMI)	25
1.6.2	L'indice de pourcentage (IP)	26
1.6.3	La déviation des centres (CD)	26
1.6.4	L'indice de Rand (RI)	26
1.6.5	La validation croisée	27
1.7	Conclusion	28
2	Les séparateurs à vaste marge (SVM)	29
2.1	Introduction	29
2.2	La théorie d'apprentissage	30
2.2.1	L'apprentissage supervisé	30
2.2.2	L'apprentissage non supervisée	31
2.2.3	L'apprentissage semi supervisé	31
2.2.4	L'apprentissage actif	32
2.3	les séparateurs à vaste marge (SVM)	34
2.3.1	Cas linéairement séparable	34
2.3.2	Cas linéairement non séparable	37
2.4	SVM multi-classes	39
2.4.1	Un contre tous (One versus All)	39
2.4.2	Un contre un (One versus One)	39
2.5	Conclusion	39
3	Classification des textures par le modèle 2D-RCA	42

3.1	Introduction	42
3.2	Extraction de caractéristiques de textures	43
3.2.1	Analyse statistique	43
3.2.2	Analyse par ondelettes	47
3.2.3	Filtrage	48
3.2.4	Analyse par modèles	48
3.3	Modèles autorégressif à coefficients aléatoires bidimensionnels (2D-RCA)	49
3.3.1	Concepts et définitions	50
3.3.2	Définition du modèle 2D-RCA	54
3.3.3	L'estimation des paramètres et propriétés asymptotiques	55
3.4	Conclusion	58
4	Approches développées pour la classification dans les sous espaces	59
4.1	Introduction	59
4.2	Classification floue basée sur l'apprentissage actif et les séparateurs à vaste marge	60
4.2.1	Initialisation par les séparateurs à vaste marge (SVM)	64
4.2.2	Apprentissage actif et séparateurs à vaste marge	64
4.2.3	Réglage de paramètres	66
4.3	Classification utilisant le modèle autorégressif bidimensionnel à coefficients aléatoires	68
4.3.1	Extraction des caractéristiques par le modèle 2D-RCA	68
4.3.2	Classification utilisant le modèle 2D-RCA	68
4.3.3	Textures générées par le modèle 2D-RCA	70
4.3.4	Mesures de similarité	71

4.4	Conclusion	72
5	Analyse des résultats et discussions	74
5.1	Introduction	74
5.2	Description des bases utilisées	74
5.2.1	Bases de données synthétiques	74
5.2.2	Images réelles	76
5.2.3	Base de textures (Brodatz)	76
5.3	Résultats obtenus par l'algorithme ALISSC	77
5.3.1	Évaluation sur des bases de données synthétiques	77
5.3.2	Évaluation sur des images	87
5.4	Classification utilisant le modèle autorégressif bidimensionnel à coefficients aléatoires 2D-RCA	94
5.4.1	Validation croisée	97
5.4.2	Textures générées par le modèle 2D-RCA	98
5.4.3	Mesure de similarité entre les textures générées et des textures réelles	100
5.5	Conclusion	103
	Conclusion et perspectives	104
	Annexe	106
	Bibliographie	107

Table des figures

1.1	Différentes approches faisant face au problème du fléau de la dimensions. . .	10
2.1	Exemple graphique des données linéairement séparables.	35
2.2	(a) Cas des données linéairement séparables, (b) cas des données linéairement non séparables.	37
2.3	Transformation d'espace.	38
3.1	Plus proches voisins du pixel x selon 4 directions.	45
4.1	Organigramme de la méthode proposée	61
5.1	Description des images utilisées.	76
5.2	Exemple d'images à niveau de gris de texture de Brodatz.	77
5.3	Evolution des centres des classes à chaque itération, vert : Centres réels, bleu : Centres obtenus par ALISSC, rouge : Centres obtenus par ESSC.(a) : iris ; (b) : glass ,(c) :S1	81
5.4	Base Iris avec les centres réels et les centres obtenus par ALISSC et ESSC, (b) : Classification par ALISSC, (c) Classification par ESSC	82
5.5	(a) Base S1 avec les centres réels et ceux obtenus par ALISSC et ESSC, ((b) : Classification par ALISSC, (c) Classification par ESSC	83
5.6	Convergence des méthodes,(a) : l'algorithme ALISSC , (b) : l'algorithme ESSC	88

5.7	Moyenne des résultats de la classification	88
5.8	Résultats de la classification obtenus, 2 ^{eme} colonne par ALISSC, 3 ^{eme} colonne par ESSC, 4 ^{eme} colonne par FCS	90
5.9	a :image original, b,c,d : différents résultats de la classification dans les sous espaces	91
5.10	Classification finale	91
5.11	Résultats dans les sept sous espaces, ALISSC (1 ^{er} colonne), ESSC(2 ^{eme} colonne) et FCS (3 ^{eme} colonne)	93
5.12	Etude comparative,(a) : image de scanner après injection de produit , (b) : résultat final de la classification	94
5.13	Images originales(1 ^{er} colonne). Résultats du modèle 2D-RCA (2 ^{eme} colonne). Résultats du SVM sur le modèle 2D-RCA (3 ^{eme} colonne). Résultats du calcul de la distance sur les matrices de cooccurrence (4 ^{eme} colonne). Résultats du modèle AR(5 ^{eme} colonne).	95
5.14	Images bruitées(1 ^{er} colonne).Résultats du modèle 2D-RCA (2 ^{eme} colonne). Résultats du SVM sur le modèle 2D-RCA (3 ^{eme} colonne). Résultats du calcul de la distance sur les matrices de cooccurrence (4 ^{eme} colonne). Résultats du modèle AR(5 ^{eme} colonne).	96
5.15	Douze images de textures	97
5.16	Huit textures générées par 2D-RCA	99
5.17	Textures similaires aux textures générées selon SIM et GSIM	102
5.18	Textures similaires	102

Liste des tableaux

2.1	Fonctions Noyaux (Kernel function)	38
5.1	Base UCI	75
5.2	Bases S,A et DIM	75
5.3	Résultats de la classification pour la base UCI avec les valeurs de NMI et le nombre d'itérations <i>it</i>	78
5.4	Résultats de la classification pour la base UCI avec l'indice IP	78
5.5	Résultats de la classification pour les bases A, Dim et S avec les valeurs de NMI et le nombre d'itérations <i>it</i>	79
5.6	Résultats de la classification pour les bases A, Dim et S avec l'indice IP . .	79
5.7	Résultats de la classification pour les bases UCI avec l'indice CD	84
5.8	Résultats de la classification pour les bases A, Dim et S avec l'indice CD .	84
5.9	Taux de la classification dans chaque sous espace	85
5.10	La validation croisée de la base UCI	86
5.11	Temps d'exécution (seconde)	87
5.12	Taux de classification	96
5.13	Validation croisée pour douze textures	98
5.14	Différents paramètres du modèle 2D-RCA utilisé pour générer les textures	99
5.15	Meilleures mesures de similarité	101

Liste des notations

v : Centre des classes

u : Degré d'appartenance

m : Paramètre flou

c : Nombre de classes

N : Taille des données

D : Taille des attributs (sous espaces)

w : Matrice des poids

x : Objet / niveau de gris / couleur

n_{ik} : Densité de la i^{eme} classe dans le sous espace k

η : paramètre contrôlant l'influence des poids dans le terme de séparabilité des classes.

γ : paramètre contrôle l'influence de l'entropie.

v_0 : Centre des centres

ent : Entropie des données

N_c : Nombre d'objets dans une classe

T : Vecteur d'apprentissage

k : fonction de noyau

y : Classe d'une entrée x

N_i : le nombre d'éléments dans la classe i résultant de la classification.

N_{ij} : le nombre de correspondances entre la classe i et la classe j donnée par la base.

N_{ti} : le nombre des objets bien classés dans une classe i

N_{fi} : le nombre des objets mal classés dans une classe i

f_{oo} : le nombre des paires d'observation qui n'ont pas les mêmes étiquettes et qui sont

associés aux différentes classes.

f_{11} : le nombre des paires d'observation qui ont les mêmes étiquettes et qui appartiennent aux mêmes classes.

Liste des abréviations

ALISSC : l'algorithme "Active learning for improving a soft subspace clustering".

ESSC : l'algorithme "Enhanced soft subspace clustering".

FCS : l'algorithme "Fuzzy compactness Separation".

SVM : L'algorithme de Séparateur à Vaste Marge.

NMI : Information Mutuelle Normalisée (Normalized mutual information).

IP : Indice de pourcentage.

CD : la déviation des centres

CVE : Evaluation de la validation croisée.

Moy :La moyenne

CST :Le contraste

Var : la variance cor : La corrélation

IDM :Le moment inverse

AR :Modèle autorégressif.

2D-RCA : Modèles autorégressif à coefficients aléatoires bidimensionnels.

GMM :la méthode des moments généralisée.

SSIM :Indice de similarité structurelle.

GSSIM :Structurale à base de gradient

Introduction Générale

Problématique

La classification consiste à séparer différentes zones homogènes des données afin d'organiser les objets en groupes (clusters) dont les éléments ont en commun diverses propriétés (intensité, couleur, texture, etc.). Regrouper dans une même classe des objets ayant des caractéristiques similaires est une tâche très importante dans des domaines aussi varié que la médecine, la génétique, la chimie, la télédétection, etc. Après plusieurs décennies de recherche dans ce domaine, la tâche reste toujours aussi difficile du fait de l'augmentation constante du volume et de la dimension des données à traiter. En effet, les données de grande dimension présentent deux particularités qui rendent difficile l'exploitation et l'extraction de l'information qu'elles contiennent : d'une part les données de grande dimension existent dans des espaces de dimension intrinsèque plus petite que la dimension de l'espace original et, d'autre part, ces données peuvent être bruitées.

Le groupement dans les sous espaces est une extension de la classification traditionnelle qui recherche un ensemble de groupes (clusters) qui peuvent être définis dans différents sous-espaces. L'intérêt de telles techniques est important dans le cadre de données contenant un nombre important de dimensions car elles permettent de faire face au problème de la dimensionnalité. De plus, elles permettent de fournir une description réduite des classes obtenues car ceux-ci sont alors définis par un nombre restreint de dimensions. La classification dans les sous espaces (subspace clustering) est intéressante en ce qu'elle permet d'identifier des groupes dans différents sous espaces et elle est capable de faire

ressortir les comportements locaux de l'ensemble de données analysée.

Différentes approches sont proposées dans la littérature pour gérer ce problème. Et, pour l'appréhender nous nous sommes intéressées aux méthodes basées sur l'optimisation de fonctions objectives. Cependant, ces méthodes posent le problème de la stabilité des résultats due à l'étape d'initialisation et la gestion des données mal et non classées. Dans ce sens, nous pensons apporter une solution à ces problèmes en introduisant les méthodes noyaux avec différents types d'apprentissage au niveau de toutes de toutes les étapes ; à savoir initialisation et la classification proprement dite.

Différents domaines grand public utilisent les techniques de traitement et d'analyse d'images. L'exploitation de ces images dépend fortement de leurs textures. La classification de ces textures joue un rôle important dans l'interprétation de ces images. Elle consiste à sélectionner une classe de texture (prédéfinie) pertinente à chaque région de texture dans une image. Pour résoudre ce problème, nous pouvons le diviser en sous-problèmes comme l'extraction de caractéristiques d'une texture, l'extraction de primitives et le partitionnement de données. Pour ces deux opérations, nous devons réaliser la phase commune : extraction de caractéristiques de texture. Plusieurs travaux ont été menés dans ce sens et la littérature est très riche. Dans ce travail, nous nous sommes intéressés à l'analyse par modèle. L'approche envisagée est basée sur un nouveau modèle spatial non linéaire qui généralise au cadre spatial le modèle auto-régressif à coefficients aléatoires (RCA) standard noté 2D-RCA. L'estimation de ces coefficients est opérée par la méthode (GMM). C'est une méthode supervisée. La classification peut être effectuée par le calcul d'une distance ou par l'utilisation d'un algorithme de classification supervisée, en l'occurrence l'algorithme des séparateurs à vaste marge.

Organisation de la thèse

Cette thèse est organisée en cinq chapitres :

La première partie de ce travail présente un état de l'art des méthodes de classification

dans les sous espaces. Après une brève définition et concepts théoriques de la classification dans les sous espaces ainsi que la présentation des propriétés souhaitées, un rappel des algorithmes classique de classification, le principe de la classification dans les sous espaces tel que la sélection et la réduction d'attributs sont énumérés. Le chapitre se termine par les différents types de classification dans les sous espaces et la validation des classes obtenues.

Le deuxième chapitre est dédié aux séparateurs à vaste marge (SVM). Tous les modes d'apprentissage sont présentés. Les différents type de SVM sont exposés avec leurs différents modes d'utilisation à savoir "*un contre tous*" et "*un contre un*".

La classification d'images texturées par des modèles statistiques constitue l'essentiel du chapitre trois. Le modèle étudié et implémenté est un modèle auto-régressif bidimensionnel à coefficients aléatoires estimés par la méthode des moments généralisée (GMM). Les concepts mathématiques de ce modèle sont donnés et les propriétés du modèle sont démontrées.

Les approches développées pour la classification dans les sous espaces sont présentées dans le chapitre quatre. Une description méthodologique de ces approches est effectuée.

Le cinquième chapitre est consacré à la présentation des résultats expérimentaux. Après la présentation des critères d'évaluation adoptés pour la comparaison des résultats, les premières comparaisons sont réalisées sur des bases synthétiques. Elle montre l'efficacité de l'approche proposée. La deuxième partie est consacrée aux résultats obtenus par la classification dans les sous espaces sur des bases de données synthétiques, des images de synthèse texturées et des images réelles. Les résultats obtenus sont significatifs et prouvent l'efficacité des idées proposées en comparant avec des algorithmes établis dans l'état de l'art. La validation des résultats ont été faites pour toutes les méthodes proposées. L'étude conforte les résultats précédents et prouve la robustesse des approches proposées. Enfin, ce manuscrit s'achève par une conclusion générale et des perspectives.

Chapitre 1

Classification dans les sous espaces :

État de l'art

1.1 Introduction

Regrouper dans une même classe des objets ayant des caractéristiques similaires est une tâche très importante dans des domaines aussi varié que la médecine, la génétique, la chimie, la télédétection, etc. Après plusieurs décennies de recherche dans ce domaine, la tâche reste toujours aussi difficile du fait de l'augmentation constante du volume et de la dimension des données à traiter.

Une problématique supplémentaire a apparue faisant face aux problèmes des bases de données de grande dimension : dans ce cas, les groupes peuvent être caractérisés uniquement par certains sous-ensembles de dimensions. Et, ces dimensions pertinentes peuvent être différentes d'un groupe à l'autre. Sur de tels problèmes, les techniques classiques de partitionnement fonctionnent mal car, fondées sur une distance entre objets définie globalement dans l'espace de description, elles ne peuvent pas appréhender le fait que la notion de similarité varie d'un groupe à l'autre.

La classification dans les sous espaces répond à cette problématique dont l'enjeu est de cibler les groupes d'objets et, pour chacun, le sous espace spécifique dans lequel il est défini.

1.2 Le problème de la classification dans les sous espaces

1.2.1 Concepts et définitions :

Dans cette section, seront présentées quelques notations utilisées pour les différentes définitions de la classification dans les sous espaces.

Soit \mathbb{O} un ensemble des objets et \mathbb{A} un ensemble d'attributs. Les attributs peuvent être de type binaire, catégories discrètes ou continus.

Dans la littérature, le terme dimension est souvent utilisé de manière interchangeable avec l'attribut. Aussi un ensemble des attributs \mathbb{A} est également connu comme un sous espace des données.

Les données sont représentées par une matrice $\mathbb{D} = \mathbb{O} \times \mathbb{A}$. Soit $o \in \mathbb{O}$ un ensemble d'objets et $a \in \mathbb{A}$ un ensemble d'attributs. On note C une classe dans un sous espace : $C = (O, A)$.

Soit $ALL = \{C_1, \dots, C_m\}$ l'ensemble possible des classes dans les sous espaces. Soit $M = \{C_1, \dots, C_n\} = \{(O_1, A_1), \dots, (O_n, A_n)\} \subseteq ALL$ est un ensemble des classes dans les sous espaces.

Le problème de la classification dans les sous espaces recherche un ensemble de classes (clusters) qui sont homogènes dans l'ensemble des attributs. La classification dans les sous espaces est caractérisée par les fonctions suivantes :

Définition 1.1. *La fonction d'homogénéité $h(c)$: cette fonction mesure l'homogénéité dans la matrice $O \times A$. On dit que cette matrice est homogène quand $h(c)$ satisfait un seuil donné.*

L'homogénéité est basée sur une mesure de similarité qui peut être une distance, ou une

densité etc...Elle dépend du problème de la classification. Après la sélection de la mesure, le prochain problème à résoudre est de définir la fonction d'homogénéité. La solution générale est de fixer un seuil et la classe est homogène si la fonction $h(c)$ dépasse ce seuil.

Définition 1.2. La fonction du support $\pi(c)$. Cette fonction mesure la taille de la matrice $O \times A$, on dit que la taille de cette matrice est significative quand $\pi(c)$ satisfait un seuil donné. De façon similaire à l'homogénéité, la solution générale pour décider de la taille significative d'une classe est de fixer un seuil. $C = (O \times A)$ est une classe (cluster) dans un sous espace quand :

1. L'ensemble d'objets O sont homogènes dans l'ensemble des attributs A selon la fonction $h(c)$.
2. La taille de la matrice $O \times A$ est significative selon la fonction $\pi(c)$.

Différentes approches de classification dans les sous espace peuvent résoudre différents problème, qu'elles ont donc leurs propres fonctions.

Définition 1.3. La classification dans les sous espaces :

Soit $C = (O \times A)$, la matrice C forme une classe dans les sous espaces si :

1. La fonction d'homogénéité $h(c)$ est satisfaite.
2. La fonction du support $\pi(c)$ est satisfaite.
3. $\forall o \in O$: peut associer a un autre sous espace C' .

1.2.2 Propriétés souhaitées des algorithmes dans les sous espaces

Résultat complet : il est souhaitable d'exploiter l'ensemble complet des classes dans les sous espaces, de sorte que les classes ne soient pas négligées.

Résultat stable : Dans chaque exécution il faut avoir les mêmes classes, ce qui est important que le résultat est inutilisable s'il est instable.

L'efficacité : il est important d'avoir un algorithme efficace pour gérer des ensembles de données de grande dimension.

1.3 Les algorithmes de base de la classification

La classification dans les sous espaces [1] est une extension du partitionnement traditionnel [2] qui recherche un ensemble de clusters qui peuvent être définis dans différents sous-espaces. La majorité de ces méthodes sont des extensions des algorithmes de classification traditionnelle. Dans cette section, nous présentons quelques algorithmes qui seront les noyaux des algorithmes améliorés de la classification dans les sous espaces.

1.3.1 L'algorithme des k-moyennes

k-means est l'algorithme de regroupement (clustering) le plus connu et le plus utilisé [3], du fait de sa simplicité de mise en œuvre. Il partitionne les données d'une image en classes (cluster). Chaque cluster de la partition est défini par ses objets et son centroïde. Le k-means est un algorithme itératif qui minimise la somme des distances entre chaque objet et le centroïde de son cluster. La position initiale des centroïdes conditionne le résultat final de sorte que les centroïdes doivent être initialement placés le plus loin possible les uns des autres de façon à optimiser l'algorithme. K-means change les objets de la classe jusqu'à ce que la somme ne puisse plus diminuer. Le résultat est un ensemble de classes compactes et clairement séparées, sous réserve qu'on ait choisi la bonne valeur K du nombre des classes.

1.3.2 L'algorithme des k-médoïdes

est un algorithme de partitionnement plus robuste [4] vis-à-vis les données aberrantes (outliers) que celui des k-means (k-moyennes). En statistiques, un médoïde est le représentant le plus central d'une classe. Comme les k-moyennes, l'algorithme des k-médoïdes minimise l'erreur quadratique moyenne qui est la distance entre les points de la classe et le point central (ou médoïde). La grandeur à optimiser n'est donc plus la somme des distances au centre mais la somme des distances des points du jeu de données à l'un d'entre eux. Dans la phase d'initialisation, chaque point est assigné au médoïde qui lui est le plus proche (la plus similaire). Il s'ensuit alors un processus itératif où la distance moyenne

entre les points et le médoïde de leur groupe est minimisée, conduisant ainsi à un meilleur regroupement des points entre eux. Une itération se décompose en trois étapes :

- On choisit un médoïde au hasard et on le remplace par un point choisi au hasard dans les données.
- On recalcule les clusters avec le nouveau médoïde et on compare la nouvelle distance moyenne avec l'ancienne distance moyenne.
- Si la nouvelle distance moyenne est meilleure (i.e. plus petite) on conserve le nouveau médoïde.
- Le processus est alors itéré jusqu'à ce qu'aucune nouvelle substitution n'améliore la classification.

Cet algorithme est ainsi plus robuste aux outliers. En effet, les algorithmes fondés sur le centroïde ont la propriété de grouper tous les points même les plus éloignés, qui peuvent ainsi déplacer le centroïde assez loin de l'idée intuitive que l'on se fait du centre du groupe. (Il est donc une pratique courante d'exclure les outliers avant l'analyse).

1.3.3 L'algorithme des C-moyens flous

l'algorithme FCM (Fuzzy C-means) [5] est un algorithme de classification floue fondé sur l'optimisation d'un critère quadratique de classification où chaque classe est représentée par son centre de gravité. L'algorithme nécessite de connaître le nombre de classes au préalable et génère les classes par un processus itératif en minimisant une fonction objective. Ainsi, il permet d'obtenir une partition floue de l'image en donnant à chaque pixel un degré d'appartenance à une région donnée.

1.3.4 Les cartes de Kohonen

self-organizing maps SOM ou les cartes de Kohonen dans la littérature [6] peuvent être définies comme un algorithme de classification non supervisé issu du domaine des réseaux de neurones artificiels.

Ces cartes se distinguent par la propriété commune suivante : pour un espace de données

fixé, par exemple les signaux lumineux sur la rétine, la carte corticale est une représentation à deux dimensions telle que des données "voisines" aient des représentations voisines donc, un certain nombre des fonctions sensorielles sont dépendantes de la réalisation d'applications qui conservent la topologie entre l'espace des données (sur les capteurs) et l'espace des représentations (dans la zone corticale correspondante).

Du point de vue informatique, on peut traduire cette propriété de la façon suivante :

Supposons que l'on dispose de données que l'on désire classifier. On cherche un mode de représentation tel que des données voisines soient classées dans la même classe ou dans des classes voisines. L'algorithme proposé par Kohonen produit un réseau qui a cette propriété [6]. Cet algorithme projette l'espace des données X sur un espace discret de faible dimension. Cet espace, qu'on appelle la "carte" sera noté A . L'ensemble A est constitué par un ensemble de neurones inter-connectés, les liens entre neurones se faisant par l'intermédiaire d'une structure de graphe non orienté. La structure de graphe induit une distance discrète δ sur A : pour toute paire de neurones (c, r) de cette "carte", $\delta(c, r)$ est définie comme étant la longueur du plus court chemin entre le neurone c et r . Cette distance permet de définir la notion de voisinage d'ordre d de c . La notion de distance entre neurones ou de voisinage découle alors directement de cette structure graphique, et définit la topologie discrète de la carte.

1.4 La classification dans les grands espaces

La classification de données situées dans un espace de grande dimension est un problème délicat qui apparaît dans de nombreuses sciences telles que l'analyse d'images. Un problème plus connu sous le terme de "fléau de la dimension" [7-9], implique la perte du pouvoir discriminant de la notion de distance au fur et à mesure que la dimension augmente. Aggarwal et al. [10] montrent théoriquement que quand l'espace devient très grand les mesures classiques de distance deviennent inefficaces, car elles ne permettent pas de distinguer les points proches des points éloignés. Dans un espace de dimension p , si l'on considère que les variables ont une distribution aléatoire d'observations et si l'on note

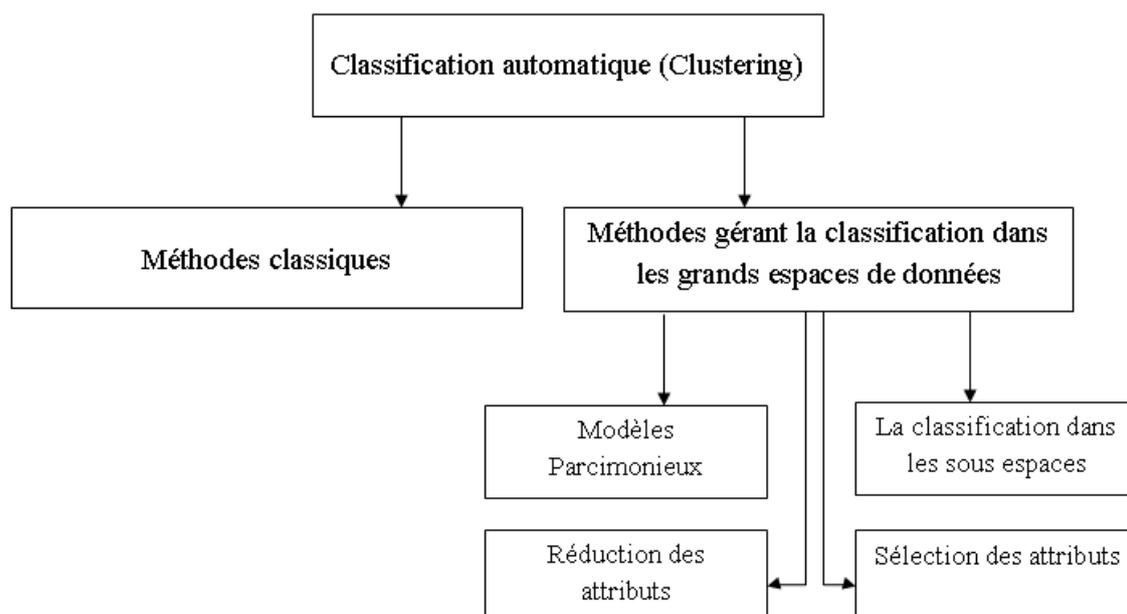


FIGURE 1.1 – Différentes approches faisant face au problème du fléau de la dimensions.

d_{min} la distance entre les deux points les plus proches et d_{max} la distance entre les deux points les plus éloignés alors $\lim_{p \rightarrow \infty} \frac{d_{max} - d_{min}}{d_{min}} = 0$. Plusieurs approches ont été proposées pour la classification dans les grands espaces [11,12].

La figure 1.1 montre les différentes approches pour s'affranchir du problème du la dimension (curse of dimensionality). Dans cette section, nous discutons brièvement ces méthodes.

1.4.1 Réduction de dimension

La réduction de dimension consiste à transformer des données représentées dans un espace de grande dimension en une représentation dans un espace de dimension plus faible. Idéalement, la nouvelle représentation a une dimension égale au nombre de paramètres nécessaires pour décrire les données observées. La réduction de dimension est importante dans de nombreux domaines étant donné qu'elle facilite la classification, la visualisation ou encore la compression de données de grande dimension. Elle permet également souvent de limiter l'effet de la malédiction de la dimension et d'autres propriétés non désirées des espaces de grande dimension [13].

De nombreuses méthodes utilisent une réduction de dimension globale pour s'affranchir

le problème du fléau de la dimension [14,15]. Les plus simples d'entre elles consistent à réduire la dimension préalablement à une classification classique. A ce titre, l'analyse en composantes principales (ACP) est très souvent utilisée en analyse d'images.

L'analyse en Composantes Principales (ACP) [16], aussi connue sous le nom de transformée de Karhunen-Loève[17] est une méthode très utilisée en statistique. Introduite par Pearson [18] puis plus tard par Hotelling [19], sa principale idée est de réduire la dimension d'un jeu de données tout en gardant un maximum d'informations. Cela est réalisé grâce à une projection qui maximise la variance tout en minimisant l'erreur quadratique moyenne de la reconstruction. L'ACP ne prenant en compte que les dépendances linéaires entre les variables, de nombreuses alternatives ont été proposées telles que l'ACP par noyaux [20], ACP par variétés [21], ou réseaux de neurones [22,23]. La réduction de dimension globale offre souvent de bonnes performances mais au prix d'une perte d'information qui aurait pu être discriminante. En effet, lorsque les classes sont localisées dans des sous-espaces différents, toute approche globale est inadaptée. Parmi les méthodes de réduction de dimension, on trouve aussi L'Analyse en Composantes Indépendantes (ou ICA pour Independent Component Analysis) a été introduite par Héroult et al dans [24] dans le contexte de la neurophysiologie. Elle devint populaire lors de son utilisation dans le domaine du traitement du signal pour la séparation de sources aveugles. Le but est d'exprimer un ensemble de n variables aléatoires x_1, \dots, x_n comme une combinaison linéaire de n variables aléatoires statistiquement indépendantes S , sous forme matricielle :

$$x = A_s$$

où $x = [x_1, \dots, x_n]^T$, $s = [s_1, \dots, s_n]^T$ et A est une matrice contenant les coefficients a_{ij} . Le but de l'Analyse en Composantes Indépendantes est l'estimation des composantes originales si, ou de manière équivalente des coefficients a_{ij} .

Si les données d'apprentissage sont labellisées, ces informations peuvent être utilisées pour l'apprentissage du sous-espaces. Ainsi, pour assurer une classification plus efficace.

l'Analyse Discriminante Linéaire de Fisher (LDA pour Linear Discriminant Analysis) a pour but de maximiser la distance entre les classes tout en minimisant la variance intra-

classe. Plus formellement, soient $\{x_1, \dots, x_n\}$ n échantillons appartenant à une classe parmi $c \{X_1, \dots, X_c\}$. L'Analyse Discriminante Linéaire calcule une fonction de classification $g(x) = W^T x$, où la matrice W est choisie comme la projection linéaire minimisant la variance intra-classe[25].

On trouve aussi les méthodes non linéaires de réduction de dimension. Les techniques non linéaires peuvent être catégorisées en trois principaux types : les techniques essayant de préserver les propriétés globales des données d'apprentissage dans l'espace de faible dimension, les techniques s'attachant à préserver les propriétés locales des données d'apprentissage, et les techniques réalisant un alignement global de modèles linéaires.

La réduction de dimension globale offre souvent de bonnes performances mais au prix d'une perte d'information qui aurait pu être discriminante. En effet, lorsque les classes sont localisées dans des sous-espaces différents, toute approche globale est inadaptée.

1.4.2 Sélection des attributs

Dans ces méthodes, il n'y a pas de transformation des attributs. Elles consistent à identifier le sous-ensemble d'attributs de taille minimale nécessaire et suffisant pour définir le concept cible [26,27]. Les techniques de sélection d'attributs sont divisibles en trois catégories, selon la manière dont elles interagissent avec le classifieur.

Approche par filtre (Filter model)[28] : opèrent directement sur le jeu de données et fournissent une pondération, un classement ou un ensemble de variables en sortie. Ces méthodes ont l'avantage d'être rapides et indépendantes du modèle de classification, mais au prix de résultats inférieurs.

Approche symbiose (Wrapper model) [29] : effectuent une recherche dans l'espace des sous-ensembles des objets, guidée par le résultat du modèle, par exemple les performances en validation croisée sur les données d'apprentissage. Elles ont souvent de meilleurs résultats que les méthodes de filtrage, mais au prix d'un temps de calcul plus important.

Les méthodes embarquées (Embedded methods) utilisent l'information interne du modèle de classification par exemple, le vecteur de poids dans le cas des SVM(support

vector machines)[30]. Ces méthodes sont donc proches des méthodes d'enveloppes, du fait qu'elles combinent le processus d'exploration avec un algorithme d'apprentissage sans étape de validation, pour maximiser la qualité de l'ajustement et minimiser le nombre d'attributs [31]. La différence de ces dernières, avec les méthodes enveloppes est que le classifieur sert non seulement à évaluer un sous-ensemble candidats mais aussi à guider le mécanisme de sélection. Selon Elisseff et al [31], ces méthodes seraient bien plus avantageuses en terme de temps de calcul que les méthodes de type wrapper et seraient robustes face au problème de sur-ajustement.

1.4.3 Les modèles parcimonieux

Une solution alternative consiste à utiliser des modèles nécessitant l'estimation de peu de paramètres. Ainsi, il est possible de re-paramétriser les matrices de covariance des classes à partir de leur décomposition en éléments propres [32-33] et, en contraignant certains paramètres à être commun à toutes les classes, on obtient alors des modèles parcimonieux. D'autres modèles parcimonieux gaussiens sont introduits dans [34]. Ils forment une hiérarchie du plus complexe (une matrice de covariance pleine affectée à chaque groupe) au plus simple (une matrice de covariance identité commune à tous les groupes - modèle de l'algorithme des moyennes mobiles). Cependant, ces modèles ne peuvent rendre compte de l'existence d'un sous-espace propre spécifique à chaque classe.

1.4.4 bi-classification

Les méthodes de bi-partitionnement ou de Co-clustering recherchent une partition C des observations z_i et une partition C_0 des variables z_j dont les classes caractérisent les classes de la partition C . Cette famille de méthodes a suscité beaucoup d'intérêts dans les domaines telles que l'analyse de données textuelles et la génétique où l'objectif est de définir des classes de documents par des classes de mots ou de gènes. Elles visent à obtenir des blocs individus/variables ou lignes/colonnes les plus homogènes selon des critères métriques ou probabilistes.

Il existe plusieurs algorithmes de bi-partitionnement [35] :

Les méthodes basées sur des algorithmes de partitionnement simple [36] : ces méthodes appliquent un algorithme de classification simple sur les lignes et sur les colonnes séparément. Les bi-classes sont construites à partir des classes obtenues sur les lignes et sur les colonnes. Cette famille inclut les méthodes Croeuc, Croki, Crobin et Cromul présentées par Govaert[37].

Les méthodes divisives : ces méthodes procèdent par découpage itératif afin d'aboutir à des classes. Cette famille inclut les méthodes one-way splitting et two-way splitting de Hartigan.[38]

L'inconvénient de ces méthodes est que elles ont besoin d'un large nombre d'attributs binaires ce qui augmente le coût informatique ainsi que l'espace des données.

1.5 Méthodes de classification dans les sous espaces

De nombreuses méthodes de la classification dans les sous espaces existent dans la littérature[1,2] ,[39-41]. Généralement on peut les scinder en deux grandes catégories : la classification dure et la classification floue dans les sous espaces.

1.5.1 Classification dure dans les sous espaces

Dans ce type de classification la probabilité d'un point (objet) appartenant à une autre classe que la sienne est nulle. Parmi ces méthodes, on trouve deux types d'algorithmes de recherche dans sous-espaces : les méthodes de recherche ascendante dites bottom-up et descendante (Top-down) et les méthodes basées sur la densité [1].

Les Méthodes de recherche ascendante (Bottom-up) :

Ces méthodes utilisent les méthodes de classification basées sur un maillage de l'espace des observations en définissant pour chaque dimension un histogramme. Puis, les intervalles ayant une densité d'observations supérieure à un seuil fixé a priori définissent

des classes pour chaque variable. Les auteurs font l'hypothèse que si un espace de q dimensions présente une forte densité d'observations alors tout espace composé de $q - 1$ dimensions de cet espace est aussi dense. L'algorithme CLIQUE : CLustering In QUEst[42] fut l'un des premiers algorithmes bottom-up proposé pour rechercher des groupes dans des sous-espaces de l'espace original. L'idée est basée sur la notion de densité. Il recherche automatiquement des sous-espaces de plus grande dimensionnalité contenant des classes de forte densité. CLIQUE partitionne l'espace des observations en unités rectangulaires denses. Les unités denses voisines sur la grille sont regroupées pour former des classes. Étant donné un espace initial à p attributs, CLIQUE procède comme suit :

- Découpage de chaque dimension de l'espace en I intervalles de même largeur.
- Détermination de l'ensemble des cellules denses dans l'ensemble des sous-espaces de l'espace des données.
- Détermination des classes comme un ensemble maximal de cellules denses contiguës.

L'algorithme ENCLUS : ENTropy-based CLUStering[43] une approche semblable à CLIQUE. Elle utilise un critère basé sur la notion d'entropie pour sélectionner des sous-espaces de données denses d'observations. Comme CLIQUE, ENCLUS utilise des unités rectangulaires fixées à l'avance. Cependant, ENCLUS recherche directement des sous-espaces contenant potentiellement une ou plusieurs classes d'observations contrairement à CLIQUE.

Les méthodes de recherche descendante (Top-down) :

Les techniques de recherche de ce type sont des méthodes itératives qui, partant de l'espace entier comme solution initiale déterminent une première classification à l'aide de l'ensemble des variables. Un poids associé à chaque variable est ensuite utilisé dans une nouvelle phase d'un processus itératif pour réaffecter les observations aux classes. La difficulté principale dans cette catégorie est la définition du nombre de classes et du nombre de variables formant le sous-espace associé à une classe.

L'algorithme PROCLUS : PROjected CLUStering[44] est la première méthode de type Top-down qui évalue à chaque itération la qualité de la classification en calculant la distance moyenne entre les centres des groupes. Il se présente comme une extension de

l'algorithme des k-médoïdes à la classification dans les sous espaces[3] en déterminant les représentants les plus centraux des classes. Son processus se déroule en trois étapes : initialisation, itération et raffinement. La première phase utilise les techniques "gloutonnes" pour échantillonner et sélectionner les K-médoïdes potentiels. L'idée consiste ensuite à définir autour des médoïdes, des groupes homogènes d'observations. Dans la phase d'itération l'algorithme calcule la pertinence des médoïdes à travers la moyenne des distances entre les observations d'une classe et le médoïde correspondant. Les médoïdes les moins pertinents sont ensuite remplacés par des nouveaux médoïdes choisis aléatoirement. Dans la phase de raffinement, pour chaque classe formée, un ensemble de variables de taille $l < p$ fixée a priori est choisi tel qu'il minimise la distance des observations au médoïde. Enfin PROCLUS réaffecte les observations aux nouveaux médoïdes. La procédure s'arrête lorsque la qualité d'un résultat ne change pas après un certain nombre de changements des médoïdes. Comme l'algorithme des K-moyennes, PROCLUS se spécialise dans la détermination des classes à structure sphérique qui sont cette fois représentées par un ensemble de médoïdes et de sous-espace correspondants.

L'algorithme ORCLUS [45] : étend la méthode PROCLUS à la recherche de classes dans des sous-espaces formés par des combinaisons linéaires des variables initiales. Cet algorithme contient les phases d'affectation des observations, de détermination des sous-espaces et de regroupement des classes proches dans l'espace. La phase d'affectation est identique à PROCLUS. Pour chaque classe, un sous-espace de vecteurs orthonormés est formé de vecteurs propres de la matrice de variance-covariance des observations qu'elle contient.

L'algorithme COSA : Clustering Objects on Subsets of Attributes[46] est un algorithme itératif de type "top-down" qui associe à chaque observation des poids sur les dimensions. L'algorithme commence par affecter un poids identique à l'ensemble des attributs, puis il détermine les k-plus proches voisins (kNN) de chaque observation. Les poids forts, obtenus par optimisation d'un critère objectif, sont affectés aux variables dont la dispersion est faible dans les groupes de l'algorithme kNN. Ce processus est alors répété jusqu'à la stabilisation des poids.

Les méthodes basées sur la densité

Dans ces méthodes, les classes sont considérées comme des régions de haute densité séparée par des régions de faible densité. La densité est représentée par le nombre d'objets de données dans la classe. L'objectif des méthodes basées sur la densité est de gérer les classes de forme sphérique et gérer le bruit qui peut exister dans les données. Un algorithme très populaire dans ce type d'algorithme de classification est DBSCAN [47].

L'algorithme DBSCAN : Density-Based Spatial Clustering of Applications with Noise est conçu pour regrouper les données spatiales issues de grandes bases de données avec possibilité de bruit dans les classes (clusters). Ces derniers étant de forme arbitraire, il fait intervenir un nombre minimum de paramètres d'entrée, correspondant au rayon du groupe (Eps) et au nombre minimum de points requis à l'intérieur du groupe (Minpts). Pour créer un groupe, l'algorithme commence par un objet arbitraire p et recherche tous les objets accessibles par densité à partir de p . Si p est un noyau, la procédure crée un groupe et ajoute p à ce dernier ; si p est un objet périphérique, aucun objet ne peut être atteint à partir de p et l'algorithme visitera l'objet suivant dans l'échantillon. L'ajout de chaque objet crée ainsi une zone d'extension du groupe, qui s'élargira progressivement. Les valeurs globales Eps et MinPts permettent à l'algorithme de fusionner deux groupes voisins mais de densité différente.

Il convient de souligner que cet algorithme montre certaines particularités/avantages :

- Son application nécessite un minimum de connaissances sur les domaines objets de la classification, afin de pouvoir définir les paramètres d'entrée ;
- Il est très efficace lors du traitement des bases de données larges (milliers d'objets) ;
- Il crée des groupes de forme arbitraire, pouvant être sphériques, étirés, linéaires ou allongés.

1.5.2 Classification floue dans les sous espaces

À l'inverse de la classification dure, on reconnaît qu'une observation peut effectivement appartenir simultanément à plusieurs classes à des degrés divers dont la somme est égale

à 1. Cette attitude reflète le fait que dans de nombreuses situations réelles, les définitions des classes (supervisées ou non) ne rendent pas obligatoire l'appartenance à une classe et une seule (exemple des classes : "Chaud" et "Très chaud"). La classification floue affecte alors à chaque observation des degrés d'appartenance aux diverses classes d'une façon cohérente avec la répartition géométrique des données. Pour chaque observation, la somme des degrés d'appartenance à chacune des classes est égale à 1. Plusieurs méthodes de ce type ont été trouvées dans la littérature basées sur la minimisation des différentes combinaisons d'une fonction objective.

L'algorithme Entropy weighting K-Means (EWKM) :

Dans [48] les auteurs ont proposé un algorithme utilisant les k-moyennes pondérées. Entropy weighting K-Means (EWKM) est basé sur une définition d'un système de pondération par modification de la fonction de coût associée à l'algorithme des K-Moyennes en introduisant des poids. Dans [49] cet algorithme a été amélioré en introduisant l'entropie pondérée. Les auteurs minimisent simultanément, l'inertie intra-classe et maximisent un terme d'entropie dans le processus d'apprentissage. EWKM calcule pour chaque variable des poids inversement proportionnels à leur variance dans chaque classe. Le sous-espace de variables pertinents pour chaque classe est défini en se basant sur ces poids, facilitant ainsi l'interprétation des classes. La fonction objective de l'algorithme EWKM est définie comme suivante :

$$J_{EWKM} = \sum_{i=1}^C \sum_{j=1}^N u_{ij} \sum_{k=1}^D w_{ik} (x_{jk} - v_{ik})^2 + \gamma \sum_{i=1}^C \sum_{k=1}^D w_{ik} \ln w_{ik} \quad (1.1)$$

Sous les contraintes :

$$u_{ij} \in \{0, 1\}, \sum_{i=1}^C u_{ij} = 1, 0 < \sum_{j=1}^N u_{ij} < N, w_{ik} \in [0, 1] \text{ and } \sum_{k=1}^D w_{ik} = 1$$

Où

U : matrice d'appartenance

w : entropie pondérée

v : les centres de classes

C : le nombre de classe

D :le nombre de dimension (attribut ou features)

Un autre algorithme basé aussi sur un terme d'entropie qui s'appelle LAC.

L'algorithme Local adaptative clustering (LAC) :

Local adaptative clustering[50] est un algorithme de classification adaptatif dans les sous espaces prenant en considération l'entropie. Sa fonction objective se présente comme suit :

$$J_{LAC} = \sum_{i=1}^C \sum_{k=1}^D W_{ik} v_{ik} + \gamma \sum_{i=1}^C \sum_{k=1}^D W_{ik} \ln w_{ik} \quad (1.2)$$

Où

$$v_{ik} = \frac{\sum_{j=1}^N u_{ij} (v_{jk} - v_{ik})^2}{\sum_{j=1}^N u_{ij}} \quad (1.3)$$

Sous les contraintes :

$$u_{ij} \in \{0, 1\}, \sum_{i=1}^C u_{ij} = 1, 0 < \sum_{j=1}^N u_{ij} < N, w_{ik} \in [0, 1] \text{ and } \sum_{k=1}^D w_{ik} = 1$$

On constate que cette fonction objective et celle de l'algorithme EWKM sont très similaires avec une seule différence est que l'effet de la taille des clusters pris en compte dans l'éq.1.1, mais manque dans l'éq.1.2.

L'algorithme Feature Group k-means (FGKM)[51]

Cet algorithme est une version étendue de la méthode EWKM[49] à la classification d'individus décrits par un grand nombre de variables structurées en blocs. Un second terme d'entropie défini au niveau des blocs permet d'établir des scores sur ces derniers et ainsi faire de la sélection des blocs. La fonction objective à minimiser associée à cet algorithme est :

$$J(u, v, w, \beta) = \sum_{i=1}^c \left(\sum_{j=1}^N \sum_{k=1}^D \sum_{b \in G^k} u_{ij} w_{ik} \beta_{ib} d(x_{ij}, v_{ik}) + \right. \\ \left. \gamma \sum_{k=1}^D w_{ik} \log(w_{ik}) + \lambda \sum_{b=1}^p \beta_{ib} \log(\beta_{ib}) \right) \quad (1.4)$$

Sous les contraintes :

$$\begin{cases} \sum_{i=1}^c u_{ij} = 1, u_{ij} \in \{0, 1\} \\ \sum_{k=1}^D w_{ik} = 1, 0 \leq w_{ik} \leq 1, 1 \leq k \leq D \\ \sum_{b \in G^k} \beta_{ib} = 1, 0 \leq \beta_{ib} \leq 1, 1 \leq k \leq D \end{cases} \quad (1.5)$$

Ou C : le nombre des classes

D : le nombre de dimension

N : taille de données

u : le degré d'appartenance

w : le poids de l'entropie de chaque dimension

v : centre des classes

$$d(x_{ij}, v_{ik}) = (x_{ij} - v_{ik})^2$$

β : matrice des poids

λ : paramètre contrôle les valeurs des poids

p : le nombre de dimension représentant les objets dans x.

G : l'ensemble des sous espaces

Les méthodes présentées dans cette section ont l'inconvénient majeur de ne pas prendre en compte la distance intra-classe, l'algorithme suivant propose une amélioration de la classification floue dans les sous espaces. Il utilise deux informations concernant l'inter et l'intra-classes.

L'algorithme Enhanced SubSpace Clustering (ESSC)[52] :

Cet algorithme est basé sur la minimisation d'une nouvelle fonction objective. Deux informations concernant l'inter et l'intra-classes ont été utilisées par cette méthode. La fonction objective contient trois termes ; le terme de compacité, le terme de l'entropie pondérée et le terme de séparabilité. Cette modification introduit par ESSC apporte une

grande amélioration sur la classification.

$$J_{ESSC} = \sum_{i=1}^C \sum_{j=1}^N u_{ij}^m \sum_{k=1}^D w_{ik} (x_{jk} - v_{ik})^2 + \gamma \sum_{i=1}^C \sum_{k=1}^D w_{ik} \ln w_{ik} - \eta \sum_{i=1}^C \left(\sum_{j=1}^N u_{ij}^m \right) \sum_{k=1}^D w_{ik} (v_{ik} - v_{0k})^2 \quad (1.6)$$

Où

C : le nombre des classes

D : le nombre de dimension

N : taille de données

u : le degré d'appartenance

w : le poids de l'entropie de chaque dimension

v : centre des classes

v_0 : moyenne des centres des classes

γ : paramètre contrôle l'influence de l'entropie.

η : paramètre contrôle l'influence du terme de séparabilité des classes.

Le multiplicateur de Lagrange a été utilisé pour déduire les expressions de degré d'appartenance u, les poids w et les centres de classes v qui minimisent la fonction d'objective (eq.1.6), données respectivement par les équations :

$$u_{ij} = \frac{[\sum_{k=1}^D w_{ik} (x_{jk} - v_{ik})^2 - \eta (v_{ik} - v_{0k})^2]^{\frac{-1}{m-1}}}{\sum_{i=1}^c [\sum_{k=1}^D w_{ik} [(x_{jk} - v_{ik})^2 - \eta (v_{ik} - v_{0k})^2]^{\frac{-1}{m-1}}} \quad (1.7)$$

$$v_{ik} = \frac{\sum_{j=1}^N u_{ij}^m (x_{jk} - \eta v_{0k})}{\sum_{j=1}^N u_{ij}^m (1 - \eta)} \quad (1.8)$$

$$w_{ik} = \frac{\exp(\frac{-\delta_{ik}}{\gamma})}{\sum_{h=1}^D \exp(\frac{-\delta_{ih}}{\gamma})} \quad (1.9)$$

Où

$$\delta_{ik} = \sum_{j=1}^N u_{ij}^m (x_{jk} - v_{ik})^2 - \eta \left(\sum_{j=1}^N u_{ij}^m \right) (v_{ik} - v_{0k})^2 \quad (1.10)$$

On peut résumer les étapes de cette méthode par l'algorithme ESSC :

Cette méthode pose un problème de localisation des centres de classes, en terme d'estimation des distances inter et intra classes. En effet l'estimation des distances se base sur

ESSC

Étape 1 : Initialisation

Entrées : le nombre de classe C , paramètres m, ε, η .

Initialisation arbitraire des centres des classe V et les poids des entropies W

Étape 2 : Traitement

Tant que $u(t+1) - u(t) > \varepsilon$ **faire**

 Calculer u selon l'équation (1.7).

 Calculer v selon l'équation (1.8).

 Calculer w selon l'équation (1.9).

Fin du tant que

Étape 3 : Classification

Assigner chaque objet x à sa propre classe i en prenant le degré d'appartenance u maximal.

le centre moyen des centres à travers les itérations de l'algorithme. Par conséquent l'erreur d'estimations des distances est propagée et amplifiée à travers les itérations. Ceci implique un taux de classification faible et un temps de traitement élevé. Un autre problème est posé, quand la densité des classes varie. Bien que la densité est un outil très important pour la classification des données de grande dimension qui donne de bon résultats [53-55], mais ESSC l'a ignorée.

L'algorithme Fuzzy Compactness Separation (FCS) :

Dans[56]les auteurs ont utilisé deux informations très importantes; la compacité et la séparabilité des classes pour construire leur algorithme. Sa fonction d'objective est basée sur une matrice de dispersion floue de sorte que le FCS puisse être construit par la minimisation de la compacité et en même temps la maximisation de la séparabilité entre les classes. Sa fonction d'objective est donnée par l'eq.1.11 :

$$J_{FCS} = \sum_{i=1}^C \sum_{j=1}^N u_{ij}^m (x_j - v_i)^2 - \sum_{i=1}^C \sum_{j=1}^N w_{ik} (u_{ij}^m (v_i - v_0)^2) \quad (1.11)$$

Où

C : le nombre des classes

D : le nombre de dimension

N : taille de données

U : le degré d'appartenance

V : centre des classes

w : le poids des classes v_0 : moyenne des centres des classes

η : paramètre de contrôle.

Cet algorithme est robuste pour des données bruitées. La solution du problème de minimisation s'obtient à l'aide du multiplicateur de Lagrange.

Les solutions qui minimisent cette fonction objective sont sous la forme :

$$u_{ij} = \frac{[(x_j - v_i)^2 - \eta_i(v_i - v_0)]^{\frac{-1}{m-1}}}{\sum_{h=1}^C [(x_j - v_h)^2 - \eta_h(v_h - v_0)]^{\frac{-1}{m-1}}} \quad (1.12)$$

$$v_{ik} = \frac{\sum_{j=1}^N u_{ij}^m (x_{jk} - w_i v_0)}{\sum_{j=1}^N u_{ij}^m (1 - w_i)} \quad (1.13)$$

$$w_i = \frac{(\varphi/4) \min_{i \neq h} (v_i - v_h)^2}{\max_i (v_i - v_0)^2} \quad (1.14)$$

Où $\varphi \in [0, 1]$

L'algorithme FCS résume les étapes de cette méthode :

La méthode Soft Subspace Self OrganizingMap (2S-SOM)[57] :

Cette approche de type classification dans sous espace basée sur une version modifiée de la fonction de coût de SOM en introduisant des poids adaptatifs sur les blocs et sur les variables et un terme d'entropie permettant de définir les contributions relatives des variables et des blocs inspirée des travaux de Huang et Ng [58]; Jing et al. [49] et de Chen et al. [51]. L'idée de base consiste à rechercher itérativement une partition des observations et à déterminer pour chaque cellule des attributs et des blocs spécifiques. Pour surmonter la dépendance des résultats de la méthode SOM par rapport aux paramètres d'initialisation, les auteurs proposent deux approches de recherche de consensus de SOM, CSOM (Consensus SOM) et Rv-CSOM, qui prennent en compte la structuration en bloc

FCS

Étape 1 : Initialisation

le nombre de classe C , paramètres m, ε, η .

Initialisation arbitraire des centres des classe V

Étape 2 : Traitement

Tant que $u(t+1) - u(t) > \varepsilon$ **faire**

 Calculer u selon l'équation (1.12).

 Calculer v selon l'équation (1.13).

 Calculer w selon l'équation (1.14).

Fin du tant que

Étape 3 : Classification

Assigner chaque objet x à sa propre classe i en prenant le degré d'appartenance u maximal.

des variables. Dans l'approche CSOM, l'objectif est de favoriser les cartes les meilleures au sens d'un certain critère de validation, alors que Rv-CSOM va plutôt privilégier les cartes similaires par évaluation de la liaison entre cartes.

1.6 Les critères d'évaluation d'une classification

L'évaluation de la qualité d'une classification est un aspect très important pour valider les classes obtenues. La qualité d'une classification non supervisée dépend de plusieurs paramètres, à savoir le nombre de classe attendus par l'utilisateur. Une variation de cette tâche est de ne pas utiliser le nombre attendu de classes comme une donnée du problème. Dans ce cas, l'algorithme construit plusieurs partitions candidates et choisit la meilleure. La meilleure partition est celle qui optimise un critère de qualité des partitions. Donc une étape d'évaluation les résultats de la classification est indispensable.

Plusieurs critères d'évaluation existent dans la littérature. On peut les regrouper en deux grandes familles. La famille des critères non-supervisés ou internes qui utilisent uniquement les informations internes aux données telles que la distance entre les ob-

servations, pour quantifier l'adéquation entre les classes obtenues avec un algorithme de classification et l'idée que l'on se fait d'une bonne classification à savoir la séparabilité et la compacité des classes. La famille des critères externes s'utilisant pour la validation d'un algorithme de classification fait appel à des connaissances externes à la classification. En effet, il est habituel d'utiliser des données étiquetées ; ces étiquettes ou labels peuvent être obtenus selon l'avis d'expert ou suite à l'application d'un algorithme de classification sur les données. On cherche alors, à évaluer la capacité d'un algorithme, à définir des classes dans lesquelles on retrouve des données ayant des labels identiques.

Nous présentons quelques critères d'évaluations que nous avons utilisés dans ce travail :

1.6.1 L'information mutuelle normalisée (NMI)

"the normalized mutual information" mesure la dépendance statistique entre deux variables, en effet l'information mutuelle correspond à la similitude qu'une image possède sur une autre image. Elle est définie à partir des valeurs d'entropie des images. Une condition préalable pour calculer cette mesure est d'attribuer à chaque classe du résultat une classe de référence où label Les valeurs de NMI sont comprise entre 0 et 1, plus les valeurs sont grande plus les résultats sont meilleurs. Le critère NMI est donné par la formule suivantes [59] :

$$NMI = \frac{\sum_{i=1}^c \sum_{j=1}^c N_{ij} \log((N \cdot N_{ij}) / (N_i N_j))}{\sqrt{\sum_{i=1}^c N_i \log(N_i / N) \cdot \sum_{j=1}^c N_j \log(N_j / N)}} \quad (1.15)$$

Où

N : taille des données.

N_i : le nombre d'éléments dans la classe i résultant de la classification.

N_{ij} : le nombre de correspondances entre la classe i et la classe j donnée par la base.

C : le nombre de classes.

Les valeurs de NMI sont comprise entre 0 et 1, plus les valeurs sont grande plus les résultats sont meilleurs.

1.6.2 L'indice de pourcentage (IP)

Ce paramètre est le taux de bonne classification. Il est basé sur le nombre des éléments appartenant à chaque classe selon la formule suivante :

$$IP = \sum_{i=1}^c \left(\frac{Nt_i}{Nt_i + Nf_i} \right) 100 \quad (1.16)$$

où

Nt_i : le nombre des objets bien classés dans une classe i

Nf_i : le nombre des objets mal classés dans une classe i

1.6.3 La déviation des centres (CD)

Cet indice [60] est utilisé pour évaluer la qualité des centres des classes pour chaque méthode.

$$CD = \frac{1}{c} \sum_{i=1}^c \|v_i - tv_i\|_2^2 \quad (1.17)$$

Où tv : les vrai centres données par les bases.

1.6.4 L'indice de Rand (RI)

L'indice de Rand indique la proportion de paires d'observations pour lesquelles deux partitions sont en accord. Il prend ses valeurs dans l'intervalle $[0, 1]$ et il est défini de la manière suivante [61] :

$$RI = \frac{f_{00} + f_{11}}{N(N-1)/2} \quad (1.18)$$

Où f_{00} : le nombre des paires d'observation qui n'ont pas les mêmes étiquettes et qui sont associés aux différentes classes.

f_{11} : le nombre des paires d'observation qui ont les mêmes étiquettes et qui appartiennent aux mêmes classes.

N : la taille des données.

1.6.5 La validation croisée

La validation croisée[62] est une méthode très populaire pour estimer la performance d'une méthode. Elle peut être utilisée pour choisir de façon optimale un algorithme d'apprentissage plutôt qu'un autre, ou pour comparer statistiquement deux algorithmes.

Pour minimiser l'influence du choix du partitionnement de l'ensemble des exemples, la validation croisée subdivise l'ensemble d'entraînement initial en k sous ensemble disjoints D_1, D_2, \dots, D_k de même taille. L'entraînement et le test sont effectués k fois. A l'itération i le sous-ensemble D_i est réservé pour le test et le reste des exemples sont utilisés pour entraîner le modèle. La précision finale du modèle est égale à la moyenne des k précisions de test. Deux grandes techniques de validation croisée existent dans la littérature : « k -fold cross-validation » et « leave-one-out cross-validation » (LOOCV).

La validation croisée par la méthode k -fold :

Elle consiste à diviser l'échantillon original en k échantillons. Puis on sélectionne un des k échantillons comme ensemble de teste et les $(k-1)$ autres échantillons constitueront l'ensemble d'apprentissage. Le taux de la classification est calculé puis l'opération est répétée en sélectionnant un autre échantillon de teste parmi les $(k-1)$ échantillons qui n'ont pas encore été utilisés pour la validation du modèle. L'opération est répétée ainsi k fois pour qu'en fin de compte chaque sous-échantillon ait été utilisé exactement une fois comme ensemble de validation. La moyenne des k erreurs quadratiques moyennes est enfin calculée pour estimer l'erreur de prédiction.

La méthode Leave-One-Out :

Cet est un cas particulier du K -fold où K est égal au nombre des exemples N . A chaque itération, le modèle est entraîné par les $N - 1$ exemples et testé sur l'exemple exclu de l'entraînement. On obtient à la fin N précisions. La précision du modèle est égale à leur moyenne.

1.7 Conclusion

Dans ce chapitre, nous avons exposé les différentes méthodes de classification dans les sous espaces. En premier lieu quelques algorithmes de classification de base ont été présentés. Ces algorithmes sont constitués le noyau de plusieurs travaux actuels. Ces méthodes classiques ont l'inconvénient majeur d'être incapable de gérer les données de grande dimension. Elles remédient à quelques inconvénients avec une perte d'information. Les résultats sont très sensibles à l'initialisation des paramètres par l'utilisateur.

Nous avons présenté ensuite les méthodes de classification dans les sous espaces qui permettant de surmonter les limites des méthodes classiques faisant face aux données de grande dimension. Deux grandes familles de classification dure et floue ont été présentées. L'intérêt de ces méthodes est important car elles permettent de faire face à la malédiction de la dimensionnalité. De plus, elles permettent de fournir une description réduite des clusters obtenus car ceux-ci sont alors définis par un nombre restreint de dimensions.

Chapitre 2

Les séparateurs à vaste marge (SVM)

2.1 Introduction

Les machines à vecteur support souvent traduit par l'appellation de Séparateur à Vaste Marge (SVM), sont des algorithmes de classification binaire par apprentissage supervisé développés par Vapnik [30]. Ils permettent d'aborder des problèmes très divers dont la classification. SVM est une méthode particulièrement bien adaptée pour traiter des données de très haute dimension telles que les textes et les images. Depuis leur introduction dans le domaine de la reconnaissance de formes, plusieurs travaux ont pu montrer l'efficacité de ces techniques principalement en traitement d'image [63].

Cette technique est une méthode de classification à deux classes qui tente de séparer linéairement les exemples positifs des exemples négatifs dans l'ensemble des exemples. Chaque exemple doit être représenté par un vecteur de dimension n .

Le principe de l'algorithme SVM est la recherche de l'hyperplan qui sépare les exemples positifs des exemples négatifs, en garantissant que la marge entre le plus proche des positifs et des négatifs soit maximale. Intuitivement, cela garantit un bon niveau de généralisation car de nouveaux exemples pourront ne pas être trop similaires à ceux utilisés pour trouver l'hyperplan mais être tout de même situés franchement d'un côté ou l'autre de la frontière.[64]

Le problème de la classification rentre dans le cadre de l'apprentissage. La notion d'appren-

tissage inclut toute méthode permettant de produire une fonction, des règles générales, ou plus généralement un modèle qui représente les informations présentes dans les données d'apprentissage.

Premièrement, nous présentons une généralité sur la théorie d'apprentissage avant de parler des concepts de la méthode SVM.

2.2 La théorie d'apprentissage

La théorie d'apprentissage statistique étudie les propriétés mathématiques des machines d'apprentissage, qui représentent les propriétés de la classe de fonctions ou modèles que peut implémenter la machine. L'apprentissage statistique utilise un nombre limité d'entrées (appelées exemples) d'un système avec les valeurs de leurs sorties pour apprendre une fonction qui décrit la relation fonctionnelle existante, mais non connue, entre les entrées et les sorties du système [65].

Une machine d'apprentissage permet d'implémenter une famille de fonctions $f_\alpha(x)$, $\alpha \in A$ ou A est un ensemble de paramètres. Ces fonctions doivent produire pour chaque vecteur d'entrée x une sortie \hat{y} la plus proche possible de la sortie y du superviseur qui associe pour chaque vecteur d'entrée x une sortie y (la classe) suivant une distribution de probabilité également inconnue $P(x, y)$.

Il existe plusieurs types d'apprentissage automatique pour la classification : supervisé, non-supervisé, semi supervisé et actif.

2.2.1 L'apprentissage supervisé

L'apprentissage supervisé nécessite une base d'apprentissage où chaque instance est préalablement étiquetée avec sa classe respective.

Soit O un ensemble d'individus décrit par un nombre fixe n de caractéristiques. Prenons un sous-ensemble S de O . Étant donnée les paramètres d'un objet de O/S on veut pouvoir estimer le mieux possible la classe de cet objet (-1 ou +1) avec la seule connaissance

de S . S est l'ensemble d'apprentissage (training set) et O/S est l'ensemble de test. On représente S par un ensemble de m couples $(x_i, y_i)_{1 \leq i \leq m}$ où x_i est un point de R^n qui représente les paramètres et $y_i = 1$ représente la classe de x_i . Les $(x_i, y_i)_{1 \leq i \leq m}$ sont issus d'une distribution de probabilité inconnue $P(x, y)$, les données sont supposées i.i.d (indépendamment identiquement distribuées). Le but de l'apprentissage est de trouver une estimation de l'application $x_i \rightarrow y_i$. Il existe principalement deux familles de méthodes d'apprentissage supervisé pour la classification ; les méthodes discriminatives et les méthodes génératives. Les méthodes discriminatives sont celles qui sont les plus utilisées et incluent, entre autres : SVM, KNN (K nearest neighbor) [66,67].

2.2.2 L'apprentissage non supervisée

L'apprentissage non supervisé consiste à tirer de la valeur de données dans lesquelles l'attribut à prédire n'apparaît pas à l'opposé de l'apprentissage supervisé, en apprentissage non supervisée, il n'y a pas de sortie, et il s'agit alors de construire un modèle permettant de représenter au mieux les observations X_1, \dots, X_n , de manière à la fois précise et compacte.

2.2.3 L'apprentissage semi supervisé

Dans l'apprentissage semi-supervisé, l'idée de base est de réduire le coût d'étiquetage en utilisant peu de données étiquetées et en exploitant les données non étiquetées. À titre d'exemple, une méthode d'apprentissage semi-supervisée très basique consiste à apprendre un modèle de classification à partir des données étiquetées et prédire les classes des données non étiquetées. Les données non étiquetées les plus sûres (pour lesquelles le classifieur est le moins incertain), avec leurs étiquettes prédites, sont alors ajoutées à la base d'apprentissage pour apprendre un nouveau modèle de classification, et ainsi de suite.

L'hypothèse faite par l'apprentissage semi-supervisé est que la distribution des données non étiquetées $p(x)$ partage des caractéristiques en commun avec la distribution des don-

nées dans leurs classes respectives $p(x|y)$. Ainsi, l'apprentissage semi-supervisé peut aider à améliorer les résultats d'apprentissage si cette hypothèse est vraie, mais peut conduire à une dégradation des performances du classifieur appris dans le cas contraire.

2.2.4 L'apprentissage actif

L'apprentissage est dit actif si l'apprenant peut influencer le choix des exemples d'apprentissage. L'apprentissage actif permet de demander à un opérateur l'étiquetage de quelques données "importantes" sélectionnées de manière pertinente, ce qui réduit considérablement le coût d'étiquetage. Le principal problème à résoudre pour un apprentissage actif concerne la façon de sélectionner les données à étiqueter. Il est une extension de l'apprentissage semi-supervisé, en permettant de choisir les données à étiqueter. Le choix des instances à étiqueter pour l'apprentissage, peut influencer considérablement la qualité du classifieur appris. En effet, les étiquettes de classes associées aux instances, dirigent le système vers la règle de classification désirée. Il est donc important d'étiqueter des instances informatives qui permettent de converger plus rapidement vers cette règle de classification. Par exemple, le gain d'étiqueter manuellement une instance qui est très bien classée par le classifieur (avec une grande confiance) est faible, comparé à d'autres instances plus informatives (pour lesquelles le classifieur est incertain).

L'apprentissage actif consiste ainsi à se demander quelles seraient les instances les plus informatives à présenter à l'opérateur, afin de les utiliser comme données étiquetées pour l'apprentissage.[68]

Il existe principalement deux cas d'apprentissage actif; un apprentissage actif statique consiste à choisir pour étiquetage l'instance (parmi toutes les instances) qui maximise une mesure d'informativité, tandis que celui séquentiel consiste à demander l'étiquetage d'une instance dont l'informativité est supérieure à un seuil [69].

Il existe différentes stratégies pour mesurer l'informativité des instances et donc pour demander ou non leurs étiquettes de classes à un opérateur.

Les stratégies présentées dans cette section sont utilisables telles quelles dans le cas d'un apprentissage actif statique mais nécessitent d'introduire un seuil dans le cas séquentiel.

Stratégies basées sur l'incertitude

Les stratégies les plus simples et les plus courantes sont celles basées sur l'incertitude. Avec ces stratégies, l'apprentissage actif demande l'étiquetage des instances pour lesquelles il est le moins certain (ou le plus incertain) de leur classe. Un des avantages des stratégies basées sur l'incertitude est qu'elles sont faciles à mettre en œuvre et peuvent être utilisées avec n'importe quel classifieur. Elles permettent aussi de réduire le coût d'étiquetage.[69]

Stratégies basées sur un ensemble de modèles

L'approche consiste à maintenir un ensemble de C modèles du même type (issus d'un même classifieur) qui sont tous entraînés sur un ensemble d'instances étiquetées L , mais représentent des modèles concurrents. Chaque modèle de l'ensemble est alors autorisé à voter sur l'étiquetage d'instances candidates. L'instance la plus informative est considérée comme l'instance pour laquelle les modèles sont le plus en désaccord.[70]

L'apprentissage actif peut alors être vu comme la recherche du "meilleur" modèle dans l'espace de versions. Dans ce cas, le but de l'apprentissage actif est de réduire la taille de cet espace autant que possible, de sorte que la recherche devienne plus précise, en utilisant le moins possible d'instances étiquetées. Contrairement aux stratégies basées sur l'incertitude, cette stratégie n'est pas indépendante du type de modèle utilisé.

Autres stratégies

D'autres stratégies qui sont différentes des précédentes, sont appelées « Exploitation de la structure des clusters » elles consistent à partitionner l'ensemble des données non étiquetées U (qui est supposé être disponible à l'avance) en clusters et à exploiter la structure de ces clusters pour un apprentissage actif. L'idée est de demander l'étiquetage de quelques instances de chaque cluster, d'attribuer à chaque cluster l'étiquette majoritaire, puis d'utiliser l'ensemble de ces données étiquetées pour l'apprentissage .en général, la

structure des clusters n'est pas aussi clairement définie, et on peut obtenir des résultats médiocres lorsque les clusters ne coïncident pas du tout avec les limites de décision des vraies classes. La maximisation de l'espérance de changement du modèle est une autre stratégie moins courante consiste à sélectionner les instances qui auraient une grande influence sur le modèle courant si elles sont étiquetées. Cette influence est estimée par le degré de changement du modèle avant et après l'introduction de l'instance qui est candidate pour l'étiquetage. Cette stratégie reste aussi coûteuse et dépend fortement du type de modèle utilisé [71-72].

2.3 les séparateurs à vaste marge (SVM)

2.3.1 Cas linéairement séparable

Un problème de classification est dit linéairement séparable lorsqu'il existe une fonction de décision linéaire (appelé aussi séparateur linéaire) qui permet de classifier les données ou classes distinct, voici un exemple d'apprentissage, $X = [x_1, x_2, \dots, x_n]$ le nuage d'observation dans l'espace d'attributs correspondant et $Y = [y_1, y_2, \dots, y_n]$ l'ensemble de leurs étiquettes qui appartiennent toutes à l'ensemble $\{-1, 1\}$. En supposant qu'il existe un hyperplan permettant de séparer les exemples positifs des exemples négatifs qu'on peut le représenter par l'équation suivante :

$$H(x) = w^T x + b \quad (2.1)$$

Tel que

$$w_i x_i + b \geq 1 \quad \text{Si } y_i = 1 \quad (2.2)$$

$$w_i x_i + b \leq -1 \quad \text{Si } y_i = -1 \quad (2.3)$$

Où w est un vecteur de m dimensions et b est un terme.

En divisant les deux parties des inégalités par y , on trouve les inégalités précédentes qui sont équivalentes à l'équation :

$$y_i(w_i x_i + b) \leq +1 \quad (2.4)$$

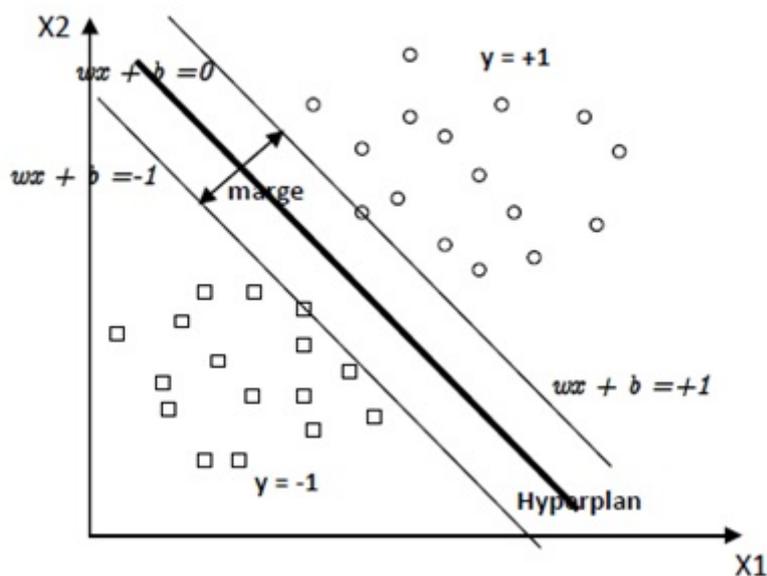


FIGURE 2.1 – Exemple graphique des données linéairement séparables.

La distance entre cet hyperplan et l'exemple le plus proche s'appelle la marge (figure 2.1). La région qui se trouve entre les deux hyperplans $wx + b = -1$ et $wx + b = +1$ est appelée la région de généralisation de la machine d'apprentissage. Les données qui se trouvent sur les deux droites $+1$ et -1 représentent les vecteurs supports (figure 2.1).

La distance perpendiculaire de l'origine à l'hyperplan :

$$H_1 : w.x + b = +1 \quad \text{est} \quad \frac{|1 - b|}{\|w\|}$$

De même pour

$$H_2 : w.x + b = -1 \quad \text{est} \quad \frac{|1 + b|}{\|w\|}$$

La distance entre un point situé sur H_1 et l'hyperplan H_0 est donnée par :

$$\frac{|wx + b|}{\|w\|} = \frac{1}{\|w\|}$$

Donc la marge (la distance entre les deux hyperplans H_1 et H_2 est $\frac{2}{\|w\|}$

Dans le cas linéairement séparable, on peut trouver plus d'un seul hyperplan, donc nous allons chercher parmi ceux-ci celui qui passe aux milieux des points à séparer. Cela revient à chercher l'hyperplan optimal passe par la détermination de la distance euclidienne mini-

male entre l'hyperplan et l'exemple le plus proche des deux classes ($\|w\|$) ou de maximiser la marge ($\frac{1}{\|w\|}$), soit à résoudre le problème suivant :

$$\begin{cases} \text{minimiser} & \frac{1}{2}\|w\|^2 \\ \text{sous contrainte} & Y(W.X + b) \geq 1 \end{cases} \quad (2.5)$$

Le problème de l'équation (2.5) est un problème de programmation quadratique avec contraintes Linéaires appelée formulation primale. Dans ce problème, les variables sont w et b , c-à-d que le nombre de variables est égal à $d + 1$ ou d est la dimension de x . Généralement, le nombre de variables est important ce qui ne permet pas d'utiliser les techniques classiques de programmation quadratique. Donc, il existe une transformation de ce problème dans une formulation duale. Après transformation, le problème équivalent sans contraintes de l'équation Eq.(2.6) qui introduit les multiplicateurs de Lagrange :

$$Q(w, b, \alpha) = \frac{1}{2}w^T w - \sum_{i=1}^n (y_i(w^T x_i + b) - 1) \quad (2.6)$$

Où les α_i sont les multiplicateurs non négatifs de Lagrange. L'optimum de la fonction objective Q peut être obtenu en la minimisant par rapport à w et b et en la maximisant par rapport aux α_i . À l'optimum de la fonction objective, ses dérivées par rapports aux variables w et b s'annulent ainsi que le produit des α_i aux contraintes, donc on obtient le problème dual à maximiser suivant :

$$\begin{cases} \text{maximiser} & Q(\alpha) = \sum_{i=1}^n \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{j=1}^n \alpha_i \alpha_j y_i y_j x_i^T x_j \\ \text{sous contrainte} & \sum_{i=1}^n \alpha_i y_i = 0 \\ & \alpha_i \geq 0 \end{cases} \quad (2.7)$$

L'hyperplan solution correspondant peut alors s'écrire :

$$H(x) = \sum_S \alpha_i y_i x^T x_i + b \quad (2.8)$$

Où S représente l'ensemble des vecteurs supports. b peut être calculé à partir de n'importe quel vecteur support par l'équation suivante :

$$b = y_i - w^T x_i \quad (2.9)$$

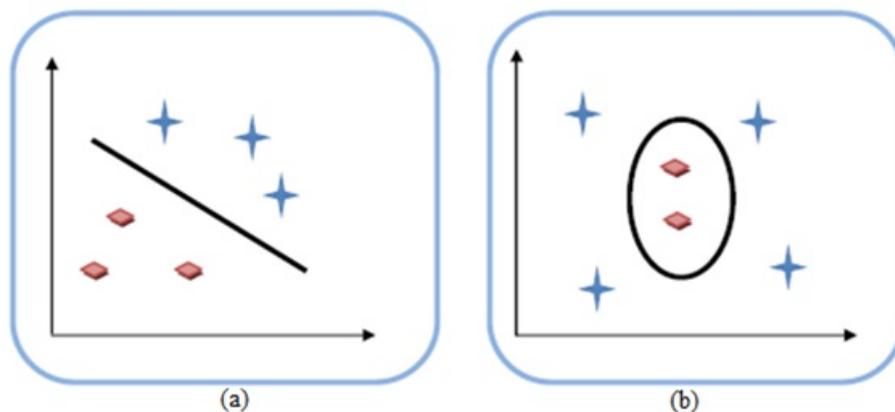


FIGURE 2.2 – (a) Cas des données linéairement séparables, (b) cas des données linéairement non séparables.

D'un point de vue précision, on prend la moyenne de b pour tous les vecteurs supports :

$$b = \frac{1}{|S|} \sum_{i \in S} y_i - w^T x_i \quad (2.10)$$

La fonction de décision H peut être calculée, donc, pour chaque nouvel exemple x par l'équation (2.8) et la décision peut être prise comme suit :

$$\begin{cases} x \in \text{Classe} + 1 & \text{si } H(x) > 0 \\ x \in \text{Classe} - 1 & \text{si } H(x) < 0 \\ x \text{ inclassable} & \text{si } H(x) = 0 \end{cases} \quad (2.11)$$

2.3.2 Cas linéairement non séparable

Dans la plupart des problèmes réels il n'y a pas de séparation linéaire possible entre les données, le classificateur de marge maximale ne peut pas être utilisé car il fonctionne seulement si les classes de données d'apprentissage sont linéairement séparables. Pour surmonter les inconvénients des cas non linéairement séparable, l'idée des SVM est de changer l'espace des données. La transformation non linéaire des données peut permettre une séparation linéaire des exemples dans un nouvel espace. On va donc avoir un changement de dimension. Cette transformation non linéaire est réalisée via une fonction noyau. En pratique, quelques familles de fonctions noyau paramétrables sont connues et il revient à l'utilisateur de SVM d'effectuer des tests pour déterminer celle qui convient le mieux pour

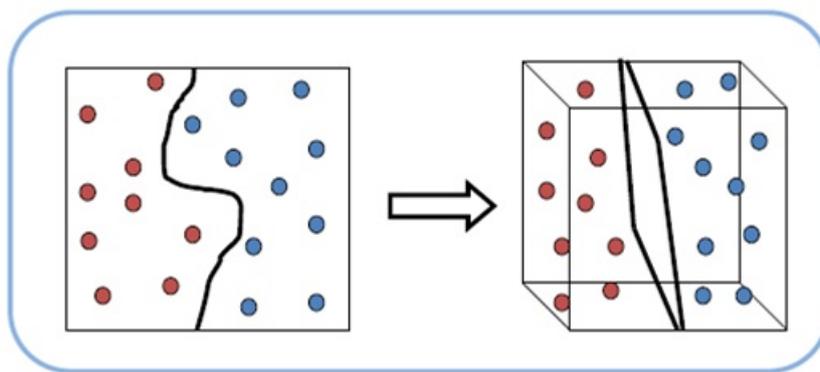


FIGURE 2.3 – Transformation d’espace.

TABLE 2.1 – Fonctions Noyaux (Kernel function)

Type noyau	Expression analytique
Noyau linéaire	$K(x, z) = xz$
Noyau polynomial	$K(x, z) = (a * \langle x, z \rangle + b)^d$
Noyau Gaussien (Radial Basis functions RBF)	$K(x, z) = e^{-\frac{\ x-z\ ^2}{2\sigma^2}}$

son application. Dans ce nouvel espace de caractéristiques, la fonction objective devient :

$$Q(\alpha) = \sum_{i=1}^n \sum_{j=1}^n \alpha_i - \frac{1}{2} \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \langle \phi(x_i), \phi(x_j) \rangle \quad (2.12)$$

Où $\phi(x_i), \phi(x_j)$ est le produit scalaire des deux images des vecteurs x_i et x_j dans le nouvel espace et dont le résultat est un scalaire. Il existe des fonctions positives $K(x, y)$, appelées fonctions noyaux, faciles à calculer et dont on peut montrer qu’elles correspondent à un produit scalaire dans un espace de grande dimension. Cette fonction permet de surmonter le problème de la détermination de la transformation ϕ et permet d’apprendre des relations non linéaires par des machines linéaires.

Une famille de ces fonctions noyaux qui sont très appropriées aux besoins des SVM sont présentées par le tableau suivant : La fonction objective peut être calculée comme suit :

$$Q(\alpha) = \sum_{i=1}^n \sum_{j=1}^n \alpha_i - \frac{1}{2} \sum_{j=1}^n \alpha_i \alpha_j y_i y_j K(x_i, x_j) \quad (2.13)$$

Et la fonction de décision H devient :

$$H(x) = \sum_S \alpha_i y_i K(x_i, x) + b \quad (2.14)$$

Où S représente l'ensemble des vecteurs supports.

L'algorithme (2.1) résume les étapes du SVM :

2.4 SVM multi-classes

Les machines à vecteur support sont dans leur origine binaire. Pour étendre cette méthode et l'appliquer à un problème multi-classes il existe des approches qui permettent de considérer le SVM multi-classes comme un seul problème d'optimisation, ou le décomposer en plusieurs sous problèmes binaire [73]. On trouve dans la littérature plusieurs méthodes de décomposition :

2.4.1 Un contre tous (One versus All)

C'est la méthode la plus simple et la plus ancienne. Elle consiste à entraîner un SVM bi-classes en utilisant les éléments d'une classe contre tous les autres. Donc pour un problème à K classes, on a K SVM binaires [74].

2.4.2 Un contre un (One versus One)

L'approche un contre un : consiste à utiliser un classifieur pour chaque paire de classes. Au lieu d'apprendre K fonctions de décisions, la méthode 1vs1 discrimine chaque classe de chaque autre classe, ainsi $K(K-1)/2$ fonctions de décisions sont apprises. La décision de la classe gagnante soit par un vote majoritaire soit en post traite les résultats grâce à l'estimation de probabilités a posteriori [75].

2.5 Conclusion

Dans ce chapitre, nous avons donné une généralité sur les séparateurs à vaste marge, dont nous avons présenté le principe de SVM binaire dans les cas des données linéairement séparables et non linéairement séparables. Nous avons ensuite décrit la démarche suivie

Algorithme 2.1 SVM**Entrée(s) :**

1. Une matrice $X \in R^{n \times d}$ de n exemples d'apprentissage.
2. Un vecteur $Y \in \{+1, -1\}^n$ des étiquettes de classe correspondants.
3. Un vecteur $C \in R_{\geq 0}^n$ des facteurs de pénalité :
($x_i \in X$ est pénalisé par le facteur $c_i \in C$ si mal classée).
4. Un fonction handle K d'un noyau tel que :
 $K : R^d \times R^d \rightarrow R, (x_i, x_j) \rightarrow \phi(x_i) * \phi(x_j)$

Sortie(s) :

1. Une fonction handle H du classifier.
2. Une matrice $S_X \subset R^{s \times d}$ de s vecteurs de support ($S_X \subseteq X$).
3. Un vecteur $S_Y \subset \{+1, -1\}^s$ des étiquettes correspondantes ($S_Y \subseteq Y$).
4. Un vecteur $S_\alpha \subset R_{\geq 0}$ de multiplicateurs de Lagrange non négatifs ($x_i \in S_X$ est multiplié par $\alpha_i \in S_\alpha$).

Traitement

1. Calculer la matrice symétrique $H1 \in R^{n \times n} : H1_{ij} = y_i y_j k(x_i, x_j)$
2. Utilise l'optimisation quadratique(QP) pour :

$$\left\{ \begin{array}{l} \text{Minimise} \quad 1/2 \alpha^T H1 \alpha - \sum_{i=1}^n \alpha_i \\ \text{sous condition} \quad 0 \leq \alpha_i \leq c_i \\ \sum_{i=1}^n \alpha y_i = 0 \end{array} \right.$$
3. Détermine ensemble des indices de vecteurs de support S_i par :
 $S_i = \{i | x_i \in X \wedge 0 < \alpha_i < c_i\}$
4. Détermine les matrices S_X , S_Y , and S_α sachant que :
 $(x_i \in S_X \wedge y_i \in S_Y \wedge \alpha_i \in S_\alpha) \Leftrightarrow (x_i \in X \wedge i \in S_i)$
5. Calcule b par : $b = \frac{1}{|S_i|} \sum_{i \in S_i} (y_i - \sum_{j \in S_i} (\alpha_j y_j K(x_j, x_i)))$
6. Retourne la fonction H par : $H(x) = \sum_{j \in S_i} \alpha_j y_j K(x_j, x) + b$

pour généraliser cette technique pour le cas de multiclassés à partir du cas binaire en utilisant deux méthodes ; Un contre tous (One versus All) et Un contre un (One versus One). L'intérêt principal de la méthode SVM est sa facilité d'emploi, Il est très efficace sur les données à nombre élevé d'attributs. Les capacités de généralisation et la simplicité d'entraînement des SVMs sont bien au-delà des autres méthodes. Parmi d'autres avantages des SVMs on cite : qu'ils possèdent des fondements théoriques solides, une optimisation très simple, et un temps de calcul court. La qualité d'un modèle de décision obtenu par la méthode SVM dépend de plusieurs paramètres, à savoir le paramètre de pénalisation sur les multiplicateurs de Lagrange C, le noyau utilisé et ses paramètres. L'apprentissage actif force l'algorithme SVM à se restreindre aux données qui apportent plus d'information (les plus intéressantes). Cette option peut être appliquée à tous les modèles SVMs (classification, régression et « one-class ») et peut être désactivée.

Chapitre 3

Classification des textures par le modèle 2D-RCA

3.1 Introduction

La texture est une des plus importantes caractéristiques utilisées pour identifier les différentes régions d'une image. Son étude est difficile, car il n'existe pas de définition universelle permettant d'appréhender la notion de texture. Néanmoins, de nombreuses méthodes, utilisant des modèles statistiques ou structurelles, ont été développées pour caractériser les textures.

Dans le traitement d'images, l'analyse de textures joue un rôle très important. Elle permet l'identification des objets. En effet, il est inévitable pour distinguer les différentes régions d'une image de même couleur, d'extraire une information texturale. Il est difficile de caractériser précisément la texture. La notion de texture ne peut être précisément définie [76]; elle sert à qualifier mathématiquement des termes de langage utilisés pour quantifier les images, telles que : la régularité, l'homogénéité, contraste etc...Par exemple dans le cadre des images médicales, ce genre d'analyse peut être utilisée pour diverses raisons : Distinction entre des zones saines et pathologiques, aide au diagnostic sur l'évolution d'une pathologie et Caractérisation de zones informatives en vue d'opération de fusion. La texture a suscité de nombreuses recherches et travaux d'application [77,78] qui

ont engendré une multitude de méthodes d'analyse [78-82]. Ces méthodes ont pour objet de caractériser de décrire, de discriminer, de segmenter ou de synthétiser les textures.

La classification de texture sélectionne une classe de texture (prédéfinie) pertinente pour chaque région de texture de l'image. Pour résoudre ce problème, nous pouvons le diviser en sous-problèmes comme l'extraction de caractéristiques de texture, l'extraction de primitives et le partitionnement de données. Pour ces deux opérations, nous devons réaliser la phase commune : extraction de caractéristiques de texture.

3.2 Extraction de caractéristiques de textures

On peut partager les méthodes d'extraction de texture en quatre grandes familles : analyse statistique, analyse par Ondelettes, Filtres et analyse par modèle. Dans ce chapitre nous nous intéressons à l'analyse par modèle. Nous présentons le modèle autorégressif à coefficients aléatoires (RCA) standard et sa version bidimensionnelle, le modèle 2D-RCA [83] pour la classification des textures.

3.2.1 Analyse statistique

Les méthodes statistiques étudient les relations entre un pixel et ses voisins. Elles sont particulièrement adaptées à l'étude des textures stochastiques. L'ordre de la statistique est donné par le nombre de pixels voisins mis en jeu dans le calcul des paramètres. Il est utile de préciser que si certains paramètres correspondent à des propriétés visuelles de la texture, certaines autres ne reflètent que des propriétés mathématiques auxquelles il n'y a pas forcément de propriété visuelle associée [78].

Paramètres de 1^{er} ordre

Les paramètres texturaux du premier ordre sont calculés à partir de l'histogramme. L'histogramme représente le nombre de pixels présents dans la région d'intérêt RI étudiée pour un niveau de gris donné [78].

- La moyenne donne la valeur moyenne des niveaux de gris appartenant à la RI.

$$Moy = \frac{1}{K} \sum_{i=0}^{N-1} i * p(i) \quad (3.1)$$

avec :

$p(i), i \in [0..N - 1]$ histogramme de la RI considérée.

N : Nombre total de niveaux de gris,

K : Nombre total de pixels de la RI.

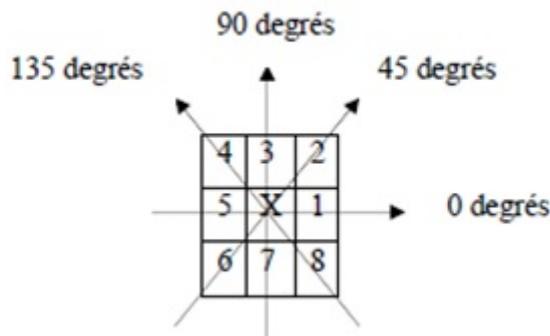
Paramètres d'ordres supérieurs

Les méthodes du premier ordre ne procurent aucune information sur l'organisation locale des pixels. Des méthodes d'ordre supérieur sont donc utilisées afin de faire une analyse plus précise.

Les matrices de co-occurrence

La matrice de co-occurrence est largement utilisée dans l'analyse de texture [79]. Du fait de sa richesse en information de texture, elle est devenue la plus connue et la plus utilisée pour extraire ces caractéristiques de textures. Elle estime des propriétés des images relatives à des statistiques de second ordre. Une matrice de co-occurrence est basée sur le calcul des fréquences non normalisées $P(i, j, \delta, \theta)$ qui représente le nombre de fois où un pixel de niveau de couleur i apparaît à une distance relative δ d'un pixel de niveau de couleur j et selon une orientation θ donnée.

Les directions angulaires θ classiquement utilisées sont 0, 45, 90 et 135 degrés. Les relations de voisinage entre pixels, nécessaires au calcul des matrices, sont illustrées en figure 3.1 ; par exemple, les plus proches voisins de x selon la direction $\theta = 135$ degrés sont les pixels 4 et 8. Les caractéristiques extraites à partir de ces matrices contiennent des informations notamment sur l'homogénéité, les dépendances linéaires entre les niveaux de gris, le contraste et la complexité de cette image. Les matrices obtenues selon ces quatre directions sont alors calculées par les équations Eq.(3.2), (3.3), (3.4) et (3.5) où (k, l) sont

FIGURE 3.1 – Plus proches voisins du pixel x selon 4 directions.

les coordonnées d'un pixel de niveau de couleur $i \in [0, n_{max} - 1]$ et (m, n) celles du pixel de niveau de couleur $j \in [0, n_{max} - 1]$.

$$P(i, j, \delta, 0) = \text{card}\{((k, l), (m, n)) \in (N \times M)^2 \text{ tel que} \\ (k - m = 0, |l - n| = \delta, I_{k,l} = i, I_{m,n} = j)\} \quad (3.2)$$

$$P(i, j, \delta, 45) = \text{card}\{((k, l), (m, n)) \in (N \times M)^2 \text{ tel que} \\ (k - m = \delta, l - n = -\delta) \vee (k - m = -\delta, l - n = \delta), \\ I_{k,l} = i, I_{m,n} = j)\} \quad (3.3)$$

$$P(i, j, \delta, 90) = \text{card}\{((k, l), (m, n)) \in (N \times M)^2 \text{ tel que} \\ (|k - m| = \delta, l - n = 0, I_{k,l} = i, I_{m,n} = j)\} \quad (3.4)$$

$$P(i, j, \delta, 135) = \text{card}\{((k, l), (m, n)) \in (N \times M)^2 \text{ tel que} \\ (k - m = \delta, l - n = \delta) \vee (k - m = -\delta, l - n = -\delta), \\ I_{k,l} = i, I_{m,n} = j)\} \quad (3.5)$$

La plupart des images sont codées sur 256 niveaux de gris, par conséquent, la taille des matrices de co-occurrence est de 256×256 . On s'aperçoit ainsi que ces matrices comptabilisent une très grosse quantité d'informations difficile à exploiter directement. C'est pourquoi, un certain nombre d'auteurs comme Zucker [84] ont essayé d'extraire de

l'information de ces matrices afin de mettre en évidence la structure des textures. Mais c'est Haralick et al dans [79] qui ont proposé 14 paramètres, caractérisant les textures, issus de ces matrices. Les six paramètres les plus utilisés est considérés les plus pertinents sont :

1. **L'énergie :**

$$ENE = \sum_i \sum_j (P_{ij}(\delta, \theta)^2) \quad (3.6)$$

où $P_{ij}(\delta, \theta)$ est la fréquence relative avec la quelle deux voisins séparés par une distance δ dans une direction θ apparaissent dans l'image.

Ce paramètre mesure l'uniformité de la texture. Il atteint de fortes valeurs lorsque la distribution des niveaux de gris est constante ou de forme périodique.

2. **Le contraste :**

$$CST = \sum_i \sum_j ((i - j)^2 P_{ij}(\delta, \theta)) \quad (3.7)$$

La valeur en est d'autant plus élevée que la texture présente un fort contraste. Ce paramètre est fortement non corrélé à l'énergie.

3. **L'entropie :**

$$ENT = \sum_i \sum_j (\log P_{ij}(\delta, \theta) P_{ij}(\delta, \theta)) \quad (3.8)$$

Ce paramètre mesure le désordre dans l'image. Contrairement à l'énergie, l'entropie atteint de fortes valeurs lorsque la texture est complètement aléatoire (sans structure apparente). Elle est fortement corrélée (par l'inverse) à l'énergie.

4. **La variance :**

$$Var = \sum_i \sum_j ((i - \mu)^2 P_{ij}(\delta, \theta)) \quad (3.9)$$

La variance mesure l'hétérogénéité de la texture. Elle augmente lorsque les niveaux de gris sont différents de leur moyenne. La variance est indépendante du contraste.

5. **La corrélation :**

$$COR = \sum_i \sum_j \left(\frac{(i - \mu)(j - \mu) P_{ij}(\delta, \theta)}{\delta^2} \right) \quad (3.10)$$

COR mesure la dépendance linéaire (relativement à (δ, θ)) des niveaux de gris de l'image. La corrélation n'est corrélée ni à l'énergie, ni à l'entropie.

6. Le moment inverse :

$$IDM = \sum_i \sum_j \left(\frac{P_{ij}(\delta, \theta)}{1 + (i - j)^2} \right) \quad (3.11)$$

IDM (Inverse Difference Moment) mesure l'homogénéité de l'image. Ce paramètre est corrélé à une combinaison linéaire des variables ENE (entropie) et CST (contraste). De nombreuses études ont été menées afin de caractériser, classifier, modéliser les textures à l'aide de ces paramètres ([78,79]).

La méthode d'extraction de ces paramètres basée sur le calcul des matrices de co-occurrence est une des méthodes les plus proches de la notion de texture. Elles mettent effectivement en avant les relations qui existent entre les pixels de l'image en faisant intervenir l'aspect local (les niveaux de gris) et l'aspect spatial (δ, θ) .

Dans la plupart des études, les auteurs calculent ces matrices de co-occurrence sur les images brutes ou filtrées, donc toujours à partir des niveaux de gris. Dans [85] les auteurs ont généralisé cette utilisation à tous les types d'information (incluant les niveaux de gris). Ils ont transformé les images en codant certains pixels par une autre information que celle des niveaux de gris. Ensuite, ils ont calculé les matrices de co-occurrence à partir de ces nouvelles images codées.

3.2.2 Analyse par ondelettes

La décomposition en ondelettes (DWT : Discrete Wavelet Transform) d'une image permet une analyse multi-résolution. Ce modèle est justifié [86] en effectuant un parallèle avec le système visuel humain (SVH). Celui-ci procède à une analyse multi-échelle : une scène est d'abord perçue globalement puis dans les détails. Cette analyse peut être exploitée dans le cadre de l'analyse de texture, via divers paramètres.

3.2.3 Filtrage

Si certains paramètres caractéristiques d'une texture sont récupérés directement sur les niveaux de gris de l'image, on passe souvent par un filtrage de l'image qui permet de se focaliser sur certaines caractéristiques importantes de l'image ; nous pouvons trouver des exemples qui utilisent des filtres simples comme le filtre de Roberts[87], le filtre laplacien [88] ou les filtres plus élaborés comme le filtre de Fourier ou celui de Gabor [89] qui permet d'avoir des informations précises à la fois en espace et en fréquence à propos de la texture.

Filtre de Fourier

L'extraction de paramètres de texture à partir de la transformée de Fourier permet d'établir un modèle compact pour les textures périodiques. Normalement, nous choisissons quatre directions (0 deg; 45 deg; 90 deg; 135 deg) et certaines fréquences. Le nombre de fréquences dépend de la taille de l'image.

Filtre de Gabor

Un filtre de Gabor est une sinusoïde modulée par une gaussienne. Dans le domaine fréquentiel, il s'exprime comme étant une gaussienne centrée sur une fréquence déterminée. En 2D, un banc de filtres de Gabor va s'exprimer comme étant un ensemble de filtres, chacun sélectionnant une fréquence particulière dans une dimension particulière. L'utilisation d'un banc de filtres de Gabor permet d'extraire de l'image considérée des informations pertinentes, à la fois en espace et en fréquence, relatives à la texture. Avec les filtres de Gabor, nous pouvons analyser la texture à différentes échelles et différentes orientations.

3.2.4 Analyse par modèles

Les méthodes basées sur les modèles supposent que la texture est modélisée par un processus stochastique. On considère que les paramètres de ce modèle sont des caracté-

ristiques de la texture. Par exemple, on peut modéliser le niveau de gris d'une image par un champ aléatoire de Markov. La probabilité qu'un pixel reçoive une intensité doit être positive et markovienne. La caractéristique markovienne émet l'hypothèse que la distribution d'intensité des pixels dépend uniquement de ses voisins. Elle est indépendante du reste de l'image. Ce modèle vise à capturer des statistiques de ces voisinages et de les représenter comme des paramètres du modèle. Parmi les modèles les plus couramment utilisés pour modéliser les propriétés texturales de chaque RI, on trouve le modèle autorégressif (AR)[90]. Les vecteurs de paramètres ainsi définis permettent d'envisager une segmentation supervisée de l'image.

Modèle autorégressif (AR)

On suppose, dans ce modèle, qu'il existe une interaction entre le niveau de gris de chaque pixel de l'image et ses voisins. Ainsi l'intensité d'un pixel dépend de celles de ses voisins. Un modèle AR d'un signal $y(s)$ considère ce dernier comme le résultat d'un filtrage linéaire d'un bruit $e(s)$. Le modèle s'écrit de la manière suivante :

$$y(s) = \sum_{l \in V} (a_l y(s-l) + e(s)) \quad (3.12)$$

s représente un site de l'image donc un ensemble de 2 coordonnées. V est le voisinage du pixel considéré. $e(s)$ est un bruit blanc de moyenne nulle et $y(s)$ est la valeur du pixel s de l'image source considérée. Les paramètres à déterminer sont donc les a_l ainsi que la variance du bruit $e(s)$.

3.3 Modèles autorégressif à coefficients aléatoires bidimensionnels (2D-RCA)

S.Kharfouchi [83] a introduit le modèle spatiale non linéaire qui généralise le modèle autorégressif à coefficients aléatoires (RCA) standard à deux dimensions notées 2D-RCA pour l'extraction de paramètres caractéristiques d'une texture en vue de sa classification. Nous explicitons dans la suite les concepts de cette méthode.

Glossaire

- Les variables en gras sont des vecteurs dans Z^2
- Les variables souligné sont des vecteurs
- $\sharp A$ désigne le cardinal de A
- Pour chaque vecteur $x(t)$, $\hat{X}(N)$ est le moment empirique.
- Pour chaque vecteur $\mathbf{s}_1 = (s_1, s_2)$ et $\mathbf{t}_1 = (t_1, t_2)$, nous écrivons $s \ll t$ si et seulement si $(s_1 < t_1) \vee (s_1 = t_1)$ et $s_2 \leq t_2$

3.3.1 Concepts et définitions

Dans cette section, nous allons présenter les principaux concepts méthodologiques qui ont été développés afin de permettre l'étude des champs aléatoires, ces derniers étant l'outil probabiliste permettant de modéliser des données spatiales. Nous donnons ainsi les définitions, notations et outils qui seront utilisés tout au long de ce chapitre. Nous nous intéressons en particulier à la classe de champs aléatoires discrets qui s'adaptent parfaitement au cas de la modélisation des images.

Champs aléatoires

Définition 3.1. *Soit $(\Omega, \mathcal{A}, \mathcal{P})$ un espace probabilisé et D un domaine de \mathbb{R}^d : Un champ aléatoire sur D à valeurs dans E est une collection de variables aléatoires sur $(\Omega, \mathcal{A}, \mathcal{P})$; tel qu'à chaque élément s de D est associé X_s ; ce que l'on note $\{X_s, s \in D\}$.*

Un champ aléatoire est totalement caractérisé par les distributions fini-dimensionnelles $(X_{s_1}, X_{s_2}, \dots, X_{s_n})$ pour toute partie finie $\{s_1, s_2, \dots, s_n\}$ de D . Ces distributions ne peuvent pas être quelconques, elles doivent vérifier des conditions minimales de cohérence, appelées conditions de Kolmogorov. Un champ aléatoire discret X est un processus prenant des valeurs aléatoires sur le réseau \mathbb{Z}^d : Les sites représentent en général des réseaux structurés à partir d'un graphe de voisinage et X_s est une donnée dite latticielle "intégrée" sur la région s En dimension 2, le réseau est appelé "grille".

Stationnarité

Dans les séries chronologiques, l'inférence des paramètres se fait par la répétition indépendante des données. Pour les séries spatiales, en pratique, il faudra inférer les propriétés de cette dernière à une seule réalisation, par exemple un épisode de pollution de l'ozone, une région agricole particulière, une image satellitaire ect... Pour pouvoir réaliser l'inférence statistique pour un événement unique, il faut donc jouer sur la répétition de certaines caractéristiques d'un point à l'autre de l'espace, au lieu de compter sur les répétitions indépendantes d'un moment à l'autre. On pose donc les hypothèses de stationnarité et d'ergodicité suivantes.

1. Stationnarité stricte :

Définition 3.2. *Un champ aléatoire est dit strictement stationnaire si toutes les lois fini-dimensionnelles sont invariantes pour toute translation, c'est à dire si $(X_{s_1}, X_{s_2}, \dots, X_{s_n})$ et $((X_{s_1+h}, (X_{s_2+h}, \dots, (X_{s_n} + h)$ ont les mêmes loi pour tout n -uplet s_1, \dots, s_n et tout $h \in D$.*

Cette hypothèse difficilement vérifiable, n'est utilisée que dans certains cas, rencontrés en pratique comme celui des champs de carrés non intégrables. Elle est rarement utilisée ailleurs car elle présente un défaut majeur : elle ne dit rien sur l'existence ou non des moments, alors que l'espérance et la variance sont des grandeurs nécessaires pour l'analyse statistique des données, et d'autre part elle doit être vérifiée pour des n supérieurs au nombre de données disponibles. On lui préfère l'hypothèse plus faible de stationnarité d'ordre 2.

2. Stationnarité du second ordre (ou faible)

Définition 3.3. *Un champ de carré intégrable, X_s sur \mathbb{R}^d est dit stationnaire d'ordre 2 si les deux premiers moments existent et sont invariants par translation :*

$$E(X_s) = m \quad \forall s \in \mathbb{R}^d \quad (3.13)$$

$$Cov(X_u, X_v) = \gamma(u, v) = Cov(X_{u+\tau}, X_{v+\tau}) \quad (3.14)$$

quelques soient u, v et $\tau \in \mathbb{R}^d$

Notons que puisque $\gamma(u, v) = \gamma(u - v, 0)$ pour tout u, v et τ dans \mathbb{R}^d pour un champ faiblement stationnaire, il sera plus commode de redéfinir la fonction d'auto-covariance comme une fonction d'une seule variable de la manière suivante :

$$\gamma(h) = \gamma(h, 0) \quad (3.15)$$

Si X est stationnaire au sens strict et si $X \in L^2$, (L : l'espace des variables aléatoires de carré intégrable) alors X est stationnaire au second ordre. L'inverse n'est pas vrai en général mais les deux notions coïncident si X est Gaussien.

Exemple 1 : (Bruit Blanc Fort (BBF), BB faible (BBf), BB Gaussien (BBG))

X est un BBF si les v.a. $\{X_s, s \in D\}$ sont centrées, indépendantes et identiquement distribuées (i.i.d.).

X est un BBf si les v.a. sont centrées, et décorréliées.

Un BBF sur D est strictement stationnaire ; un BBf sur D est faiblement stationnaire.

X est un BBG si les v.a. $\{X_s, s \in D\}$ sont i.i.d. gaussiennes centrées.

3. Ergodicité

L'ergodicité est une propriété qui renforce la notion de stationnarité, qui permet de contourner la restriction d'une réalisation unique et qui permet d'obtenir la convergence presque sûre d'une moyenne empirique spatiale lorsque le domaine "tend vers l'infini".

Exemple 2 $X = \{X_s, s \in \mathbb{Z}^d\}$, X_s i.i.d est un exemple de processus ergodique. Après avoir introduit les notions de champs aléatoires, de stationnarité et d'ergodicité dans le cas le plus général (champ indexé par \mathbb{R}^d), on revient au contexte de ce travail doctoral qui est le domaine des champs indexés par \mathbb{Z}^d . Sur \mathbb{Z}^d on définit dans le paragraphe suivant, l'ordre partiel usuel et l'ordre lexicographique. Ces notions d'ordre vont permettre, même artificiellement, de structurer l'espace par un certain ordre qui jouera le rôle de passé et de présent ou plus précisément de précédence et

de succession des observations.

Ordre total

Dans les champs indexés par \mathbb{Z} , la notion de futur et de passé est claire et naturelle. En revanche lorsque le champ est indexé par \mathbb{Z}^d ($d \geq 2$); plusieurs formulations peuvent être définies selon l'ordre utilisé sur \mathbb{Z}^d . Nous donnons ici l'ordre que nous utiliserons pour définir certains ensembles utiles à la description des modèles que nous introduisons dans ce chapitre. L'ordre lexicographique sur \mathbb{Z}^d (ou total) est défini pour deux points $s = (s_1, s')$ et $t = (t_1, t')$ de \mathbb{Z}^d tels que s_1 et t_1 appartiennent à \mathbb{Z} par :

$$s \ll t \quad (3.16)$$

Si et seulement si $s_1 < t_1$ et $s' \ll t'$ dans \mathbb{Z}^{d-1} . Cet ordre est aussi appelé ordre demi plan.

Ensemble indéxant

Pour deux éléments a, b de \mathbb{Z}^d ; tel que $a < b$ et $a \neq b$ nous considérons maintenant les ensembles de \mathbb{Z}^d suivants :

$$\begin{aligned} S[a, b] &= \{x \in \mathbb{Z}^d : a \ll x \ll b\} \\ S[a, \infty] &= \{x \in \mathbb{Z}^d : a \ll x\} \\ S[\infty, a] &= \{x \in \mathbb{Z}^d : x \ll a\} \\ S]a, b] &= S[a, b] \setminus a \\ S]a, \infty] &= S[a, \infty] \setminus a \end{aligned} \quad (3.17)$$

$S[a, b]$ nous permet de définir une notion de fenêtre spatiale qui étend la notion classique d'intervalle du temps utilisée dans le cadre temporel.

$S[a, \infty]$ généralise la notion de futur au cadre spatial.

Enfin $S[\infty, a]$ permet de donner un équivalent à la notion du passé d'un instant d'observation lorsqu'il s'agit du passé d'un site d'observation a .

3.3.2 Définition du modèle 2D-RCA

Le modèle spatial non linéaire qui généralise au cadre spatiale le modèle autorégressif à coefficients aléatoires (RCA) standard noté 2D-RCA [83] est généré par l'équation :

$$X(\mathbf{t}) = \sum_{\mathbf{s} \in S[\mathbf{0}, \mathbf{p}]} a_{\mathbf{s}}(\mathbf{t})X(\mathbf{t} - \mathbf{s}) + e(\mathbf{t}), \mathbf{t} \in \mathbf{Z}^2 \quad (3.18)$$

où $\mathbf{p} = (p_1, p_2)$, $\mathbf{0} = (0, 0)$, $a_{\mathbf{s}}(\mathbf{t}) = \alpha_{\mathbf{s}}(\mathbf{t}) + \beta_{\mathbf{s}}(\mathbf{t})$ and $d = \#S[\mathbf{0}, \mathbf{p}]$.

Pour ce modèle, les hypothèses suivantes ont été posées :

- (a) $(e(\mathbf{t}))$; $\mathbf{t} \in \mathbf{Z}^2$ est une suite indépendante de variables aléatoires faiblement stationnaires de moyenne zéro et de variance σ_e^2
- (b) Les $\alpha_{\mathbf{s}}$, $\mathbf{s} \in S[\mathbf{0}, \mathbf{p}]$ sont des constantes réelles.
- (c) si $\underline{\beta}(\mathbf{t}) = (\beta_{\mathbf{s}}(\mathbf{t}); \mathbf{s} \in S[\mathbf{0}, \mathbf{P}])'$, pris dans l'ordre lexicographique, alors $(\underline{\beta}(\mathbf{t}), \mathbf{t} \in \mathbf{Z}^2)$ est une suite indépendante de vecteurs aléatoires de dimensions $d \times 1$ de moyenne zéro et $E \{ \underline{\beta}(\mathbf{t})\underline{\beta}'(\mathbf{t}) \} = C$
- (d) $\underline{\beta}(\mathbf{t})$ and $e(\mathbf{t})$ sont indépendants.

Les modèles 2D-RCA proposés dans [83] sont définis sur un réseau régulier. Ils sont unilatéraux par construction, seulement la plupart des images que nous traitons sont constituées de grilles avec pixels irréguliers. Heureusement, avec l'utilisation croissante de la technologie informatique, dans certaines situations au moins, les données avec des pixels irrégulièrement espacés, peuvent être remplacés par une grille régulière en utilisant des techniques d'interpolation d'images et les programmes de ré-échantillonnage.

Dans la partie estimation, nous avons besoin de la stationnarité du second et du quatrième ordre du modèle défini par l'Eq.(3.18). Des conditions de stationnarité

du second et du quatrième ordres ont été établies dans [83].

Exemple 3 : Considérons un modèle 2D-RCA simple du premier ordre défini par :

$$\begin{aligned} X(i, j) &= \alpha X(i, j-1) + \beta X(i-1, j) + \gamma X(i-1, j-1) + a_1(i, j)X(i, j-1) \\ &\quad + a_2(i, j)X(i-1, j) + a_3(i, j)X(i-1, j-1) + \varepsilon(i, j) \\ \underline{\theta} &= (\alpha, \beta, \gamma) \\ \underline{x} &= (X(i, j-1), X(i-1, j), X(i-1, j-1)) \end{aligned} \quad (3.19)$$

où $\varepsilon(i, j); (i, j) \in \mathbb{Z}^2$ est une suite i.i.d de variables aléatoires avec moyenne nulle et une variance σ^2 , α, β et γ sont constantes et $\{a_l(i, j); (i, j) \in \mathbb{Z}^2\}$ sont des suites indépendantes de variables aléatoires centrées, $E[\alpha_l(i, j)] = \eta_l^2$ et $a_l(i, j)$ sont indépendants de $\{\varepsilon(i, j); (i, j) \in \mathbb{Z}^2\}$ pour tout $l=1, 2, 3$. Dans ce cas :

$$\underline{x} = (X(i, j-1), X(i-1, j), X(i-1, j-1)) \quad (3.20)$$

La condition de stationnarité du second ordre est donnée par :

$$(i) \Delta \geq 0$$

$$(ii) \frac{1}{2} \left(\alpha^2 + \beta^2 + \eta_1^2 + \eta_2^2 + \sqrt{\Delta} \right) < 1.$$

où

$$\Delta = (\alpha^2 + \beta^2 + \eta_1^2 + \eta_2^2)^2 + 4(\gamma^2 + \eta_3^2 + 2\alpha\beta\gamma) \geq 0,$$

Pour plus de détails voir [83].

3.3.3 L'estimation des paramètres et propriétés asymptotiques

Dans cette partie, nous allons nous intéresser à l'estimation des paramètres du modèle 2D-RCA défini par Eq.3.18 en utilisant la méthode des moments généralisée (GMM)[91]. La méthode des moments généralisée repérée par l'acronyme MMG en français et GMM en anglais, consiste à substituer les moments empiriques des observations dans un système d'équations obtenu à partir de conditions dites "conditions d'orthogonalités (voir Eq.3.22). Pour introduire la classe des estimateurs MMG, soit $\underline{w}(i, j)$ un vecteur $N_1 N_2 \times 1$ de

variables aléatoires qui sont observées à la position (i, j) . Soit $\underline{\theta} \in \Omega$ un vecteur à x 1 de paramètres, et soit $h(\underline{\theta}, \underline{w}(i, j))$ une fonction vectorielle de dimension r . On définit pour $N = (N_1, N_2)$

$$g_{\mathbf{N}}(\underline{\theta}) = \frac{1}{N_1 N_2} \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} h(\underline{\theta}, \underline{w}(i, j)) \quad (3.21)$$

En supposant de plus que $\underline{\theta}_0$ est la vraie valeur de $\underline{\theta}$ et que les conditions des moments peuvent s'écrire :

$$E[h(\underline{\theta}, \underline{w}(i, j))] = 0 \quad (3.22)$$

ce qui constitue r conditions d'orthogonalité : L'estimateur MMG est défini alors comme solution du problème de minimisation

$$\hat{\underline{\theta}}_{\mathbf{N}} = \underset{\underline{\theta} \in \Omega}{\operatorname{argmin}} [g_{\mathbf{N}}(\underline{\theta})]' W_{\mathbf{N}} [g_{\mathbf{N}}(\underline{\theta})] \quad (3.23)$$

où la matrice de pondération $W_{\mathbf{N}}$ est définie positive. En pratique, cette matrice est évaluée à partir des observations. Sous de convenables conditions, l'estimateur MMG est consistant et asymptotiquement normal[92].

Pour le modèle 2D-RCA, en plus des conditions (a), (b), (c) et (d), on suppose que les paramètres $\alpha_s, s \in S] \mathbf{0}, \mathbf{p}]$ et C sont telles qu'une solution stationnaire et $\mathcal{F}_{\mathbf{t}}$ mesurable existe. L'objectif est d'estimer les paramètres du modèle 2D-RCA sur la base des observations $\{X(t), t \in S] \mathbf{0}, \mathbf{N}]\}$, $\mathbf{N} = (N, N)$. Le modèle défini par l'Eq.3.18 est réécrit comme un modèle de régression linéaire :

$$X(\mathbf{t}) = \underline{x}'(\mathbf{t}) \underline{\theta} + U(\mathbf{t}) \quad (3.24)$$

$$\text{où } \underline{\theta} = (\alpha_s; s \in S] \mathbf{0}, \mathbf{p}])', U(\mathbf{t}) = \underline{\beta}(\mathbf{t}) \underline{x}(\mathbf{t}) + e(\mathbf{t}),$$

$$\underline{\beta}(\mathbf{t}) = (\beta_s(\mathbf{t}); s \in S] \mathbf{0}, \mathbf{p}])' \text{ et } \underline{x}(\mathbf{t}) = (X(\mathbf{t} - \mathbf{s}); s \in S] \mathbf{0}, \mathbf{p}])'.$$

Comme $\underline{\beta}(\mathbf{t})$ et $e(\mathbf{t})$ sont indépendants de $(\underline{\beta}(\mathbf{t} - \mathbf{k}), e(\mathbf{t} - \mathbf{k}), \mathbf{0} \preceq \mathbf{k}, \mathbf{k} \neq (0, 0))$ et $\underline{x}(\mathbf{t})$ est fonction mesurable de cet ensemble seulement ; on obtient :

$$E[(X(\mathbf{t}) - \underline{x}'(\mathbf{t}) \underline{\theta}_0) \setminus \mathcal{F}_{\mathbf{t}}] = 0,$$

de plus,

$$\begin{aligned}
 & E [\underline{x}(\mathbf{t}) (X(\mathbf{t}) - \underline{x}'(\mathbf{t}) \underline{\theta}_0)] \\
 &= E [E [\underline{x}(\mathbf{t}) (X(\mathbf{t}) - \underline{x}'(\mathbf{t}) \underline{\theta}_0) \setminus \mathcal{F}_{\mathbf{t}}]] \\
 &= E [\underline{x}(\mathbf{t})] E [(X(\mathbf{t}) - \underline{x}'(\mathbf{t}) \underline{\theta}_0) \setminus \mathcal{F}_{\mathbf{t}}],
 \end{aligned}$$

On pose $\underline{w}(\mathbf{t}) = X(\mathbf{t}) \underline{x}(\mathbf{t})$ et $Z(\mathbf{t}) = \underline{x}(\mathbf{t}) \underline{x}'(\mathbf{t})$ on obtient les conditions des moments suivantes :

$$\begin{cases} h(\underline{\theta}) = \underline{x}(\mathbf{t}) [X(\mathbf{t}) - \underline{x}'(\mathbf{t}) \underline{\theta}] \\ E[h(\underline{\theta}_0)] = 0, \end{cases}$$

ce qui représente un système de \mathbf{d} conditions d'orthogonalité. Dans la suite, $A(N)$ est la matrice des poids $\mathbf{d} \times \mathbf{d}$ (peut-être aléatoire) de telle sorte que $A(N)$ converge en probabilité vers la matrice non aléatoire A quand $N \rightarrow \infty$. Pour un choix donné de la matrice de pondération $A(N)$, l'estimateur GMM $\hat{\underline{\theta}}$ de $\underline{\theta}$ peut être choisi de manière à rendre les moments empiriques $\hat{h}_{\mathbf{N}}(\underline{\theta})$ aussi proches que possible de zéro, c'est-à-dire $\hat{\underline{\theta}}$ est la valeur de $\underline{\theta}$ qui minimise la distance pondérée de $\hat{h}_{\mathbf{N}}(\underline{\theta})$ à zéro.

$$\hat{\underline{\theta}}_{\mathbf{N}} = \text{Arg min} \left\| A(\mathbf{N}) \hat{h}_{\mathbf{N}}(\underline{\theta}) \right\|^2 = \text{Arg min} \hat{h}'_{\mathbf{N}}(\underline{\theta}) H(\mathbf{N}) \hat{h}_{\mathbf{N}}(\underline{\theta}) \quad (3.25)$$

Où $H(\mathbf{N}) := A'(\mathbf{N})A(\mathbf{N})$ est une matrice de poids définie positive et symétrique qui converge en probabilité vers la matrice non aléatoire H quand $N \rightarrow \infty$. Donc après un choix judicieux de la matrice $H(\mathbf{N})$, on obtient :

$$\hat{\underline{\theta}}_{\mathbf{N}} = \left[\hat{Z}(\mathbf{N}) \right]^{-1} \hat{w}(\mathbf{N}) \quad (3.26)$$

$$= \left(\sum_{\mathbf{t} \in S[1, \mathbf{N}]} \underline{x}(\mathbf{t}) \underline{x}'(\mathbf{t}) \right)^{-1} \sum_{\mathbf{t} \in S[1, \mathbf{N}]} \underline{x}(\mathbf{t}) X(\mathbf{t}). \quad (3.27)$$

Les propriétés asymptotiques de $\hat{\underline{\theta}}_{\mathbf{N}}$ sont données dans les lemmes suivants :

Lemme 1 : soit $\hat{\underline{\theta}}_{\mathbf{N}}$ la suite des estimateurs GMM satisfaisant Eq.(3.25) alors, sous les conditions de stationnarité données dans [83]

$\hat{\underline{\theta}}_{\mathbf{N}} \rightarrow \infty$ quand $N \rightarrow \infty$ presque sûrement.

Lemme 2 : avec les conditions de stationnarité, on suppose que $(X(t), t \in \mathbb{Z}^2)$ possède un moment d'ordre quatre. Alors

$N \left(\hat{\underline{\theta}}_{\mathbf{N}} - \underline{\theta} \right) \stackrel{\mathcal{L}}{\rightsquigarrow} \mathcal{N}(0, \mathcal{V})$ où $\mathcal{V} = \mathcal{D}^{-1} \mathcal{S} \mathcal{D}^{-1}$, avec $\mathcal{S} = E \left[\underline{\mathbf{x}}(\mathbf{t}) \underline{\mathbf{x}}'(\mathbf{t}) \left(\sigma_e^2 + [\underline{\mathbf{x}}(\mathbf{t})^{\otimes 2}]' \underline{\Sigma} \right) \right]$ et $\mathcal{D} = -E \underline{\mathbf{x}}(\mathbf{t}) \underline{\mathbf{x}}'(\mathbf{t})$.

Exemple Considérons le modèle 2D-RCA du premier ordre défini par Eq.(3.19)

$$\underline{\theta} = (\alpha, \beta, \gamma)$$

$$\underline{\mathbf{x}} = (X(i, j-1), X(i-1, j), X(i-1, j-1))'$$

Sous les conditions de stationnarité d'ordre deux et quatre, l'estimateur GMM est :

$$\begin{aligned} \hat{\underline{\theta}}_{\mathbf{N}} = (\hat{\alpha}_N, \hat{\beta}_N, \hat{\gamma}_N) &= \left(\sum_{i=1}^N \sum_{j=1}^N [\underline{\mathbf{x}}(i, j) \underline{\mathbf{x}}'(i, j)] \right)^{-1} \\ &\times \sum_{i=1}^N \sum_{j=1}^N \underline{\mathbf{x}}(i, j) X(i, j) \end{aligned} \quad (3.28)$$

Un estimateur convergent de σ_U^2 est

$$\hat{\sigma}_U^2 = \frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^N \left[X(i, j) - \underline{\mathbf{x}}'(i, j) \hat{\underline{\theta}}_{\mathbf{N}} \right]^2 \quad (3.29)$$

3.4 Conclusion

L'objectif de ce chapitre est la présentation d'un nouveau modèle non linéaire pour la caractérisation de textures. L'étude théorique du modèle 2D-RCA a été exposée d'une façon succincte pour justifier l'utilisation des estimateurs des coefficients du modèle comme caractéristiques pertinentes d'images texturées. Ces paramètres ont été estimés par la méthode des moments généralisée. Ces estimateurs seront utilisés dans différents schémas de classification dans les chapitres suivants.

Chapitre 4

Approches développées pour la classification dans les sous espaces

4.1 Introduction

Dans ce chapitre, nous présentons les approches de classification développées dans le cadre de cette thèse. La première méthode présente la classification floue dans les sous espaces basée sur la minimisation d'une nouvelle fonction objective [93]. Les méthodes existantes dans la littérature posent un problème de localisation des centres de classes, en terme d'estimation des distances inter et intra classes. En effet l'estimation des distances basée sur le centre moyen des centres à travers les itérations de l'algorithme induit une erreur d'estimations des distances propagée et amplifiée à travers les itérations. Dans la majorité de ces algorithmes, l'initialisation arbitraire implique une instabilité des résultats pour chaque exécution. Ceci produit un taux de classification faible et un temps de traitement élevé. Dans un deuxième volet, nous avons développé une nouvelle approche utilisant les statistiques spatiales pour la classification des images texturées. Elle est basée sur un modèle autorégressif bidimensionnels 2D-RCA pour l'extraction des caractéristiques des textures [92]. Elle opère une classification supervisée. Plusieurs variantes d'algorithmes sont proposées.

4.2 Classification floue basée sur l'apprentissage actif et les séparateurs à vaste marge

Avant de détailler toutes les approches développées, nous donnons les notations utilisées dans toute la suite de ce mémoire.

Notation

ALISSC : Active learning for improving a soft subspace clustering

ESSC : Enhanced soft subspace clustering

FCS : Fuzzy compactness Separation

v : Centre des classes

u : Degré d'appartenance

m : Paramètre flou

c : Nombre de classes

N : Taille des données

D : Taille des attributs (sous espaces)

w : Matrice des poids

x : Objet / niveau de gris / couleur

n_{ik} : Densité de la i^{eme} classe dans le sous espace k

η : paramètre contrôlant l'influence des poids dans le terme de séparation

v_0 : Centre des centres

ent : Entropie des données

N_c : Nombre d'objets dans une classe

T : Vecteur d'apprentissage

k : fonction de noyau

y : Classe d'une entrée x

NMI : Information Mutuelle Normalisée (Normalized mutual information)

IP : Indice de pourcentage

CD : la déviation des centres

CVE : Evaluation de la validation croisée.

Dans l'approche proposée la classification se déroule en deux temps :

- Une étape d'initialisation utilisant l'algorithme d'apprentissage supervisé SVM pour le calcul optimal des centres des classes et des degrés d'appartenance.
- La classification dans les sous espaces avec l'utilisation conjointe de l'algorithme SVM et l'apprentissage actif.

Une présentation simple de l'approche proposée est donnée par la Figure.(4.1)

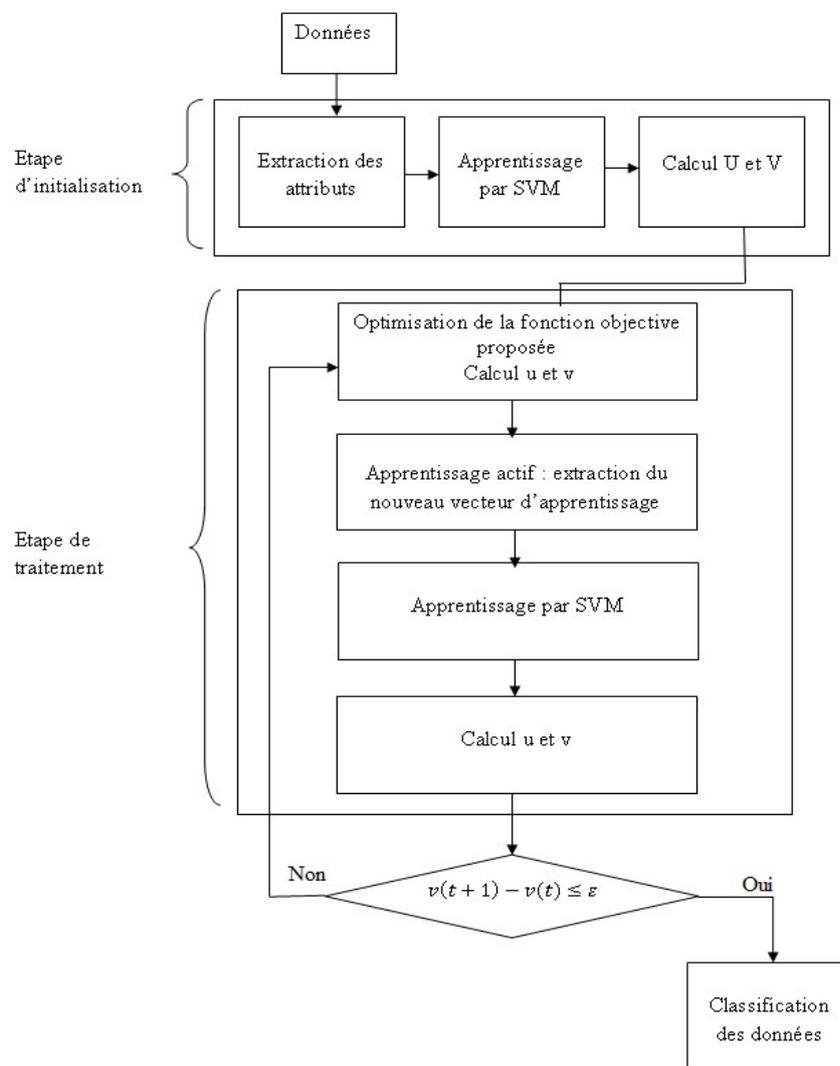


FIGURE 4.1 – Organigramme de la méthode proposée

Nous minimisons une nouvelle fonction objective J_{ALISSC} . Elle se compose de deux termes : un terme de séparabilité et un terme de compacité J_{Ct} dans lequel nous avons introduit la notion de la densité des classes \mathbf{n} :

$$J_{ALISSC}(v, u) = \sum_{i=1}^c \sum_{j=1}^N u_{ij}^m \sum_{k=1}^D w_{ik} \frac{(x_{jk} - v_{ik})^2}{n_{ik} + 1} - \eta \sum_{i=1}^c \sum_{j=1}^N u_{ij}^m \sum_{k=1}^D w_{ik} (v_{ik} - v_{0k})^2 \quad (4.1)$$

Où le terme de compacité est :

$$J_{Ct}(v, u) = \sum_{i=1}^c \sum_{j=1}^N u_{ij}^m \sum_{k=1}^D w_{ik} \frac{(x_{ik} - v_{ik})^2}{n_{ik} + 1}$$

et le terme de séparabilité est donnée par :

$$J_{St}(v, u) = \eta \sum_{i=1}^c \sum_{j=1}^N u_{ij}^m \sum_{k=1}^D w_{ik} (v_{ik} - v_{0k})^2$$

et Sous la contrainte :

$$\sum_{i=1}^c u_{ij} = 1 \quad \text{et} \quad 0 \leq u_{ij} \leq 1 \quad (4.2)$$

Les contraintes du problème posé par l'Eq.(4.1) découlent directement du principe de la logique floue, à savoir le degré d'appartenance d'un pixel à une classe ne peut dépasser 1 et la somme des degrés d'appartenance à toutes les classes doit valoir 1. Le problème d'optimisation de la fonction objective sous la contrainte Eq.(4.2) est un problème mal-posé, car les degrés d'appartenance u_{ij} et les centres des classes v_{ik} ne peuvent être déterminés simultanément. Afin de résoudre ce problème, une procédure d'alternance a été utilisée. En effet, dans une première étape, les degrés d'appartenance sont déterminés en supposant les centres des classes fixes et dans une deuxième étape, les centres des classes sont déterminés en supposant les degrés d'appartenance fixes. Ces deux étapes sont répétées jusqu'à convergence de l'algorithme. Pour résoudre le problème d'optimisation avec contraintes, le problème est transformé en un problème sans contraintes à l'aide des multiplicateurs de Lagrange Eq.(4.3) et Eq.(4.6)

$$\phi_1(u, \lambda) = \sum_{i=1}^c \sum_{j=1}^N u_{ij}^m \sum_{k=1}^D w_{i,k} \frac{(x_{jk} - v_{ik})^2}{n_{ik} + 1} - \eta (v_{ik} - v_{0k})^2 - \sum_{j=1}^N \lambda_j^u \sum_{i=1}^c u_{ij} - 1 \quad (4.3)$$

La dérivée par rapport aux degrés d'appartenance est calculée en supposant les centres

des classes v constants :

$$\begin{aligned} \frac{\partial \phi_1}{\partial u_{ij}} &= m u_{ij}^m - 1 \sum_{k=1}^D w_{ik} \frac{(x_{jk} - v_{ik})^2}{n_{ik} + 1} - \eta (v_{ik} - v_{0k})^2 - \lambda_j^u = 0 \\ \frac{\partial \phi_1}{\partial \lambda_j^u} &= \sum_{i=1}^c u_{ij} - 1 = 0 \end{aligned} \quad (4.4)$$

Donc le degré d'appartenance qui minimise la fonction objective de l'Eq.(4.1) est donné par la formule suivante Eq.(4.5 :

$$u_{ij} = \frac{\sum_{k=1}^D w_{ik} \left[\frac{(x_{jk} - v_{ik})^2}{n_{ik} + 1} - \eta (v_{ik} - v_{0k})^2 \right]^{\frac{-1}{m-1}}}{\sum_{i=1}^c \sum_{k=1}^D w_{ik} \left[\frac{(x_{jk} - v_{ik})^2}{n_{ik} + 1} - \eta (v_{ik} - v_{0k})^2 \right]^{\frac{-1}{m-1}}} \quad (4.5)$$

avec

$$\sum_{i=1}^c u_{ij} = 1$$

La dérivée par rapport les centres des classes en supposant les degrés d'appartenance constants sont données par Eq.(4.8) :

$$\phi_2(v) = \sum_{i=1}^c \sum_{j=1}^N u_{ij}^m \sum_{k=1}^D w_{ik} \frac{(x_{jk} - v_{ik})^2}{n_{ik} + 1} - \eta \sum_{i=1}^c \sum_{j=1}^N u_{ij}^m \sum_{k=1}^D w_{ik} ((v_{ik} - v_{0k})^2) \quad (4.6)$$

$$\frac{\partial \phi_2}{\partial v_{ik}} = w_{ik} \sum_{j=1}^N u_{ij}^m \left(-\frac{(x_{jk} - v_{ik})^2}{n_{ik} + 1} \right) - \eta (v_{ik} - v_{0k}) = 0 \quad (4.7)$$

Les centres des classes qui minimisent la fonction (Eq.(4.1) sont données par :

$$v_{ik} = \frac{\sum_{j=1}^N u_{ij} \left(\frac{x_{jk}}{n_{ik} + 1} - \eta v_{0k} \right)}{\sum_{j=1}^N u_{ij} \left(\frac{1}{(n_{ik} + 1)} - \eta \right)} \quad (4.8)$$

L'entropie pondérée est basée sur le calcul local de l'entropie selon les équations suivantes :

$$w_{ik} = \frac{sm_{ck}}{\sum_{k=1}^D sm_{ck}} \quad (4.9)$$

Avec

$$sm_{ck} = \sum_{h=1}^{N_c} ent(h, k)$$

sous la contrainte : $\sum_{k=1}^D w_{ik} = 1$

Avec $i=1 \dots c$ et $k = 1 \dots D$

4.2.1 Initialisation par les séparateurs à vaste marge (SVM)

Pour pallier aux problèmes d'initialisation aléatoire des algorithmes de classification, nous proposons d'utiliser un SVM multi-classes pour estimer les centres des classes \mathbf{v} ainsi que le degré d'appartenance \mathbf{u} . Un vecteur d'apprentissage \mathbf{T} est sélectionné arbitrairement.

$$\mathbf{T}_{xy} = (x_1, y_1), \dots, (x_{ne}, y_{ne})$$

où

ne : le nombre des exemples choisis.

L'algorithme SVM est entraîné pour produire les résultats initiaux de la classification. Selon ces résultats, les valeurs adéquates des centres des classes ainsi que les degrés d'appartenance seront obtenus.

Nous avons utilisé le noyau RBF (Radial Basis function) pour les SVM donné par :

$$k(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|}{2\sigma^2}\right) \quad (4.10)$$

Les centres des classes \mathbf{v} et les degrés d'appartenance \mathbf{u} peuvent être mis sous les formes suivantes :

$$v_{ik} = \frac{\sum_{l=1}^{N_i} x_{lk}}{N_i} = \frac{\sum_{j=1}^N u'_{ij} x_{jk}}{N_i} \quad i = 1..c, k = 1..D \quad (4.11)$$

Avec

$$\begin{cases} u'_{ij} = 1 & \text{si la classe } x_j \text{ est gale a } i \\ u'_{ij} = 0 & \text{autrement} \end{cases} \quad i = 1..c; j = 1..N \quad (4.12)$$

Les centres des classes \mathbf{v} et les degrés d'appartenance \mathbf{u} obtenus seront injectés dans l'étape d'optimisation.

4.2.2 Apprentissage actif et séparateurs à vaste marge

Dans cette partie nous proposons un critère basé sur une nouvelle formulation de l'expression des degrés d'appartenance et des centres des classes. Cette nouvelle formulation est basée sur l'estimation des centres de classes par l'apprentissage actif et la méthode

SVM multi-classes avec la nouvelle fonction objective (J_{ALISSC}) Eq.(4.1)

A la première itération, les paramètres \mathbf{w}, \mathbf{u} et \mathbf{v} sont calculés respectivement selon les équations données par : (Eq.(4.9), (Eq.(4.8) et (Eq.(4.5)). Nous utilisons le modèle d'apprentissage par SVM à chaque itération pour bien localiser les nouveaux centres de classes.

L'utilisation de l'apprentissage actif sert à choisir les meilleurs vecteurs d'apprentissage pour entraîner l'algorithme SVM.

La sélection de ce vecteur est basée sur les centres des classes \mathbf{v} et les degrés d'appartenance \mathbf{u} . Les objets \mathbf{x} sélectionnés sont les points avec une valeur de \mathbf{u} supérieur à 0.5 ou à $1/c$ ou c est le nombre de classes. Ce vecteur constitué par les centres des classes ainsi que leurs voisinages est utilisé comme vecteur d'apprentissage pour entraîner l'algorithme SVM. La construction de ce vecteur \mathbf{T} se fait comme suit :

- La distance Euclidienne est calculée entre chaque objet \mathbf{x} et les centres des classes

\mathbf{v} :

$$dis(i, j) = \sqrt{\sum_{j=1}^N \sum_{i=1}^c (x_{jk} - v_i)^2}$$

- Les étiquettes y_j données par :

$$y_j = argmin(dis(i, j))$$

- Le vecteur \mathbf{T}' constitué par les objets \mathbf{x} qui ont une valeur de degré d'appartenance supérieur ou égale à $1/c$ est formé. \mathbf{T}' est donné donc par :

$$\mathbf{T}' = \{(x_{jk}, y_j) \text{ if } u_{ij} \geq 1/c$$

- Le vecteur d'apprentissage final pour entraîner le SVM est donné par l'équation Eq.(4.13)

$$\mathbf{T}_{xy} = \{(v_1, 1), \dots, (v_c, c), (x_i, y_i) \dots (x_l, y_l)\} \quad l = 1 \dots ne \quad (4.13)$$

Les résultats de la classification donnés par SVM seront utilisés pour recalculer les centres des classes, les degrés d'appartenance, les poids de pondération des centres et le centre moyen. Ce dernier sert à vérifier la séparabilité des classes. Ce processus est itéré jusqu'à convergence.

La nouvelle formulation des degrés d'appartenance est :

$$u_{ij} = \frac{\sum_{k=1}^D w_{ik} [som1 - \eta(som2)]^{\frac{-1}{m-1}}}{\sum_{i=1}^c [\sum_{k=1}^D w_{ik} (som1) - (som2)]^{\frac{-1}{m-1}}} \quad (4.14)$$

Sachant que

$$som1 = \frac{(x_{jk} - \frac{\sum_{j=1}^N u_{ij} x_{jk}}{N_i})^2}{n_{ik} + 1}$$

$$som2 = (\frac{\sum_{j=1}^N u_{ij} x_{jk}}{N_i} - v_{0k})^2$$

Avec $i=1\dots c$ et $j = 1\dots N$ et $k=1\dots D$

4.2.3 Réglage de paramètres

Dans l'algorithme proposé, les paramètres du flou m et de contrôle de la séparabilité et la compacité η doivent être réglés à des valeurs adéquates. La meilleure valeur de m (le paramètre flou) est calculée comme suit [94] :

si $\min(N, D - 1) \geq 3$ **alors**

$$1 < m < \frac{\min(N, D - 1)}{\min(N, D - 1) - 2}$$

sinon $m = 2$

fin si Avec N : taille des données et D : nombre des sous espaces

Le paramètre η est utilisé pour faire une balance entre l'influence des termes de séparabilité et de compacité de l'Eq.(4.1). On peut déterminer ce paramètre par la distance entre $som1$ et $som2$ de l'Eq.(4.14) comme suit :

$d_{i,j} = som1 - \eta som2$ Si le paramètre η prend une valeur égale à $som1/som2$ donc on ne peut pas calculer le degré d'appartenance u l'Eq.(4.14). Pour s'assurer que la valeur de u prend sa valeur entre 0 et 1 il faut mettre une condition sur le paramètre η comme suit :

$$\eta = \min(\min(\eta, \frac{som1}{som2} * F))$$

où $F \neq 1$

On peut résumer les étapes de cette méthode (ALISSC) dans l'algorithme (4.1) ALISSC :

Algorithme 4.1 ALISSC

Entrée(s) : Initialisation :

1. Entrées : nombre de classes c , paramètre flou m , seuil ε
2. Normalisation des données
3. Entraînement de l'algorithme SVM
4. Calcul des degrés d'appartenance \mathbf{u} , des centres des classes \mathbf{v} selon les équations Eq.(4.12 et Eq.(4.11) respectivement.

Sorties : v_0, u_0 **traitement :****Tant que** $\|v_{ik}^t - v_{ik}^{t+1}\| \geq \varepsilon$ **faire**

1. Calcul de \mathbf{w} , \mathbf{v} , \mathbf{u} selon les équations Eq.(1.9, Eq.(4.8) et Eq.(4.5) respectivement
2. Sélection d'un nouveau vecteur d'apprentissage T utilisant l'apprentissage actif Eq.(4.13)
3. Entraînement de l'algorithme SVM
4. Recalcul des centres des classes \mathbf{v} , \mathbf{u} , par les équations Eq.(4.11) et Eq.(4.14) respectivement

Fin Tant que**Classification :**

Assigner chaque objet x à sa propre classe i en prenant le degré d'appartenance \mathbf{u} maximal

4.3 Classification utilisant le modèle autorégressif bidimensionnel à coefficients aléatoires

Cette approche a été développée pour la classification supervisée des textures. Cette tâche est réalisée en deux étapes. Des caractéristiques de textures sont extraites en utilisant le modèle autorégressif à coefficients aléatoire bi-dimensionnel 2D-RCA [83]. Ces paramètres sont utilisés dans différents schémas de classification.

4.3.1 Extraction des caractéristiques par le modèle 2D-RCA

Après extraction des paramètres, leur estimation est effectuée par la méthode des moments généralisée (GMM) [91] sur une fenêtre d'analyse de taille fixée à priori selon l'algorithme 4.2.

4.3.2 Classification utilisant le modèle 2D-RCA

Dans cette étape nous avons utilisé deux algorithmes de classification supervisée. Le premier algorithme est basé sur un simple calcul de distance entre les paramètres de la fenêtre et ceux du pixel de l'image. Un vecteur de paramètres vt_i est formé pour toutes les textures présentes dans une image (la sortie de l'algorithme (4.2)). Le deuxième algorithme utilise comme classifieur, un séparateur à vaste marge (SVM) entraîné par le vecteur vt_i . Les étapes de l'algorithme proposé de classification sont présentées par l'algorithme 4.3.

Des points aléatoires pour chaque classe ont été sélectionnés, autour de ces points des fenêtres carrées ($n \times n$) ont été constituées, sur lesquelles on va estimer les caractéristiques de chaque texture, dans ce travail nous avons utilisé quatre paramètres. Cette opération est répétée pour toute l'image, en remplaçant chaque pixel par un vecteur de quatre paramètres. On obtient une matrice de la même dimension que l'image en quatre dimensions. Le deuxième algorithme de classification basé sur SVM se déroule en deux étapes qu'on peut résumer par l'algorithme (4.4) :

Algorithme 4.2 Estimation par GMM**Entrée(s) : Pour** objet x **faire**

1. Sélection d'une fenêtre FE de taille (nxn) autour de cet objet.
2. Estimation des caractéristiques de la fenêtre FE (considéré comme une réalisation du processus 2D-RCA(1,1) selon l'estimateur GMM ayant comme formule explicite :

$$\hat{\underline{\theta}}_N = \left(\hat{\alpha}_N, \hat{\beta}_N, \hat{\gamma}_N \right) = \left(\sum_{i=1}^N \sum_{j=1}^N [\underline{x}(i, j) \underline{x}'(i, j)] \right)^{-1} \\ \times \sum_{i=1}^N \sum_{j=1}^N \underline{x}(i, j) X(i, j)$$

Un estimateur convergent de σ_U^2 est

$$\hat{\sigma}_U^2 = \frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^N \left[X(i, j) - \underline{x}'(i, j) \hat{\underline{\theta}}_N \right]^2$$

Sortie : Constituer le vecteur $v_j = (\hat{\alpha}_N, \hat{\beta}_N, \hat{\gamma}_N)$ **Algorithme 4.3** Classification supervisée basée sur le calcul d'une distance**Entrée(s) : Pour** chaque pixel pix de l'image IMG de taille $(N \times M)$ **faire**

- sélection d'une fenêtre autour de chaque pixel
- Estimation des caractéristiques de chaque fenêtre utilisant l'algorithme.(4.2) pour obtenir v_i^{pix}
- Calcul du degré $D_{W_{\text{pix}}}^{C^i}$

$$D_{W_{\text{pix}}}^{C^i} = \frac{\sqrt{\sum_{l=1}^4 (Vt_l^i - Vt_l^{\text{pix}})^2}}{\max_{1 \leq s, j \leq k} \left\{ \sqrt{\sum_{l=1}^4 (Vt_l^s - Vt_l^j)^2} \right\}},$$

Sortie : Affectation de chaque pixel à sa classe C^{i_0} selon la condition suivante :

$$D_{W_{\text{pix}}}^{C^{i_0}} = \min_{1 \leq i \leq k} \left\{ D_{W_{\text{pix}}}^{C^i} \right\}$$

Algorithme 4.4 Classification par SVM

Entrée(s) :**Initialisation** :

- Image IMG de taille (NxM), nombre de classes c , taille de la fenêtre (nxn), paramètres de l'algorithme SVM (σ, vt_i) .

Extraction des paramètres du modèle 2D-RCA par l'estimateur GMM :**Pour** chaque classe C_i $i=1\dots c$ **faire**

- Sélectionner NE pixels d'une façon aléatoire
- Calcul du vecteur vt_i l'algorithme(4.2) pour obtenir les vecteurs vt_{ij} ou $j=1\dots NE$

Fin**Pour** chaque pixel pix de l'image IMG de taille (NxM) **faire**

- sélection d'une fenêtre autour de chaque pixel
- Estimation des caractéristiques de chaque fenêtre utilisant l'algorithme(4.2) pour obtenir vt_{ipix} $i=1\dots 4$
- Construction d'une matrice G de dimension (NxMx4)

Classification :Utiliser l'algorithme SVM multi-classes sur la matrice G avec le vecteur d'apprentissage vt_{ij} **Sortie** : Image classifiée

4.3.3 Textures générées par le modèle 2D-RCA

On peut considérer quelques textures comme une réalisation du modèle 2D-RCA. Donc ce modèle est capable de représenter différentes textures selon l'équation suivante avec différentes valeurs de α, β, γ :

$$X(i, j) = \alpha X(i, j - 1) + \beta X(i - 1, j) + \gamma X(i - 1, j - 1) + a_1(i, j)X(i, j - 1) \\ + a_2(i, j)X(i - 1, j) + a_3(i, j)X(i - 1, j - 1) + \varepsilon(i, j)$$

Pour évaluer la qualité des textures générées par rapport à une base des textures réelles. Nous avons opté pour les mesures de similarité largement utilisé en qualité d'image.

4.3.4 Mesures de similarité

a-Indice de similarité structurelle (SSIM) : SSIM [95] est une mesure de similarité entre deux images numériques. La métrique SSIM est calculée sur plusieurs fenêtres d'une image. On dénote x et y l'image originale et l'image déformée respectivement. La similarité compare la luminance, le contraste et structure entre chaque couple de fenêtres.

La luminance est estimée par la mesure de l'intensité moyenne de chaque fenêtre. La fonction mesurant la similarité, est de la forme :

$$SSIM = \frac{2\mu_x\mu_y + (K_1L)^2}{\mu_x^2 + \mu_y^2 + (K_1L)^2} * \frac{2\sigma_{xy} + (K_2L)^2}{\sigma_x^2 + \sigma_y^2 + (K_2L)^2} \quad (4.15)$$

sachant que

La luminance est estimée par la mesure de l'intensité moyenne de chaque fenêtre comme suit :

$$\mu_x = \frac{1}{N} \sum_{i=1}^N x_i$$

où

N : le nombre de pixels de chaque fenêtre.

X_i : l'intensité d'un pixel.

Le contraste pour chaque fenêtre est mesuré par :

$$\sigma_x = \left(\frac{1}{N-1} \sum_{i=1}^N (x_i - \mu_x)^2 \right)^{1/2}$$

K_1, K_2 : est une constante de très faible valeur.

b-structurale à base de gradient (GSSIM) : GSSIM [96] est une version améliorée de SSIM qui compare l'information de contour entre les blocs de l'image déformée et de l'image originale. Elle remplace la comparaison de contraste $c(x,y)$ et la comparaison

de structure $s(x,y)$, par la comparaison de contraste à base gradient $cg(x,y)$, et la comparaison de la structure $sg(x,y)$, respectivement. Il y a un certain nombre de manières d'obtenir l'information de contour, telle que l'algorithme simple de détection de contour. Deux cartes de gradient X' et Y' représentant respectivement l'image originale et déformée sont générées en utilisant l'opérateur de Sobel. En plus les blocs des images X' et Y' sont donnés par x' et y' . La comparaison de contraste à base de gradient $cg(x,y)$, et la comparaison de structure $sg(x, y)$ sont décrites par :

$$C_g(x, y) = \frac{2\sigma_{x'}\sigma_{y'} + c_2}{\sigma_{x'}^2 + \sigma_{y'}^2 + c_2} \quad (4.16)$$

$$S_g(x, y) = \frac{\sigma_{x'y'} + c_3}{\sigma_{x'}\sigma_{y'} + c_3} \quad (4.17)$$

Où $\sigma_{x'}$ et $\sigma_{y'}$ sont l'écart type des blocs x' et y' respectivement, $\sigma_{x'y'}$ est la covariance des blocs x' et y' .

C_2 , C_3 sont des petites constantes évitant la dénomination de la valeur zéro. La similitude structurale à base de Gradient (GSSIM) est décrite comme suit :

$$GSSIM(x, y) = [l(x, y)^\alpha], [C_g(x, y)]^\beta, [S_g(x, y)]^\gamma \quad (4.18)$$

Sachant que la fonction de comparaison de la luminance de la fenêtre de la forme :

$$l(x, y) = \frac{2\mu_x\mu_y + (K_1L)^2}{\mu_x^2 + \mu_y^2 + (K_1L)^2}$$

4.4 Conclusion

Dans ce chapitre nous avons présenté les approches développées pour apporter des solutions aux problèmes de classification dans les sous espaces liées à l'utilisation d'algorithme basé sur l'optimisation de fonctions objectives nécessitant une étape d'initialisation. Dans ce sens nous avons proposé une approche basée sur l'optimisation d'une nouvelle fonction objective. La classification a été améliorée par une étape d'initialisation judicieuse basée sur l'algorithme SVM. L'apprentissage actif pour l'algorithme SVM a été introduit dans

chaque itération pour améliorer la localisation des centres ainsi l'accélération de la convergence de la méthode. Une méthode de classification supervisée des textures basée sur un modèle spatiale non linéaire 2D-RCA a été proposée. L'estimation des paramètres de ce modèle a été faite par la méthode des moments généralisée (GMM). Ces paramètres représentent les caractéristique de texture qui constituent les différents attributs nécessaires pour l'étape de classification. Nous avons développé deux algorithmes pour classifier des images texturées.

Chapitre 5

Analyse des résultats et discussions

5.1 Introduction

Dans ce chapitre, nous présentons les résultats des approches proposées sur des bases de données synthétiques, des images de synthèse texturées et des images réelles. Les résultats obtenus ont été comparés avec ceux fournis par les algorithmes ESSC [52] et FCS [56] qui sont établis dans l'état de l'art. Les matrices de co-occurrence [79] et le modèle AR [90] ont été utilisés pour extraire les caractéristiques des textures. Ces caractéristiques ont été utilisées dans les mêmes schémas de classification pour tester l'efficacité des algorithmes proposés. Les tests ont été menés sur un ordinateur personnel core Duo, 2,2 GHz, 2 Go de RAM.

5.2 Description des basées utilisées

5.2.1 Bases de données synthétiques

Plusieurs bases de données ont été utilisées pour évaluer la performance de la méthode proposée, à savoir UCI [97], S, A and DIM [98] avec différents nombre de classes qui compris entre 2 et 50, nombre d'attributs variant entre 2 et 128 à différents degrés de chevauchement. La description de ces bases est présentée par les tableaux suivants

Table.(5.1) et Table.(5.2) :

TABLE 5.1 – Base UCI

base de données	nombre de classes	nombre d'objets	nombre d'attributs
Abdolan	3	4177	8
Australian	2	690	14
Balance	3	625	4
Car	4	946	18
Glass	6	214	9
Heart1	5	270	13
Heart	2	270	13
iris	3	150	4
Wine	3	178	13

TABLE 5.2 – Bases S,A et DIM

base de données	nombre de classes	nombre d'objets	nombre d'attributs
A1	20	3000	2
A2	35	5250	2
A3	50	7500	2
Dim32	16	1024	32
Dim64	16	1024	64
Dim128	16	1024	128
S1	15	5000	2
S2	15	5000	2
S3	15	5000	2
S4	15	5000	2

5.2.2 Images réelles

Les images réelles utilisées sont des images de scènes et des images médicales Figure.(5.1). Les Figures.(5.1.a,b et c) sont des images de scanner abdominal à différents coupes. L'image de la Figure.(5.1.d) est une image à résonance magnétique (IRM) du crâne qui met en évidence un processus expansif de la région frontale droite exerçant un effet de masse sur le ventricule latéral droit. Les différents organes sont repérés par des flèches.

1-le rein droit, 2- le rein gauche, 3-colonne vertébrale, 4- veines rénales, 5-rein droit avec

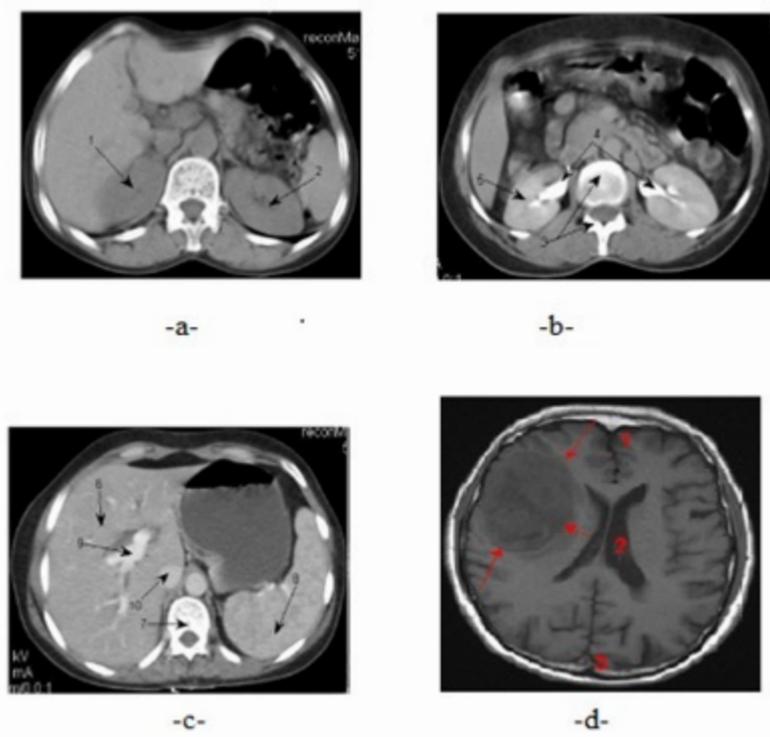


FIGURE 5.1 – Description des images utilisées.

contraste, 6- foie ,7- colonne vertébrale, 8- la rate, 9-veine, 10-Aorte, 1-d - masse kystique.

5.2.3 Base de textures (Brodatz)

Les textures de Brodatz sont une collection d'images de textures [99]. La figure.(5.2) montre un échantillon des images que nous avons utilisées. Dans nos tests, nous avons utilisé un ensemble d'images constitué par deux, trois ou quatre textures.

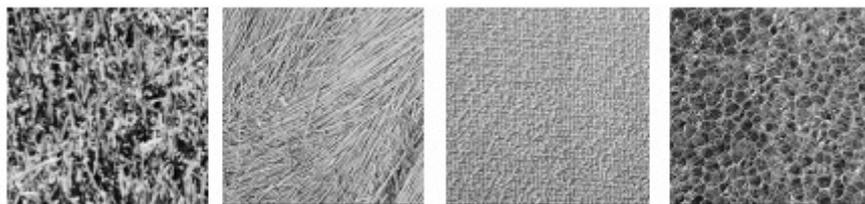


FIGURE 5.2 – Exemple d’images à niveau de gris de texture de Brodatz.

5.3 Résultats obtenus par l’algorithme ALISSC

5.3.1 Évaluation sur des bases de données synthétiques

Pour les tests sur les bases de données synthétiques, nous avons utilisé les métriques suivantes : l’Information Mutuelle Normalisée (NMI), l’indice de pourcentage (IP) ou le taux de bonne classification et la déviation des centres (CD). Toutes ces mesures ont été utilisées pour comparer et valider les résultats finaux. Les résultats obtenus par la méthode proposée ALISSC, l’algorithme ESSC [52], l’algorithme FCS [56] et SVM multi-classe ont été analysés. Ces algorithmes ont été exécutés dix fois, la moyenne et la déviation standard des différents indices sont présentées sur les tableaux Tab.(5.3),(4.4),(5.4),(5.5)et(5,6) avec les nombres d’itérations pour chaque méthode : ALISSC donne les valeurs de NMI les plus grandes pour les bases Abdolan (NMI=0.176), Australian (NMI=0.451), Balance (NMI=0.389), Glasses (NMI=0.668), Iris (NMI=0.873) (Table.5.3), A, Dim et S, avec taux de 0.98 pour les bases Dim32 et Dim64 (Table.5.5). Toutes ces bases présentent différents nombre de classes, nombre d’attributs, nombre d’objets, et où le nombre de classes est compris entre 2 et 16 et un nombre des attributs variant entre 2 et 128.

Les valeurs de l’indice IP confirment les résultats précédents. Ceux-ci montrent l’efficacité de la méthode proposée sur différents type des bases. Les meilleurs résultats obtenus par l’algorithme SVM multi classes sont donnés pour quelque bases (UCI) à l’opposé de ceux obtenus avec les bases ayant un nombre d’attributs élevé (A, Dim, S) où elles présentent taux de classification très faible.

Les tests sur les bases de données UCI, A, DIM et S confirment que la méthode proposée est la meilleure par rapport aux autres algorithmes.

TABLE 5.3 – Résultats de la classification pour la base UCI avec les valeurs de NMI et le nombre d'itérations it

base de données	ALISSC		ESSC		FCS		SVM Multi-classes
	NMI	it	NMI	it	NMI	it	NMI
Abdolan	0.176	2	0.163±0.02	11	0.150±0.014	20	0.133
Australian	0.451	2	0.361±0.016	12	0.105±0.218	20	0.319
Balance	0.389	3	0.287±0.14	9	0.159±0.09	20	0.14
Car	0.181	3	0.123±0.197	12	0.141±0.04	20	0.655
Glass	0.668	3	0.350±0.012	12	0.307±0.047	20	0.352
Heart1	0.213	2	0.111±0.153	12	0.046±0.052	20	0.547
Heart	0.245	3	0.306±0.047	16	0.131±0.040	20	0.337
iris	0.873	4	0.741±0.005	16	0.717±0.184	20	0.873
Wine	0.834	2	0.862±0.05	12	0,733±0,005	20	0.853

TABLE 5.4 – Résultats de la classification pour la base UCI avec l'indice IP

base de données	ALISSC	ESSC	FCS	SVM Multi-classes
Abdolan	48.00	45.37±0.60	41.63±5.01	65.86
Australian	87.00	80.65±2.10	51.90±0.70	93.00
Balance	60.00	59.20±7.32	45.19±7.13	54.51
Car	44.00	34.04±1.50	31.38±2.53	80.87
Glass	65.89	41.50±1.77	43.47±7.41	74.79
Heart1	48.51	21.45±2.33	47.51±0.45	92.73
Heart	78.00	72.43±3.51	31.76±5.81	70.30
iris	95.33	90.00±0.51	88.67±6.18	95.33
Wine	94.94	95.26±0.54	88.20±0.50	96.05

La convergence de l'algorithme ALISSC est plus rapide que les algorithmes ESSC et FCS. Ce ci est confirmé par le nombre itération (it) qui est très faible par rapport aux autres algorithmes.

TABLE 5.5 – Résultats de la classification pour les bases A, Dim et S avec les valeurs de NMI et le nombre d'itérations *it*.

base de données	ALISSC		ESSC		FCS		SVM Multi-classes
	NMI	it	NMI	it	NMI	it	NMI
A1	0.900	7	0.795±0.206	19	0.723±0.046	20	0.455
A2	0.912	16	0.784±0.207	23	0.720±0.055	20	0.350
A3	0.914	8	0.910±0.009	33	0.718±0.028	20	0.291
Dim32	0.983	4	0.741±0.005	16	0.831±0.032	20	0.485
Dim64	0.982	5	0.954±0.005	45	0.826±0.085	20	0.442
Dim128	0.900	6	0.900±0.023	20	0.841±0.114	20	0.399
S1	0.770	17	0.689±0.134	24	0.725±0.019	20	0.490
S2	0.756	13	0.84±0.128	21	0.798±0.053	20	0.586
S3	0.777	18	0.60±0.164	10	0.731±0.045	20	0.481
S4	0.710	17	0.60±0.128	18	0.669±0.057	20	0.443

TABLE 5.6 – Résultats de la classification pour les bases A, Dim et S avec l'indice IP

Base de donnée	ALISSC	ESSC	FCS	SVM Multi-classes
A1	74.33	62.87± 0.94	45.10±6.00	20.00
A2	77.81	56.59±1.04	38.82±1.27	11.43
A3	76.43	74.94±0.79	28.49±0.90	8.00
Dim32	95.33	90.00± 1.95	65.13±7.40	24.90
Dim64	93.00	84.00±1.34	70.59±1.28	23.14
Dim128	75.74	73.14±1.76	71.00±1.40	18.75
S1	84.62	66.53±1.94	58.25±1.06	38.01
S2	78.98	75.03±3.05	65.17±6.21	28.84
S3	79.50	53.18±2.86	52.50±7.21	28.94
S4	75.16	52.48±2.34	59.38± 5.80	27.65

Il a été observé que les valeurs obtenues par l'algorithme ALISSC sont les mêmes durant les dix exécutions à l'opposé de ceux obtenues par les deux algorithmes ESSC et FCS. En effet les résultats de la classification obtenus par la méthode proposée sont stables à cause de l'étape d'initialisation proposée.

La modification proposée apporte une grande amélioration dans différents types de bases et le temps de traitement diminue.

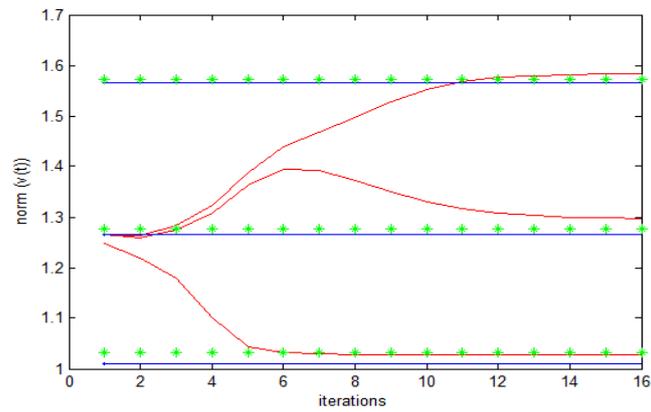
Évaluation des centres des classes

Puisque l'objectif de notre approche est basé essentiellement sur une amélioration de la localisation des centres des classes, nous avons besoin d'évaluer la qualité de ces centres. Pour réaliser cette tâche, nous avons tracé l'évolution des centres à chaque itération en les comparant avec les centres réels donnés dans les différentes bases. Les courbes sont représentées dans par la Figures.(5.3). Deuxièmement, pour mieux voir la localisation des centres des classes par rapport aux centres réels, nous avons présenté deux bases de données iris et S1 Figure.(5.4) et Figure.(5.5) avec les centres réels ainsi que les centres obtenus par ALISSC et ESSC. Enfin, nous avons utilisée la mesure de la déviation des centres (CD) pour confirmer les observations visuelles présentés par les figures (5.4)et(5.5), les résultats sont présentés par les tableaux Table.(5.7) et Table.(5.8)

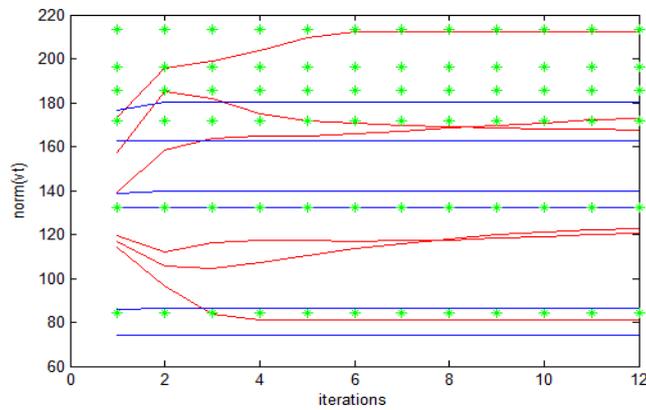
Les centres des classes obtenus par ALISSC illustrés sur la Figure.(5.3) convergent rapidement et ils ont aussi très proches des centres réels.

La Figure.(5.4.b) montre qu'ALISSC localisent bien les centres en les comparant avec les centres réels combien même les classes sont chevauchantes. L'algorithme ESSC produit des plus proches des centres réels dans les classes bien séparées Figure.(5.4.c) (classe bleu IRIS). Ceci est confirmé par la base S1 illustré par la figure.(5.5). Cette base présente 15 classes qui se chevauchent. Les centres obtenus par la méthode proposée sont très proches des centres réels l'opposé de ESSC qui crée un faux centre. Ce centre a conduit à la formation d'une mauvaise classe (représentée par le cercle rouge dans la figure.(5.5.c) et au même temps il a occulté une autre classe (représentée par le cercle noire).

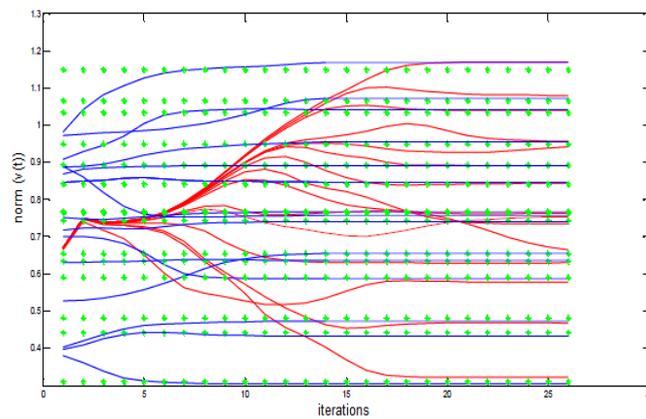
Les tableaux Table.(5.7) et Table.(5.8) représentent les valeurs de CD. Les valeurs



(a)

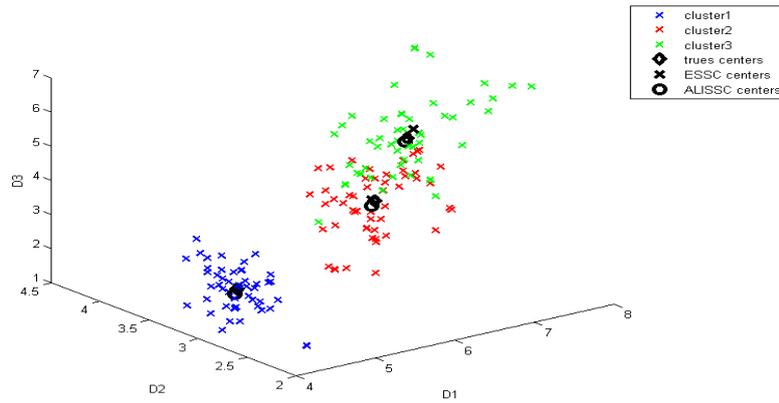


(b)

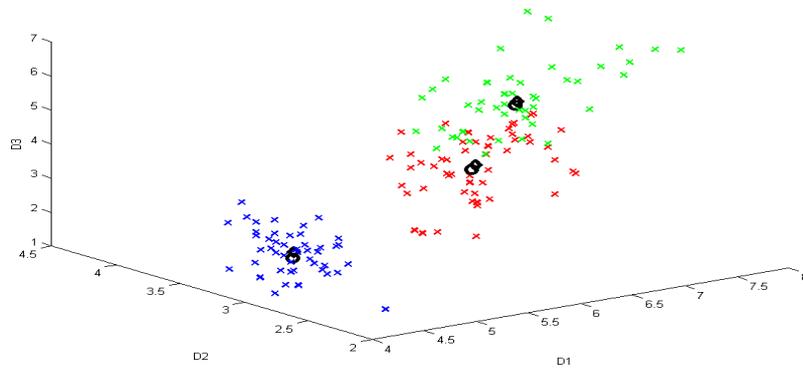


(c)

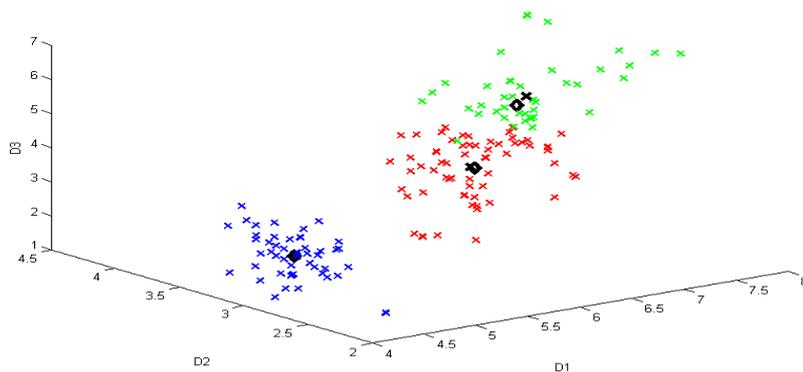
FIGURE 5.3 – Evolution des centres des classes à chaque itération, vert : Centres réels, bleu : Centres obtenus par ALISSC, rouge : Centres obtenus par ESSC.(a) : iris ; (b) : glass ,(c) :S1



(a)

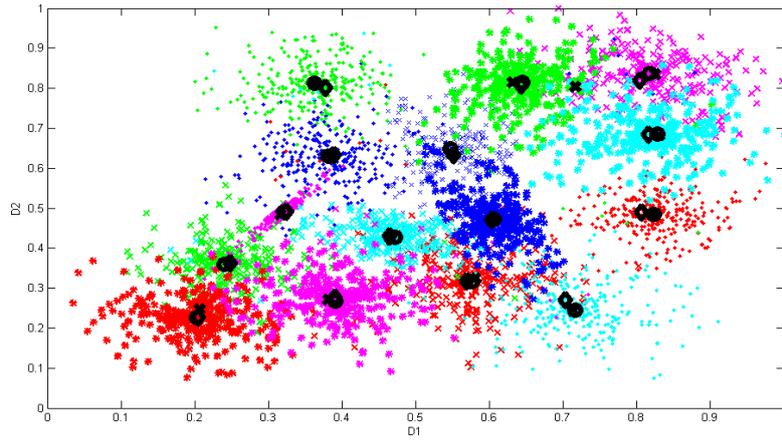


(b)

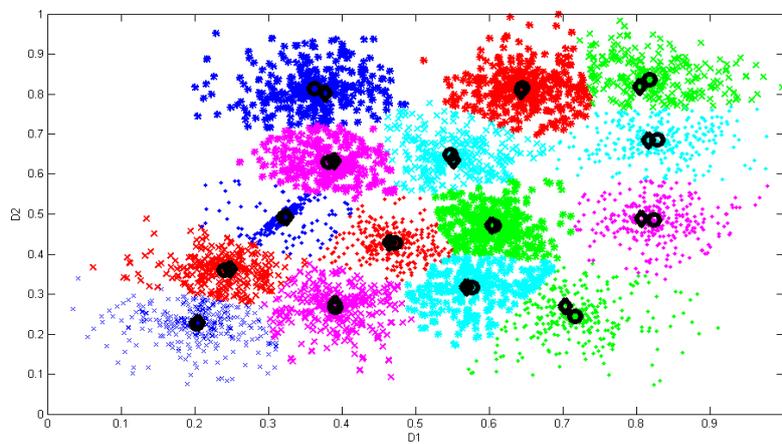


(c)

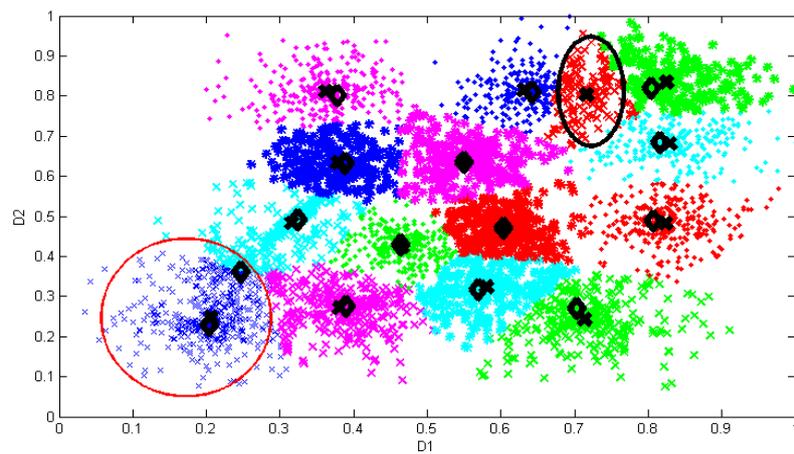
FIGURE 5.4 – Base Iris avec les centres réels et les centres obtenus par ALISSC et ESSC, (b) : Classification par ALISSC, (c) Classification par ESSC .



(a)



(b)



(c)

FIGURE 5.5 – (a) Base S1 avec les centres réels et ceux obtenus par ALISSC et ESSC, ((b) : Classification par ALISSC, (c) Classification par ESSC

TABLE 5.7 – Résultats de la classification pour les bases UCI avec l'indice CD

base de données	ALISSC	ESSC	FCS
Abdolan	0.210	0.171±0.01	0.083±0.015
Australian	0.229	0.243±0.03	0.804±0.061
Balance	0.026	0.012±0.0016	0.321±0.250
Car	0.760	0.07±0.10	2.276±0.838
Glass	0.027	0.064±0.077	0.968±0.355
Heart1	0.013	0.073±0.02	0.047±0.143
Heart	0.003	0.073±0.02	1.539±0.188
iris	0.0086	0.048±0.03	0.933±0.294
Wine	0.0043	0.03±0.02	0.163±0.392

TABLE 5.8 – Résultats de la classification pour les bases A, Dim et S avec l'indice CD

Base de données	ALISSC	ESSC	FCS
A1	0.016	0.180±0.243	0.098±0.144
A2	0.028	0.162±0.13	0.072±0.071
A3	0.10	0.145±0.12	0.057±0.071
Dim32	0.052	0.048±0.03	7.846±0.594
Dim64	0.012	0.05±0.126	15.863±1.277
Dim128	0.231	0.228±0.194	40.910±2.617
S1	0.035	0.175±0.04	0.036±0.043
S2	0.03	0.184±0.178	0.055±0.063
S3	0.028	0.14±0.098	0.042±0.047
S4	0.035	0.199±0.271	0.309±0.179

obtenues par ALISSC sont plus petites pour la plupart des bases utilisées que celles obtenues par les autres méthodes. Ceci signifie que les centres des classes obtenus par la méthode proposée sont plus proches des centres réels que ceux donnés par les autres

méthodes.

Évaluation de l'entropie pondérée pour chaque sous espace

L'entropie pondérée représente l'influence de chaque sous espace dans la distribution des classes. La valeur la plus élevée de l'entropie indique la meilleure partition dans le sous espace correspondant. Quand la valeur du poids w_{ik} dans le sous espace k est plus élevée, le centre de classe v_{ik} correspondant donne la meilleure localisation aboutissant à une meilleure classification. Nous calculons le taux de classification dans les différents sous espaces.

Le tableau (Table.(5.9)) représente le taux de classification de chaque classe dans les quatre sous espaces de la base IRIS. La deuxième classe est bien classée dans le 4eme sous espace, aussi la valeur élevée du poids w correspond au même sous espace ce qui signifie que les poids de pondération indiquent les meilleurs sous espaces capable de révéler le plus grand nombre possible d'objets dans une classe.

TABLE 5.9 – Taux de la classification dans chaque sous espace

	classes	S espace 1	S espace 2	S espace 3	S espace 4	meilleur w_i
IRIS	C1	92	98	100	100	3
	C2	98	94	98	100	4
	C3	100	94	100	96	3

Evaluation par la validation croisée

Puisque la méthode proposée est un modèle d'exploration de données, pour valider sa robustesse, nous avons utilisé la validation croisée. Dans ce travail, la validation croisée par la méthode des K-fold a été utilisée.

Les trois méthodes de classification ont été testées ; la méthode proposée, l'algorithme ESSC et SVM Multi-classes. La base de données UCI a été utilisée pour ce test. Le nombre de portions (Folds) a été fixé à 10. Dans ce cas 90%des données avec leurs étiquettes ont été utilisées pour calculer les centres des classes v et le degré d'appartenance u . Dans

TABLE 5.10 – La validation croisée de la base UCI

Base de données	ALISSC	ESSC	SVM Multi-classes
Abdolan	0.55	0.33	0.44
Australian	0.53	0.53	0.80
Balance	0.55	0.38	0.98
Car	0.99	0.48	0.77
Glass	0.97	0.42	0.94
Heart1	0.64	0.56	0.70
Heart	0.74	0.29	0.73
iris	0.99	0.82	0.96
Wine	0.97	0.77	0.94

l'étape de traitement seulement les 10% de données sont testées jusqu'à la convergence. Un taux de classification est calculé pour chaque test. L'opération est répétée pour les différents k dossiers puis la moyenne des taux obtenus est calculée à chaque fois. Les valeurs sont représentées par Table.(5.10). L'évaluation par la validation croisée confirme que la méthode proposée donne une meilleure classification pour les bases : Abdolan, Car, Glass, Heart, Iris et Wine. Un taux de classification arrive jusqu'à 99% pour les bases de donnée Iris et Car où l'algorithme SVM Multi-classes présente les meilleurs résultats pour Australian, Balance et Heart1.

La complexité

La complexité de la méthode proposée a été étudiée. Elle dépend du nombre de données N , le nombre d'attributs D , le nombre de classes C et la complexité de l'algorithme SVM multi classes. Celle-ci est de $O(C, Nsv)$ où Nsv est le nombre de vecteurs séparateurs. Pour chaque itération la complexité de notre méthode est de $O(NDc)$, donc pour n itérations la complexité est de $O(nt, N, D, C)$. Le temps d'exécution pour chaque algorithme a été calculé. Table.(5.11) avec une exécution pour les algorithmes ALISSC et SVM multi-

TABLE 5.11 – Temps d'exécution (seconde)

Base de données	ALISSC (1 exécution)	ESSC (10 exécution)	FCS (10 exécution)	SVM Multi-classes (1 exécution)
Abdolan	12.308	57.905	616.697	0.339
Australian	1.865	6.655	10.469	0.887
Balance	1.559	6.50	12.176	0.812
Car	0.981	5.72	3.580	0.429
Glass	3.864	8.350	7.931	0.91
Heart1	2.577	6,212	9.214	1.019
Heart	0.833	3.194	3.344	0.449
iris	0.842	3.72	3.295	0.452
Wine	1.127	5.32	3.849	0.511

classes et dix exécutions pour les algorithmes ESSC et FCS.

L'algorithme ALISSC présente le temps le moins faible car il est exécuté une seule fois dû à la stabilité des résultats grâce à l'initialisation proposée. Ce qui aide à la convergence rapide de la méthode qui est présentée par la figure.(5.6). Nous avons résumé tous les résultats des tests dans la figure.(5.7). Dans le but de simplifier la visualisation des résultats de la classification obtenue ; la moyenne de tous les indices ont été calculées.

5.3.2 Évaluation sur des images

Plusieurs images réelles ont été utilisées pour évaluer l'algorithme ALISSC. Avant d'appliquer les méthodes de classification, il faut d'abord extraire les sous espaces. Cinq paramètres de Haralick [79] sont utilisés. Le contraste, l'homogénéité, la corrélation, l'entropie et l'énergie ont été extraits par la matrice de co-occurrence. Ils constituent les premiers sous espaces utilisés. Le niveau de gris d'un pixel ainsi que filtre de Canny pour détecter les contours sont utilisés. On obtient ainsi sept sous espaces pour chaque image

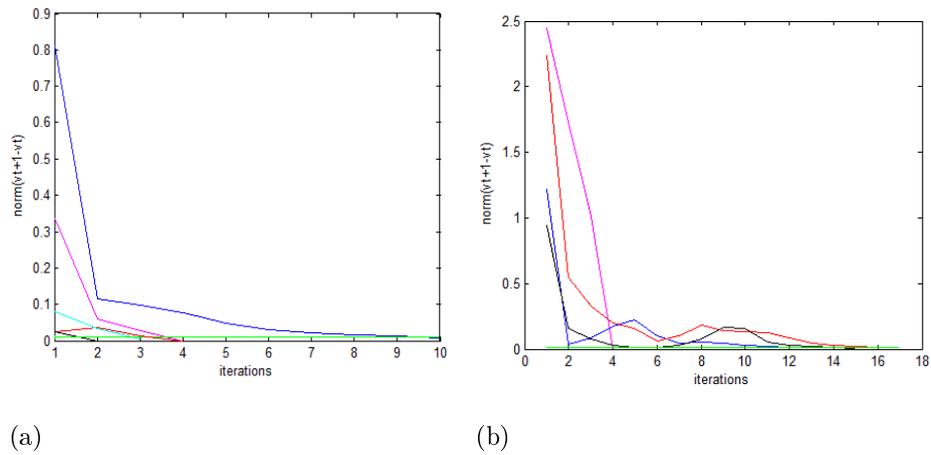


FIGURE 5.6 – Convergence des méthodes, (a) : l’algorithme ALISSC , (b) : l’algorithme ESSC

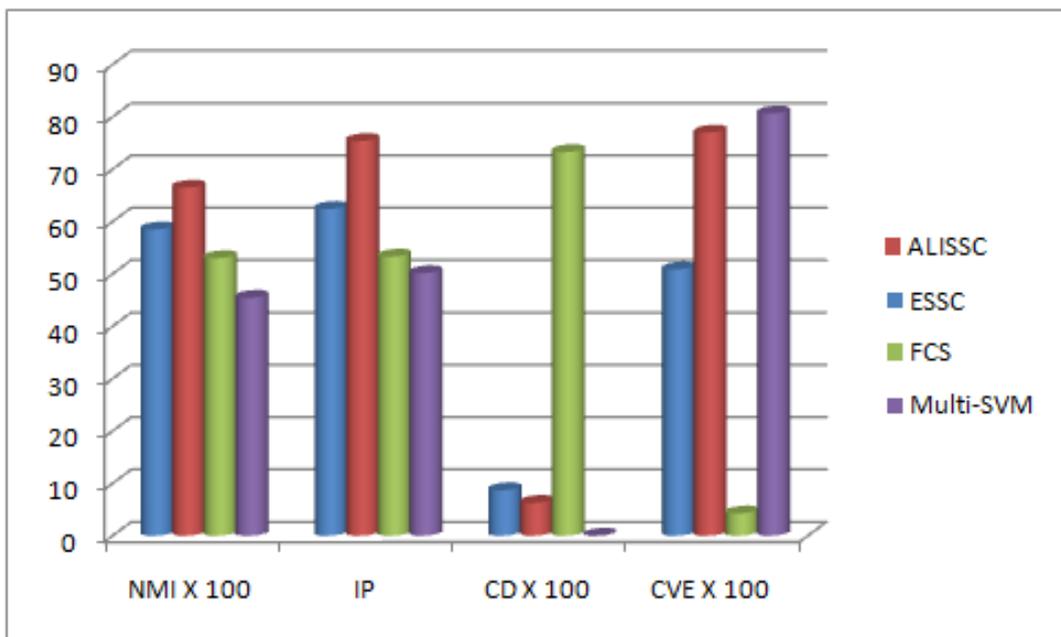


FIGURE 5.7 – Moyenne des résultats de la classification

testée.

Le nombre de classe est initialisé à la même valeur pour tous les algorithmes testés. Les résultats de la classification sont représentés par la Figure.(5.8).

Généralement, on peut remarquer que les formes des différentes classes obtenues par ALISSC sont similaires à celles obtenues par ESSC et FCS. Les Figure.(5.8.b), (5.8.c) et Figure.(5.8.d) sont les résultats de la classification de l'image de synthèse obtenus par ALISSC, ESSC et FCS respectivement. Le nombre de classes est initialisé à $c=5$. On remarque que le triangle et le fond de l'image sont fusionnés dans les résultats obtenus par ESSC et FCS. Les différentes classes présentées par la Figure(5.8.b) sont bien détectées et elles sont bien séparées.

Les résultats de la classification des images réelles sont représentés à partir de la Figure(5.8.f) Le nombre de classes pour la figure.(5.8.e) est égal à 6. $C=7$ pour les Figures.(5.8.i) et la figure.(5.8.m). Pour la Figure(5.8.q) le nombre de classe est 5. Dans la Figure. (5.8.g) la forme du rein droit n'est pas respectée, dans la Figure(5.8.h) le rein gauche (1) n'est pas détecté. Tous les organes sont bien détectés dans le résultat obtenu par ALISSC (Figure.(5.8.f)). Dans la Figure.(5.8.k), le rein (5) et la colonne vertébrale sont confondus dans une même classe. La veine rénale(4) et le rein (5) sont aussi fusionnés alors qu'ils sont bien séparés dans la Figure.(5.8.j) (flèche bleu). Le foie (6), la colonne vertébrale (7) et la rate (8) sont fusionnés dans une même classe. La veine rénale aussi(10) dans la partie haute de l'abdominale n'est pas détectée dans la figure(5.8.o) mais ils sont présentés clairement dans la figure.(5.8.n), la veine(9) est bien limité (flèche verte). Dans la figure.(5.8.r) toutes les classes sont bien détectées.

*** La classification des images dans les différents sous espaces**

Elle permet de détecter les différentes classes qui sont cachées dans les sous espaces.

Résultats sur des images de synthèse : Plusieurs images ont été utilisées pour évaluer et démontrer l'intérêt de la classification dans les sous espaces. Nous présentons une image de synthèse avec trois classes. La Figure.(5.9)représente les différents résultats de la classification dans différents sous espaces. Sur cette image de synthèse,

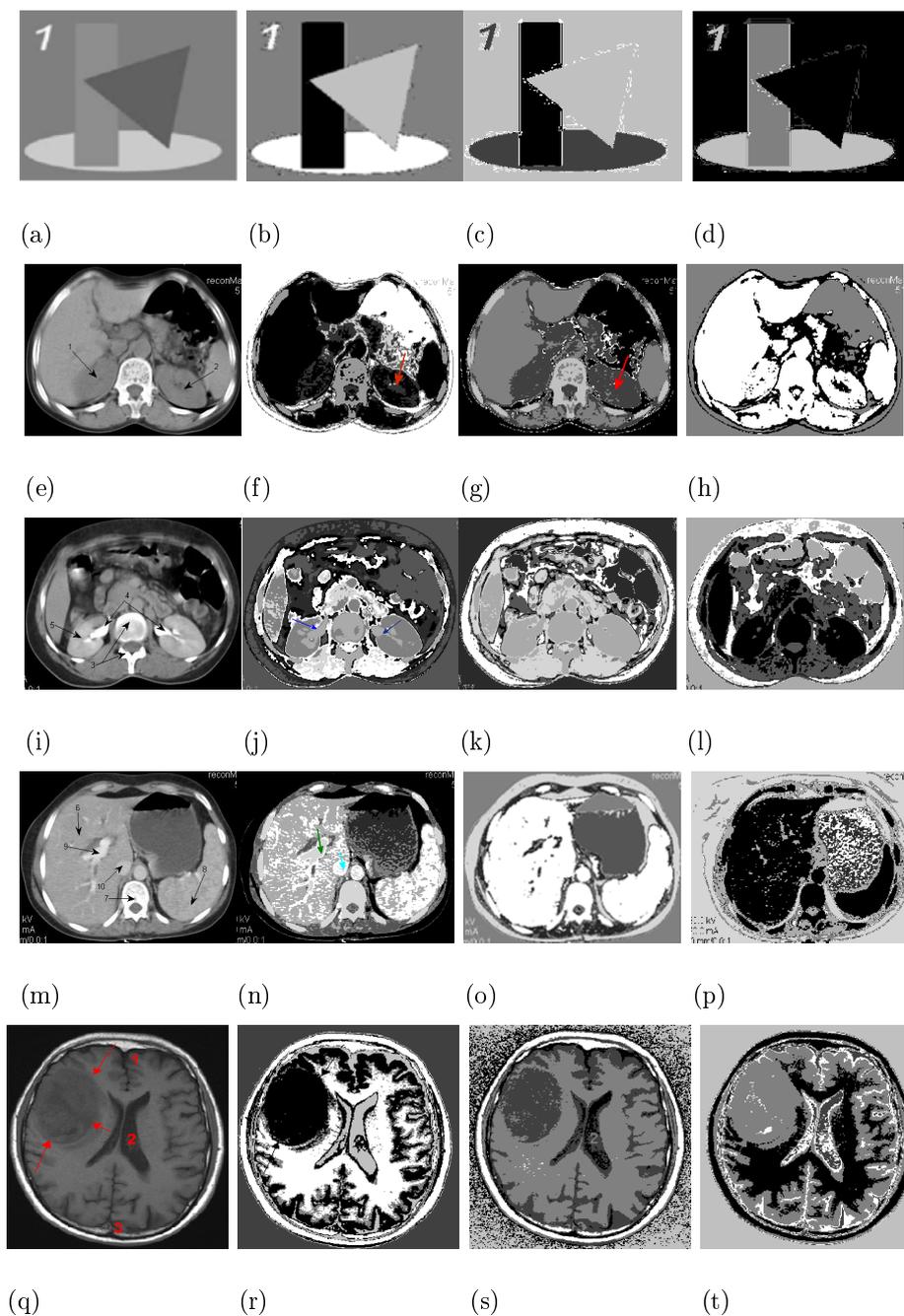


FIGURE 5.8 – Résultats de la classification obtenus, 2^{eme} colonne par ALISSC, 3^{eme} colonne par ESSC, 4^{eme} colonne par FCS

chaque sous espace détecte clairement une classe. Après l’obtention des résultats de la classification dans chaque sous espace, les résultats sont fusionnés Figure.(5.10).

Les images présentées par la figure.(5.9) montrent que dans chaque sous espace on obtient des classes différentes aux autres sous espaces. Dans le sous espace 1 une

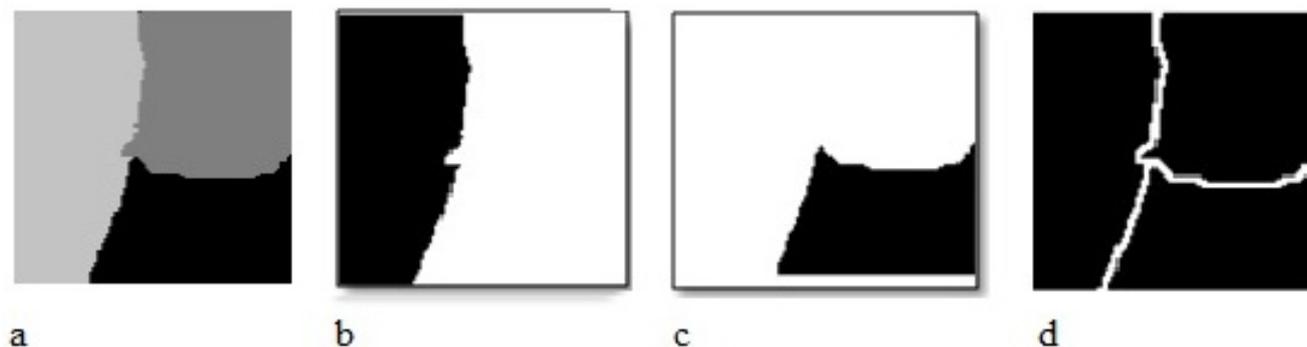


FIGURE 5.9 – a :image original, b,c,d : différents résultats de la classification dans les sous espaces

classe a été détectée figure.(5.9.b) et les autres sont masquées. Elles sont présentées dans le 2^{eme} sous espace (Figure.(5.9.b). Le dernier sous espace ne détecte que les contours de l'image (Figure.(5.9.d)). Les classes sont rassemblées dans la Figure.(5.10). Toutes les classes sont bien classifiées et que leurs contours sont bien limités.



FIGURE 5.10 – Classification finale

Application sur des images médicales Plusieurs images médicales sont utilisées pour tester cette méthode, précisément des images de scanner. Les renseignements apportés par cet examen sont très souvent déterminants pour porter un diagnostic précis. Certains examens nécessitent, selon les cas, une injection intraveineuse, le plus souvent au pli du coude, parfois de boire une boisson ou d'avoir un lavement. Des produits à base d'iode, opaques aux rayons X, seront utilisés. Le type de ces produits et leurs modalités d'administration seront adaptés au cas du patient par le médecin radiologue. Ce choix dépendra notamment des informations sur l'état de santé du patient qu'il aura donné avant l'examen. L'injection intra-veineuse du

produit iodé permet de mieux étudier les vaisseaux, l'appareil urinaire et de mieux distinguer les tissus anormaux des tissus normaux.

Dans ce travail nous avons proposé une méthode pour réduire les risques et les effets indésirable de l'injection intra-veineuse de produit iodé. Deux types d'images du scanner sont utilisés ; images du scanner avec et sans injection de produit de contraste. Ces images seront classifiées et les résultats seront comparés.

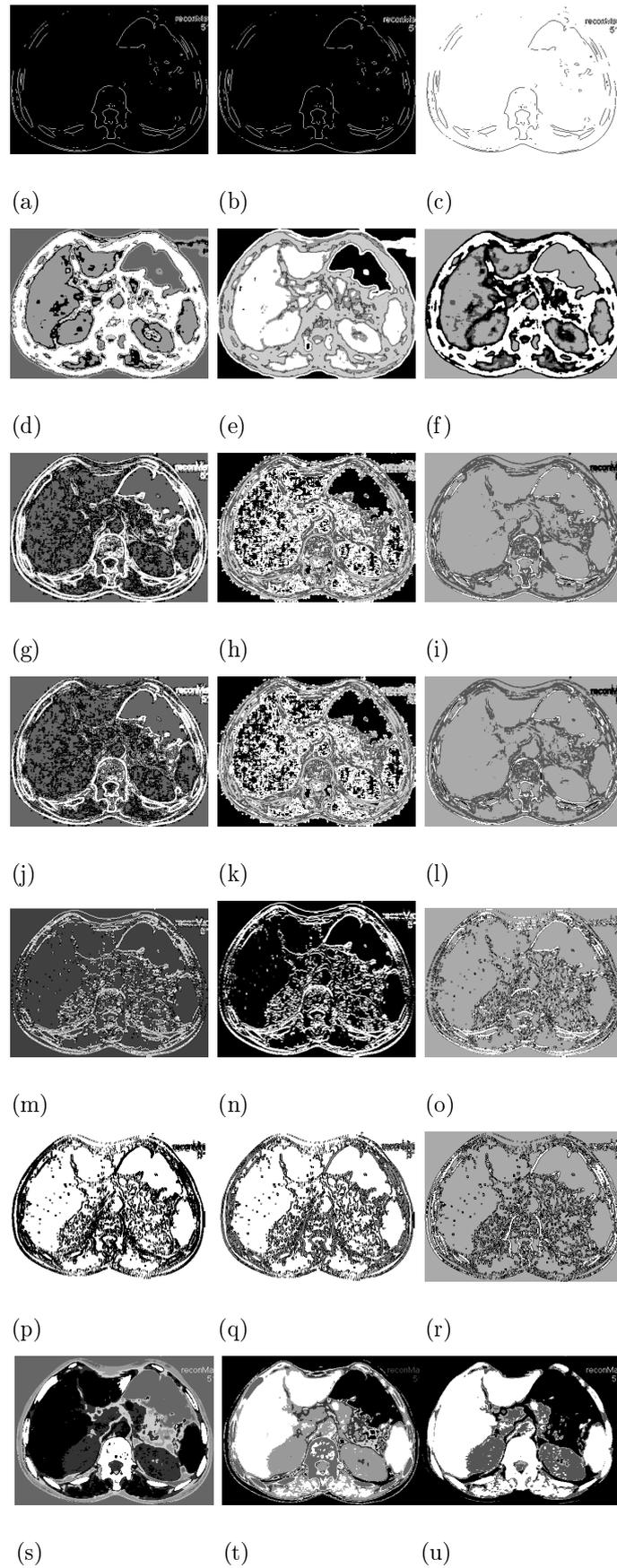


FIGURE 5.11 – Résultats dans les sept sous espaces, ALISSC (1^{er} colonne), ESSC (2^{eme} colonne) et FCS (3^{eme} colonne)

On remarque dans un sous espace que les deux reins, le foie, la rate, l'estomac sont les seuls organes représentés, les cotes et la colonne vertébrale sont masquées (Figure.(5.11.d)). Dans un autre sous espace on remarque que les cotes et la colonne vertébrale sont les seuls organes représentés et les autres sont masqués Figure.(5.11.a).

Les résultats des différentes sous espaces ont été fusionnés pour obtenir la classification finale que nous allons comparer avec l'image du scanner après injection du produit de contraste.

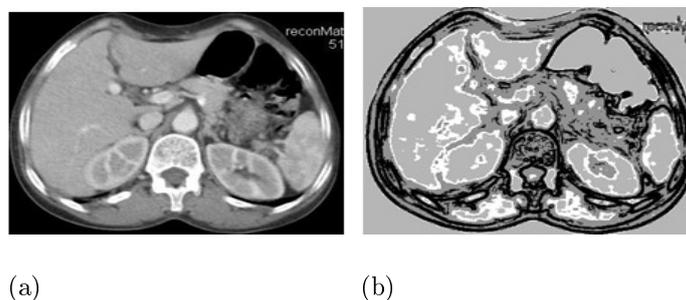


FIGURE 5.12 – Etude comparative,(a) : image de scanner après injection de produit , (b) : résultat final de la classification

On remarque que les organes repérés par les flèches sont bien détectés et bien séparés des autres organes dans le résultat de la classification alors qu'ils sont mal séparés dans l'image originale Figure.(5.12).

5.4 Classification utilisant le modèle autorégressif bidimensionnel à coefficients aléatoires 2D-RCA

La base de données (Brodatz, [99]) a été utilisée dans les tests préliminaires. Pour évaluer les performances de l'approche proposée, nous avons comparé les résultats obtenus avec les caractéristiques extraites par le modèle AR [90] et la matrice co-occurrence [79] utilisant les mêmes algorithmes de classification. Quatre paramètres de la matrice co-occurrence ont été utilisés ; le contraste, l'homogénéité, la corrélation et l'entropie. La taille de la fenêtre que nous avons choisie est de 80x 80 pixels. Les résultats obtenus sont

représentés par la Figure.(5.13). Le taux de classification a été calculé aussi pour différents textures et méthodes. Les résultats sont présentés par le tableau Table.(5.12).

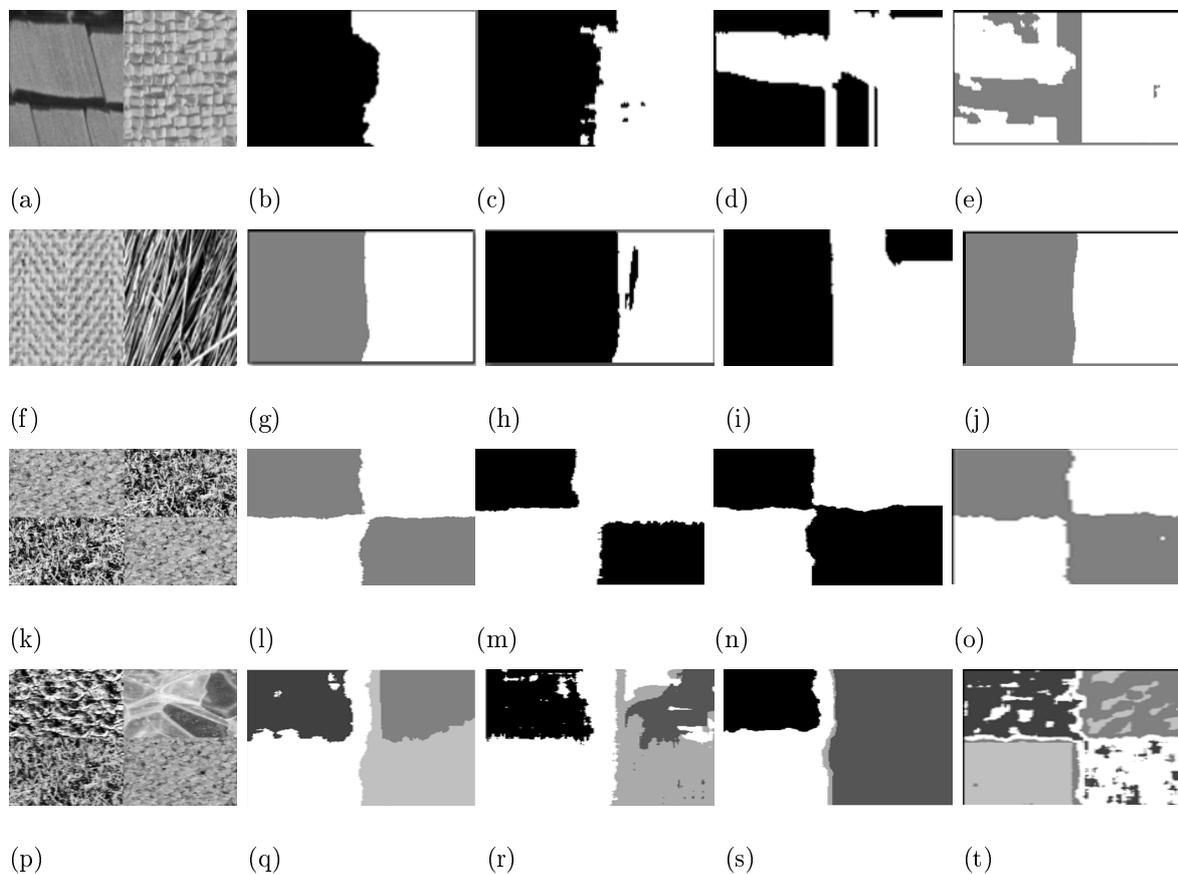


FIGURE 5.13 – Images originales(1^{er} colonne). Résultats du modèle 2D-RCA (2^{ème} colonne). Résultats du SVM sur le modèle 2D-RCA (3^{ème} colonne). Résultats du calcul de la distance sur les matrices de cooccurrence (4^{ème} colonne). Résultats du modèle AR(5^{ème} colonne).

Les résultats de la classification illustrés par la Figure.(5.13) montrent que les résultats obtenus par l'algorithme basé sur le calcul de la distance et l'algorithme basé sur les SVM utilisant les paramètres du 2D-RCA sont les meilleurs pour différents textures par rapport aux autres algorithmes. Le modèle AR donne de bons résultats pour certaines textures seulement, ce qui signifie que les paramètres extraits par ce modèle ne sont pas toujours pertinents et que le modèle AR ne peut pas décrire tout type de textures à l'opposé du modèle 2D-RCA qui présente une généricité par rapport aux textures ; d'où une meilleure

classification.

Nous avons étudié la robustesse de cette approche par rapport au bruit. Des images de textures bruitées ont été testées. Les résultats sont présentés par la Figure.(5.14)

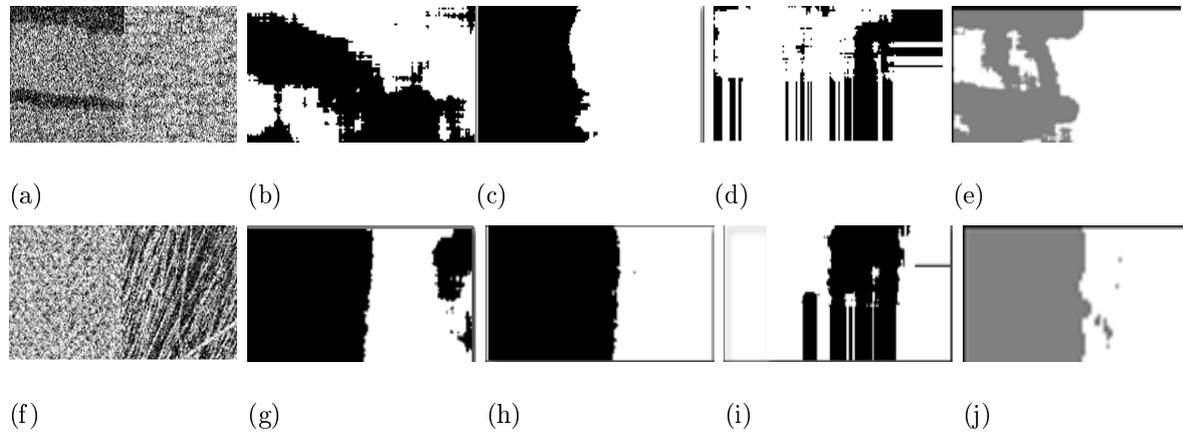


FIGURE 5.14 – Images bruitées(1^{er} colonne).Résultats du modèle 2D-RCA (2^{eme} colonne). Résultats du SVM sur le modèle 2D-RCA (3^{eme} colonne). Résultats du calcul de la distance sur les matrices de cooccurrence (4^{eme} colonne). Résultats du modèle AR(5^{eme} colonne).

Nous avons remarqué que l’algorithme SVM donne de très bon résultats par rapport aux autres algorithmes. L’algorithme basé sur le calcul de la distance avec 2D-RCA est influencé par le bruit pour certaines textures.

TABLE 5.12 – Taux de classification

images	calcul par distance			SVM	
	2D-RCA	2D-AR	matrice Co-occurrence	2D-RCA	2D-AR
Fig.5.13.a	95.45	95.28	73.52	95.95	94.56
Fig.5.13.f	97.5	90.41	90.41	90.68	95.44
Fig.5.13.k	97.5	81.33	94.5	95.00	55.12
Fig.5.13.p	87.06	55.9	72.25	73.57	65.36
Fig.5.14.a	63.39	63.21	67.75	94.18	94.15
Fig.5.14.f	87.72	56.6	50.95	92.51	67.75

5.4.1 Validation croisée

La validation croisée est utilisé pour valider les résultats de cette approche. Douze textures sont testées. La validation croisée par la méthode k -fold est aussi utilisée. Dans ce cas k est fixé à 10. Le partitionnement à 10 dossiers de chaque texture parmi les 12 présentées par la Figure.(5.15) a été fait. Le même principe utilisé auparavant est appliqué. Les résultats sont présentés sur Table.(5.13).

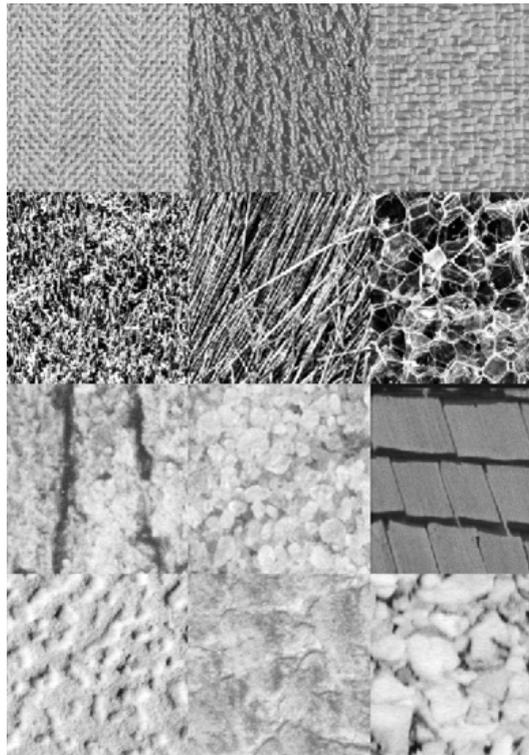


FIGURE 5.15 – Douze images de textures

TABLE 5.13 – Validation croisée pour douze textures

textures	2D-RCA		matrice Co-occurrence		SVM	
	means	std	means	std	means	std
texture1	0.8809	8.5×10^{-4}	0.9167	0.005	0.9412	3.595×10^{-5}
texture2	0.999	0.0017	0.08	0.001	0.9023	1.03×10^{-5}
texture3	0.8940	0.0555	0.547	0.117	0.9517	3.751×10^{-6}
texture4	0.999	5.77×10^{-4}	0.8268	1.25×10^{-4}	0.9109	0.01×10^{-5}
texture5	0.991	2.12×10^{-4}	0.594	0.1076	0.9167	0.2×10^{-6}
texture6	0.999	5.86×10^{-4}	0.8365	0.04	0.9236	1.01×10^{-5}
texture7	0.7067	0.0095	0.8150	0.0968	0.9269	1.538×10^{-5}
texture8	0.7709	0.0012	0.9167	1.52×10^{-4}	0.9610	1.944×10^{-5}
texture9	0.8700	0.0024	0.9168	1.6×10^{-4}	0.9871	1.242×10^{-5}
texture10	0.5142	0.0016	0.1914	0.065	0.9706	9.59×10^{-5}
texture11	0.6786	4.94×10^{-4}	0.595	0.1416	0.9443	1.205×10^{-5}
texture12	0.8559	0.0037	0.915	0.4×10^{-4}	0.9655	6.91×10^{-5}

La moyenne et la déviation standard de la classification est calculée pour chaque résultat sur les différentes textures. On peut remarquer que les meilleures valeurs sont données par l'algorithme proposé et l'algorithme SVM. Un taux de classification arrive jusqu'à 99% pour la texture numéro 6 avec une déviation standard très faible.

5.4.2 Textures générées par le modèle 2D-RCA

Différentes textures ont été obtenues utilisant le modèle 2D-RCA avec différentes valeurs de α , β et γ représentées sur Table.(5.14). La Figure(5.16) montre huit images générées par ce modèle.

TABLE 5.14 – Différents paramètres du modèle 2D-RCA utilisé pour générer les textures

Textures	a	b	c	d	e	f	j	h
α	0.04	0.09	0.3	0.7	0.1	0.7	0.4	0.3
β	0.8	0.1	0.32	0.12	0.01	0.2	0.3	0.32
γ	0.06	0.7	0.39	0.029	0.09	0.09	0.1	0.39
η_1	0.2	0.2	0.001	0.2	0.2	0.2	0.2	0.2
η_2	0.2	0.2	0.001	0.2	0.2	0.2	0.2	0.2
η_3	0.2	0.2	0.001	0.2	0.2	0.2	0.2	0.2

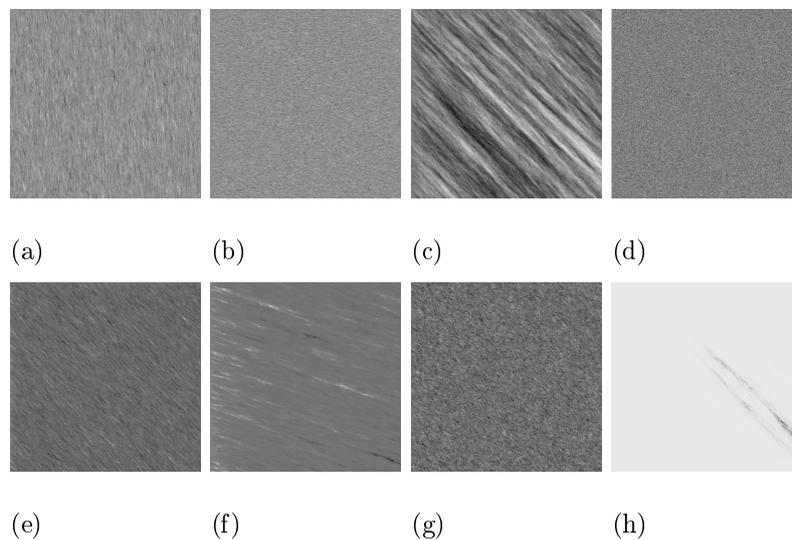


FIGURE 5.16 – Huit textures générées par 2D-RCA

La Figure.(5.16) Montre différents types de textures générées ; des textures directives, grossières et aussi fines. On peut justifier la différence entre les textures obtenues par le choix des valeurs des paramètres 2D-RCA utilisées pour chaque expérience.

- Si la variance de μ_i , $i = 1..3$ est petite par rapport au constantes α, β et γ les textures générées ressemblent à des textures générées par des coefficients auto régressifs constants. Cependant, si la variance est plus grande que les autres paramètres, les textures présentent certaines valeurs d'intensité plus élevées Figure.(5.16.c). Particulièrement, si les valeurs de α, β et γ sont proches de zéro, les textures obtenues ressemble à du bruit Figure.(5.16.e).

- Les textures obtenues sont soit directives si une des trois valeurs α , β et γ est large par rapport aux autres. En effet si la valeur de $\beta \gg \alpha$ et γ , la texture obtenue suit une direction verticale Figure.(5.16.a), Alors quand la valeur de α est très grande, la texture synthétique suit une direction horizontale Figure.(5.16.d). Enfin si la valeur de γ est très grande, nous avons une direction oblique (voir la Figure.(5.16.b)).

5.4.3 Mesure de similarité entre les textures générées et des textures réelles

Pour tester la similarité des textures obtenues par le modèle 2D-RCA nous avons utilisé deux indices de similarité; SSIM et GSIM. 112 textures sont classées dans 32 classes. Chaque classe contient entre 2 et 6 textures similaires. Cette classification a été faite manuellement par des chercheurs dans un laboratoire de recherche en vision [100]. Chaque texture parmi les huit générées a été comparée avec les 112 textures utilisant SSIM et GSIM. Table.(5.15) et la Figure.(5.17) présentent les taux les plus élevés et les textures correspondantes. Par exemple, les textures (a et b) sont similaires aux images D8 et D21 selon les mesures.

TABLE 5.15 – Meilleures mesures de similarité

Images	Indice	Images similaires	Résultats
Textures a	SSIM	D8.gif	0.9219
	GSIM	D21.gif	0.9771
Textures b b	SSIM	D8.gif	0.9373
	GSIM	D21.gif	0.9534
Textures c	SSIM	D71.gif	0.9412
	GSIM	D21.gif	0.9750
Textures d	SSIM	D71.gif	0.957
	GSIM	D21.gif	0.9692
Textures e	SSIM	D71.gif	0.9197
	GSIM	D21.gif	0.9647
Textures f	SSIM	D8.gif	0.9207
	GSIM	D21.gif	0.9727
Textures g	SSIM	D71.gif	0.9572
	GSIM	D21.gif	0.9690
Textures h	SSIM	D96.gif	0.9395
	GSIM	D48.gif	0.9395

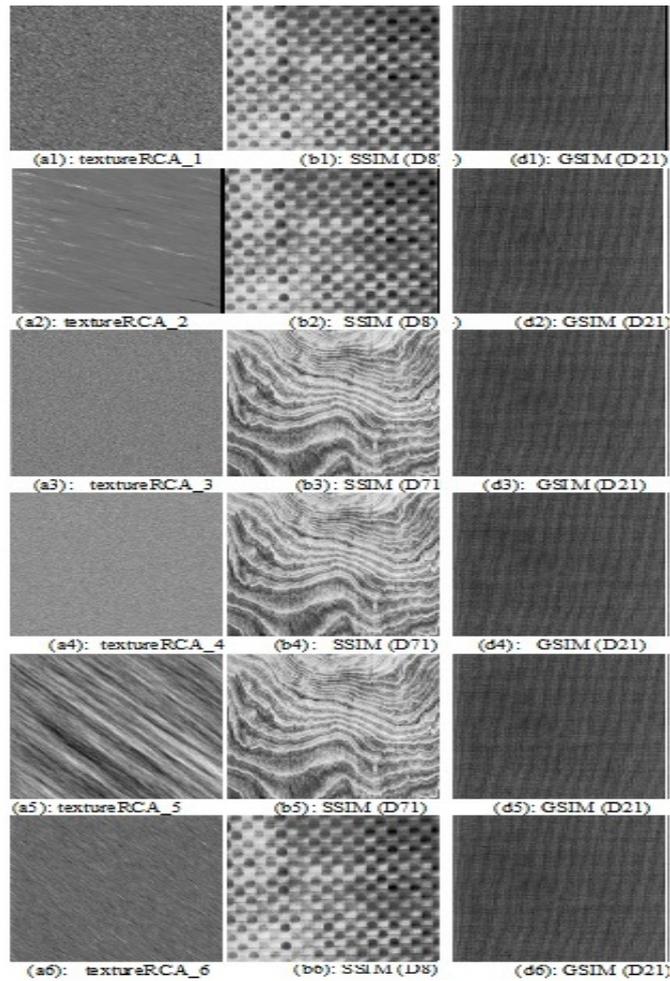


FIGURE 5.17 – Textures similaires aux textures générées selon SIM et GSIM

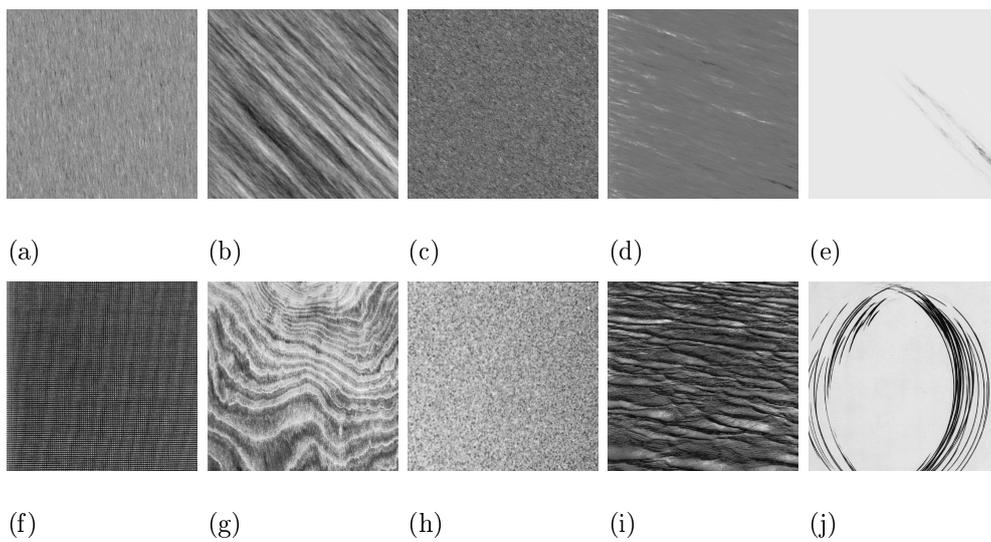


FIGURE 5.18 – Textures similaires

La Figure.(5.18) montre les textures générées qui sont similaires aux textures de la base Brodatz selon l'observation humaine. Quatre textures générées ont été identifiées comme des textures réelles. On remarque que la texture "a" est similaire à la toile française présentée par la texture D21, la texture "c" est similaire à la texture D72 (bois d'un arbre), texture "f" est similaire à l'eau, la texture "g" est similaire au sable et la texture "h" est similaire à D44 (lumière d'une pièce sombre). Ces résultats confirment la capacité de ce modèle de représenter différents types de textures

5.5 Conclusion

Dans ce chapitre nous avons présenté tous les résultats des approches proposées. La classification proposée dans les sous espaces montre une grande amélioration dans différents type de bases. L'amélioration de l'étape d'initialisation permet une stabilisation des résultats ainsi qu'une automatisation des paramètres initialisés.

L'apprentissage actif pour l'algorithme SVM dans chaque itération permet de construire une nouvelle formulation du degré d'appartenance et les centres des classes. Ainsi la localisation des centres des classes a été améliorée. Le nombre d'itérations a été diminué ce qui conduit à la convergence rapide de la méthode. Cette méthode présente de grands avantages surtout dans la classification des images médicales. Notons qu'elle est générique et elle peut être appliquée à n'importe quel problème en classification dans les sous espaces.

Les résultats de la classification supervisée utilisant le modèle 2D-RCA ont montré que les paramètres de ce modèle permettent d'extraire une information pertinente qui facilite les étapes de la classification même en présence de bruit. La stabilité des résultats de la classification sur des images bruitées confirme l'efficacité de ce modèle.

Conclusion et perspectives

Dans ce travail, nous avons abordé les problèmes de la classification des données. L'une des questions principales que nous avons considérée concerne la classification des données dans les grands espaces. L'extraction et l'exploitation de l'information dans les espaces des données de grande dimension est difficile, sachant que ces données peuvent être bien représentées dans un espace de dimensions intrinsèques plus petites que l'espace original. Pour répondre à cette question nous avons développé une méthode efficace opérant la classification dans les sous espaces. Dans de nombreux algorithmes de classification basés sur une étape d'initialisation arbitraire, il se pose le problème de la localisation des centres de classes, en terme d'estimation des distances inter et intra classes. En effet l'estimation des distances se base sur le centre moyen des centres à travers les itérations de l'algorithme. Par conséquent l'erreur d'estimations des distances est propagée et amplifiée à travers les itérations. Ceci affaiblit le taux de classification et nécessite un temps de traitement élevé. Dans l'objectif d'améliorer les performances de tels algorithmes, une nouvelle fonction objective a été proposée dans laquelle la densité a été introduite pour permettre de détecter différentes formes de classes. L'algorithme SVM est utilisé pour estimer les paramètres initiaux de la classification de façon automatique. Nous avons utilisé l'apprentissage actif pour l'algorithme SVM utilisé dans le processus de classification afin d'accélérer la convergence de la méthode et améliorer la qualité de la classification accompagnée de la réduction du coût de calcul.

La classification des textures reste une tâche nécessaire pour l'exploitation des images. Donc, il est primordial de chercher des méthodes rapides pour caractériser ces textures. Pour améliorer la qualité de la classification des textures, nous avons développé une mé-

thode pour l'extraction des caractéristiques de texture . Nous avons développé différents algorithmes utilisant le modèle spatial non linéaire qui généralise au cadre spatial bidimensionnel le modèle autorégressif à coefficients aléatoires (RCA) standard ; noté 2D-RCA. Ce modèle permet d'extraire des informations pertinentes qui facilitent les étapes de la classification même avec des images bruitées.

En perspectives pour des travaux futurs, plusieurs axes seront abordés tels que l'amélioration de l'étape d'initialisation par hybridation de l'approche proposée avec des métaheuristiques d'optimisation et introduction de l'information spatiale dans la dernière étape de la classification. La gestion de l'information texturale de l'image peut aider à apporter des modifications dans le calcul des coefficients pour le modèle 2D-RCA. Le modèle 2D-RCA du second ordre peut être envisagé.

Annexe

(Ω, \mathbf{A}, P) : espace de probabilité : est un triplet formé d'un ensemble Ω appelé aussi espace fondamental ou univers), d'une tribu ou σ -algebre A sur $(\Omega$ et d'une mesure P sur cette σ -algebre tel que $P(\Omega)=1$.

Série chronologique (ou série temporelle) : Une série chronologique, est une suite de variables aléatoires (ou de valeurs numérique) représentant l'évolution d'une quantité spécifique au cours du temps.

L'inférence statistique Consiste à tirer des conclusion concernant un groupe auquel on ne peut pas accéder directement (généralement parce qu'il est trop grand) à partir d'un sous groupe auquel on a accès et que l'on considère comme un échantillon aléatoire provenant de cette population.

Convergence faible : Convergence en loi

Régression linéaire : Un modèle de régression linéaire est un modèle qui explique les variation d'une variable expliquée sur une ou plusieurs variable explicatives dans lequel on fait l'hypothèse que la fonction qui relie les variables explicatives à la variable expliquée est linéaire dans ses paramètres.

Champ aléatoire : un champ aléatoire X est un processus prenant des valeurs aléatoires sur le réseau \mathbb{R}^2 (ou \mathbb{Z}^2). Lorsque $d=2$ et le champ aléatoire $Xx(i, j)$ est définie sur une grille, le dernier représente une vraie image mais qui est non observable où $X(i,j)$ est le niveau de grille à chaque position du pixel(i,j).

Mesure de Lebeque :ensemble mesurable au sens de Lebesgue.

Champ de carré intégrable :Champ aléatoire $(X_t, t \in D)$ tel que $\int |X_t|^2 d\mu < \infty(D \subset \mathbb{R}^d(\text{ou } \mathbb{Z}^d))$.

Distribution fini-dimensionnelle : loi gégrant un vecteur de variable aléatoire de dimension finie.

Bibliographie

- [1] L. Parsons, E. Haque, H. Liu, *et al.*, “Evaluating subspace clustering algorithms.” *In Workshop on Clustering High Dimensional Data and its Applications. SIAM Int. Conf. on Data Mining*, 2004.
- [2] P. Berkhin, “A survey of clustering data mining techniques,” in *Grouping Multidimensional Data - Recent Advances in Clustering*, pp. 25–71, 2006.
- [3] D. M. Kanungo, Tapas et Mount, C. D. Netanyahu, Nathan S et Piatko, R. Silverman, and A. Y. Wu, “An efficient k-means clustering algorithm : Analysis and implementation,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 7, pp. 881–892, 2002.
- [4] H.-S. Park and C.-H. Jun, “A simple and fast algorithm for k-medoids clustering,” *Expert Systems with Applications*, vol. 36, no. 2, pp. 3336–3341, 2009.
- [5] B. James C, R. Ehrlich, and W. Full, “Fcm : The fuzzy c-means clustering algorithm,” *Computers and Geosciences*, vol. 10, no. 2, pp. 191–203, 1984.
- [6] T. Kohonen, “Self-organizing maps,” *Springer Series in Information Sciences*, 1997.
- [7] R. Bellman, “Dynamic programming treatment of the travelling salesman problem,” *Journal of the ACM (JACM)*, vol. 9, no. 1, pp. 61–63, 1962.
- [8] T. Pavlenko, “On feature selection, curse-of-dimensionality and error probability in discriminant analysis,” *Journal of Statistical Planning and Inference*, vol. 115, no. 2, pp. 565–584, 2003.
- [9] T. Pavlenko and D. Von Rosen, “Effect of dimensionality on discrimination,” *Statistics*, vol. 35, no. 3, pp. 191–213, 2001.
- [10] C. C. Aggarwal, A. Hinneburg, and D. A. Keim, *On the surprising behavior of distance metrics in high dimensional space*. Springer, 2001.

- [11] C. Bouveyron, *Modélisation et classification des données de grande dimension : application à l'analyse d'images*. PhD thesis, Grenoble 1, 2006.
- [12] L. Bai, J. Liang, C. Dang, and F. Cao, "A novel attribute weighting algorithm for clustering high-dimensional categorical data," *Pattern Recognition*, vol. 44, no. 12, pp. 2843–2861, 2011.
- [13] M. Ouattara, *Development and implementation of a multi-block clustering methods : apply to OQAI data sets*. Theses, Conservatoire national des arts et metiers - CNAM, Mar. 2014.
- [14] I. K. Fodor, "A survey of dimension reduction techniques," 2002.
- [15] M. A. Carreira-Perpinán, "A review of dimension reduction techniques," *Department of Computer Science. University of Sheffield. Tech. Rep. CS-96-09*, vol. 9, pp. 1–69, 1997.
- [16] I. T. Jolliffe, "Principal component analysis : a beginner's guide ii. pitfalls, myths and extensions," *Weather*, vol. 48, no. 8, pp. 246–253, 1993.
- [17] M. Gastpar, P. L. Dragotti, and M. Vetterli, "The distributed karhunen–loève transform," *Information Theory, IEEE Transactions on*, vol. 52, no. 12, pp. 5177–5196, 2006.
- [18] K. Pearson, "Liii. on lines and planes of closest fit to systems of points in space," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 2, no. 11, pp. 559–572, 1901.
- [19] H. Hotelling, "Analysis of a complex of statistical variables into principal components.," *Journal of educational psychology*, vol. 24, no. 6, p. 417, 1933.
- [20] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural computation*, vol. 10, no. 5, pp. 1299–1319, 1998.
- [21] S. Girard, "A nonlinear pca based on manifold approximation," *Computational Statistics*, vol. 15, no. 2, pp. 145–167, 2000.
- [22] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.
- [23] J. B. Tenenbaum, V. De Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

- [24] C. Jutten and J. Herault, “Blind separation of sources, part i : An adaptive algorithm based on neuromimetic architecture,” *Signal processing*, vol. 24, no. 1, pp. 1–10, 1991.
- [25] R. A. Fisher, “The use of multiple measurements in taxonomic problems,” *Annals of eugenics*, vol. 7, no. 2, pp. 179–188, 1936.
- [26] I. Guyon and A. Elisseeff, “An introduction to variable and feature selection,” *The Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.
- [27] Y. Slimani, M. L. Essegir, Mohamed Amir et Samb, F. Camara, and S. Ndiaye, “Approche de sélection d’attributs pour la classification basée sur l’algorithme rfe-svm,” *Revue Africaine de la Recherche en Informatique et Mathématiques Appliquées*, vol. 17, pp. 197–219, 2014.
- [28] H. Zhao, Zheng et Liu, “Spectral feature selection for supervised and unsupervised learning,” in *Proceedings of the 24th international conference on Machine learning*, pp. 1151–1157, ACM, 2007.
- [29] H. Dash, Manoranjan et Liu, “Feature selection for clustering,” in *Knowledge Discovery and Data Mining. Current Issues and New Applications*, pp. 110–121, Springer, 2000.
- [30] V. Vapnik, “An overview of statistical learning theory,” *IEEE Trans. on Neural Networks*, 1999.
- [31] C. O. W. J. e. E. A. Lal, Thomas Navin, “Embedded methods,” in *Feature extraction*, pp. 137–165, Springer, 2006.
- [32] S. Bouveyron, Charles et Girard, “Classification supervisée et non supervisée des données de grande dimension,” *La revue de Modulad*, vol. 40, pp. 81–102, 2009.
- [33] G. Celeux, Gilles et Govaert, “Gaussian parsimonious clustering models,” *Pattern recognition*, vol. 28, no. 5, pp. 781–793, 1995.
- [34] A. E. Fraley, Chris et Raftery, “Model-based clustering, discriminant analysis, and density estimation,” *Journal of the American statistical Association*, vol. 97, no. 458, pp. 611–631, 2002.
- [35] M. S. et Oliveira.A.L, “Biclustering algorithms for biological data analysis : a survey,” *IEEE Transactions on Computational Biology and Bioinformatics*, 2004.
- [36] e. a. Busygin.S, “Biclustering in data mining,” *Computers and Operations Research*, 2008.

- [37] P. e. a. Bhatia, “blockcluster : An r package for model based co-clustering,” 2014.
- [38] J. A. Hartigan, “Direct clustering of a data matrix,” *Journal of the american statistical association*, vol. 67, no. 337, pp. 123–129, 1972.
- [39] K. e. a. Sim, “A survey on enhanced subspace clustering,” *Data mining and knowledge discovery*, vol. 26, no. 2, pp. 332–397, 2013.
- [40] Z. D. et al, “A survey on soft subspace clustering,” *Information Sciences*, vol. 348, pp. 84 – 106, 2016.
- [41] P. Favaro and A. Vidal, René et Ravichandran, “A closed form solution to robust subspace estimation and clustering,” in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pp. 1801–1807, IEEE, 2011.
- [42] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan, *Automatic subspace clustering of high dimensional data for data mining applications*, vol. 27. ACM, 1998.
- [43] C.-H. Cheng, A. W. Fu, and Y. Zhang, “Entropy-based subspace clustering for mining numerical data,” in *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 84–93, ACM, 1999.
- [44] C. C. Aggarwal, J. L. Wolf, P. S. Yu, C. Procopiuc, and J. S. Park, “Fast algorithms for projected clustering,” in *ACM SIGMoD Record*, vol. 28, pp. 61–72, ACM, 1999.
- [45] C. C. Aggarwal and P. S. Yu, *Finding generalized projected clusters in high dimensional spaces*, vol. 29. ACM, 2000.
- [46] J. H. Friedman and J. J. Meulman, “Clustering objects on subsets of attributes (with discussion),” *Journal of the Royal Statistical Society : Series B (Statistical Methodology)*, vol. 66, no. 4, pp. 815–849, 2004.
- [47] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, “A density-based algorithm for discovering clusters in large spatial databases with noise.,” in *Kdd*, vol. 96, pp. 226–231, 1996.
- [48] J. Z. Huang, M. K. Ng, H. Rong, and Z. Li, “Automated variable weighting in k-means type clustering,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, no. 5, pp. 657–668, 2005.
- [49] L. Jing, M. K. Ng, and J. Z. Huang, “An entropy weighting k-means algorithm for subspace clustering of high-dimensional sparse data,” *Knowledge and Data Engineering, IEEE Transactions on*, vol. 19, no. 8, pp. 1026–1041, 2007.

- [50] C. Domeniconi, D. Gunopulos, S. Ma, B. Yan, M. Al-Razgan, and D. Papadopoulos, "Locally adaptive metrics for clustering high dimensional data," *Data Mining and Knowledge Discovery*, vol. 14, no. 1, pp. 63–97, 2007.
- [51] X. Chen, Y. Ye, X. Xu, and J. Z. Huang, "A feature group weighting method for subspace clustering of high-dimensional data," *Pattern Recognition*, vol. 45, no. 1, pp. 434–446, 2012.
- [52] Z. Deng, K.-S. Choi, F.-L. Chung, and S. Wang, "Enhanced soft subspace clustering integrating within-cluster and between-cluster information," *Pattern Recognition*, vol. 43, no. 3, pp. 767–781, 2010.
- [53] R. W. Sembiring and J. M. Zain, "Cluster evaluation of density based subspace clustering," *arXiv preprint arXiv :1012.6009*, 2010.
- [54] S. Jahirabadkar and P. Kulkarni, "IsC-intelligent subspace clustering, a density based clustering approach for high dimensional dataset," *World Academy of Science, Engineering and Technology*, vol. 55, pp. 69–73, 2009.
- [55] D. R. Edla, P. K. Jana, and I. S. Member, "A prototype-based modified dbscan for gene clustering," *Procedia Technology*, vol. 6, pp. 485–492, 2012.
- [56] K.-L. Wu, J. Yu, and M.-S. Yang, "A novel fuzzy clustering algorithm based on a fuzzy scatter matrix with optimality tests," *Pattern Recognition Letters*, vol. 26, no. 5, pp. 639–652, 2005.
- [57] M. Ouattara, N. Niang, F. Badran, and C. Mandin, "2s-som : une méthode de soft-subspace clustering pour données multi-blocs basée sur les cartes topologiques auto-organisées," in *Fouille de données complexes, 2014*, pp. 65–88, 2014.
- [58] Z. Huang and M. K. Ng, "A fuzzy k-modes algorithm for clustering categorical data," *Fuzzy Systems, IEEE Transactions on*, vol. 7, no. 4, pp. 446–452, 1999.
- [59] J. Liu, J. Mohammed, J. Carter, S. Ranka, T. Kahveci, and M. Baudis, "Distance-based clustering of cgh data," *Bioinformatics*, vol. 22, no. 16, pp. 1971–1978, 2006.
- [60] R. J. Hathaway, J. C. Bezdek, and Y. Hu, "Generalized fuzzy c-means clustering strategies using l p norm distances," *Fuzzy Systems, IEEE Transactions on*, vol. 8, no. 5, pp. 576–582, 2000.
- [61] J. M. Santos and M. Embrechts, "On the use of the adjusted rand index as a metric for evaluating supervised classification," in *Artificial neural networks-ICANN 2009*, pp. 175–184, Springer, 2009.

- [62] P. A. Devijver and J. Kittler, *Pattern recognition : A statistical approach*, vol. 761. Prentice-Hall London, 1982.
- [63] B. M. Keller, D. L. Nathan, Y. Wang, Y. Zheng, J. C. Gee, E. F. Conant, and D. Kontos, “Estimation of breast percent density in raw and processed full field digital mammography images via adaptive fuzzy c-means clustering and support vector machine segmentation,” *Medical physics*, vol. 39, no. 8, pp. 4903–4917, 2012.
- [64] A. Djeflal, *Utilisation des méthodes Support Vector Machine (SVM) dans l’analyse des bases de données*. PhD thesis, Thèse de doctorat, Université Mohamed Khider-Biskra. Cité, 2012.
- [65] S. Lecomte, *Classification partiellement supervisée par SVM : application à la détection d’événements en surveillance audio*. PhD thesis, Troyes, 2013.
- [66] R. Langone, C. Alzate, B. De Ketelaere, J. Vlasselaer, W. Meert, and J. A. Suykens, “Ls-svm based spectral clustering and regression for predicting maintenance of industrial machines,” *Engineering Applications of Artificial Intelligence*, vol. 37, pp. 268–278, 2015.
- [67] S. Taneja, C. Gupta, K. Goyal, and D. Gureja, “An enhanced k-nearest neighbor algorithm using information gain and clustering,” in *Advanced Computing & Communication Technologies (ACCT), 2014 Fourth International Conference on*, pp. 325–329, IEEE, 2014.
- [68] C.-H. Ho, M.-H. Tsai, and C.-J. Lin, “Active learning and experimental design with svms,” *Active Learning Challenge Challenges in Machine Learning, Volume 6*, p. 77, 2011.
- [69] M.-R. Bouguelia, *Classification et apprentissage actif à partir d’un flux de données évolutif en présence d’étiquetage incertain*. PhD thesis, Université de Lorraine, 2015.
- [70] H. S. Seung, M. Opper, and H. Sompolinsky, “Query by committee,” in *Proceedings of the fifth annual workshop on Computational learning theory*, pp. 287–294, ACM, 1992.
- [71] S. Dasgupta and D. Hsu, “Hierarchical sampling for active learning,” in *Proceedings of the 25th international conference on Machine learning*, pp. 208–215, ACM, 2008.
- [72] S. Dasgupta, “Two faces of active learning,” *Theoretical computer science*, vol. 412, no. 19, pp. 1767–1781, 2011.

- [73] X. He, Z. Wang, C. Jin, Y. Zheng, and X. Xue, "A simplified multi-class support vector machine with reduced dual optimization," *Pattern Recognition Letters*, vol. 33, no. 1, pp. 71–82, 2012.
- [74] Y. Liu and Y. F. Zheng, "One-against-all multi-class svm classification using reliability measures," in *Neural Networks, 2005. IJCNN'05. Proceedings. 2005 IEEE International Joint Conference on*, vol. 2, pp. 849–854, IEEE, 2005.
- [75] C.-W. Hsu and C.-J. Lin, "A comparison of methods for multiclass support vector machines," *Neural Networks, IEEE Transactions on*, vol. 13, no. 2, pp. 415–425, 2002.
- [76] J. P. Cocquerez and S. Philipp-Foliguet, "Analyse d'images : filtrage et segmentation," 1995.
- [77] F. Agüera, F. J. Aguilar, and M. A. Aguilar, "Using texture analysis to improve per-pixel classification of very high resolution images for mapping plastic greenhouses," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 63, no. 6, pp. 635–646, 2008.
- [78] M. Hanifi, *Extraction de caractéristiques de texture pour la classification d'images satellites*. PhD thesis, Université de Toulouse, Université Toulouse III-Paul Sabatier, 2009.
- [79] R. M. Haralick, "Statistical and structural approaches to texture," *Proceedings of the IEEE*, vol. 67, no. 5, pp. 786–804, 1979.
- [80] A. Khotanzad and R. Kashyap, "Feature selection for texture recognition based on image synthesis," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 17, no. 6, pp. 1087–1095, 1987.
- [81] R. M. Haralick, "Statistical image texture analysis," *Handbook of pattern recognition and image processing*, vol. 86, pp. 247–279, 1986.
- [82] H.-J. Bunge, *Texture analysis in materials science : mathematical methods*. Elsevier, 2013.
- [83] S. KHarfouchi, *Statistical inference In spatial bilinear processes*. PhD thesis, Université Mentouri Constantine.
- [84] S. W. Zucker and D. Terzopoulos, "Finding structure in co-occurrence matrices for texture analysis," *Computer graphics and image processing*, vol. 12, no. 3, pp. 286–308, 1980.

- [85] L. S. Davis, S. A. Johns, and J. Aggarwal, "Texture analysis using generalized co-occurrence matrices," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, no. 3, pp. 251–259, 1979.
- [86] S. Arivazhagan and L. Ganesan, "Texture classification using wavelet transform," *Pattern recognition letters*, vol. 24, no. 9, pp. 1513–1521, 2003.
- [87] R. Azencott, J.-P. Wang, and L. Younes, "Texture classification using windowed fourier filters," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 19, no. 2, pp. 148–153, 1997.
- [88] Q. Wu and Y. Yu, "Feature matching and deformation for texture synthesis," *ACM Transactions on Graphics (TOG)*, vol. 23, no. 3, pp. 364–367, 2004.
- [89] S. Reis and K. Taşdemir, "Identification of hazelnut fields using spectral and gabor textural features," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 66, no. 5, pp. 652–661, 2011.
- [90] S. Y. Hwang and I. Basawa, "Explosive random-coefficient ar (1) processes and related asymptotics for least-squares estimation," *Journal of Time Series Analysis*, vol. 26, no. 6, pp. 807–824, 2005.
- [91] H. H. Kelejian and I. R. Prucha, "A generalized moments estimator for the autoregressive parameter in a spatial model," *International economic review*, vol. 40, no. 2, pp. 509–533, 1999.
- [92] A. Boulemnadjel, F. Hachouf, and S. Kharfouchi, "Gmm estimation of 2d-rca models with applications to texture image classification," *Image Processing, IEEE Transactions on*, vol. 25, no. 2, pp. 528–539, 2016.
- [93] A. Boulemnadjel, F. Hachouf, A. Hebboul, and K. Djemal, "Active learning for improving a soft subspace clustering algorithm," *Engineering Applications of Artificial Intelligence*, vol. 46, pp. 196–208, 2015.
- [94] J. Yu, Q. Cheng, and H. Huang, "Analysis of the weighting exponent in the fcm," *Systems, Man, and Cybernetics, Part B : Cybernetics, IEEE Transactions on*, vol. 34, no. 1, pp. 634–639, 2004.
- [95] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment : from error visibility to structural similarity," *Image Processing, IEEE Transactions on*, vol. 13, no. 4, pp. 600–612, 2004.

-
- [96] G.-H. Chen, C.-L. Yang, and S.-L. Xie, "Gradient-based structural similarity for image quality assessment," in *Image Processing, 2006 IEEE International Conference on*, pp. 2929–2932, IEEE, 2006.
- [97] <http://archive.ics.uci.edu/ml/>.
- [98] P. Fränti and O. Virtajoki, "Iterative shrinking method for clustering problems," *Pattern Recognition*, vol. 39, no. 5, pp. 761–775, 2006.
- [99] P. Brodatz, "Textures," 1966.
- [100] W.-Y. Ma and B. S. Manjunath, "Texture features and learning similarity," in *Computer Vision and Pattern Recognition, 1996. Proceedings CVPR'96, 1996 IEEE Computer Society Conference on*, pp. 425–430, IEEE, 1996.