Modélisation et classification des données de grande dimension : application à l'analyse d'images.

Charles Bouveyron

UNIVERSITÉ JOSEPH FOURIER – GRENOBLE 1

THÈSE

présentée par

Charles BOUVEYRON

pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ JOSEPH FOURIER

Spécialité : Mathématiques Appliquées

MODÉLISATION ET CLASSIFICATION DES DONNÉES DE GRANDE DIMENSION

APPLICATION À L'ANALYSE D'IMAGES

---

réalisée sous la direction de
Cordelia SCHMID et Stéphane GIRARD

---

soutenue publiquement le 28 septembre 2006

JURY

Christophe	BIERNACKI	Professeur	Président
Gilles	CELEUX	Directeur de Recherche	Rapporteur
Fionn	MURTAGH	Professeur	Rapporteur
Tinne	TUYTELAARS	Chargé de Recherche	Examinateur
Cordelia	SCHMID	Directeur de Recherche	Directeur
Stéphane	GIRARD	Maître de Conférence	Directeur

Thèse préparée au sein des équipes SMS (LMC-IMAG) et Mistis (INRIA Rhône-Alpes)

Table des matières

Principales notations

k : nombre total de classes

$\mathcal{P}$: partition des données en k classes

C_i : $i$ème classe de la partition $\mathcal{P}$, $i = 1, ..., k$

x_j : observation dans $\mathbb{R}^p$, $j = 1, ..., n$

z_j : classe d'appartenance de x_j, $j = 1, ..., n$

s_{ij} : vaut 1 si x_j appartient à C_i, 0 sinon

π_i, μ_i, Σ_i : proportion, moyenne et matrice de covariance de la classe C_i, $i = 1, ..., k$

$\theta_i = \{\mu_i, \Sigma_i\}$: paramètres de la $i$ème composante du mélange, $i = 1, ..., k$

$f(x, \theta_i)$: densité paramétrée de la $i$ème composante du mélange, $i = 1, ..., k$

$L(\theta)$, $l(\theta)$: vraisemblance et log-vraisemblance de $\theta = \{\pi_1, ..., \pi_k, \theta_1, ..., \theta_k\}$

$\nu(m)$: nombre de paramètres à estimer du modèle m

W_i : matrice de variance empirique de la classe C_i, $i = 1, ..., k$

W : matrice de variance intra-classe empirique

$\mathcal{N}(\mu, \Sigma)$: loi normale de paramètres μ et Σ

Principales abbréviations

QDA : *Quadratic Discriminant Analysis*

LDA : *Linear Discriminant Analysis*

RDA : *Regularized Discriminant Analysis*

EDDA : *Eigenvalue Decomposition Discriminant Analysis*

SVM : *Support Vector Machines*

Full-GMM : modèle gaussien général

Com-GMM : modèle gaussien à matrices de covariance égales

Diag-GMM : modèle gaussien à matrices de covariance diagonales

Sphe-GMM : modèle gaussien à matrices de covariance sphériques

HDDA : *High Dimensional Discriminant Analysis*

HDDC : *High Dimensional Data Clustering*

ACP : Analyse en Composantes Principales

AUC : *Area Under the ROC Curve*

EER : *Equal-Error Rate*

Chapitre 1

Introduction

L'apprentissage statistique, et en particulier la classification, a pour but la résolution automatique de problèmes complexes par la prise de décisions sur la base d'observations de ces problèmes. Par exemple, le tri du courrier est actuellement réalisé de manière automatique par des machines réussissant à lire et interpréter les caractères écrits correspondant au code postal du destinataire. Cette spécialité de la statistique, qu'est l'apprentissage statistique, joue de nos jours un rôle croissant dans de nombreux domaines scientifiques. En effet, les progrès scientifiques réalisés ces dernières années ont permis d'augmenter sensiblement les capacités de mesures et il est à présent difficile pour un opérateur humain de traiter ces données dans un temps raisonnable. L'apprentissage statistique prend alors le relai sur l'humain pour analyser de façon automatique ces données. Les méthodes d'apprentissage statistique fournissent en général soit une représentation simple des données permettant à un opérateur humain de prendre rapidement une décision, soit une décision assortie d'un risque d'erreur qui pourra éventuellement être entérinée par un opérateur humain.

1.1 Modélisation et classification des données modernes

Le thème d'étude de ce mémoire est le traitement des données modernes dans un but de classification. Les données fournies par les applications modernes sont généralement de grande dimension et leur classification peut en être perturbée.

Spécificité des données modernes

La spécificité principale des données modernes est certainement leur grande dimension. Par exemple, si l'on considère le cas des images numériques, la résolution des images est passée de 0.1 à 12 mégapixels[1] entre l'apparition du capteur CCD (acronyme de *Charged Coupled Device*) en 1973 et les appareils haut de gamme d'aujourd'hui. La dimension des images numériques a donc été multiplié

[1] 1 mégapixel = 1 000 000 pixels.

par plus de 100 en l'espace d'une vingtaine d'années et cette augmentation s'est particulièrement accentuée ces dernières années. Il va de même pour les appareils de mesure utilisés dans de nombreux domaines scientifiques et, ainsi, la dimension des données fournie par ces appareils croît chaque jour. Nous verrons dans ce mémoire que la grande dimension des données pose cependant des problèmes spécifiques aux méthodes se proposant de les analyser. En parallèle, le coût d'acquisition et de supervision de ces données est lui aussi croissant. En effet, dans des domaines comme la génétique, les mesures sont extrêmement difficiles à réaliser et, de surcroît, la supervision humaine est complexe. Par conséquent, dans certains domaines scientifiques, les méthodes d'apprentissage statistique ne disposent que d'un nombre très limité de mesures pour assimiler la tâche qu'elles auront à effectuer ensuite de manière automatique. Cette difficulté vient donc s'ajouter au problème de la grande dimension des données modernes et fait de l'analyse de ce type de données un défi important qui doit être relevé si l'on veut pouvoir traiter les données du futur.

Modélisation et classification

Les méthodes d'apprentissage statistique sont usuellement divisées en deux catégories : les méthodes génératives et les méthodes discriminatives. L'idée de l'approche générative est de modéliser le système complexe qui a généré les données observées et, à partir de cette modélisation, construire une règle de décision. L'approche discriminative construit, quant à elle, directement la règle de décision à partir des données observées et ne s'intéresse pas aux caractéristiques du système qui a donné naissance aux données. L'avantage de l'approche générative est de fournir une règle de décision facilement compréhensible par un opérateur humain. Toutefois, ces dernières années, les méthodes génératives ont été distancées par les méthodes discriminatives en terme de performance de classification. Cela est principalement dû aux caractéristiques des données modernes : grande dimension et nombre d'exemples limités. En effet, les méthodes génératives les plus utilisées, basées sur le modèle de mélange gaussien, sont particulièrement sensibles à ces deux facteurs alors que les méthodes discriminatives ne sont généralement pas affectées dans les espaces de grande dimension. Actuellement, les méthodes discriminatives sont donc particulièrement utilisées pour résoudre des problèmes où les données sont de grande dimension car elles fournissent des résultats quantitativement très bons mais on peut déplorer que ce soit au dépend de la qualité d'interprétation des décisions prises.

1.2 Problématique et contributions de la thèse

La problématique, à laquelle cette thèse se propose de répondre, est l'adaptation des modèles gaussiens aux spécificités des données modernes afin de fournir des méthodes de classification qui soient performantes et dont les résultats soient facilement interprétables par l'utilisateur.

Une famille de modèles du plus complexe au plus parcimonieux

La principale contribution de la thèse est l'introduction d'une famille de modèles gaussiens parcimonieux prenant en compte les caractéristiques des données de grande dimension. Nous verrons, au cours du chapitre 2, que les données de grande dimension vivent dans des sous-espaces de dimension inférieure à la dimension de l'espace d'origine. Il est donc naturel de penser que, dans le cadre de la classification, les données de groupes différents vivent dans des sous-espaces distincts dont les dimensions intrinsèques peuvent ne pas être égales. De plus, nous verrons que la discrimination de données avec un classifieur adapté est plus aisée dans un espace de grande dimension que dans un espace de faible dimension. Sur la base de ce constat, nous proposerons une modélisation gaussienne qui prenne en compte ces caractéristiques des données de grande dimension. Cette re-paramétrisation du modèle gaussien permettra de contrôler la complexité du modèle par les dimensions des sous-espaces des classes et engendrera ainsi une famille de modèles allant du plus complexe au plus parcimonieux.

Une approche unifiée de la classification des données de grande dimension

Cette famille de mélanges gaussiens sera utilisée en classification supervisée pour donner naissance à une méthode de discrimination, baptisée *High Dimensional Discriminant Analysis* (HDDA), adaptée aux données de grande dimension. De même, nous utiliserons ces modèles parcimonieux en classification non-supervisée pour donner le jour à une méthode de *clustering* de données de grande dimension, appelée *High Dimensional Data Clustering* (HDDC). Nous appliquerons ces méthodes de classification à des problèmes réels, dont nous donnons un aperçu au paragraphe suivant. En outre, la re-paramétrisation du modèle de mélange gaussien, proposée dans ce mémoire, permettra d'unifier les approches existantes de classification des données de grande dimension. Nous montrerons, en effet, que les modèles associés aux méthodes existantes de classification dans des espaces de grande dimension appartiennent à notre famille de modèles.

Approche probabiliste de la reconnaissance d'objets

L'application de notre méthode de *clustering* au problème de la reconnaissance d'objets nous amènera à proposer une modélisation basée sur le mélange gaussien pour la reconnaissance d'objets dans des images. L'approche qui sera présentée dans ce mémoire permettra de localiser des objets dans des images de manière probabiliste. En outre, cette approche pourra être soit supervisée, *i.e.* les objets sont segmentés dans les images d'apprentissage, soit faiblement supervisée, *i.e.* les objets ne sont pas segmentés dans les images d'apprentissage. Pour ce faire, nous introduirons la notion de pouvoir discriminant afin d'évaluer la contribution d'une partie d'objet à la discrimination de cet objet. Cette modélisation ouvrira donc la voie à la classification faiblement supervisée qui nécessite une intervention humaine limitée.

1.3 Domaines d'application

La modélisation et la classification des données de grande dimension intervient dans de nombreux domaines d'application. Le champ de mise en œuvre de ces techniques va des applications courantes, telles que le traitement des courriers électroniques indésirables ou la reconnaissance optique de caractères, aux applications professionnelles, telles que l'aide au diagnostic en imagerie médicale ou la catégorisation automatique des images d'un satellite. En particulier, l'analyse d'images est un domaine d'application où l'apprentissage statistique a une place privilégiée. Nous considérerons, dans ce mémoire, les applications suivantes en analyse d'images.

Reconnaissance optique de caractères écrits

La reconnaissance optique de caractères écrits (OCR en anglais) est certainement l'application de vision par ordinateur la plus ancienne. En effet, la poste des Etats-Unis utilise depuis 1965 des machines OCR pour lire le code postal du destinataire et orienter ainsi le courrier vers le bon centre de tri. Ce problème, qui est bien connu à présent, reste encore difficile puisque les données fournies sont de grande dimension (typiquement, plusieurs centaines de dimensions) et qu'aucune méthode automatique n'arrive à ce jour à être aussi performante que l'œil humain.

Reconnaissance de visages

La reconnaissance de visages est certainement l'une des applications de la biométrie les plus utilisées actuellement. Elle peut par exemple être utilisée pour sécuriser l'accès à un bâtiment ou pour identifier des personnes dans une foule. C'est d'ailleurs cette technique qui a été retenue pour l'établissement du nouveau passeport biométrique français auquel est intégré une image numérique du titulaire. Toutefois, cette technique reste moins sûre que les techniques de reconnaissance basées sur l'iris ou les empreintes digitales pour l'identification de personnes, car les données fournies par cette technique sont de très grande dimension et qu'il existe une très grande variabilité intra-sujet.

Localisation d'objets dans des images

La localisation d'objets dans des images est un problème clé à l'heure actuelle en vision par ordinateur et qui ouvre la voie à un très grand nombre d'application comme le pilotage autonome de véhicules. La difficulté de ce problème est certainement due à la grande dimension des données utilisées et à la variabilité très forte qui existe dans une même classe d'objets. Pour ces raisons, nous nous intéresserons particulièrement à ce problème qui fera l'objet du chapitre 6 de ce mémoire. Le but de la localisation d'objets dans des images est de décider pour chaque point d'une nouvelle image s'il appartient ou non à un objet donné. On peut ensuite déterminer la localisation globale de l'objet dans l'image en utilisant un cadre. Cette application est encore au stade expérimental mais il est raisonnable de penser que, dans quelques années, le grand public bénéficiera des résultats de ces recherches. L'équipementier automobile Volkswagen envisage d'ailleurs d'embarquer un système de

reconnaissance automatique des panneaux de signalisation dans ces futurs véhicules pour assister le conducteur.

Analyse minéralogique du sol d'une planète

Les imageurs embarqués dans les satellites actuels fournissent des images hyper-spectrales du sol de certaines planètes du système solaire. Ces images transmises par les satellites sont de grande dimension et en très grand nombre, ce qui fait que le traitement manuel de ces données n'est pas envisageable. Des techniques de catégorisation automatiques sont alors utilisées pour affecter chaque point des images hyper-spectrales à une des classes minéralogiques définies par les experts. Il est d'un intérêt majeur de pouvoir traiter correctement de façon automatique de telles données car, dans un futur proche, les imageurs de nouvelle génération enverront des données de plus grande taille et en nombre encore plus important que les données déjà disponibles.

1.4 Organisation de la thèse

Le chapitre 2 présentera l'état de l'art en classification des données de grande dimension. Au cours de ce chapitre, nous considérons tout d'abord les problèmes de classification supervisée et non supervisée dans une approche unifiée. Nous exposerons ensuite les méthodes classiques d'analyse discriminante et de classification automatique avant de nous intéresser plus particulièrement aux spécificités des données de grande dimension et aux problèmes qu'elles posent en classification. Nous présenterons en outre, dans ce chapitre, les solutions existantes pour classer de telles données ainsi que les développements récents dans ce domaine.

Après avoir dressé une analyse critique des approches existantes pour classer des données de grande dimension, nous proposerons au chapitre 3 une re-paramétrisation du modèle de mélange gaussien prenant en compte le fait que les données de grande dimension vivent dans des sous-espaces de dimensions intrinsèques inférieures à la dimension de l'espace original. Cette re-paramétrisation donnera naissance à une famille de modèles gaussiens adaptés aux données de grande dimension allant du modèle le plus général au modèle le plus parcimonieux. De plus, les règles de décision associées à certains modèles de cette famille pourront être interprétées d'un point de vue géométrique.

Au cours du chapitre 4, nous utiliserons ces modèles gaussiens pour la discrimination et la classification automatique de données de grande dimension. Les classifieurs supervisés et non supervisés associés aux modèles gaussiens, proposés au cours de ce chapitre, seront baptisés respectivement *High Dimensional Discriminant Analysis* (HDDA) et *High Dimensional Data Clustering* (HDDC). La construction du classifieur supervisé HDDA et du classifieur non supervisé HDDC nécessitera l'estimation par *maximum* de vraisemblance des paramètres des différents modèles issus de notre re-paramétrisation. Nous verrons en outre que la nature de notre re-paramétrisation permettra aux méthodes HDDA et HDDC de ne pas être perturbées quand le nombre d'observations disponible pour

l'apprentissage est limité. Nous aborderons également le problème de l'estimation des dimensions intrinsèques des classes en proposant une méthode empirique pour les estimer.

Le chapitre 5 sera consacré à la validation des méthodes de classification des données de grande dimension présentées dans ce mémoire. Cette validation sera réalisée sur des jeux de données réelles et simulées. Ces expérimentations mettront en évidence que les méthodes de classification HDDA et HDDC présentent l'avantage d'être performantes aussi bien dans des espaces de grande dimension que de faible dimension. La comparaison sur données réelles avec les méthodes de classification existantes démontrera l'efficacité de notre approche pour la modélisation et la classification des données de grande dimension.

Dans le chapitre 6, nous proposerons une approche probabiliste de la reconnaissance d'objets dans des images. Cette approche, qui sera basée sur le modèle de mélange gaussien, permettra à la fois d'exploiter au mieux les résultats de nos méthodes de classification et de localiser de manière probabiliste un objet dans une image inconnue. En outre, cette approche pourra être supervisée ou faiblement supervisée. Nous mettrons également en œuvre notre modélisation sur des bases de d'images récentes et ces expérimentations montrerons que notre approche probabiliste est plus efficace que les méthodes existantes.

Le chapitre 7 conclura ce mémoire et présentera les travaux en cours ainsi que les perspectives de recherches qui s'ouvrent à nous.

Chapitre 2

État de l'art

La classification est une méthode d'analyse des données qui vise à regrouper en classes homogènes un ensemble d'observations. Ces dernières années, les besoins d'analyse de données et en particulier de classification ont augmenté significativement. En effet, de plus en plus de domaines scientifiques nécessitent de catégoriser leurs données dans un but descriptif ou décisionnel. Dans ce chapitre, nous présenterons tout d'abord au paragraphe 2.1 le problème de la classification et sa modélisation probabiliste. Nous verrons notamment que la classification se divise généralement en deux sous-problèmes distincts : la classification supervisée, appelée également analyse discriminante, et la classification non supervisée, dénommée aussi classification automatique. Les méthodes classiques d'analyse discriminante seront présentées au paragraphe 2.2 et les méthodes de classification automatique le seront au paragraphe 2.3. D'autre part, les processus d'acquisition des données ayant aussi progressé rapidement, la dimension des données à étudier est devenue très grande. Nous verrons au paragraphe 2.4 que la grande dimension des données pose des problèmes spécifiques en apprentissage statistique et particulièrement en classification. Le problème de la dimension est généralement appelé « fléau de la dimension » et nous verrons dans ce paragraphe quelles solutions existent pour pallier cette limitation des méthodes de classification. Ce chapitre sera illustré par la mise en œuvre d'un certain nombre des méthodes présentées sur données simulées ou sur données réelles. Les données réelles seront principalement issues d'applications en analyse d'images.

2.1 Modélisation probabiliste en classification

Les premières approches qui ont été proposées en classification étaient algorithmiques, heuristiques ou géométriques et reposaient essentiellement sur la dissimilarité entre les objets à classer. L'approche statistique, plus récente, se base sur des modèles probabilistes qui formalisent l'idée de classe. Cette approche permet en outre d'interpréter de façon statistique la classification obtenue. Nous présenterons tout d'abord le problème de la classification et l'approche probabiliste de la classification. Nous verrons ensuite que la modélisation la plus classique est celle du modèle de mélange

fini qui peut être paramétrique ou non. Enfin, l'estimation des paramètres des modèles de mélange paramétriques sera abordée au travers de la méthode du *maximum* de vraisemblance.

2.1.1 Le problème de la classification

Le problème de la classification étant d'organiser un ensemble d'objets en classes homogènes, il nous faut donc définir ce que sont une partition, une classe (on utilisera également la notion de groupe) et ce que sont les éléments que l'on cherche à classer. On définira ainsi le cadre d'étude de ce mémoire.

Partition et classe

La question se pose alors de savoir ce qu'est une classe et quel type de structure va être utilisé pour modéliser l'espace des objets. D'un point de vue mathématique, une classe est un sous-ensemble de l'ensemble E des objets à classer. Nous verrons qu'il existe deux approches différentes pour décrire une classe : l'approche générative et l'approche discriminative. La première décrit une classe par les propriétés caractéristiques des objets qui la composent alors que la seconde décrit une classe par sa frontière avec ses voisines. La structure la plus couramment utilisée est celle de la « partition » qui se définit de la façon suivante :

Définition 2.1.1. L'ensemble $P = \{C_1, ..., C_k\}$ est une partition de l'ensemble E en k classes si et seulement si :

(i) $C_i \neq \emptyset$ pour $i = 1, ..., k$,

(ii) $\bigcup_{i=1}^{k} C_i = E$,

(iii) $C_i \cap C_\ell = \emptyset$ pour tout $i \neq \ell$.

On remarque tout de suite que le paramètre k jouera un rôle important dans les problèmes de classification. Dans ce mémoire, la structure de classes employée sera celle de la « partition » que nous venons de définir. Notons toutefois qu'il existe d'autres structures de classes qui s'obtiennent à partir de la « partition » par relaxation des contraintes (ii) et (iii). En effet, la relaxation de la contrainte (ii) fournit une structure de classes qui permet de ne pas être contraint de classer des points aberrants. D'autre part, si l'on relâche la contrainte (iii), alors chaque objet peut appartenir à plusieurs classes et l'on a ainsi des classes empiètantes. Cette notion de classes empiètantes est par exemple très utile en bio-informatique. On pourra consulter [4] pour une mise en œuvre de cette structure à classes empiètantes en classification automatique.

Cadre théorique

Dans ce mémoire, un objet de l'ensemble E des objets à classer sera modélisé par le vecteur X, décrit lui même par p variables quantitatives. Le but de la classification étant d'associer ce vecteur X à une des k classes, nous introduisons la variable auxiliaire Z à valeurs dans $\{1, ..., k\}$ et telle que

$Z = i$ si X appartient à la $i$ème classe. Ainsi, le problème de la classification revient à établir une règle de décision δ qui associe au vecteur $X \in \mathbb{R}^p$ un vecteur $Z \in \{1, .., k\}$:

$$\begin{aligned} \delta : \mathbb{R}^p &\longrightarrow \{1, ..., k\}, \\ X &\longmapsto Z. \end{aligned}$$

Cette règle de décision est généralement construite à partir d'un jeu de données (dit d'apprentissage) et c'est la nature de ce jeu de données qui différencie les deux types d'apprentissage possibles : l'apprentissage supervisé et l'apprentissage non supervisé.

L'apprentissage supervisé Les données utilisées pour l'apprentissage supervisé, notées $y_1, ..., y_n$, sont dites « complètes » car elle contiennent à la fois les valeurs $x_1, ..., x_n$ prises par les p variables explicatives et leur appartenance aux k classes $z_1, ..., z_n$. Les données complètes sont donc l'ensemble des couples observations–labels, *i.e.* $\{y_1, ..., y_n\} = \{(x_1, z_1), ..., (x_n, z_n)\}$.

L'apprentissage non supervisé Les données utilisées pour l'apprentissage non supervisé ne sont pas « complètes » car elles ne contiennent que les valeurs $x_1, ..., x_n$ prises par les p variables explicatives.

2.1.2 La classification probabiliste

Nous allons à présent formaliser de façon probabiliste le cadre théorique de la classification que nous avons défini au paragraphe précédent. Cela nous permettra notamment d'introduire la règle de classification optimale au sens probabiliste.

Modélisation probabiliste

La classification probabiliste suppose que les observations $x_1, ..., x_n$ de l'ensemble E des observations à classer sont des réalisations d'un vecteur aléatoire X à valeurs dans $\mathbb{R}^p$. Elle suppose en outre que les valeurs $z_1, ..., z_n$, décrivant l'origine des observations $x_1, ..., x_n$, sont des réalisations de la variable aléatoire Z à valeurs dans $\{1, ..., k\}$. Ainsi, le fait de dire que x est une réalisation de la variable aléatoire X conditionnellement au fait que $Z = i$ revient à dire que l'observation x appartient à la $i$ème classe C_i. Nous introduisons également le vecteur aléatoire $S \in \{0, 1\}^k$ tel que si $Z = i$ alors $S = (0, ..., 0, 1, 0, ..., 0)$ où le $i$ème terme vaut 1, et ce pour $i = 1, ..., k$.

La règle de Bayes

Ce cadre probabiliste permet de construire la règle de décision optimale δ^*, dite également règle de Bayes, qui minimise le risque conditionnel $R(\delta|x)$ pour chaque observation x. En associant un coût

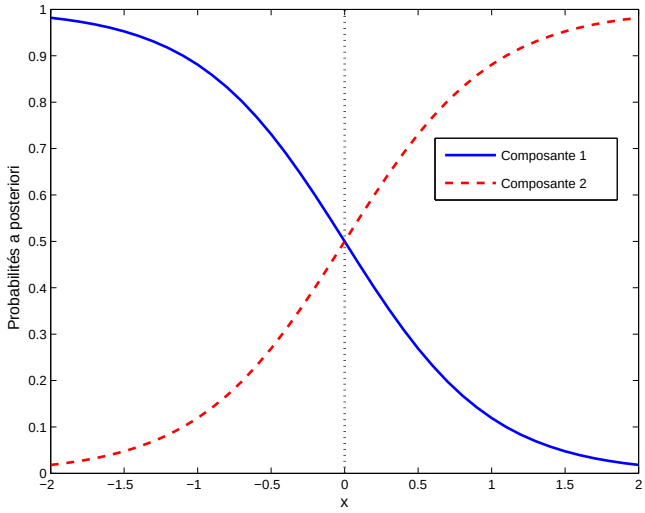


FIG. 2.1 – Principe du *maximum a posteriori* pour un mélange à 2 composantes en dimension 1.

nul à une bonne affectation et un coût de 1 à une mauvaise affectation, le risque conditionnel s'écrit :

$$R(\delta|x) = 1 - P(Z = \delta(x)|X = x).$$

La règle δ^* consiste donc à affecter l'observation x à la classe la plus probable *a posteriori* :

$$\delta^*(x) = \underset{i=1,\ldots,k}{\operatorname{argmax}} P(Z = i|X = x).$$

Cette règle porte également le nom de MAP pour *maximum a posteriori*. La figure 2.1 illustre la règle de décision du *maximum a posteriori* dans le cas d'un mélange à 2 composantes en dimension 1. La courbe bleue et la courbe rouge représentent respectivement la probabilité *a posteriori* que l'observation x appartienne à la première composante et à la seconde composante du mélange, cela en fonction de la valeur de x. On peut observer que si l'observation x a une valeur inférieure à 0 alors la probabilité *a posteriori* la plus forte est celle associée à la première composante et l'observation x sera donc affectée à la classe C_1. A l'inverse, si x est supérieure à 0 alors la probabilité *a posteriori* la plus forte est celle associée à la seconde composante et l'observation x sera donc affectée à la classe C_2.

Approches génératives et discriminatives

En classification probabiliste, la règle de décision repose donc sur les probabilités *a posteriori* et c'est la manière de calculer ces probabilités qui différencie les deux approches de la classification probabiliste : l'approche discriminative et l'approche générative. La première modélise directement la probabilité *a posteriori* $P(Z|X)$ en cherchant à définir la frontière entre les classes. L'approche générative, quant à elle, cherche tout d'abord à modéliser la distribution jointe $P(X, Z)$ et en déduit

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

$$P(Z|X) = \frac{P(Z)P(X|Z)}{P(X)} \propto P(Z)P(X|Z).$$

Les méthodes de classification que nous proposerons dans ce mémoire relèveront de cette dernière approche qui modélise donc chacune des classes par une densité de probabilité.

2.1.3 Modélisation par mélange de lois

Les données multi-dimensionnelles que l'on retrouve dans les applications modernes sont complexes et elles ne peuvent généralement pas être modélisées par une loi classique. Le modèle de mélange est un moyen de modéliser des données complexes en s'appuyant sur des lois simples comme des lois normales.

Le modèle de mélange

Le modèle de mélange suppose que chaque groupe est caractérisé par une distribution de probabilité. Nous verrons dans la suite que cette modélisation est très souple et permet de prendre en compte un grand nombre de situations. Dans un modèle de mélange, on considère que les données $x_1, ..., x_n$ constituent un échantillon de n réalisations indépendantes du vecteur aléatoire X à valeur dans $\mathbb{R}^p$ dont la fonction de densité peut s'écrire de la façon suivante :

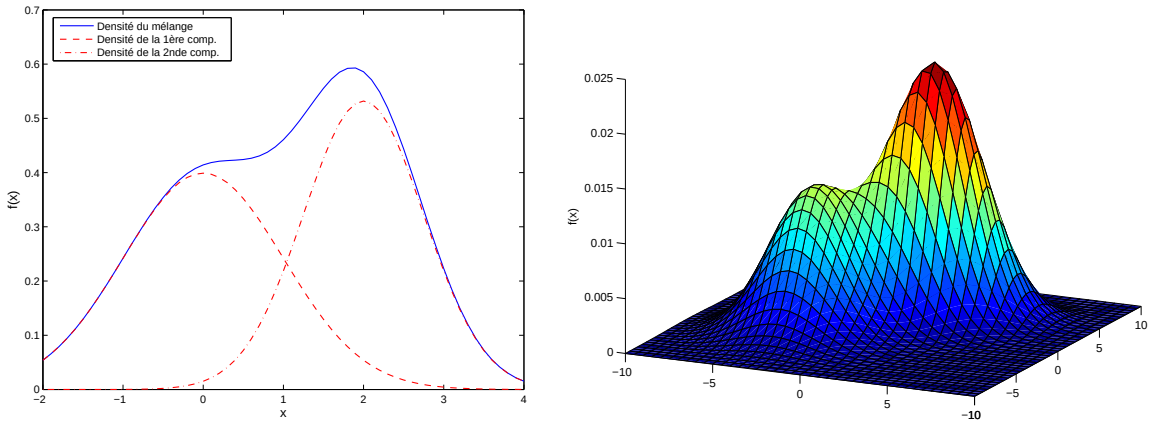
$$f(x) = \sum_{i=1}^{k} \pi_i f_i(x),$$

où k est le nombre de classes (connu dans le cas de l'analyse discriminante mais inconnu dans le cas de la classification automatique), f_i est la densité de la distribution de X conditionnellement à $Z = i$ (de la $i$ème composante du mélange) et les π_i sont les proportions du mélange ($\pi_i \in [0, 1]$ et $\sum_{i=1}^{k} \pi_i = 1$). Notons que l'identifiabilité d'un modèle de mélange est tout d'abord conditionnée à la numérotation des classes. Dans le cadre de l'analyse discriminante, la numérotation des classes sera évidente puisque l'on connaît les labels des observations. En revanche, dans le cas de la classification automatique, les labels ne sont pas connus et les classes sont donc numérotées de façon arbitraire.

Le modèle de mélange paramétrique

De plus, on suppose généralement que les densités f_i des classes appartiennent à une famille paramétrée, *i.e.* $f_i(.) = f(., \theta_i)$. Le modèle de mélange s'écrit alors :

$$f(x) = \sum_{i=1}^{k} \pi_i f(x, \theta_i),$$



(a) (b)

FIG. 2.2 – Exemple d'un mélange gaussien (a) univarié et (b) bivarié.

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

Le modèle de mélange paramétrique gaussien

Parmi les modèles de mélange paramétriques, le modèle gaussien est certainement le plus utilisé en classification et est de ce fait très classique. Dans ce cas, les densités de probabilité des variables explicatives conditionnellement aux classes $f(x, \theta_i), \forall i = 1, ..., k$, sont supposées être celles de lois normales $\mathcal{N}(\mu_i, \Sigma_i)$ de moyennes μ_i et de matrice de variance Σ_i :

$$f(x, \theta_i) = \frac{1}{(2\pi)^{p/2}(\det \Sigma_i)^{1/2}} \exp\left(-\frac{1}{2}(x - \mu_i)^t \Sigma_i^{-1}(x - \mu_i)\right), \tag{2.1}$$

où $\theta_i = \{\mu_i, \Sigma_i\}$. La figure 2.2 présente des exemples de mélanges gaussiens univariés et bivariés. Une des raisons de la popularité de ce modèle de mélange est que, au même titre que les mélanges exponentiels et de Poisson, il est identifiable alors qu'il est possible de vérifier qu'un mélange de lois uniformes ou binomiales ne l'est pas (*cf* [42, chap. 9] pour plus de détails).

Règle de Bayes et modèle de mélange paramétrique

Si l'on se place dans le cadre du modèle de mélange paramétrique (gaussien ou non), la formule de Bayes permet d'écrire :

$$P(Z = i|X = x, \theta) = \frac{\pi_i f(x, \theta_i)}{f(x)},$$

où $f(x) = \sum_{i=1}^{k} \pi_i f(x, \theta_i)$ est une quantité commune à chacune des classes. La règle de décision de Bayes peut alors être écrite de la façon suivante :

$$\delta^*(x) = \underset{i=1,...,k}{\operatorname{argmax}} \{\pi_i f(x, \theta_i)\}.$$

A la vue de cette re-formulation, il apparaît clairement que, dans le cadre du modèle de mélange paramétrique, le problème de la classification se résume à l'estimation des paramètres du modèle.

2.1.4 Estimation des paramètres d'un modèle de mélange

L'estimation des paramètres du modèle de mélange paramétrique (qu'il soit gaussien ou non) a fait l'objet de nombreux travaux depuis l'introduction du modèle de mélange par Pearson [71] en 1894 et la méthode la plus fréquemment utilisée aujourd'hui est celle du *maximum* de vraisemblance (que nous noterons parfois MV). Nous allons tout d'abord définir dans ce paragraphe les notions de données complètes et de vraisemblance complète, puis nous aborderons le problème de l'estimation des paramètres par la méthode du *maximum* de vraisemblance.

Données complètes et vraisemblance complète

Nous considérerons que les données observées $x_1, ..., x_n$ ne correspondent qu'à une connaissance partielle des données complètes $y_1, ..., y_n$ qui sont supposées être de la forme $y_j = (x_j, z_j)$, $j = 1, ..., n$, où z_j est le numéro de la classe auquel appartient l'observation x_j. Nous verrons que dans le cas de la classification non supervisée cette dernière information sera absente et z_j sera appelée une *donnée manquante*. On appellera *vraisemblance complète*, la vraisemblance calculée à partir des données complètes $y_1, ..., y_n$ et elle sera notée $L(y; \theta)$, ou plus simplement $L(\theta)$, dans la suite de ce mémoire.

Estimation par *maximum* de vraisemblance

La méthode du *maximum* de vraisemblance propose d'estimer le paramètre θ du modèle par $\hat{\theta}_{MV}$:

$$\hat{\theta}_{MV} = \underset{\theta}{\operatorname{argmax}}\, L(\theta), \tag{2.2}$$

où $L(\theta)$ est la vraisemblance complète du modèle. Les n données y_j, $j = 1, ..., n$, étant supposées indépendantes, on peut exprimer la vraisemblance du paramètre θ par le produit de toutes les densités marginales. Pour des raisons calculatoires, nous préférons dans la suite du document utiliser le logarithme naturel de la vraisemblance du modèle. Dans le cadre du modèle de mélange, la log-vraisemblance s'écrit alors de la façon suivante :

$$\log(L(\theta)) = \sum_{j=1}^{n} \log(\sum_{i=1}^{k} \pi_i f(x_j, \theta_i)). \tag{2.3}$$

Si de plus, les labels des observations sont connues, *i.e.* dans le cas supervisé, la log-vraisemblance du modèle de mélange peut également s'écrire sous la forme suivante :

$$\log(L(\theta)) = \sum_{j=1}^{n} \sum_{i=1}^{k} s_{ij} \log(\pi_i f(x_j, \theta_i)), \tag{2.4}$$

où $s_{ij} = 1$ si l'observation x_j appartient à la classe C_i et $s_{ij} = 0$ sinon. L'expression de la log-vraisemblance étant différente que l'on soit dans le cadre supervisé ou dans le cadre non supervisé, il est alors clair que la technique d'estimation du paramètre θ par maximisation de la vraisemblance devra être différente dans les cas supervisé et non supervisé. La méthode de calcul de $\hat{\theta}_{MV}$ dans le cas supervisé sera basée sur l'expression (2.4) de la log-vraisemblance et sera détaillée au paragraphe 2.2. Dans le cas non supervisé, l'estimation de l'ensemble des paramètres du modèle utilisera l'expression (2.3) de la log-vraisemblance et sera traitée au paragraphe 2.3.

Propriétés de l'estimateur du *maximum* de vraisemblance

Une des raisons de la popularité de la méthode d'estimation du *maximum* de vraisemblance est que l'estimateur qu'elle fournit possède de nombreuses propriétés statistiques appréciables. En effet, il a été montré que, sous certaines conditions, l'estimateur du *maximum* de vraisemblance est consistant, *i.e.* il converge en probabilité vers les vraies valeurs du paramètre, sans biais et asymptotiquement gaussien. On pourra consulter [79, chap. 14] à titre de référence sur ce sujet.

2.2 Analyse discriminante

L'analyse discriminante est le nom donné à la classification dans le cadre supervisé. La classification supervisée se distingue de la classification non supervisée par le fait que des observations dont on connaît l'appartenance aux classes sont disponibles pour apprendre la règle de décision (on parlera aussi parfois de classifieur). Ces observations, dites d'apprentissage, « supervisent » la construction du classifieur. Après avoir rappelé les objectifs et le problème de la discrimination, nous présenterons les principales méthodes génératives dont la très connue analyse discriminante linéaire. Enfin, nous dresserons un panorama des méthodes discriminatives dont certaines présentent des performances de prédiction remarquables.

2.2.1 Le problème de la discrimination

On distingue classiquement deux objectifs principaux en analyse discriminante : l'aspect descriptif et l'aspect décisionnel. L'aspect descriptif vise à trouver une représentation qui permette l'interprétation des groupes grâce aux variables explicatives. Cette tâche est rendue difficile quand le nombre de variables explicatives est plus grand que 3. Toutefois, des techniques existent pour visualiser la classification de données ayant un grand nombre de dimensions. On peut citer par exemple la méthode de

visualisation hiérarchique de Bishop et Tipping [11]. Dans le cas de l'aspect décisionnel, on cherche à définir la meilleure affectation d'un nouvel individu dont on ne connaît que les valeurs des variables explicatives. Cet aspect est particulièrement apprécié dans des domaines où la notion de diagnostic est essentielle. Dans ce mémoire, nous nous intéresserons plus particulièrement à l'aspect décisionnel qui est le plus important et souvent le plus délicat. Le problème de l'analyse discriminante est de prédire l'appartenance d'une observation x, décrit par p variables explicatives, à une classe parmi k classes C_1, ..., C_k définies *a priori*. Afin de prédire l'appartenance de l'observation x à une des k classes, nous disposons d'un ensemble d'apprentissage :

$$A = \{(x_1, z_1), ..., (x_n, z_n),\ x_j \in \mathbb{R}^p,\ z_j \in \{1, ..., k\}\},$$

où le vecteur x_j est la $j$ème observation et z_j indique le numéro de la classe à laquelle x_j appartient. Nous allons donc utiliser l'échantillon A pour construire une règle de décision δ qui associe tout vecteur x de $\mathbb{R}^p$ à une des k classes $C_1, ..., C_k$.

2.2.2 L'approche générative

Nous allons nous intéresser dans ce paragraphe aux méthodes génératives d'analyse discriminante qui, rappelons-le, proposent de modéliser la densité de chacune des classes. Les méthodes génératives peuvent être non-paramétriques et basées par exemple sur la méthode du noyau ou paramétriques comme le sont les méthodes basées sur un modèle de mélange gaussien. Il est en effet possible d'utiliser le mélange gaussien présenté précédemment pour modéliser les données des classes. Nous verrons que son utilisation dans le cadre de l'analyse discriminante a d'ailleurs donné naissance aux deux méthodes de discrimination les plus populaires : l'analyse discriminante quadratique et l'analyse discriminante linéaire.

Exemple d'approche non-paramétrique

L'approche non-paramétrique ne fait pas d'hypothèse spécifique sur la densité de chacune des classes et propose d'estimer les densités des classes grâce à la méthode du noyau qui est une des méthodes non-paramétriques d'estimation les plus utilisées. Les densités $f_i(x)$ sont donc estimées par :

$$f_i(x) = \frac{1}{n_i h^p} \sum_{j=1}^{n} s_{ij} K\left(\frac{x - x_j}{h}\right),$$

où K est une densité multi-dimensionnelle (la densité gaussienne est fréquemment choisie), h est le paramètre de lissage et $n_i = \sum_{j=1}^{n} s_{ij}$. Ce paramètre peut être choisi par validation croisée sur le jeu d'apprentissage. Si l'on choisit le noyau K tel que :

$$K(x) = 1_{B(0,1)}(x),$$

où $B(0,1)$ représente la boule unité de centre 0. La densité est alors estimée par :

$$f_i(x) = \frac{1}{n_i h^p} \sum_{j=1}^{n} s_{ij} 1_{B(x,h)}(x_j),$$

et la méthode consiste alors à rechercher, pour un nouveau point x à classer, les points du jeu d'apprentissage appartenant à la boule de rayon h et de centre x et affecter l'observation x à la classe majoritaire. Le choix du rayon h, qui joue alors le rôle du paramètre de lissage, est clairement primordial et celui-ci peut également être choisi par validation croisée sur le jeu d'apprentissage. Cette méthode ne fournit généralement pas des résultats stables quand la taille d'échantillon est petite et la recherche des voisins peut s'avérer longue si le rayon h est grand.

Analyse Discriminante Quadratique (QDA)

Nous allons à présent nous placer dans le cadre du modèle de mélange gaussien et donc considérer des méthodes paramétriques de discrimination. Nous avons vu au paragraphe 2.1.3, que la règle de décision de Bayes se réduisait, avec les hypothèses du modèle de mélange paramétrique, à la maximisation de la quantité $\pi_i f(x, \theta_i)$ qui ne dépend que des paramètres du modèle. Dans le but de faciliter l'écriture des règles de décision par la suite, nous allons introduire la fonction de coût K :

Définition 2.2.1. La fonction de coût K_i est définie conditionnellement à la classe C_i, $\forall i = 1, ..., k$, de la façon suivante :

$$\begin{aligned} K_i \ : \ \mathbb{R}^p &\longrightarrow \mathbb{R}, \\ x &\longmapsto -2\log(\pi_i f(x, \theta_i)). \end{aligned}$$

Avec cette notation et dans le cadre du modèle de mélange paramétrique, la règle du MAP s'écrit simplement de la façon suivante :

$$\delta^*(x) = \underset{i=1,...,k}{\operatorname{argmin}} \{K_i(x)\}.$$

Si l'on suppose que les densités conditionnelles des classes $f(x, \theta_i)$ sont gaussiennes, on obtient alors l'Analyse Discriminante Quadratique (QDA en anglais) qui doit son nom au fait qu'elle réalise des séparations quadratiques entre les classes. La fonction de coût K_i prend dans ce cas la forme suivante :

$$K_i(x) = (x - \mu_i)^t \Sigma_i^{-1} (x - \mu_i) + \log(\det \Sigma_i) - 2\log(\pi_i) + C^{te},$$

où la constante représente une quantité commune à toutes les classes et n'intervient donc pas dans la règle de décision. L'estimation des paramètres par *maximum* de vraisemblance conduit aux estima-

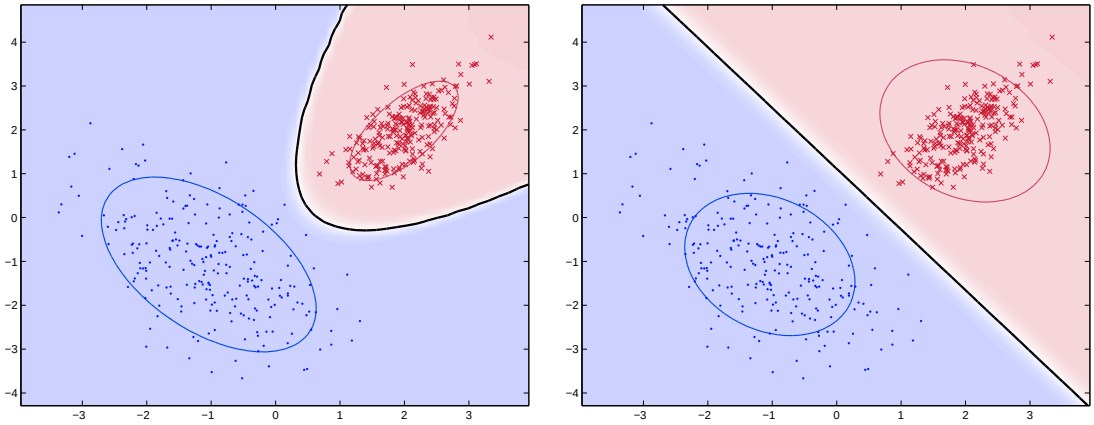


FIG. 2.3 – Frontières de décision de l'Analyse Discriminante Quadratique (QDA) et de l'Analyse Discriminante Linéaire (LDA) sur un même jeu de données en dimension 2.

teurs classiques suivants :

$$\begin{aligned} \hat{\pi}_i &= \frac{n_i}{n}, \\ \hat{\mu}_i &= \frac{1}{n_i}\sum_{j=1}^{n} s_{ij}x_j, \\ \hat{\Sigma}_i &= W_i = \frac{1}{n_i}\sum_{j=1}^{n} s_{ij}(x_j - \hat{\mu}_i)(x_j - \hat{\mu}_i)^t, \end{aligned} \tag{2.5}$$

où n_i est le nombre d'observations de la classe C_i, $i = 1, ..., k$. En pratique, cette méthode est pénalisée par l'estimation des nombreux paramètres quand la dimension p devient grande. Cette méthode requiert en effet l'estimation de k matrices de covariance de taille $p \times p$. La figure 2.3 montre la frontière de décision de QDA pour un jeu de données artificielles de dimension 2.

Analyse Discriminante Linéaire (LDA)

En anticipant sur la présentation des méthodes de régularisation, nous avons choisi de présenter ici l'Analyse Discriminante Linéaire (LDA en anglais) car elle est généralement considérée comme une méthode d'analyse discriminante à part entière plutôt que comme une méthode de régularisation de QDA. Pourtant, LDA est une régularisation de QDA car elle fait, par rapport à QDA, l'hypothèse supplémentaire d'égalité des matrices de variances, *i.e.* $\forall i = 1, ..., k$, $\Sigma_i = \Sigma$. L'analyse discriminante linéaire doit également son nom au fait qu'elle réalise des séparations linéaires entre les classes. Les termes $\log(\det \Sigma)$ et $x^t\Sigma^{-1}x$ ne dépendant plus des classes, ils peuvent être omis pour le calcul

de la règle de décision et la fonction de coût K_i prend la forme suivante :

$$K_i(x) = \mu_i^t \Sigma^{-1} \mu_i - 2\mu_i^t \Sigma^{-1} x - 2\log(\pi_i) + C^{te}.$$

Les estimateurs des proportions et des moyennes sont les même que ceux de QDA et l'estimateur du *maximum* de vraisemblance de la matrice Σ est la matrice de covariance intra-classe empirique W :

$$\hat{\Sigma} = W = \frac{1}{n} \sum_{i=1}^{k} \sum_{j=1}^{n} s_{ij}(x_j - \hat{\mu}_i)(x_j - \hat{\mu}_i)^t. \tag{2.6}$$

La figure 2.3 montre la frontière de décision de LDA pour un jeu de données artificielles de dimension 2. En pratique, l'analyse discriminante linéaire est fréquemment utilisée car elle offre un bon compromis entre pertinence et complexité (voir le paragraphe 2.4.1 dédié au phénomène du sur-apprentissage). D'autre part, elle fournit des résultats robustes aux fluctuations sur les hypothèses de normalité des classes et d'égalité des matrices de variance (voir [33, 66] à ce sujet). Pour toutes ces raisons, elle doit être considérée comme une méthode de référence et nous comparerons fréquemment les méthodes de discrimination proposées dans ce mémoire à LDA.

2.2.3 L'approche discriminative

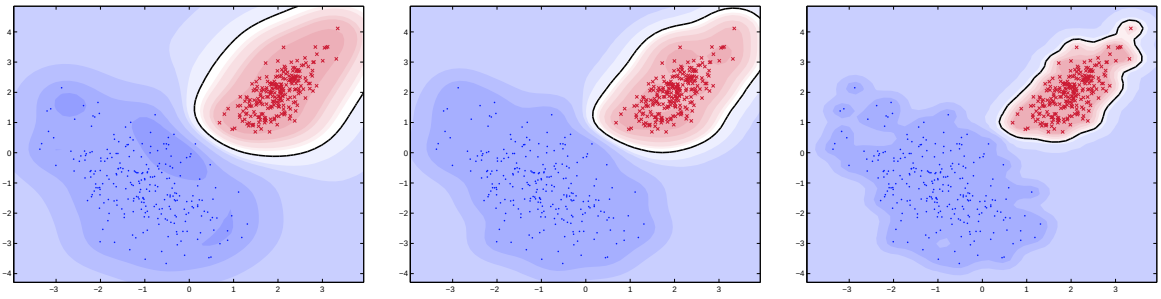
Au contraire de l'approche générative, les méthodes discriminatives estiment directement la probabilité *a posteriori* $P(Z|X)$ par la minimisation d'un coût de classification qui peut être pénalisé afin d'éviter le sur-apprentissage (voir paragraphe 2.4.1). Nous allons présenter ici les principales méthodes discriminatives qui s'avèrent être souvent très efficaces en pratique. Il est à noter en revanche que ces méthodes ne sont généralement pas nativement multi-classes, *i.e.* elles ne considèrent que le cas binaire (de 2 classes). Au chapitre 5, nous comparerons à ce type de méthodes l'approche générative adaptée aux données de grande dimension qui sera proposée dans ce mémoire.

La régression logistique

Cette approche, que l'on peut qualifier de semi-paramétrique, modélise donc directement la probabilité *a posteriori* et non les densités de probabilité des groupes. L'hypothèse qui est faite dans le cas de la régression logistique est que la différence entre les logarithmes des densités des deux classes est linéaire par rapport à x, c'est-à-dire que :

$$\log\left(f_1(x)\right) - \log\left(f_2(x)\right) = \beta_0 + \beta' x,$$

où β_0 et $\beta = (\beta_1, ..., \beta_p)$ sont des coefficients réels à déterminer à partir du jeu d'apprentissage. Leur estimation est généralement faite en utilisant la méthode du *maximum* de vraisemblance sur le jeu d'apprentissage. Les équations de vraisemblance n'ayant pas de solution analytique, il est nécessaire d'utiliser une méthode numérique de type Newton-Raphson. Une fois les β estimés, il est possible



$\gamma = 1$ $\gamma = 2$ $\gamma = 5$

FIG. 2.4 – Le classifieur SVM avec un noyau RBF : influence du paramètre γ.

d'appliquer la règle du MAP comme règle de décision. La règle de classification revient alors à assigner la nouvelle observation x à la classe C_1 si $\beta_0 + \beta' x > \log(\frac{\pi_2}{\pi_1})$ et de l'affecter à C_2 dans le cas contraire. Le principal avantage de la régression logistique est qu'elle est très générale car elle ne fait pas d'hypothèse sur la distribution de chacune des classes. De plus, la régression logistique requiert l'estimation d'un faible nombre de paramètres (de l'ordre de p) comparé à LDA (de l'ordre de p^2). En revanche, cette méthode de discrimination n'est pas nativement multi-classes et les estimateurs des paramètres n'ont pas de forme explicite. Pour de plus amples détails sur cette méthode, on pourra consulter [79, chap. 18].

Les *support vector machines*

Les « machines à vecteurs supports », traduction de l'anglais *Support Vector Machines* (SVM), sont une famille de classifieurs binaires qui a été introduite par Vapnik [89] au milieu des années 90 et qui connaît un franc succès dans la communauté du *machine learning*. Les SVM recherchent le meilleur hyperplan séparant deux groupes de sorte que cette frontière linéaire produise une marge maximale, *i.e.* tel que la distance des deux groupes à la frontière soit maximale. L'originalité des SVM est de ne pas contraindre cette recherche à l'espace d'origine. En effet, la recherche de l'hyperplan séparateur est en général faite dans des espaces de très grande dimension. La règle de décision des SVM affecte une nouvelle observation x en fonction du signe de la quantité $M(x)$:

$$M(x) = \sum_{j=1}^{n} \alpha_j \omega_j K(x, x_j) + \beta_0,$$

où les α_i et β_0 sont les coefficients des vecteurs supports, $\omega_j = 1$ si l'observation d'apprentissage appartient à la classe C_1 et $\omega_j = -1$ sinon. La fonction K est appelée le noyau et est définie par $K(x, x') =< h(x), h(x') >$, où $h(.)$ est l'opérateur de transformation des données. L'estimation des coefficients des vecteurs supports est un problème d'optimisation convexe qui peut être résolu par des outils d'optimisation classique. Enfin, un paramètre, usuellement noté γ, contraint le problème d'optimisation et doit être réglé par l'utilisateur. Le succès de ce type de méthodes est principalement

du à leur performance et à leur développement qui est très actif. D'autre part, les SVM ne subissent pas le fléau de la dimension (*cf.* 2.4.1) puisque le nombre de paramètres d'un classifieur SVM est de l'ordre de n et ne dépend pas de p. Ces méthodes ont pourtant quelques désavantages qui méritent d'être notés. La première limitation des SVM est la difficulté d'interprétation des règles de décision produites qui sont uniquement basées sur les observations d'apprentissage. La figure 2.4 montre la frontière de décision d'un classifieur SVM avec un noyau RBF (*Radial Basic Function*) de variance 1, *i.e.* $K(x, x') = \exp(-\gamma\|x-x'\|^2)$, et ce pour différentes valeurs de γ. On remarque que la frontière de décision d'une SVM peut être très dépendante des données d'apprentissage ce qui peut poser problème quand, par exemple, on ne dispose que de peu d'observations d'apprentissage. D'autre part, comme le fait remarquer Burges en conclusion de [15], les SVM sont des méthodes relativement coûteuses en temps de calcul et dont le paramétrage (choix du noyau, paramètres du noyau, contrainte de violation) est souvent difficile. Enfin, les SVM sont des classifieurs qui ne considèrent que la discrimination entre deux classes. Si l'on souhaite discriminer entre plus de deux classes, il est nécessaire d'utiliser la procédure proposée par Friedman [39] qui consiste à construire tous les classifieurs possibles entre deux groupes et ensuite d'affecter la nouvelle observation au groupe qui aura remporté le plus de matchs « un contre un ».

Les autres méthodes discriminatives

Il existe bien sûr un grand nombre d'autres méthodes discriminatives dédiées à la classification supervisée. Parmi ces dernières, on peut notamment citer la méthode des k plus proches voisins, les arbres de décision et le *perceptron* multi-couche. La méthode des k plus proches voisins [31] est certainement une des méthodes les plus anciennes de discrimination et est basée sur une stratégie locale similaire à celle de la méthode non-paramétrique d'estimation par noyau présentée au paragraphe 2.2.2. Cette méthode consiste pour une nouvelle observation à rechercher ses k plus proches voisins et à l'affecter au groupe majoritaire. Le paramètre k, dont le choix peut s'avérer crucial, peut être trouvé par validation croisée sur le jeu d'apprentissage. Cette méthode s'avère être stable et efficace dans le cas où l'on dipose d'un grand nombre d'observations d'apprentissage mais la recherche des voisins est alors relativement coûteuse. Les arbres de décision sont quant à eux plus utilisés en *data mining* et en économie et se basent sur un enchaînement de décisions binaires. La principale qualité des arbres de décision est la facilité d'interprétation de la règle de décision due à la hiérarchie des décisions. En revanche, ils fournissent des règles de décision relativement instables (car dépendantes du choix du premier noeud) et sont relativement lents. L'instabilité des arbres de décision peut être palliée par une approche, appelée *boosting* [37], qui consiste à appliquer la méthode de discrimination de manière répétée sur l'échantillon d'apprentissage en donnant de plus en plus d'importance aux points mal classés. Enfin, le *perceptron* multi-couche est une méthode neuronale qui recherche directement la meilleure séparation entre les groupes par minimisation d'un critère des moindres carrés. Cette méthode est également assez sensible à l'initialisation et doit être pénalisée pour éviter le

phénomène du sur-apprentissage (voir le paragraphe 2.4.1). Ces méthodes et leurs développements récents sont détaillés dans [42, chap. 7].

2.3 La classification automatique

La classification automatique, ou *clustering* en anglais, est la partie non-supervisée de la classification. La principale différence avec la classification supervisée est que le jeu de données, à partir duquel va être appris la règle de décision, ne comprend pas l'information de l'appartenance des observations aux classes. Autrement dit, en utilisant les notations du paragraphe 2.1, le jeu d'apprentissage ne contient que les observations $x_1, ..., x_n$ et les données $z_1, ..., z_n$ sont manquantes. Après avoir présenté le problème de la classification automatique, nous nous intéresserons à l'approche générative basée sur le modèle de mélange. Nous verrons que l'estimation des paramètres du mélange nécessite dans ce cas l'utilisation d'un algorithme itératif dénommé EM. Enfin, nous étudierons les limitations et extensions de cet algorithme. Nous décrirons enfin brièvement les méthodes discriminatives existantes.

2.3.1 Le problème de la classification automatique

Le problème de la classification automatique est de prédire les labels $z_1, ..., z_n$ des observations $x_1, ..., x_n \in \mathbb{R}^p$ sur la seule connaissance des valeurs prises par les p variables explicatives. Au contraire de l'analyse discriminante, la classification automatique ne dispose pas d'un jeu d'apprentissage pour apprendre les caractéristiques des classes. Une difficulté supplémentaire en classification automatique est que l'on ne connaît pas nécessairement le nombre k de groupes (nous y reviendrons au chapitre 4). Nous considérons principalement dans ce mémoire la classification automatique dans le cadre du modèle de mélange introduit précédemment. Il existe bien entendu d'autres approches que celles des modèles de mélange pour la classification automatique. De même qu'en analyse discriminante, les méthodes de *clustering* peuvent être divisées en méthodes génératives et méthodes discriminatives. Les méthodes génératives de *clustering* sont, de façon quasi-exclusive, basées sur le modèle de mélange et l'algorithme d'estimation EM. Les méthodes discriminatives, quant à elles, utilisent toutes une structure de classification hiérarchique.

2.3.2 Le modèle de mélange et l'algorithme EM

La maximisation de la log-vraisemblance d'un modèle de mélange dans le cas non-supervisé conduit en général à des équations de vraisemblance qui ne possèdent pas de solutions analytiques. Il existe toutefois différents algorithmes permettant de maximiser la log-vraisemblance quand les labels sont inconnus. Le plus utilisé d'entre eux est l'algorithme itératif *Expectation-Maximization* (EM) de Dempster, Laird et Rubin [26] que nous allons détailler dans ce paragraphe.

L'idée de l'algorithme EM

L'algorithme EM repose sur l'idée qu'il est plus facile de maximiser la vraisemblance complète $L(y;\theta)$ que la vraisemblance $L(x;\theta)$ et se base sur la relation suivante entre les deux vraisemblances :

$$L(x;\theta) = L(y;\theta) - \log f(y|x,\theta). \tag{2.7}$$

Cependant, la vraisemblance complète n'est pas non plus calculable du fait que y n'est pas totalement connue. Dempster, Laird et Rubin [26] ont proposé, pour maximiser cette vraisemblance, une procédure itérative qui se base sur la maximisation de l'espérance conditionnelle de la vraisemblance pour une valeur du paramètre courant θ'. L'algorithme consiste donc simplement à construire une suite $(\theta^{(q)})_q$ qui vérifie :

$$\theta^{(q+1)} = \operatorname*{argmax}_{\theta} Q(\theta, \theta^{(q)}),$$

où $Q(\theta,\theta') = E[L(y;\theta)|x,\theta']$. Si l'on note $t_{ij} = E[S_i|X = x_j,\theta']$ l'espérance du $i$ème élément du vecteur aléatoire S conditionnellement à $X = x_j$ et à θ', on peut alors écrire :

$$Q(\theta,\theta') = \sum_{j=1}^{n}\sum_{i=1}^{k} t_{ij}\log(\pi_i f(x_j,\theta_i)).$$

De plus, il a été montré que la suite ainsi générée fait croître la vraisemblance $L(x;\theta)$ et converge vers un *maximum* local de la vraisemblance sous certaines conditions de régularité. On pourra consulter [42, chap. 9] pour un plus ample développement du principe de cet algorithme d'estimation.

Les étapes de l'algorithme EM

Partant d'une solution initiale $\theta^{(0)}$, l'algorithme EM alterne entre deux étapes baptisées respectivement E, pour *expectation*, et M, pour *maximisation* :

Étape E : cette étape consiste à calculer, à l'itération q, l'espérance de S_i sachant la valeur du paramètre estimé à l'étape précédente et que $X = x_j$:

$$\begin{aligned} t_{ij}^{(q)} &= E[S_i|X = x_j, \hat{\theta}^{(q-1)}] \\ &= P(Z = i|X = x_j, \hat{\theta}^{(q-1)}), \end{aligned}$$

puisque S_i est à valeurs dans $\{0,1\}$. La formule de Bayes permet finalement de formuler les probabilités t_{ij} d'appartenance des x_j aux classes, et ce conditionnellement au paramètre courant $\hat{\theta}^{(q-1)}$, de la façon suivante :

$$t_{ij}^{(q)} = \frac{\hat{\pi}_i^{(q-1)} f(x_j, \hat{\theta}_i^{(q-1)})}{\sum_{\ell=1}^{k} \hat{\pi}_\ell^{(q-1)} f(x_j, \hat{\theta}_\ell^{(q-1)})},$$

Étape M : cette étape consiste quant à elle à maximiser, à chaque étape q, l'espérance de la vraisemblance complète conditionnellement aux $t_{ij}^{(q)}$. Les proportions du mélange sont obtenues simplement par la relation :

$$\hat{\pi}_i^{(q)} = \frac{n_i^{(q)}}{n},$$

où $n_i^{(q)} = \sum_{j=1}^{n} t_{ij}^{(q)}$ et les estimateurs des paramètres $\theta_1, ..., \theta_n$ sont obtenus en résolvant les équations de vraisemblance correspondantes au modèle de mélange retenu.

L'algorithme EM pour le mélange gaussien

De la même manière qu'en analyse discriminante, le modèle de mélange gaussien est fréquemment utilisé en classification automatique. L'algorithme EM est utilisé pour estimer les paramètres du mélange gaussien et l'étape M revient alors à calculer, à l'étape q, les estimateurs suivants :

$$\begin{aligned}
\hat{\pi}_i^{(q)} &= \frac{n_i^{(q)}}{n}, \\
\hat{\mu}_i^{(q)} &= \frac{1}{n_i^{(q)}} \sum_{j=1}^{n} t_{ij}^{(q)} x_j, \\
\hat{\Sigma}_i &= W_i = \frac{1}{n_i^{(q)}} \sum_{j=1}^{n} t_{ij}^{(q)} (x_j - \hat{\mu}_i^{(q)})(x_j - \hat{\mu}_i^{(q)})^t.
\end{aligned}$$

La figure 2.6 présente quelques étapes de l'algorithme EM pour l'estimation des paramètres d'un mélange de trois densités gaussiennes en dimension 2. La figure 2.5 montre l'évolution de la log-vraisemblance au cours des itérations de l'algorithme EM.

Limitations et extensions de l'algorithme EM

L'algorithme EM, que nous venons de présenter, découle naturellement des équations de vraisemblance et possède de bonnes propriétés statistiques. En effet, Wu [93] a établi que sous des conditions suffisantes de régularité l'algorithme EM assure une convergence vers un *maximum* local de la vraisemblance. Cependant, l'algorithme EM possède un certain nombre de limitations. La première limitation est que la valeur de l'estimateur à la convergence peut être fortement dépendante de l'initialisation. Une autre limitation de l'algorithme EM est que la convergence peut être lente et l'algorithme peut même se trouver bloqué dans un point selle de la vraisemblance. Des solutions à ces problèmes ont également été proposées dans le passé.

Stratégies d'initialisation Pour pallier la limitation de la dépendance de la solution vis à vis de l'initialisation, il est courant dans la pratique de lancer plusieurs EM pour quelques itérations depuis des initialisations aléatoires et de choisir la valeur de θ associée à la plus grande vraisemblance comme initialisation d'un autre EM qui lui itèrera jusqu'à convergence. McLachlan et Peel [58] proposent

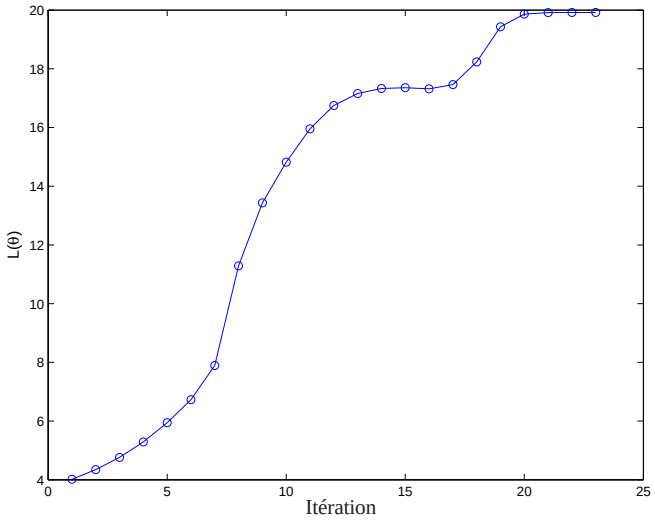


FIG. 2.5 – Évolution de la log-vraisemblance au cours des itérations de l'algorithme EM.

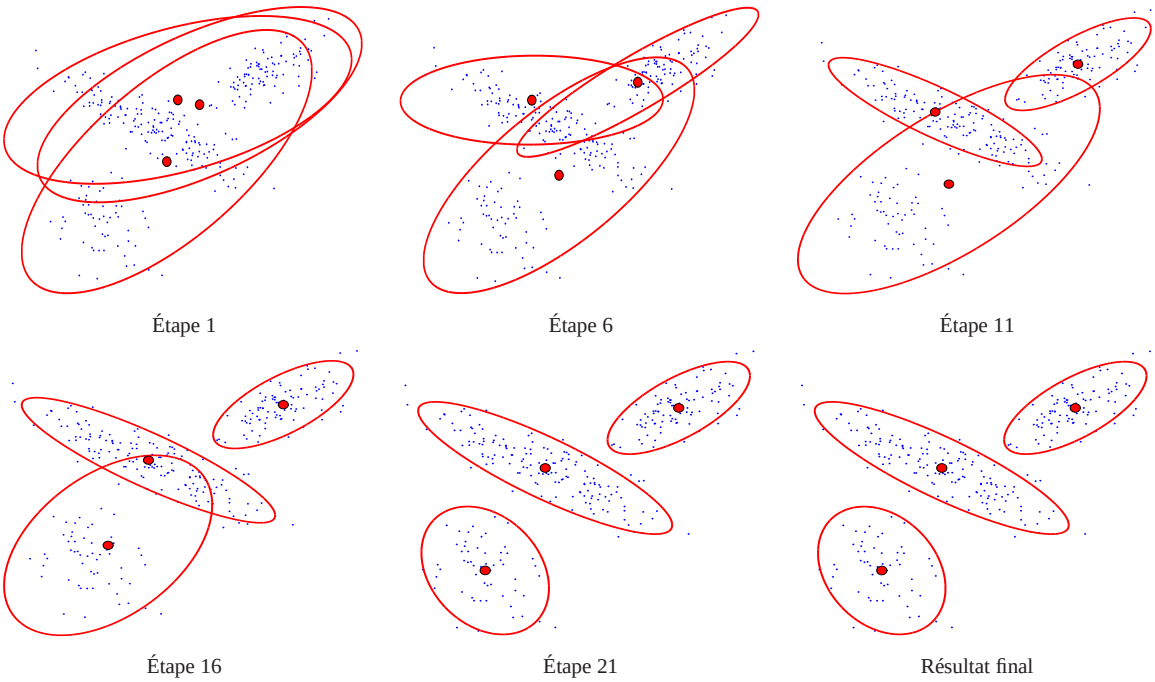


FIG. 2.6 – Quelques étapes de l'algorithme itératif EM en classification automatique. Le modèle de mélange utilisé est un mélange gaussien en dimension 2.

quant à eux une stratégie d'initialisation de θ dans le cas gaussien. Ils proposent de fixer les proportions du mélange à être égales et de générer les moyennes du mélange suivant une loi normale $\mathcal{N}(m, S)$, où m et S sont respectivement la moyenne et la matrice de covariance de l'échantillon entier. Les matrices de covariance des composantes du mélange seront également initialisées à la matrice de covariance S de l'échantillon entier. En pratique, la stratégie consistant à initialiser à plusieurs reprises de façon aléatoire et à choisir ensuite l'initialisation associée à la plus grande vraisemblance de θ est le plus souvent préférable.

Algorithme CEM Pour accélérer la convergence de l'algorithme, on peut choisir d'utiliser l'algorithme modifié CEM (Classification EM) [20] qui maximise la vraisemblance classifiante. L'algorithme CEM est obtenu à partir de l'algorithme EM classique en lui ajoutant une étape C de classification. Cette étape C affecte à chaque itération chacune des observations x_j à une des classes courantes grâce à la règle de MAP. Cela revient en fait à modifier les $t_{ij}^{(q)}$ en les remplaçant par les valeurs 1 ou 0 les plus proches. Cette approche, dite classification, fournit une estimation biaisée et inconsistante de θ et, d'un point de vue théorique, il est donc préférable d'utiliser l'approche mélange et l'algorithme EM. Cependant, la convergence de CEM est beaucoup plus rapide que EM et peut s'avérer utile lorsque l'on a des contraintes de temps ou pour traiter des jeux de données de tailles importantes. Il est intéressant de remarquer que l'algorithme des k-moyennes (k-*means*), qui est une méthode très populaire du fait de sa simplicité d'utilisation, peut être vu comme une méthode de classification automatique basée sur un modèle de mélange gaussien dont les matrices de covariance Σ_i sont toutes égales à la matrice identité et dont les proportions π_i sont également toutes égales. L'algorithme des k-moyennes procède de la façon suivante :

(i) Initialisation : on affecte aléatoirement les n observations dans les k classes,

(ii) Itération jusqu'à convergence :

  (a) on calcule les moyennes empiriques des k classes avec la partition courante,

  (b) on affecte chaque observation à la classe dont il est le plus proche de la moyenne.

(iii) Critère d'arrêt : on arrête l'algorithme quand il n'y a plus de changement d'affectation.

Algorithme SEM Afin d'éviter que l'algorithme EM se trouve bloqué dans un point selle de la vraisemblance, Celeux et Diebolt [18] ont proposé une version stochastique de EM qui évite que l'algorithme converge vers des cols de la vraisemblance et que le résultat soit trop dépendant de l'initialisation. L'algorithme SEM (Stochastique EM) s'obtient en ajoutant une étape S à l'algorithme EM classique qui modifie aléatoirement, à chaque étape q, l'appartenance des points aux classes en tenant compte des probabilités $t_{ij}^{(q)}$. Pour cela, on tire pour chaque point son appartenance à une des classes selon une loi multinomiale d'ordre 1 et de paramètre $(t_{ij}^{(q)}, i = 1, ..., k)$. Cet algorithme, qui est en fait plutôt une version stochastique de l'algorithme CEM, ne peut pas converger ponctuellement à cause des perturbations aléatoires de l'étape S mais Celeux et Diebolt ont toutefois montré qu'il converge

en loi. L'algorithme SEM doit donc être arrêté après un nombre d'itérations choisi par l'utilisateur. Il est également possible d'utiliser l'algorithme SAEM [19] qui diminue à chaque itération q l'influence des perturbations aléatoires de l'étape S grâce à une suite $(\gamma^{(q)})_q$ qui décroît vers 0 quand $q \to \infty$. Cette approche permet à l'algorithme de s'arrêter à la stationnarité de la vraisemblance.

2.3.3 Autres méthodes de classification automatique

Parmi les autres méthodes utilisées en classification automatique, la plus connue est la méthode de *clustering* hiérarchique. Cette méthode consiste à construire une hiérarchie de partitions en classes de moins en moins fines. Cette hiérarchie, représentée par un dendrogramme ou arbre de classification, est obtenue en regroupant successivement les points les plus proches au sens d'une certaine métrique. Cette stratégie, dite ascendante, regroupe tout d'abord les deux individus les plus proches au sein d'une première classe. A l'étape suivante, il ne reste alors plus que $n-1$ points à classer et l'on itère jusqu'au regroupement complet. Il ne reste plus alors qu'à définir à quelle profondeur il faut « couper » l'arbre pour obtenir la classification finale. Le lecteur pourra consulter [79, chap. 12] et [42, chap. 8] pour plus de détails sur ce type de méthodes.

2.4 Classification des données de grande dimension

Comme nous l'avons dit dans l'introduction de ce document, le monde scientifique d'aujourd'hui fournit des données qui sont chaque jour plus nombreuses et de plus grande dimension. On peut citer par exemple, le problème de la catégorisation du sol de la planète Mars (voir chapitre 7) pour lequel nous disposons à l'heure actuelle de 310 Go de données en dimension 256. Cependant, dans les mois à venir, le spectromètre imageur de nouvelle génération de l'orbiteur *Mars Reconnaissance Orbiter* devrait envoyer aux centres d'études spatiales 10 To de données multi-angulaires contenant quatre fois plus de dimensions que les données actuelles. On peut également citer l'analyse d'image où les données sont également de grande dimension, voir de très grande dimension si l'on considère les résolutions actuelles des appareils photos numériques (12 mégapixels). Dans ce paragraphe, nous allons étudier les différents problèmes posés par les données de grande dimension dans le contexte de la classification. Un exemple issu de l'analyse d'image illustrera ce paragraphe.

2.4.1 Le fléau de la dimension en classification

Dans la littérature, le terme de « fléau de la dimension » est abondamment utilisé pour caractériser les différentes manifestations de la grande dimension. Le « fléau de la dimension » est un terme que l'on doit à Bellman [6] qui l'utilisa comme principal argument en faveur de la programmation dynamique. La plupart des auteurs font référence au livre de Bellman intitulé « Adaptive Control Process : A Guided Tour » de 1961 comme la première apparition de ce terme, mais en réalité Bellman a utilisé la notion du « fléau de la dimension » dès 1957 dans la préface de son livre « Dynamic programming »

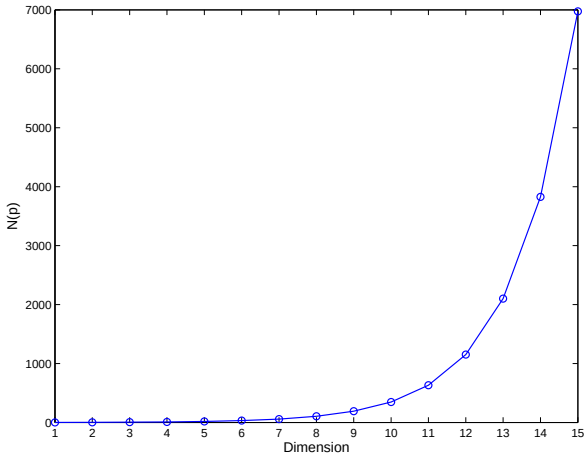


FIG. 2.7 – Nombre d'observations nécessaires à l'approximation d'une distribution gaussienne quelconque avec des noyaux gaussiens fixés avec une erreur maximale de 10% (voir [84]).

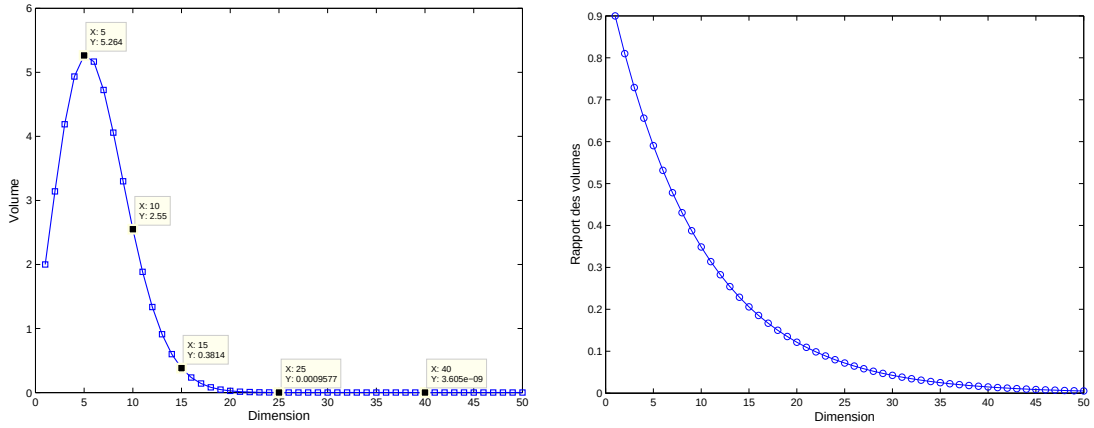
introduisant la programmation dynamique. Le site (très controversé) *books.google.com* nous a permis d'avoir accès à la préface de cet ouvrage dont voici la partie la plus intéressante pour notre propos :

> *All this* [les problèmes liés à la dimension] *may be subsumed under the heading « the curse of dimensionality ». Since this is a curse,* [...], *there is no need to feel discouraged about the possibility of obtaining significant results despite it.*

Nous verrons en effet dans la suite de ce mémoire qu'il existe des solutions à ce « fléau de la dimension » et qu'il peut même faciliter certaines tâches (dont la classification sous certaines conditions). Nous allons voir dans la suite de ce paragraphe quelles sont les principales manifestations de la grande dimension des données. Le lecteur pourra consulter [47, chap. 1], [82, chap. 7] où l'*Aide-Mémoire* de Donoho [27] pour plus de détails sur ces phénomènes.

Le fléau de la dimension à proprement parlé

Bellman utilisa le terme « fléau de la dimension » dans [6] pour parler de la difficulté d'optimiser une fonction par une recherche exhaustive de l'*optimum* dans un espace discrétisé. En effet, Bellman nous rappelle que si l'on considère une grille régulière de pas $1/10$ sur le cube unité dans un espace à 10 dimensions, nous obtenons 10^{10} points. Ainsi, pour rechercher l'optimum d'une fonction sur ce cube unité, il faut effectuer 10^{10} évaluations de la fonction. Si le cube unité en dimension 20 est considéré, alors il faudra effectuer évidemment 10^{20} évaluations de la fonction. Silverman [84] a également observé ce phénomène dans le cadre de l'approximation d'une distribution gaussienne quelconque avec des noyaux gaussiens fixés. Ses résultats montrent que le nombre N d'observations nécessaires à cette tâche avec une erreur maximale de 10% croît exponentiellement avec la dimension



(a) (b)

FIG. 2.8 – Phénomène de l'espace vide : (a) volume de la sphère unité en fonction de la dimension de l'espace, (b) rapport entre le volume des sphères de rayon 0.9 et 1 en fonction de la dimension.

et peut être approché par la relation suivante :

$$\log_{10} N(p) \simeq 0.6(p - 0.25).$$

La figure 2.7 montre l'évolution du nombre N d'observations nécessaires à l'approximation d'une distribution gaussienne quelconque avec des noyaux gaussiens fixés en fonction de la dimension p de l'espace.

Le phénomène de l'espace vide

Le « phénomène de l'espace vide », dont la paternité est usuellement attribuée à Scott et Thompson [83], met en évidence un des effets surprenants de la grande dimension et qui va à l'encontre de la représentation habituelle. En effet, nous allons voir que ce qui est naturel en dimension 1, 2 ou 3, ne peut pas être généralisé aux espaces de plus grande dimension. L'exemple dit du « volume de la boule » est classiquement utilisé pour illustrer ce phénomène. Le volume de la boule unité dans un espace de dimension p est donné par la relation :

$$V(p) = \frac{\pi^{p/2}}{\Gamma(p/2+1)},$$

où Γ est la fonction *gamma* usuelle. Le graphique de gauche de la figure 2.8 montre l'évolution du volume de la boule unité en fonction de la dimension de l'espace dans lequel elle se trouve. Il apparaît que le volume de la boule unité devient quasiment nul quand la dimension devient grande. Afin de mieux appréhender ce phénomène, il est préférable de comparer le volume de la boule à une autre valeur qui soit naturelle pour nous. Il est par exemple possible de comparer le volume de la boule de rayon 0.9 à celui de la boule unité. Le graphique de droite de la figure 2.8 présente l'évolution de ce

rapport en fonction de la dimension de l'espace. Naturellement, ce rapport décroît vers 0 quand la dimension augmente. Cet exemple a également été considéré par Huber [49] qui propose de simuler des réalisations d'un vecteur aléatoire suivant la loi uniforme dans la boule unité de $\mathbb{R}^p$. La probabilité qu'un point se trouve dans l'espace compris entre les boules de rayon 0.9 et 1 s'exprime de la façon suivante en fonction de la dimension p :

$$P(p) = 1 - 0.9^p.$$

En particulier, la probabilité de trouver un point dans la coquille comprise entre les boules de rayon 0.9 et 1 dans un espace de dimension 20 est à peu près égale à 0.88. Ces exemples montrent que l'espace de dimension p est presque vide puisque la très grande majorité des points se situe aux alentours d'un espace de dimension $p - 1$. Certains auteurs, dont Verleysen [90], utilisent d'ailleurs le phénomène de l'espace vide pour définir la limite entre les espaces de petite dimension et ceux de grande dimension. L'observation du graphique de gauche de la figure 2.8 permettrait alors de conclure que les espaces dont la dimension est plus grande que 5 sont des espaces de grande dimension. Plus simplement, certains pensent que si la dimension d'un espace est plus grande que 3, alors c'est un espace de grande dimension puisque l'humain ne peut pas naturellement se le représenter.

Le fléau de la dimension en classification générative

De manière générale, les méthodes génératives de classification requièrent l'estimation d'un nombre de paramètres qui croît avec le carré de la dimension. Cela est principalement du à l'estimation des matrices de covariance qui concentrent la plus grande part des paramètres des méthodes. En particulier, nous avons vu au paragraphe 2.1.1 que si le modèle de mélange gaussien est choisi pour discriminer des données, alors la règle de décision repose en partie sur l'inverse des matrices de covariance des classes (cela est également vrai dans le cadre de la classification automatique). En effet, le calcul de la fonction de coût K_i requiert l'inversion des k matrices Σ_i pour QDA ou de l'unique matrice Σ pour LDA. Il est donc clair que la qualité de la classification dépend directement de l'estimation de ces matrices.

Singularité des matrices de covariance Il est clair que si le nombre d'observations est trop petit devant la dimension de l'espace, alors les estimations de ces matrices seront singulières et leur inversion sera numériquement impossible. En effet, lorsque $n < p$, l'estimation de la matrice Σ est singulière et les $p - n + 1$ plus petites valeurs propres sont estimées par 0. Les vecteurs propres correspondants sont alors arbitraires. En particulier, LDA ne pourra tout simplement pas être utilisée dans ce cas.

Mauvais conditionnement des matrices de covariance Si les estimations des matrices de covariance sont mal conditionnées, leur inversion entraînera un important biais sur le calcul de la règle de décision et donc une importante erreur de prédiction. Pavlenko *et al.* [70] ont proposé un exemple dans

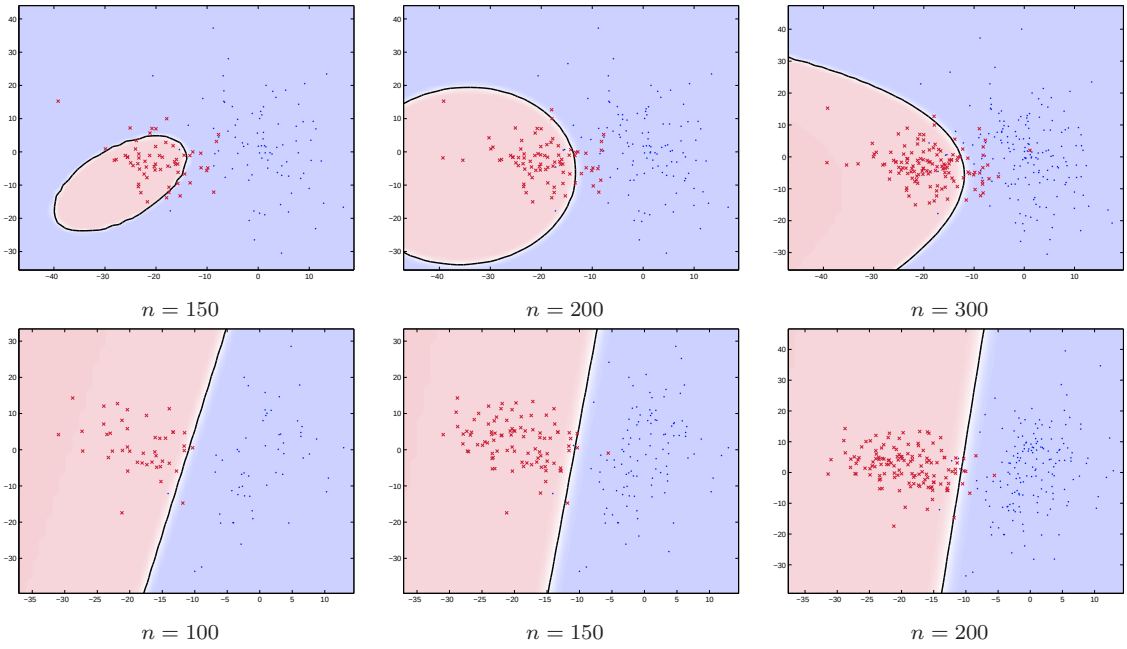


FIG. 2.9 – Influence de la taille du jeu d'apprentissage sur la règle de décision des classifieurs QDA (en haut) et LDA (en bas). Les données vivent dans un espace de dimension 50 et sont projetées sur les axes principaux uniquement pour la visualisation.

le cas gaussien illustrant ce phénomène. Considérons la trace normalisée de l'inverse de la matrice de covariance Σ d'une distribution gaussienne multivariée $N(\mu, \Sigma)$ de dimension p :

$$\tau(\Sigma) = \frac{1}{p}\,\mathrm{tr}(\Sigma^{-1}).$$

Considérons d'autre part que nous disposions d'un jeu d'observations indépendantes $x_1, ..., x_n$. Classiquement, la matrice de covariance Σ est estimée par son équivalent empirique $\hat{\Sigma} = \frac{1}{n}\sum_{j=1}^{n}(x_j - \hat{\mu})(x_j - \hat{\mu})^t$, où $\hat{\mu} = \sum_{j=1}^{n} x_j/n$ est lui aussi l'équivalent empirique de μ. Nous pouvons en déduire un estimateur de $\tau(\Sigma)$:

$$\widehat{\tau(\Sigma)} = \tau(\hat{\Sigma}) = \frac{1}{p}\,\mathrm{tr}(\hat{\Sigma}^{-1}),$$

et son espérance est :

$$E[\tau(\hat{\Sigma})] = \left(1 - \frac{p}{n}\right)^{-1}\tau(\Sigma).$$

Ainsi, si le rapport $p/n \to 0$ quand $n \to \infty$ alors $E[\tau(\hat{\Sigma})] \to \tau(\Sigma)$. L'estimateur de $\tau(\Sigma)$ est alors asymptotiquement sans biais. Par contre, si la dimension p est comparable au nombre d'observations n, alors $E[\tau(\hat{\Sigma})] \to c\,\tau(\Sigma)$ quand $n \to \infty$, où $c = \lim_{n\to\infty} p/n$ est une constante. L'estimateur de $\tau(\Sigma)$ est alors biaisé. On pourra consulter [70, 69] pour une étude théorique de l'effet de la dimension sur la classification dans un cadre strictement asymptotique, *i.e.* la dimension augmente quand le nombre d'observations augmente. Des méthodes de régularisation dédiées à la classification ont

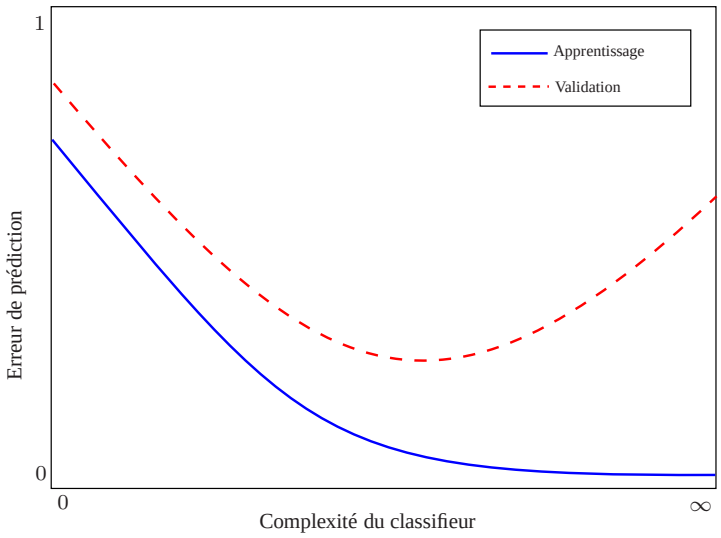


FIG. 2.10 – Phénomène du sur-apprentissage : erreur de prédiction en fonction de la complexité du classifieur.

aussi été proposées et nous les présenterons au paragraphe 2.4.3. La figure 2.9 met en évidence l'influence de la taille de l'échantillon d'apprentissage sur la règle de décision de classifieurs génératifs. On remarque notamment que QDA est beaucoup plus sensible que LDA à la taille de l'échantillon d'apprentissage.

Sur-apprentissage

Dans le cadre supervisé, le phénomène du sur-apprentissage peut survenir si le classifieur (génératif ou discriminatif) est trop complexe, *i.e.* s'il tient compte d'un trop grand nombre de paramètres. En effet, si le classifieur est très complexe, il va parfaitement « épouser » la forme des données d'apprentissage et du coup devenir très dépendant de ces données. Il aura sur-appris la forme de ces données. Dans le cas où les données d'apprentissage ne seraient pas représentatives du processus qui les a générées, le classifieur ne pourra pas être efficace pour traiter de nouvelles données, différentes des données d'apprentissage. La figure 2.10 montre le comportement typique des erreurs de prédiction des jeux d'apprentissage et de validation. Le taux d'erreur du jeu d'apprentissage a tendance à décroître quand le degré de complexité du classifieur croît : le classifieur épouse de mieux en mieux la forme des données d'apprentissage. En revanche, si le modèle tend à être trop complexe, il n'est alors plus assez général et l'erreur de prédiction sur le jeu de validation croît de nouveau. D'autre part, si le classifieur est trop simple, il ne pourra ni être efficace sur les données d'apprentissage ni sur les données de test. On parle dans ce cas de « sous-apprentissage ». Il est donc important de trouver le bon degré de complexité du classifieur pour obtenir un classifieur efficace. Ce phénomène est bien entendu accentué par la grande dimension des données puisque la complexité du classifieur est en général liée à la dimension de l'espace.

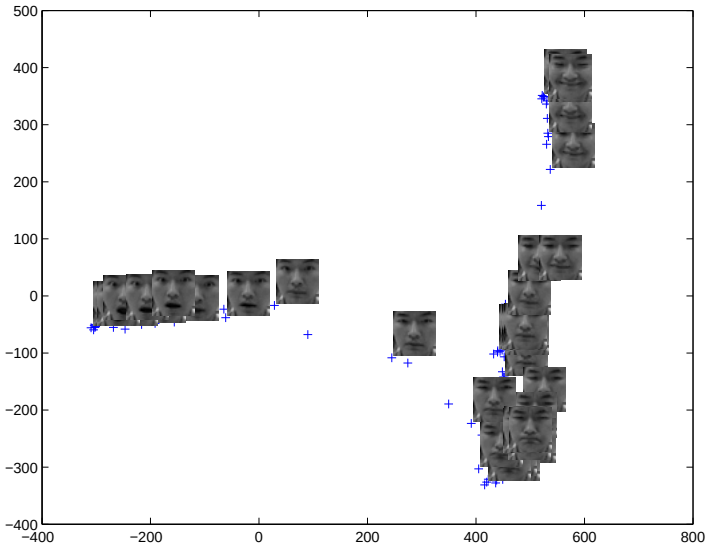


FIG. 2.11 – Projection sur les deux premiers axes principaux des données « visages » associées au sujet 1 et dont la dimension originale est 1024.

Le fléau de la dimension : un faux problème ?

Certains problèmes modernes, tels que la reconnaissance de visages, fournissent des données en très grande dimension (un millier de dimensions) et le nombre n d'observations disponibles est généralement beaucoup plus faible que la dimension p. Il est clair que, dans ce cas, la dimension des données est artificiellement augmentée par le processus d'acquisition. En effet, d'un point de vue géométrique, n points vivent dans un espace de dimension au plus $(n-1)$. Cet exemple met en évidence le fait que la dimension acquise est, en général, nettement supérieure à la dimension intrinsèque des données. Ce raisonnement induit qu'une grande part des variables est corrélée et donc qu'une grande part de l'information est redondante. Par conséquent, si nous parvenons à nous ramener à un système de d variables indépendantes, alors le fléau de la dimension sera fonction de la dimension intrinsèque d qui peut être très faible devant p. La dimension intrinsèque des données est directement liée au nombre de degrés de liberté du processus qui a généré les données. Par exemple, le jeu de données « visages »[1] est composé d'images de résolution 32x32 représentées dans un espace de dimension 1024. Pour chacun des 13 sujets, nous disposons de 75 images associées à différentes expressions du visage. La figure 2.11 montre la projection des données correspondantes au sujet 1 sur les deux premiers axes principaux. Il apparaît que ces données dont la dimension originale est 1024 vivent en réalité dans une variété dont la dimension intrinsèque est proche de 1. Cela correspond en effet au nombre de degré de liberté du processus qui a généré ces données : chaque sujet devait faire évoluer son expression faciale en ne modifiant qu'un seul « paramètre » de son visage (ouverture de la bouche, froncement des sourcils, ...).

[1] disponible à l'adresse *http ://amp.ece.cmu.edu/downloads.htm*.

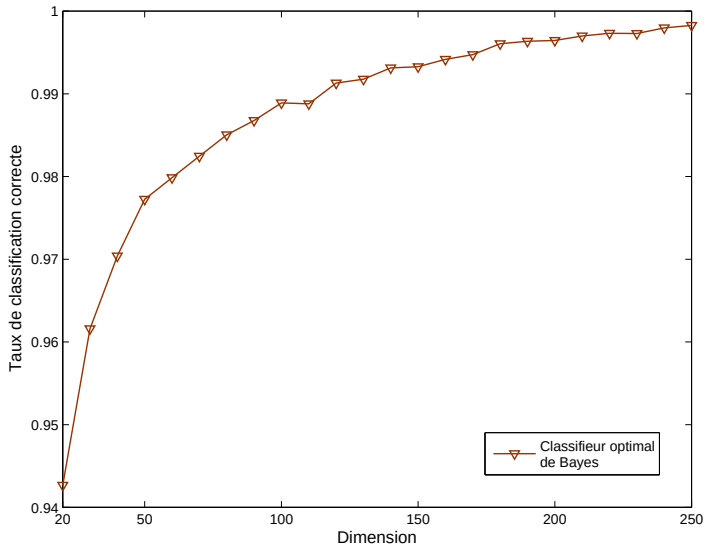


FIG. 2.12 – Taux de classification correcte du classifieur optimal de Bayes sur le jeu de validation et en fonction de la dimension de l'espace original. Ces résultats ont été obtenus sur données simulées (voir le texte pour plus de détails).

D'autre part, on observe généralement un phénomène intéressant dans le cadre de la classification : plus la dimension de l'espace est grande, plus la classification des données est facile avec un classifieur adapté. La figure 2.12 met en évidence ce phénomène en observant le taux de classification correcte du classifieur optimal de Bayes, *i.e.* basé sur les densités réelles, en fonction de la dimension de l'espace original. Pour cette étude, nous avons simulé des données issues de trois densités gaussiennes dans $\mathbb{R}^p$, $p = 20, ..., 250$. Les données de chacun de ces trois groupes vivent dans des sous-espaces de dimensions intrinsèques respectives 2, 5 et 10 et leurs moyennes sont très proches. Ces données ayant été simulées par nos soins, nous connaissons la densité réelle de chacun des groupes et nous sommes donc en mesure de calculer le taux de classification correcte du classifieur optimal de Bayes (voir paragraphe 2.1.2). Afin de moyenner les résultats obtenus, nous avons répété l'expérience à 50 reprises et chacun des jeux de validation utilisés était composé de 1000 observations. On observe clairement sur la figure 2.12 que le taux de classification correcte du classifieur de Bayes, qui est le classifieur le plus adapté à ces données, croît avec la dimension de l'espace original. Cela traduit le fait que, avec un classifieur adapté, la tâche de classification est plus facile dans un espace de grande dimension que dans un espace de faible dimension. Ce phénomène est en particulier exploité par les méthodes de discrimination SVM, présentées au paragraphe 2.2.3, qui augmentent artificiellement la dimension des données pour faciliter leur discrimination.

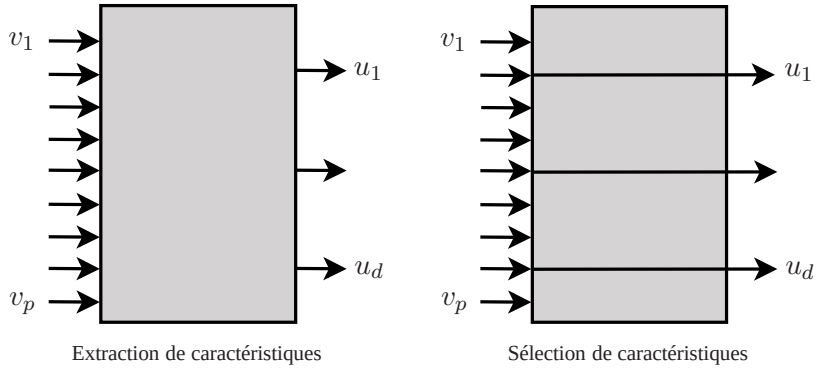


FIG. 2.13 – Principe des méthodes de réduction de dimension par extraction et sélection de caractéristiques.

2.4.2 Réduction de dimension

Une des solutions qui peut être mise en œuvre pour limiter les effets du « fléau de la dimension » est de réduire la dimension des données avant de les traiter. C'est en effet la solution la plus naturelle puisqu'elle prend le problème à la source : la dimension est trop grande, alors réduisons-la ! Nous avons vu au travers de l'exemple des « visages » du paragraphe 2.4 que la dimension de l'espace d'observation des données est le plus souvent bien supérieure à la dimension intrinsèque d des données. De fait, il est théoriquement possible de réduire la dimension de l'espace à d dimensions et ce sans entraîner de perte d'information. L'enjeu est donc d'identifier les dimensions (ou les combinaisons de dimensions) qui sont porteuses d'informations redondantes. Les techniques de réduction de dimension sont traditionnellement divisées en deux catégories : les méthodes d'extraction de caractéristiques (*feature extraction*) et les méthodes de sélection de caractéristiques (*feature selection*). Les méthodes d'extraction de caractéristiques construisent, à partir des p variables (dimensions) originales, d nouvelles variables qui contiennent la plus grande part possible de l'information initiale. Parmi toutes les techniques d'extraction de caractéristiques existantes, la plus connue et la plus utilisée est très certainement l'analyse en composantes principales (ACP ou PCA en anglais) qui est une méthode linéaire. Les méthodes de sélection de caractéristiques, quant à elles, cherchent un sous-ensemble de d variables parmi les p variables originales. La recherche peut-être optimale en utilisant une méthode de sélection exhaustive si le nombre de dimensions de l'espace original n'est pas trop grand. En pratique, ces méthodes de recherche exhaustive ne sont pas utilisables avec les données modernes qui sont décrites par un trop grand nombre de dimensions. En effet, le nombre de sous-ensembles possibles est égal à C_p^d :

$$C_p^d = \frac{p!}{(p-d)!d!}.$$

On comprend vite la nécessité d'introduire des méthodes sous-optimales. D'autre part, les différentes méthodes de sélection de variables se différencient les unes des autres de par le choix du critère mesurant la pertinence du sous-ensemble de variables. Ces méthodes sont détaillées dans [43] et [91,

chap. 9]. Il est à noter que la sélection de variables dans le cadre de la classification automatique a été notamment considérée dans [74]. Dans la suite de ce paragraphe, nous allons nous focaliser sur les méthodes d'extraction de caractéristiques qui sont certainement les plus utilisées. On pourra trouver un « tour d'horizon » détaillé des méthodes de réduction de dimension existantes dans [35] et [16].

L'analyse en composantes principales (ACP)

On doit très certainement l'analyse en composantes principales à Pearson [72] qui cherchait à approcher « un système de points dans l'espace », selon sa terminologie, par un sous-espace linéaire de dimension inférieure. Plus précisément, Pearson étudiait le problème d'approcher des données multivariées par une droite telle qu'elle minimise la somme des écarts des points à la droite au carré. Nous allons dans ce paragraphe présenter brièvement le fondement théorique de l'ACP. Le lecteur désirant de plus amples détails pourra consulter [50] ou [79, chap. 8]. Le problème de l'ACP est de trouver le sous-espace affine E de dimension $d < p$, souvent $d = 2$, tel que l'inertie J de l'ensemble des points du nuage par rapport E soit minimum. L'inertie J s'exprime de la façon suivante :

$$J = \frac{1}{n} \sum_{j=1}^{n} \|x_j - P_E(x_j)\|^2,$$

où $P_E(x_j)$ est la projection de x_j sur le sous-espace E. Généralement, M est la métrique identité. Cela revient à rechercher les axes le long desquels la variance est maximale. Nous allons donc rechercher les axes qui maximisent la variance des vecteurs dans l'espace de projection. Le long d'un axe représenté par le vecteur unitaire u, la variance totale est :

$$V = \sum_{j=1}^{n} [u^t(x_j - \bar{x})]^2,$$

où $\bar{x}$ est le barycentre de l'ensemble des vecteurs x_j, $j = 1, ..., n$. Il a été montré [79] que le vecteur u qui réalise cette maximisation est le vecteur propre associé à la plus grande valeur propre de la matrice de covariance empirique $\hat{\Sigma}_{totale} = \frac{1}{n} \sum_{j=1}^{n} (x_j - \bar{x})(x_j - \bar{x})^t$. On les note respectivement u_1 et λ_1. De même, dans l'espace orthogonal à u_1, l'axe qui maximise la variance est supporté par le vecteur propre associé à la seconde plus grande valeur propre de $\hat{\Sigma}_{totale}$, et ainsi de suite pour les p axes principaux. Les vecteurs u_j sont appelés les *facteurs principaux*. De même, les variables artificielles définies comme projections sur les facteurs principaux par la relation $c_j = X\, u_j$, sont appelées *composantes principales*. Le fait que $\hat{\Sigma}_{totale}$ soit une matrice symétrique semi-définie positive nous assure que les valeurs propres sont toutes réelles, positives ou nulles et les vecteurs propres sont orthogonaux entre eux. De plus, la valeur propre λ_ℓ, $\ell = 1, ..., d$, étant égale à la part de la variance totale portée par la composante principale associée, cela permet de sélectionner les axes formant l'espace de projection. Il suffit de retenir les d premiers vecteurs propres tels que $\sum_{\ell=1}^{d} \lambda_\ell$ représente une certaine proportion de la variance initiale (par exemple, 90%). Ainsi, les composantes principales sont les combinaisons

linéaires des variables initiales de variance maximale. L'ACP étant une méthode de réduction de dimension, il est important de savoir qu'elle ne peut pas retenir la totalité de l'information contenue dans le nuage de points initial. Enfin, l'ACP prend uniquement en compte les dépendances linéaires entre les variables et ne peut donc pas fournir une projection fidèle pour une distribution non-linéaire de points.

Combien de dimensions faut-il retenir ?

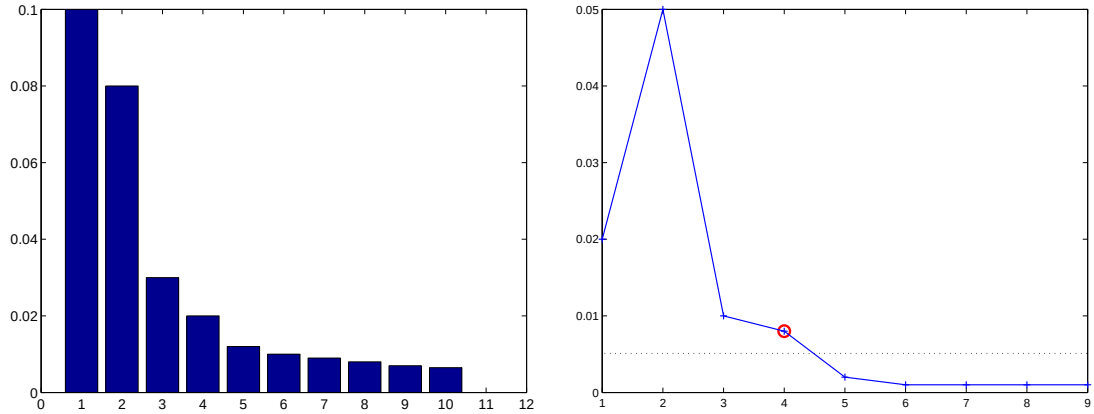
Ce point essentiel de la réduction de dimension, par extraction ou sélection de variables, ne possède malheureusement pas de solution explicite. Nous allons présenter uniquement ici des critères permettant de déterminer le nombre de dimensions à retenir dans le cadre de l'ACP. La plupart des techniques de recherche du nombre d'axes à retenir sont basées sur les valeurs propres de la matrice de covariance Σ des données. Cette approche se justifie par le fait que chaque valeur propre de Σ représente la variance portée par le vecteur propre associé.

Critère théorique Il est tout d'abord possible de tester, d'un point de vue statistique, si les $(p-d)$ dernières valeurs propres ne sont pas significativement différentes. Si elles ne le sont pas, on peut alors retenir les d premières dimensions. On fait pour cela l'hypothèse que les n observations sont les réalisations d'un vecteur aléatoire gaussien dont les $(p-d)$ dernières valeurs propres $\lambda_{d+1}, ..., \lambda_p$ de la matrice de covariance Σ sont égales. Sous cette hypothèse, la moyenne arithmétique m_a des $(p-d)$ dernières valeurs propres doit être peu différente de leur moyenne géométrique m_g. On définit Ξ :

$$\Xi = \left(n - \frac{2p+11}{6}\right)(p-d)\log\left(\frac{m_a}{m_g}\right),$$

qui suit une loi du χ^2 à $\frac{(p-d+2)(p-d-1)}{2}$ degrés de liberté. On rejettera l'hypothèse d'égalité des $(p-d)$ dernières valeurs propres si Ξ est trop grande. Notre expérience nous permet de dire que ce critère est rarement utilisable en pratique car il a tendance à surestimer le nombre de dimensions à retenir et ce, même pour une valeur élevée du seuil de confiance du test.

Critère empirique Dans la pratique, le critère empirique du *scree-test* de Cattell [17] est couramment utilisé. Ce critère est basé sur l'analyse des différences entre les valeurs propres consécutives et permet de détecter un « coude » dans l'éboulis des valeurs propres. La dimension sélectionnée par la méthode est celle pour laquelle les différences entre les valeurs propres suivantes sont toutes plus petites qu'un certain seuil. La figure 2.14 illustre cette technique : l'image de gauche présente les valeurs propres de Σ ordonnées de façon décroissante et à droite on peut observer les différences entre les valeurs propres consécutives. Dans cet exemple, le seuil a été fixé à 10% de la plus grande différence et le *scree-test* de Cattell identifie un coude au niveau de la 4ème dimension. L'observation de l'éboulis des valeurs propres (à gauche) confirme ce choix.



(a) (b)

FIG. 2.14 – Choix du nombre de dimensions à retenir grâce au critère empirique du *scree-test* de Cattell [17] : (a) éboulis des valeurs propres de Σ et (b) différences entre les valeurs propres consécutives.

Méthodes non-linéaires d'extraction de caractéristiques

Récemment, de nombreuses méthodes de réduction de dimension non-linéaire ont été proposées. Une première catégorie de méthodes se proposent d'étendre l'ACP linéaire classique au cas non-linéaire. Parmi ces méthodes, *Kernel-PCA* (KPCA) [80] utilise les fonctions noyaux des SVM (voir paragraphe 2.2.3) pour transformer les données originales avant d'appliquer une ACP classique sur les données transformées. Les méthodes dites de « courbes principales » ou de « surfaces principales » [24, 41, 46] recherchent, non plus un hyper-plan comme en ACP, mais une hyper-surface paramétrée et lisse qui approche au mieux les données. La seconde catégorie de méthodes de réduction de dimension non-linéaire est basée sur l'idée que les données sont disposées sur une variété non-linéaire de dimension intrinsèque d dans l'espace de dimension p. Ces méthodes ont généralement comme principal objectif de permettre la visualisation des données de grande dimension. Pour cela, elles cherchent à « déplier » la variété sur laquelle vivent les données. Ces méthodes font parties des méthodes neuronales dans le sens où les points (qui jouent le rôle de neurones) cherchent leur position dans l'espace de sortie tout en respectant (tout au moins localement) la topologie d'entrée. La première méthode de ce type qui a été proposée est le *Multi-Dimensional Scaling* (MDS) dont on pourra trouver la technique détaillée dans de nombreux livres tels que [47]. Ces dernières années, plusieurs méthodes dérivées ont vues le jour qui s'opposent principalement sur la question du critère de similarité entre les topologies d'entrée et de sorties. Parmi ces extensions, nous pouvons citer Analyse en Composante Curviligne (CCA) [25], la *Locally Linear Embedding* (LLE) [76] et la méthode *Isomap* [86].

Analyse Factorielle Discriminante (FDA)

Les méthodes de réduction de dimension présentées précédemment ne prennent cependant pas en compte l'objectif de classification qui est à l'origine de la nécessité pour nous de réduire la dimension. L'Analyse Factorielle Discriminante (FDA en anglais) combine quant à elle la réduction de dimension et la discrimination. En effet, effectuer une analyse factorielle discriminante consiste à projeter les données de $\mathbb{R}^p$ sur les $d = (k-1)$ axes discriminants qui maximisent le rapport de la variance inter-classe et de la variance intra-classe, puis d'apprendre la règle de décision δ^* sur les données projetées. Nous avons donc besoin de définir la matrice de variance inter-classe empirique B :

$$B = \frac{1}{n}\sum_{i=1}^{k}\sum_{j=1}^{n} s_{ij}(\hat{\mu}_i - \hat{\mu})(\hat{\mu}_i - \hat{\mu})^t,$$

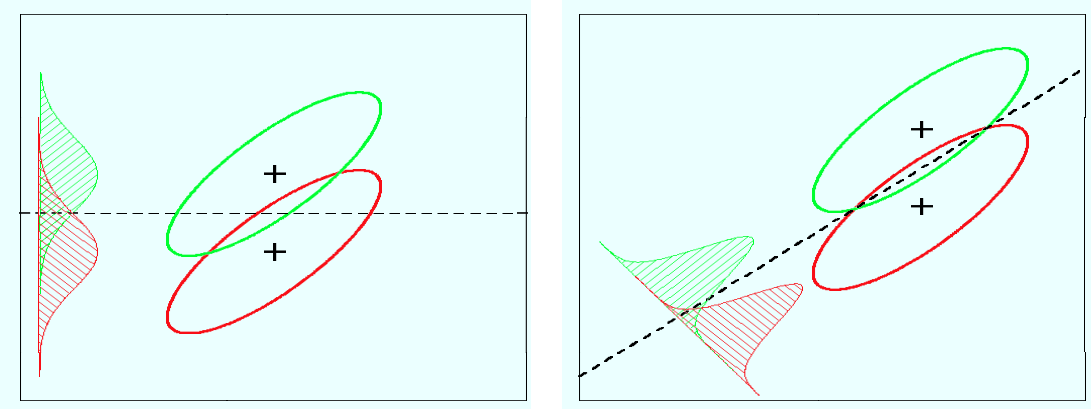
où $\hat{\mu} = \frac{1}{n}\sum_{i=1}^{k}\sum_{j=1}^{n} s_{ij}\hat{\mu}_i$. Le théorème de Huyghens nous permet d'obtenir la relation suivante liant les matrices de variance inter et intra-classe à la matrice de variance totale empirique $\hat{\Sigma}_{totale}$:

$$\hat{\Sigma}_{totale} = B + W.$$

Nous souhaitons trouver une représentation des données qui permette de discriminer les groupes le mieux possible. Pour ce faire, il faut que les projections des k centres de gravité soient les plus séparées possible, tandis que les données de chaque classe doivent se projeter de façon groupée autour du centre de gravité de leur classe. Nous recherchons donc une représentation des données qui maximise la variance inter-classe et qui minimise la variance intra-classe. Avec les notations et résultats précédents, les axes de la projection recherchée satisfont le problème d'optimisation suivant :

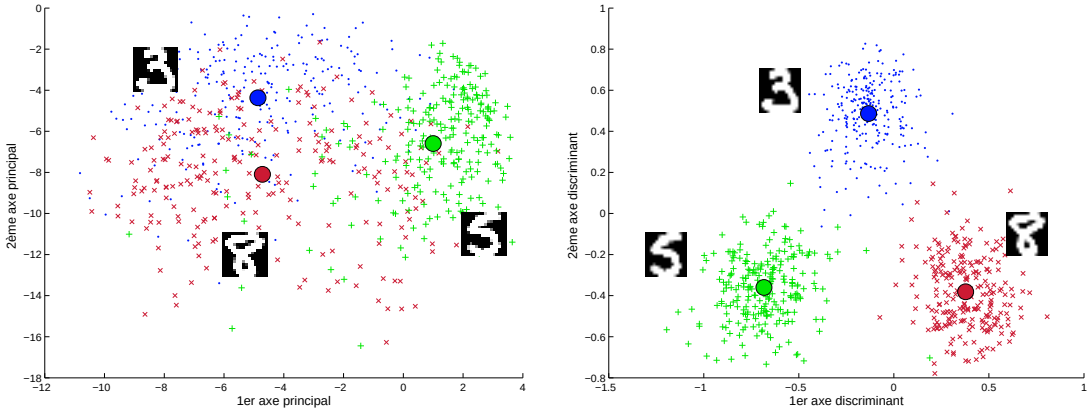
$$\max_{u} \frac{u'Bu}{u'\hat{\Sigma}_{totale}u}. \tag{2.8}$$

On sait que ce *maximum* est atteint pour u vecteur propre de $\hat{\Sigma}_{totale}^{-1}B$ associé à sa plus grande valeur propre. La figure 2.15 illustre le choix d'un axe de projection permettant de discriminer au mieux les classes. Une fois la projection déterminée, on peut alors effectuer une analyse discriminante linéaire. Cette stratégie qui combine réduction de dimension et discrimination est souvent profitable car les données de chaque classe n'occupent en général pas la totalité de l'espace et cela permet de réduire le nombre de paramètres à estimer. Cette méthode se révèle relativement efficace sur des données de grande dimension comme on peut le constater sur la figure 2.16 qui présente la projection des données USPS (voir chapitre 5) sur les axes principaux d'une part, et sur les axes discriminants d'autre part. En revanche, on peut remarquer que cette approche nécessite toutefois l'inversion de la matrice de variance totale empirique $\hat{\Sigma}_{totale}$, ce qui peut poser problème si celle-ci est mal conditionnée (voir paragraphe 2.4.1).



(a) (b)

FIG. 2.15 – L'axe principal de la figure (a) ne permet pas de discriminer efficacement les deux groupes alors que celui de la figure (b) possède un bon pouvoir discriminant (figures extraites de [47]).



(a) (b)

FIG. 2.16 – Projection des données correspondantes aux chiffres 3, 5 et 8 de la base USPS sur (a) les 2 premiers axes principaux et (b) les 2 premiers axes discriminants.

2.4.3 Méthodes de régularisation

Comme nous l'avons dit, l'analyse discriminante linéaire peut être considérée comme une méthode de référence du fait de sa robustesse. Toutefois, cette propriété de robustesse n'est plus vérifiée quand la taille de l'échantillon devient trop faible devant la dimension de l'espace. Cette remarque est encore plus vraie en ce qui concerne l'analyse discriminante quadratique. Au début des années 1990, des méthodes dites d'analyse discriminante régularisée ont vues le jour, ayant comme but de stabiliser les résultats de l'Analyse Discriminante dans ce cas limite. On pourra consulter [62] pour une synthèse sur le sujet. Nous avons vu au paragraphe 2.4.1 que dans le cas de petits échantillons les matrices de covariance sur lesquelles se basent les méthodes classiques d'analyse discriminante sont mal conditionnées voir non inversibles. Cela entraîne évidemment une détérioration de la performance du classifieur. Nous allons présenter dans ce paragraphe les principales méthodes existantes de régularisation dans le cadre de la classification. Une récente étude [53] a évalué les performances de ces méthodes de régularisation ainsi que des méthodes basées sur des modèles de mélange gaussien parcimonieux dans le cadre de la classification de puces ADN.

Régularisation simple

Pour pallier les problèmes liés au mauvais conditionnement ou à la singularité des estimations des matrices de covariance des classes, il est tout d'abord possible d'utiliser le pseudo-inverse à la place de l'inverse classique. On peut également ajouter une constante σ^2 positive à la diagonale des matrices de covariance estimées :

$$\tilde{\Sigma}_i = \hat{\Sigma}_i + \sigma_i^2 I_p.$$

Cette régularisation numérique simple est du même type que la régularisation *ridge* utilisée en régression. Zhong *et al.* [95] ont également proposé d'utiliser cette régularisation simple en régression inverse pour la détection de motifs en génétique. Enfin, il est important de noter que ce type de régularisation est généralement effectué dans les logiciels de statistique (c'est notamment le cas pour la fonction LDA de Matlab) sans que cela soit notifié à l'utilisateur.

Analyse discriminante régularisée (RDA)

Historiquement, on doit à Friedman [38] la première méthode régularisée d'analyse discriminante qu'il baptisa d'ailleurs *Regularized Discriminant Analysis* (RDA). Friedman propose de faire dépendre l'estimation des matrices de covariance des groupes de deux paramètres de régularisation, λ et γ, et ce de la façon suivante :

$$\hat{\Sigma}_i(\lambda,\gamma) = (1-\gamma)\hat{\Sigma}_i(\lambda) + \gamma \left(\frac{tr(\hat{\Sigma}_i(\lambda))}{p} \right) I_p,$$

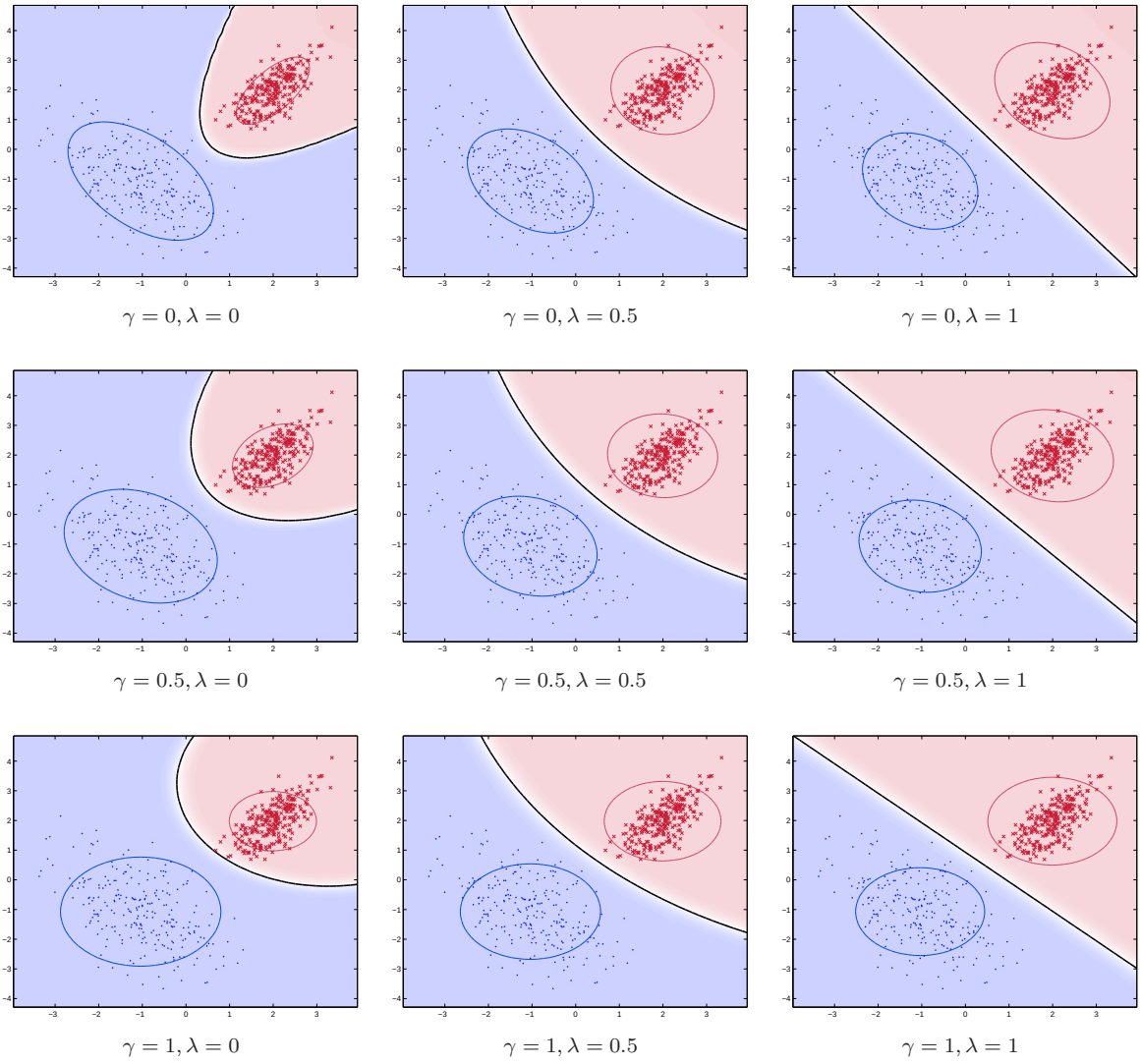


FIG. 2.17 – Analyse discriminante régularisée (RDA) : le paramètre λ permet de faire varier le classifieur entre QDA et LDA tandis que le paramètre γ contrôle l'estimation des valeurs propres des matrices de covariance.

où :

$$\hat{\Sigma}_i(\lambda) = \frac{(1-\lambda)(n_i-1)\hat{\Sigma}_i + \lambda(n-k)\hat{\Sigma}}{(1-\lambda)(n_i-1) + \lambda(n-k)}.$$

Le paramètre de complexité $\lambda \in [0,1]$ contrôle la contribution des estimateurs $\hat{\Sigma}_i$ et $\hat{\Sigma}$, qui sont donnés respectivement par les équations (2.5) et (2.6). Ainsi, l'Analyse Discriminante Régularisée engendre une règle de décision qui « varie » entre l'Analyse Discriminante Linéaire et l'Analyse Discriminante Quadratique. D'autre part, le paramètre $\gamma \in [0,1]$ contrôle l'estimation des valeurs propres des matrices de covariance. En effet, si $\gamma = 0$ alors les valeurs propres de Σ_i peuvent être différentes les unes des autres tandis que si $\gamma = 1$ alors les valeurs propres sont supposées être toutes égales. Dans ce dernier cas, cela revient à supposer que les densités des classes sont de forme sphérique. La figure 2.17 montre l'influence des deux paramètres de régularisation de RDA. On observe que, fixant γ à 0 et faisant varier λ, on obtient une méthode qui va de QDA ($\lambda = 0$) à LDA ($\lambda = 1$). A l'inverse, en fixant λ, la variation de γ conduit à des estimations plus ou moins biaisées des valeurs propres des matrices de covariances des classes. En particulier, si $\lambda = 0$, la régularisation grâce au paramètre γ est du même type que la régularisation simple présentée au paragraphe précédent. Enfin, pour $\lambda = 1$ et $\gamma = 1$, on obtient la méthode simpliste qui consiste à affecter tout nouveau point à la classe dont il est le plus proche de la moyenne au sens de la distance usuelle. Cette méthode donne généralement des résultats un peu meilleurs que QDA et LDA quand la taille de l'échantillon d'apprentissage est petite comme le fait remarquer Celeux dans [42, chap. 7]. En revanche, nous avons remarqué en expérimentant RDA que sa paramétrisation était rendue difficile par le peu de sensibilité des résultats de classification par rapport aux paramètres de régularisation. Une application de cette méthode à la reconnaissance de visage est proposée dans [73].

Régularisation de LDA par augmentation de la matrice intra-classe

Krzanowski *et al.* [52] ont proposé différentes techniques pour pallier les problèmes posés par le mauvais conditionnement des matrices de covariance dans le cadre de la discrimination de données spectroscopiques. Ce travail est tout à fait en lien avec le sujet qui nous intéresse ici puisque les données spectroscopiques sont des données de grande dimension et que la méthode de discrimination considérée est LDA. Les auteurs partent de l'hypothèse que la matrice de covariance intra-classe empirique W est singulière et que son rang est $d < p$. Leur idée est de construire un nouvelle matrice $\tilde{W}$ de rang p , et donc non-singulière, qui soit une bonne approximation de W au sens de la préservation de l'information originale. C'est l'idée inverse de l'ACP qui au contraire cherche le sous-espace de dimension d qui permet la meilleure approximation des données de dimension p. Pour construire cette nouvelle matrice, il nous faut tout d'abord considérer la décomposition spectrale de W :

$$W = LDL^t,$$

où D est la matrice diagonale composée de valeurs propres ordonnées de W, $\lambda_1 \geq ... \geq \lambda_d > \lambda_{d+1} = ... = \lambda_p = 0$ et L est la matrice orthonormale contenant les vecteurs propres correspondants. Si on

appelle D_1 la matrice diagonale composée des d premières colonnes de D, on peut alors écrire :

$$W = (L_1 L_2) \begin{pmatrix} D_1 & 0 \\ 0 & 0 \end{pmatrix} (L_1 L_2)^t,$$

où L_1 contient également les d premières colonnes de L et où L_2 contient les $(p-d)$ dernières colonnes. Les auteurs proposent alors comme matrice $\tilde{W}$ la matrice suivante :

$$\tilde{W} = \frac{1}{c}(L_1 L_2) \begin{pmatrix} D_1 + \alpha I & 0 \\ 0 & (\alpha+\beta) I \end{pmatrix} (L_1 L_2)^t,$$

où α et β satisfont les conditions suivantes :

$$\alpha \geq 0, \quad \beta < \lambda_d, \quad \alpha + \beta > 0.$$

Enfin, c est une constante de normalisation telle que $\mathrm{tr}(\tilde{W}) = \mathrm{tr}(W)$. Les paramètres de régularisation α et β, par analogie avec RDA, sont à estimer sur le jeu d'apprentissage et Krzanowski *et al.* recommandent de les estimer par validation croisée. Les expérimentations sur données simulées que nous avons mené ont montré que cette méthode devait être réservée à des cas où les autres méthodes de régularisation échouent tant la paramétrisation est difficile et la différence avec LDA est petite dans le cas standard.

Analyse discriminante pénalisée (PDA)

L'Analyse Discriminante Pénalisée (PDA) [45] a été proposée pour traiter des données dont les variables sont très corrélées ou dont la taille est petite devant le nombre de variables. PDA est au même titre que RDA une méthode de régularisation de LDA. La pénalisation introduite dans PDA est du même type que la régularisation simple présentée précédemment, à la différence que PDA pénalise également les corrélations entre les prédicteurs. L'estimateur pénalisé de Σ utilisé dans PDA est :

$$\tilde{\Sigma} = \hat{\Sigma} + \sigma^2 \Omega,$$

où la matrice Ω, de taille $p \times p$, permet de pénaliser les corrélations entre les prédicteurs. Les auteurs recommandent d'utiliser une matrice « lisse », *i.e.* deux coefficients voisins doivent avoir une valeur proche. L'ensemble des paramètres de pénalisation, σ^2 et les coefficients de Ω, peuvent être appris sur le jeu d'apprentissage par validation croisée. Ils peuvent aussi traduire des *a priori* de l'expérimentateur. Hastie *et al.* proposent également de coupler cette pénalisation à la projection des données sur les axes discriminants de Fisher (voir paragraphe 2.4.2) avant d'appliquer la règle de décision. Les auteurs proposent également d'effectuer une transformation préalable des données par l'application d'un opérateur de type noyau puis d'appliquer PDA.

Modèle	Nombre de paramètres	Ordre asymptotique	Nb de prms pour $k = 4$ et $p = 100$
Full-GMM	$\rho + kp(p+1)/2$	$kp^2/2$	20603
Com-GMM	$\rho + p(p+1)/2$	$p^2/2$	5453
Diag-GMM	$\rho + kp$	$2kp$	803
Com-diag-GMM	$\rho + p$	p	503
Sphe-GMM	$\rho + k$	kp	407
Com-sphe-GMM	$\rho + 1$	p	404

TAB. 2.1 – Propriétés des modèles gaussiens : $\rho = kp + k - 1$ est le nombre de paramètres nécessaires à l'estimation des moyennes et proportions. Pour le calcul des ordres asymptotiques, nous supposons que $k \ll p$.

2.4.4 Modèles parcimonieux

La seconde solution permettant de pallier le problème du fléau de la dimension dans le cadre de la modélisation par modèles de mélange gaussien est l'utilisation de modèles parcimonieux, *i.e.* des modèles qui requièrent un nombre « raisonnable » de paramètres à estimer. Rappelons tout d'abord que le modèle gaussien général (matrices de covariance Σ_i pleines) de QDA, noté full-GMM dans la suite, requiert l'estimation de 20603 paramètres pour des données comportant 4 classes dans un espace de dimension 100. Dans ce même cas de figure, LDA – qui est en fait déjà une méthode de régularisation de QDA – requiert quant à elle l'estimation de 5453 paramètres. Le modèle gaussien parcimonieux utilisé dans LDA sera noté dans la suite com-GMM. Le tableau 2.1 donne notamment le détail du nombre de paramètres pour ces deux modèles gaussiens. Dans ce paragraphe, nous présenterons les modèles parcimonieux dans le cadre supervisé (analyse discriminante) par souci de simplicité. Une vue d'ensemble de l'usage des modèles parcimonieux en classification automatique est disponible dans [36].

Modèles de mélange gaussien diagonaux et sphériques

Les modèles parcimonieux simples que nous allons détailler dans ce paragraphe sont obtenus en faisant des hypothèses supplémentaires par rapport aux hypothèses classiques du modèle de mélange gaussien présenté au paragraphe 2.1.3.

Modèle de mélange gaussien diagonale Si l'on fait l'hypothèse supplémentaire, par rapport au modèle de mélange gaussien classique full-GMM (matrices de covariance Σ_i pleines), que les matrices de covariance Σ_i sont diagonales, *i.e.* $\Sigma_i = \mathrm{diag}(\sigma_{i1}^2, ..., \sigma_{ip}^2)$, $i = 1, ..., k$, on obtient alors un nouveau modèle parcimonieux qui sera noté diag-GMM. Ce modèle ne requiert que l'estimation de 803 paramètres si l'on considère toujours des données comportant 4 classes dans un espace de dimension 100. Ce modèle fait en réalité l'hypothèse que les variables sont indépendantes (les termes de covariance

des matrices de variance-covariance sont tous nuls). Cette hypothèse peut toutefois s'avérer trop restrictive dans certains cas. En revanche, il peut être couplé avec succès à une ACP qui fournira des données dont les variables seront indépendantes sous hypothèse gaussienne. Dans le cadre supervisé et sous cette hypothèse, QDA donne naissance à une nouvelle méthode de discrimination que l'on peut appeler QDAd et dont la règle de décision revient à minimiser la fonction de coût :

$$K_i(x) = \sum_{\ell=1}^{p} \frac{(x^{(\ell)} - \mu_i^{(\ell)})^2}{\sigma_{i\ell}^2} + \sum_{\ell=1}^{p} \log(\sigma_{i\ell}^2) - 2\log(\pi_i) + C^{te}.$$

Cette hypothèse d'indépendance des variables peut également être combinée à l'hypothèse d'égalité des matrices de covariances (cas de LDA). Le modèle de mélange résultant, que l'on notera com-diag-GMM, ne nécessite alors que l'estimation de 503 paramètres dans le cas de l'exemple précédent. Lee *et al.* [53] ont montré que le modèle gaussien diagonal utilisé dans QDA et LDA fournit des résultats très satisfaisants en comparaison d'un grand nombre d'autres classifieurs. Toutefois, il est important de noter que la dimension des données de puces ADN utilisées pour cette évaluation avait été réduite par une procédure spécifique.

Modèle de mélange gaussien sphérique Il est également possible de faire l'hypothèse que les densités des classes sont de forme sphérique en supposant que $\Sigma_i = \sigma_i^2 I_p$, $i = 1, ..., k$, où I_p est la matrice identité de taille $p \times p$. Ce modèle, que l'on notera sphe-GMM, ne requiert alors que l'estimation de 407 paramètres si l'on considère toujours le cas de données comportant 4 classes en dimension 100. Dans le cadre supervisé et sous cette hypothèse, QDA donne naissance à une nouvelle méthode de discrimination que l'on peut appeler QDAs et dont la règle de décision revient à minimiser la fonction de coût :

$$K_i(x) = \frac{1}{\sigma_i^2}\|x - \mu_i\|^2 + p\log(\sigma_i^2) - 2\log(\pi_i) + C^{te}.$$

Il est à nouveau possible d'utiliser ce modèle parcimonieux en combinaison avec l'hypothèse d'égalité des matrices de covariances. Cela engendre alors le modèle le plus parcimonieux possible, noté com-sphe-GMM, qui ne requiert que l'estimation de 404 paramètres dans le cas de l'exemple précédent. Enfin, si l'on fait l'hypothèse supplémentaire que les proportions sont égales $\pi_i = 1/k$, alors on obtient dans le cadre supervisé la règle géométrique de l'analyse discriminante linéaire qui consiste à minimiser la fonction de coût :

$$K_i(x) = \|x - \mu_i\|^2 + C^{te}, \tag{2.9}$$

qui affecte le point x à la classe dont il est le plus proche de la moyenne. Ce classifieur a été baptisé *nearest-means classifier* par Friedman [38]. Toutefois, cette règle simple conduit à des erreurs d'affectation quand la dispersion des classes est trop différente.

Modèles de mélange gaussien à décomposition spectrale

Nous avons vu au paragraphe précédent que le modèle de mélange gaussien était à la tête d'une famille de modèles plus ou moins parcimonieux. La décomposition spectrale des matrices de covariance [5, 21] permet de paramétrer de manière unique les matrices de covariance des modèles parcimonieux précédents :

$$\Sigma_i = \lambda_i D_i A_i D_i^t,$$

où D_i est la matrice des vecteurs propres de Σ_i, A_i est une matrice diagonale contenant les valeurs propres normalisées et ordonnées de Σ_i et $\lambda_i = \det(\Sigma_i)^{1/p}$. Les quantités λ_i, D_i et A_i contrôlent respectivement le volume, l'orientation et la forme de la distribution de la classe C_i. En permettant ou non à ces trois paramètres de varier, entre et à l'intérieur des classes, les auteurs mettent en évidence 14 modèles particuliers qui vont du modèle le moins parcimonieux, le modèle gaussien classique, au modèle le plus parcimonieux. Cette paramétrisation permet de proposer de nouvelles modélisations intermédiaires qui n'étaient pas connues auparavant. Cette paramétrisation peut être naturellement utilisée en classification supervisée et non supervisée. Dans le cas supervisé, cette paramétrisation donne naissance à l'EDDA (*Eigenvalue Decomposition Discriminant Analysis*) [7] qui permet notamment d'éviter le recours aux paramètres de régularisation de la RDA. L'EDDA choisit, par validation croisée, parmi ces modèles celui qui possède le plus petit taux d'erreur. Dans le cas non supervisé, le choix du modèle se fait grâce au critère BIC [81]. Les estimateurs des paramètres de certains modèles n'ont pas une forme explicite et sont alors estimés par des méthodes itératives.

2.4.5 Classification dans des sous-espaces

Une récente extension de la classification traditionnelle consiste à rechercher les sous-espaces dans lesquels vivent les données de chacun des groupes. En effet, le phénomène de l'espace vide nous a permis de nous convaincre qu'un espace de dimension p est quasiment vide et que les données se trouvent dans des sous-espaces de dimension inférieure. De plus, il est assez naturel de penser que les données provenant de classes différentes vivent dans des sous-espaces différents. Etrangement, cette approche a plutôt été développée dans le cadre de la classification non-supervisée (*clustering*). Par conséquent, ce paragraphe sera principalement présenté dans le cadre du *clustering*. Les méthodes de *subspace clustering* peuvent être divisées en deux grandes familles de méthodes : d'une part, les méthodes heuristiques qui recherchent les dimensions permettant d'obtenir le meilleur *clustering* et, d'autre part, les méthodes basées sur des modèles de mélange qui modélisent le fait que les données vivent dans des sous-espaces.

Méthodes heuristiques

De nombreuses méthodes de *subspace clustering* utilisent des techniques heuristiques de recherche pour identifier les sous-espaces des classes. Parmi ces méthodes, on peut distinguer deux types d'algorithmes de recherche des sous-espaces : les méthodes de recherche dites « *bottom-up* » et

celles dites « *top-down* ». Les méthodes dites « *bottom-up* » utilisent des histogrammes pour sélectionner les dimensions permettant de séparer efficacement les groupes. CLIQUE [3] fut l'un des premiers algorithmes « *bottom-up* » proposés pour rechercher des groupes dans des sous-espaces de l'espace original. D'autre part, les techniques de recherche de type « *top-down* » sont des méthodes itératives qui, partant de l'espace entier comme solution initiale, pondèrent à chaque itération les dimensions ne semblant pas contenir de groupe. La première méthode de ce type fut Proclus [2] qui évalue à chaque itération la qualité du *clustering* en calculant la distance moyenne entre les centres des groupes. Une vue d'ensemble des méthodes heuristiques est disponible dans [68].

Méthodes basées sur des mélanges de *factor analyzers*

Dans le même temps, de nombreuses méthodes basées sur les mélanges de *factor analyzers* ont vu le jour. Ces méthodes génératives proposent de se placer dans les espaces propres des classes afin de prendre en compte le fait que les données vivent dans des sous-espaces de faible dimension. Pour cela, elles se basent sur le modèle de l'analyse factorielle [44]. Une vue d'ensemble de ces méthodes est notamment disponible dans [75]. Une évaluation d'une partie des méthodes présentées ci-dessous est faite dans [63].

Le modèle de l'analyse factorielle Ce modèle suppose que les observations sont des réalisations indépendantes d'un vecteur aléatoire X, de dimension p, qui est la combinaison de la transformation d'une variable latente Y, non observée, de dimension $d < p$ et d'un terme d'erreur ϵ :

$$X = HY + \mu + \epsilon.$$

La variable latente Y_j est appelée le « facteur » et est supposée suivre une loi normale $\mathcal{N}(0, I_d)$, où I_d est la matrice identité de taille d. La matrice H, de taille $p \times d$, est une matrice de transformation contenant les poids des facteurs. L'erreur ϵ est quant à elle supposée suivre une loi normale $\mathcal{N}(0, D)$ où la matrice $p \times p$ de covariance D est diagonale :

$$D = \text{diag}(\sigma_1^2, ..., \sigma_p^2).$$

Sous ces hypothèses, le vecteur aléatoire X suit une loi normale $\mathcal{N}(\mu, \Sigma)$, où la matrice de covariance Σ à la forme suivante :

$$\Sigma = HH^t + D.$$

Ainsi, l'estimation de la matrice Σ ne requiert que l'estimation de $pd - d(d-1)/2 + p$ paramètres.

Mélange de *factor analyzers* Rubin *et al.* [78], puis Ghahramani *et al.* [40] et McLachlan *et al.* [59], ont proposé d'utiliser ce modèle de variables latentes pour modéliser chacun des groupes et d'utiliser l'algorithme EM pour estimer les paramètres du modèle. La densité $f(., \theta_i)$, $i = 1, ..., k$, de chacun

des groupes est alors supposée être celle d'une loi normale $\mathcal{N}(\mu_i, \Sigma_i)$, où la matrice de covariance Σ_i a la forme suivante :

$$\Sigma_i = H_i H_i^t + D_i.$$

Ainsi, le nombre de paramètres à estimer à chaque étape de l'algorithme EM est contrôlé par la dimension d de l'espace latent. McLachlan *et al.* [59] ont notamment appliqué ce modèle de mélange pour la classification automatique de puces ADN.

Mélange de *principal component analyzers* Si l'on fait l'hypothèse supplémentaire que $D = \sigma^2 I_p$, où I_p est la matrice identité de dimension p, alors l'analyse factorielle se réduit à l'analyse en composantes principales (*cf.* paragraphe 2.4.2). Cela a été en particulier montré par Tipping et Bishop [87]. L'utilisation du mélange de *principal component analyzers* pour la classification non-supervisée a ensuite été proposée simultanément par Roweis [77] et Tipping *et al.* [88]. Ce mélange suppose donc que la densité $f(., \theta_i)$, $i = 1, ..., k$, de chacun des groupes est celle d'une loi normale $\mathcal{N}(\mu_i, \Sigma_i)$, où la matrice de covariance Σ_i à la forme suivante :

$$\Sigma_i = H_i H_i^t + \sigma_i^2 I_p.$$

Tipping *et al.* ont notamment appliqué cette technique à la compression d'images par regroupement des zones d'images similaires. Ces derniers ont également utilisé dans [11] cette modélisation en combinaison avec une approche hiérarchique pour la visualisation de données de grande dimension. La modélisation par mélange de *principal component analyzers* a également été considérée dans [13] pour le cas de la classification automatique de données de dissimilarité et par Moghadam *et al.* [64] dans le cadre supervisé pour la reconnaissance de visages.

Le sous-espace de discrimination de Flury *et al.*

Dans le cadre supervisé, Flury *et al*. [34] ont proposé une méthode de discrimination sous la contrainte que toutes les différences entre les populations ont lieu dans un sous-espace de dimension $d < p$. Cette méthode est basée sur le modèle DSM (*Discrimination Subspace Model*) et est un classifieur intermédiaire entre la discrimination quadratique (QDA) et la discrimination linéaire (LDA). Le modèle DSM combine, comme le mélange de *factor analyzers*, les idées de réduction de dimension et de contraintes sur le modèle. Pour simplifier, Flury *et al*. ont considéré uniquement le cas de deux classes. Dans ce cas, le modèle DSM suppose que X est un vecteur de dimension p distribué selon une loi normale $\mathcal{N}(\mu_i, \Sigma_i)$ pour la $i$ème population, $i = 1, 2$, et qu'il existe une matrice $p \times p$ non singulière $Q = [\tilde{Q} : \bar{Q}]$, où $\tilde{Q}$ et $\bar{Q}$ sont respectivement de taille $p \times d$ et $p \times (p - d)$, et telle que :

(i) $Q^t \Sigma_i Q$ soit diagonale pour $i = 1, 2$,

(ii) $\bar{Q}^t \Sigma_1 \bar{Q} = \bar{Q}^t \Sigma_2 \bar{Q}$,

(iii) $\bar{Q}^t \mu_1 = \bar{Q}^t \mu_2$.

Chapitre 3

Modèles de mélange gaussien pour les données de grande dimension

Dans ce chapitre, nous mènerons tout d'abord au paragraphe 3.1 une analyse critique des solutions proposées jusqu'alors pour résoudre le problème de la classification des données de grande dimension et, sur cette base, nous proposerons une approche qui exploite les bienfaits de la dimension. Au cours du paragraphe 3.2, nous proposerons une re-paramétrisation du modèle de mélange gaussien permettant de combiner les idées de réduction de dimension et de modèles parcimonieux. Cette re-paramétrisation donnera naissance à un modèle de mélange gaussien qui sera baptisé $[a_{ij}b_iQ_id_i]$, du nom de ses paramètres, et dont la complexité sera contrôlée par la dimension intrinsèque des données. En suivant l'approche de Celeux et Govaert [21], nous verrons au paragraphe 3.3 qu'en contraignant les paramètres du modèle $[a_{ij}b_iQ_id_i]$, celui-ci générera une famille de modèles allant du modèle le plus général au modèle le plus parcimonieux. Enfin, au paragraphe 3.4, nous montrerons que notre modélisation permet une approche unifiée de la classification des données de grande dimension dans le cadre gaussien. Pour cela, nous étudierons les liens existants entre les modèles proposés dans ce chapitre et les modèles gaussiens classiques.

3.1 Motivation de notre approche

Nous avons vu au chapitre 2 que la majorité des méthodes d'apprentissage souffre du « fléau de la dimension » et voit leur performance décroître quand la dimension des observations augmente. Nous allons tout d'abord analyser les principales limites des solutions existantes aux problèmes posés par la grande dimension des données en classification. Cette analyse nous permettra ensuite d'élaborer une approche adaptée à la classification de telles données dans le cadre du modèle de mélange gaussien en exploitant les « bienfaits de la dimension ».

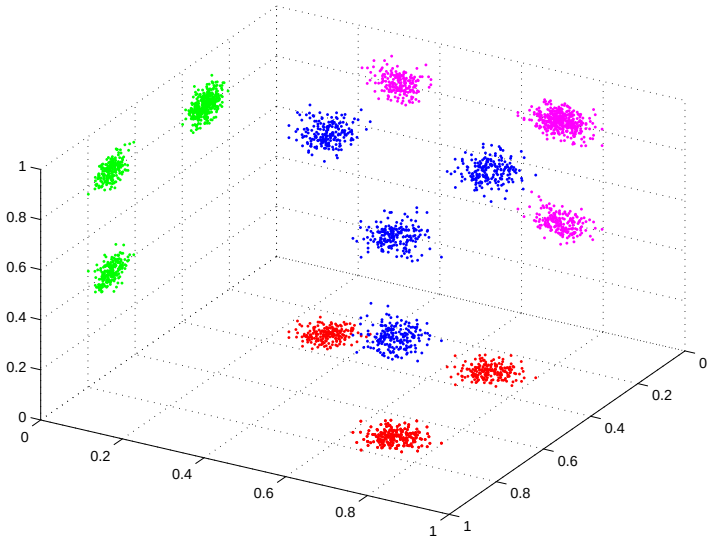


FIG. 3.1 – Un des « dangers » de la réduction de dimension dans le cadre de la classification.

3.1.1 Les limites des approches existantes

Les solutions proposées jusqu'alors pour pallier le problème de la classification des données de grande dimension sont de réduire la dimension des données de façon globale, de régulariser les estimations des matrices de covariance ou d'utiliser une modélisation des données qui soit parcimonieuse. Ces approches permettant le plus souvent d'améliorer les performances des méthodes de classification possèdent toutefois un certain nombre de limitations.

Les « dangers » de la réduction de dimension

La première des solutions existantes pour contrer le fléau de la dimension est de réduire la dimension des données. La réduction de dimension est généralement réalisée préalablement à la classification et par une approche globale qui ne tient pas compte de l'objectif de classification. Cette technique permet aux méthodes de classification de fonctionner correctement mais au prix d'une perte d'information qui aurait pu être discriminante. De plus, si l'on se place dans le cadre de la classification, cette approche est paradoxale puisque les p variables ne sont certes pas toutes nécessaires pour décrire chacune des classes mais l'ensemble des variables est le plus souvent nécessaire pour discriminer les classes les unes par rapport aux autres. La figure 3.1 illustre cela en présentant une situation caricaturale où les 4 classes ne peuvent être discriminées qu'en conservant l'ensemble des dimensions initiales. Les données originales, représentées en bleu, vivent dans un espace à 3 dimensions. Les projections des données originales sur les 3 sous-espaces de dimension 2 sont respectivement représentées en rouge, vert et violet. On observe qu'aucune des trois projections ne permet de séparer les 4 groupes et que leur discrimination n'est possible que dans l'espace d'origine. En outre, le phénomène de « l'espace vide » nous a permis de nous rendre compte que les espaces de grande dimension sont

quasiment vides et que les données vivent dans des sous-espaces de l'espace original. En étendant cette réflexion au cadre de la classification, il est assez naturel de conjecturer que les données de classes différentes vivent dans des sous-espaces différents. Il est alors clair qu'une approche globale de réduction de dimension ne peut pas prendre en compte cette spécificité des données de grande dimension. Une approche qui réduirait la dimension pour chaque classe indépendamment aurait donc déjà plus de sens qu'une approche globale. D'autre part, nous avons vu au chapitre 2 qu'il est plus aisé de discriminer, avec un classifieur adapté, des classes dans un espace de grande dimension que dans un espace de faible dimension. Les méthodes de discrimination à noyaux SVM, présentées également au chapitre 2, font cette hypothèse puisqu'elles augmentent artificiellement la dimension de l'espace de départ afin de faciliter la discrimination. Pour toutes ces raisons, nous pensons qu'il est préférable de ne pas réduire la dimension des données au préalable mais d'utiliser un modèle qui prennent en compte le fait que les données de classes vivent dans sous-espaces de dimensions intrinsèques faibles.

Les restrictions des méthodes de régularisation

Nous avons vu au chapitre précédent que la seconde solution pour que les méthodes de classification basées sur le modèle de mélange gaussien ne soient pas sujettes à la dimension des données est de régulariser *a posteriori* les estimations des matrices de covariance. Cependant, cette approche introduit un biais dans l'estimation des matrices de covariance qui améliore certes le conditionnement de la matrice estimée mais qui influe également sur la règle de décision du classifieur. De plus, si la dimension des données est très grande devant le nombre d'observations, alors la régularisation devra être importante et, par conséquent, la perturbation induite sur la règle de décision ne sera pas négligeable. D'autre part, cette approche ne fait que contourner le problème puisque le mauvais conditionnement des matrices de covariance des classes est également lié au fait que les classes vivent dans des sous-espaces de dimension intrinsèque faible. En effet, les données de chaque classe vivant dans un sous-espace de faible dimension, une grande partie des variables ne sert pas à décrire les données et les termes de la matrice de covariance associés à ces dimensions seront, dans le pire des cas, nuls ou mesureront, dans le meilleur des cas, la variance due au bruit. On comprend donc la nécessité de modéliser les données de chaque classe dans leur sous-espace spécifique.

Les limites des modèles parcimonieux

Le dernier « remède » possible au « fléau de la dimension » est l'utilisation de modèles parcimonieux. Les modèles parcimonieux, rappelons-le, font des hypothèses supplémentaires sur le modèle général afin de limiter le nombre de paramètres à estimer. Il est vrai que le modèle gaussien général requiert pour chaque classe l'estimation d'un nombre de paramètres qui croît avec le carré de la dimension. On comprend alors vite la nécessité de recourir aux modèles parcimonieux pour pouvoir traiter des données de grande dimension, en particulier quand le nombre d'observations devient petit devant la dimension. Cependant, les modèles parcimonieux les plus utilisés sont soit encore trop complexes (com-GMM), soit trop parcimonieux (diag-GMM et sphe-GMM). En effet, le nombre de paramètres à

estimer dans le cas du modèle com-GMM est encore de l'ordre du carré de la dimension. D'autre part, l'hypothèse d'indépendance conditionnelle des variables des modèles diag-GMM et sphe-GMM est le plus souvent trop restrictive pour modéliser correctement des données de grande dimension. Parmi les 14 modèles plus ou moins parcimonieux que propose leur re-paramétrisation, Celeux et Govaert recommandent d'ailleurs, dans [21], l'utilisation de ces deux modèles. Ils recommandent également l'utilisation d'un modèle moins parcimonieux, basé sur le modèle com-GMM, qui suppose que chacune des matrices de covariance des classes est proportionnelle à une même matrice B. L'utilisation des modèles parcimonieux dans le cadre de la classification de données de grande dimension peut s'avérer fructueuse car ils permettent aux méthodes génératives de fonctionner correctement. Cependant, l'utilisation des modèles parcimonieux n'est pas adaptée à la classification de données de grande dimension car ces modèles ne peuvent le plus souvent pas modéliser efficacement la structure complexe de données de grande dimension et nécessitent encore l'inversion des matrices de covariance estimées.

3.1.2 Notre approche : combiner réduction de dimension, modèles parcimonieux et régularisation

Nous avons vu précédemment que la baisse de performance des méthodes de classification dans les espaces de grande dimension est principalement due au fait que les modèles requièrent l'estimation d'un nombre trop grand de paramètres devant le nombre d'observations disponibles. De plus, les approches proposées pour s'affranchir de ce problème ne peuvent pas le résoudre efficacement car les données de grande dimension vivent dans des sous-espaces dont les dimensions intrinsèques sont inférieures à la dimension de l'espace original. Cependant, le phénomène du « fléau de la dimension » possède un antagoniste que Donoho dans [27] a baptisé les « bienfaits de la dimension ». Parmi ces aspects positifs des espaces de grande dimension, nous avons vu précédemment que la discrimination est plus aisée dans un espace de grande dimension que dans un espace de dimension plus faible. Nous allons par conséquent proposer une paramétrisation du modèle de mélange gaussien qui permette d'exploiter cette caractéristique des espaces de grande dimension. Notre idée est d'utiliser le fait que les données de grande dimension vivent dans des sous-espaces dont les dimensions intrinsèques sont faibles pour limiter le nombre de paramètres du modèle et régulariser l'estimation des matrices de covariance des classes. Pour mettre en œuvre cette idée, nous allons proposer une re-paramétrisation du modèle de mélange gaussien qui prenne en compte le fait que les données de chacune des classes vivent dans des sous-espaces différents dont les dimensions intrinsèques peuvent être différentes. Cela permettra notamment de contrôler la complexité du modèle par les dimensions des sous-espaces des classes et non plus par la dimension de l'espace d'origine. En outre, les matrices de covariance ne dépendant plus directement de la dimension totale de l'espace, leur estimation sera régularisée et ne conduira donc plus à des matrices mal conditionnées ou singulières.

3.2 Le modèle de mélange gaussien $[a_{ij}b_iQ_id_i]$

Nous allons proposer dans ce paragraphe une re-paramétrisation du modèle de mélange gaussien qui permette de combiner les idées de réduction de dimension, de contraintes sur le modèle et de régularisation. Nous étudierons ensuite les caractéristiques et la complexité du modèle engendré par cette re-paramétrisation. Nous expliciterons et interpréterons également la fonction de coût sur laquelle est basée la règle de décision associée à ce modèle.

3.2.1 Re-paramétrisation du modèle de mélange gaussien

Nous nous plaçons dans le cadre du modèle de mélange gaussien, *i.e.* nous supposons que les densités de probabilité des classes $f(x, \theta_i)$, $\forall i = 1, ..., k$, sont celles de lois normales $\mathcal{N}(\mu_i, \Sigma_i)$ de moyennes μ_i et de matrices de variance Σ_i :

$$f(x) = \sum_{i=1}^{k} \pi_i f(x, \theta_i).$$

En suivant la re-paramétrisation de Banfield et Raftery [5], basée sur la décomposition spectrale de Σ_i, on peut écrire :

$$\Sigma_i = Q_i \Delta_i Q_i^t,$$

où Q_i est la matrice orthogonale de taille $p \times p$ contenant les vecteurs propres de Σ_i et Δ_i est la matrice de covariance de la classe C_i dans son espace propre. La matrice Δ_i est alors une matrice diagonale contenant les valeurs propres de Σ_i.

Re-paramétrisation de Δ_i et définitions

Afin de modéliser le fait que les données de chacune des classes vivent dans des sous-espaces de dimensions inférieures à la dimension de l'espace, nous proposons d'écrire la matrice diagonale Δ_i sous la forme suivante :

$$\Delta_i = \begin{pmatrix} \boxed{\begin{matrix} a_{i1} & & 0 \\ & \ddots & \\ 0 & & a_{id_i} \end{matrix}} & & \mathbf{0} \\ & \boxed{\begin{matrix} b_i & & 0 \\ & \ddots & \\ & & \ddots \\ 0 & & b_i \end{matrix}} & \\ \mathbf{0} & & \end{pmatrix} \begin{matrix} \left.\vphantom{\begin{matrix} a \\ a \\ a \end{matrix}}\right\} d_i \\ \left.\vphantom{\begin{matrix} a \\ a \\ a \end{matrix}}\right\} (p - d_i) \end{matrix}$$

où $a_{ij} \geq b_i$, $j = 1, ..., d_i$, et $d_i < p$ pour tout $i = 1, ..., k$. Nous supposons donc implicitement que les d_i plus grandes valeurs propres de chaque classe C_i, $i = 1, ..., k$, sont distinctes et que les $(p - d_i)$ plus

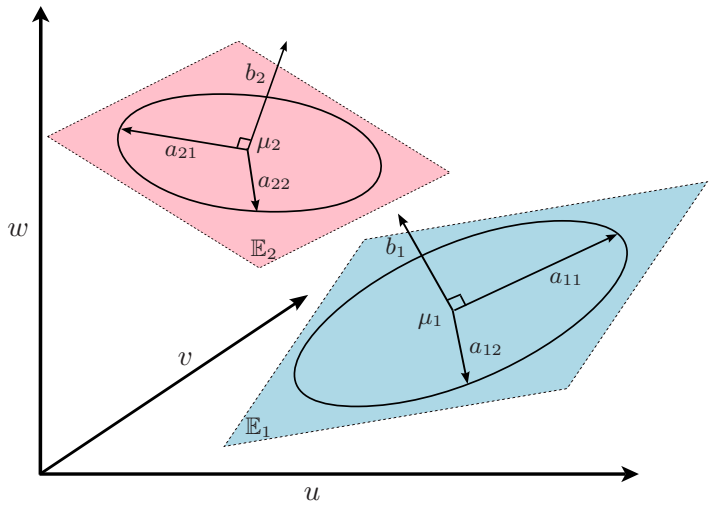


FIG. 3.2 – Les paramètres du modèle $[a_{ij}b_iQ_id_i]$ pour le cas de deux classes.

petites valeurs propres sont égales. Remarquons qu'il est toujours possible de faire cette hypothèse car dans le cas où toutes les valeurs propres de Δ_i seraient distinctes, nous pouvons choisir $d_i = (p-1)$. Nous allons également définir les sous-espaces $\mathbb{E}_i$ et $\mathbb{E}_i^{\perp}$ de la classe C_i respectivement associés aux valeurs propres $a_{i1}, ..., a_{id_i}$et b_i.

Définition 3.2.1. Le sous-espace affine $\mathbb{E}_i$ est le sous-espace engendré par les d_i vecteurs propres associés aux valeurs propres $a_{i1}, ..., a_{id_i}$ et tel que $\mu_i \in \mathbb{E}_i$. Le sous-espace $\mathbb{E}_i^{\perp}$ est le sous-espace supplémentaire de $\mathbb{E}_i$ dans $\mathbb{R}^p$, *i.e.* $\mathbb{E}_i \oplus \mathbb{E}_i^{\perp} = \mathbb{R}^p$, et tel que $\mu_i \in \mathbb{E}_i^{\perp}$.

On définit en outre les opérateurs P_i et $P_i^{\perp}$ de projection sur les sous-espaces $\mathbb{E}_i$ et $\mathbb{E}_i^{\perp}$ respectivement.

Définition 3.2.2. L'opérateur P_i de projection sur le sous-espace $\mathbb{E}_i$ est défini par :

$$\forall x \in \mathbb{R}^p,\ P_i(x) = \tilde{Q}_i\tilde{Q}_i^t(x - \mu_i) + \mu_i,$$

où $\tilde{Q}_i$ est la matrice $p \times p$ contenant les d_i premières colonnes de Q_i complétée par des zéros. De même, l'opérateur $P_i^{\perp}$ de projection sur le sous-espace $\mathbb{E}_i^{\perp}$ est défini par :

$$P_i^{\perp}(x) = \bar{Q}_i\bar{Q}_i^t(x - \mu_i) + \mu_i,$$

où $\bar{Q}_i = Q_i - \tilde{Q}_i$ est la matrice $p \times p$ contenant les $(p - d_i)$ dernières colonnes de Q_i complétée par des zéros.

La figure 3.2 illustre la paramétrisation proposée ci-dessus et permet de se représenter les sous-espaces $\mathbb{E}_i$, $i = 1, ..., k$, des classes.

Le modèle $[a_{ij}b_iQ_id_i]$ **et ses paramètres**

En suivant le système de notation introduit par Celeux *et al.* dans [21], nous proposons d'utiliser la notation $[a_{ij}b_iQ_id_i]$ pour faire référence au modèle de mélange gaussien associé à la paramétrisation que nous venons de présenter. Cette paramétrisation permet de contrôler les caractéristiques de la $i$ème composante du mélange gaussien grâce à quatre types de paramètres : le vecteur $(a_{i1}, ..., a_{id_i})$, le scalaire b_i, la matrice Q_i et la dimension d_i. Les paramètres $a_{i1}, ..., a_{id_i}$ et b_i, contenus dans la matrice diagonale Δ_i, contrôlent la forme de la classe C_i. Plus particulièrement, les d_i valeurs $a_{i1}, ..., a_{id_i}$ paramètrent la forme de la densité dans le sous-espace $\mathbb{E}_i$ où vivent les données de la classe. Ces d_i paramètres représentent donc la dispersion réelle des données de la $i$ème classe. Le paramètre b_i modélise quant à lui la variance en dehors du sous-espace $\mathbb{E}_i$ qui est par conséquent supposée être isotropique. Ce paramètre représente donc la variance qui n'est pas due aux données de la classe et qui pourrait être due au bruit. La matrice orthogonale Q_i contrôle quant à elle l'orientation de la classe C_i par rapport au système des axes originaux. En particulier, les d_i premières colonnes de la matrice Q_i engendrent le sous-espace $\mathbb{E}_i$ où les données de la classe C_i sont sensées vivre. Enfin, le paramètre d_i, qui représente la dimension intrinsèque du sous-espace de la $i$ème classe, joue un rôle clé dans la paramétrisation que nous avons présentée. Nous verrons en effet au paragraphe 3.2.3 que c'est l'ensemble des paramètres d_i qui contrôle la complexité du modèle $[a_{ij}b_iQ_id_i]$.

3.2.2 Fonction de coût K_i associée au modèle $[a_{ij}b_iQ_id_i]$

Nous avons vu au chapitre 2 que la règle de décision du MAP est entièrement déterminée dans le cadre du modèle de mélange gaussien par la fonction de coût $K_i(x) = -2\log(\pi_i f(x, \theta_i))$. Nous allons donc donner dans ce paragraphe l'expression de la fonction de coût K_i associée au modèle $[a_{ij}b_iQ_id_i]$. Nous interpréterons ensuite d'un point de vue géométrique l'effet de cette fonction de coût sur la règle de décision du modèle $[a_{ij}b_iQ_id_i]$.

Expression de la fonction de coût K_i **pour le modèle** $[a_{ij}b_iQ_id_i]$

La proposition suivante donne l'expression de la fonction de coût K_i en fonction des paramètres du modèle $[a_{ij}b_iQ_id_i]$.

Proposition 3.2.1. *La fonction de coût* K_i *associée au modèle* $[a_{ij}b_iQ_id_i]$ *a la forme suivante :*

$$K_i(x) = \|\mu_i - P_i(x)\|^2_{\mathcal{A}_i} + \frac{1}{b_i}\|x - P_i(x)\|^2 + \sum_{j=1}^{d_i}\log(a_{ij}) + (p - d_i)\log(b_i) - 2\log(\pi_i) + C^{te},$$

où $\|.\|_{\mathcal{A}_i}$ *est une norme sur* $\mathbb{E}_i$ *telle que* $\|x\|^2_{\mathcal{A}_i} = x^t\mathcal{A}_ix$ *avec* $\mathcal{A}_i = \tilde{Q}_i\Delta_i^{-1}\tilde{Q}_i^t$ *et où* $C^{te} = p\log(2\pi)$.

Démonstration. En écrivant la densité $f(x, \theta_i)$ en fonction de la matrice Δ_i, on obtient :

$$-2\log(f(x, \theta_i)) = (x - \mu_i)^t(Q_i\Delta_iQ_i^t)^{-1}(x - \mu_i) + \log(\det\Delta_i) + p\log(2\pi).$$

Or, $Q_i^t Q_i = I_p$ et donc $Q_i^{-1} = Q_i^t$. Il est alors possible d'écrire :

$$-2\log(f(x,\theta_i)) = (x-\mu_i)^t Q_i \Delta_i^{-1} Q_i^t (x-\mu_i) + \log(\det \Delta_i) + p\log(2\pi).$$

D'autre part, on peut écrire $Q_i = \tilde{Q}_i + \bar{Q}_i$ où $\tilde{Q}_i$ est la matrice $p \times p$ contenant les d_i premières colonnes de Q_i et complétée par des zéros et où $\bar{Q}_i = Q_i - \tilde{Q}_i$. En remarquant que $\tilde{Q}_i \Delta_i^{-1} \bar{Q}_i{}^t = \bar{Q}_i \Delta_i^{-1} \tilde{Q}_i^t = O_p$ où O_p est la matrice nulle, on obtient :

$$Q_i \Delta_i^{-1} Q_i^t = \tilde{Q}_i \Delta_i^{-1} \tilde{Q}_i + \bar{Q}_i \Delta_i^{-1} \bar{Q}_i^t.$$

La quantité $-2\log(f(x,\theta_i))$ peut alors être écrite comme suit :

$$\begin{aligned} -2\log(f(x,\theta_i)) &= (x-\mu_i)^t \tilde{Q}_i \Delta_i^{-1} \tilde{Q}_i^t (x-\mu_i) + (x-\mu_i)^t \bar{Q}_i \Delta_i^{-1} \bar{Q}_i^t (x-\mu_i) \\ &\quad + \log(\det \Delta_i) + p\log(2\pi). \end{aligned}$$

Les relations $\tilde{Q}_i \left[\tilde{Q}_i^t \tilde{Q}_i\right] = \tilde{Q}_i$ et $\bar{Q}_i \left[\bar{Q}_i^t \bar{Q}_i\right] = \bar{Q}_i$ permettent de re-formuler $-2\log(f(x,\theta_i))$ de la façon suivante :

$$\begin{aligned} -2\log(f(x,\theta_i)) &= (x-\mu_i)^t \tilde{Q}_i \tilde{Q}_i^t \tilde{Q}_i \Delta_i^{-1} \tilde{Q}_i^t \tilde{Q}_i \tilde{Q}_i^t (x-\mu_i) \\ &\quad + (x-\mu_i)^t \bar{Q}_i \bar{Q}_i^t \bar{Q}_i \Delta_i^{-1} \bar{Q}_i^t \bar{Q}_i \bar{Q}_i^t (x-\mu_i) \\ &\quad + \log(\det \Delta_i) + p\log(2\pi). \end{aligned}$$

Cela peut également s'écrire :

$$\begin{aligned} -2\log(f(x,\theta_i)) &= \left[\tilde{Q}_i \tilde{Q}_i^t (x-\mu_i)\right]^t \tilde{Q}_i \Delta_i^{-1} \tilde{Q}_i^t \left[\tilde{Q}_i \tilde{Q}_i^t (x-\mu_i)\right] \\ &\quad + \left[\bar{Q}_i \bar{Q}_i^t (x-\mu_i)\right]^t \bar{Q}_i \Delta_i^{-1} \bar{Q}_i^t \left[\bar{Q}_i \bar{Q}_i^t (x-\mu_i)\right] \\ &\quad + \log(\det \Delta_i) + p\log(2\pi). \end{aligned}$$

On introduit à présent la notation $\mathcal{A}_i = \tilde{Q}_i \Delta_i^{-1} \tilde{Q}_i^t$ et l'on définit la norme $\|.\|_{\mathcal{A}_i}$ sur $\mathbb{E}_i$ telle que $\|x\|^2_{\mathcal{A}_i} = x^t \mathcal{A}_i x$. Ainsi :

$$\left[\tilde{Q}_i \tilde{Q}_i^t (x-\mu_i)\right]^t \tilde{Q}_i \Delta_i^{-1} \tilde{Q}_i^t \left[\tilde{Q}_i \tilde{Q}_i^t (x-\mu_i)\right] = \|\tilde{Q}_i \tilde{Q}_i^t (x-\mu_i)\|^2_{\mathcal{A}_i}.$$

D'autre part, on a :

$$\left[\bar{Q}_i \bar{Q}_i^t (x-\mu_i)\right]^t \bar{Q}_i \Delta_i^{-1} \bar{Q}_i{}^t \left[\bar{Q}_i \bar{Q}_i^t (x-\mu_i)\right] = \frac{1}{b_i} \|\bar{Q}_i \bar{Q}_i^t (x-\mu_i)\|^2,$$

et par conséquent :

$$-2\log(f(x,\theta_i)) = \|\tilde{Q}_i\tilde{Q}_i^t(x-\mu_i)\|^2_{\mathcal{A}_i} + \frac{1}{b_i}\|\bar{Q}_i\bar{Q}_i^t(x-\mu_i)\|^2 + \log(\det\Delta_i) + p\log(2\pi).$$

En utilisant les définitions des opérateurs de projection P_i and $P_i^{\perp}$ et en remarquant que $\|\mu_i - P_i^{\perp}(x)\|^2 = \|x - P_i(x)\|^2$ (voir la figure 3.3 pour s'en persuader), nous obtenons finalement :

$$-2\log(f(x,\theta_i)) = \|\mu_i - P_i(x)\|^2_{\mathcal{A}_i} + \frac{1}{b_i}\|x - P_i(x)\|^2 + \log(\det\Delta_i) + p\log(2\pi).$$

La relation $\log(\det\Delta_i) = \sum_{j=1}^{d_i}\log(a_{ij}) + (p-d_i)\log(b_i)$ permet d'obtenir l'expression finale de la fonction de coût associée au modèle $[a_{ij}b_iQ_id_i]$. □

Sur la base de l'expression de K_i dans le cadre du modèle $[a_{ij}b_iQ_id_i]$, nous allons faire deux remarques qui mettent en évidence les qualités de ce modèle et qui seront très utiles dans la suite de ce chapitre.

Remarque 3.2.1. Nous avons vu au chapitre 2 que la fonction de coût K_i des classifieurs associés aux modèles gaussiens classiques, qu'ils soient supervisés ou non, nécessite l'inversion de la matrice de covariance Σ_i et que cela peut s'avérer difficile si l'estimation de cette matrice est mal conditionnée. Or, en examinant l'expression de K_i associée au modèle $[a_{ij}b_iQ_id_i]$, on s'aperçoit qu'elle ne nécessite pas l'inversion de Σ_i. En effet, les termes de variance étant résumés dans les paramètres a_{ij} et b_i, il n'est plus nécessaire d'inverser pour chaque classe la matrice de covariance Σ_i. Ainsi, le classifieur associé au modèle $[a_{ij}b_iQ_id_i]$ ne rencontrera pas les problèmes posés par l'inversion des matrices de covariance des classes.

Remarque 3.2.2. De même, il est particulièrement intéressant de noter que la fonction de coût K_i du modèle $[a_{ij}b_iQ_id_i]$ n'utilise jamais la projection sur le sous-espace $\mathbb{E}_i^{\perp}$ et de ce fait ne requiert pas la détermination des $(p-d_i)$ dernières colonnes des matrices Q_i, $i = 1, ..., k$.

Interprétation de la fonction de coût K_i associée au modèle $[a_{ij}b_iQ_id_i]$

Nous allons à présent nous intéresser à l'interprétation géométrique de la fonction de coût K_i qui, dans le cadre supervisé, détermine totalement la règle de décision du MAP. Rappelons que dans le cadre non supervisé, l'étape E de l'algorithme EM qui calcule à chaque étape les probabilités conditionnelles d'appartenance aux classes repose également en totalité sur la fonction de coût K_i. La figure 3.3 permet de visualiser les quantités présentes dans l'expression de la fonction de coût K_i. Il apparaît tout d'abord que la fonction de coût K_i associée au modèle $[a_{ij}b_iQ_id_i]$ dépend principalement de deux distances : la distance $\|\mu_i - P_i(x)\|^2_{\mathcal{A}_i}$ associée à la matrice $\mathcal{A}_i = \tilde{Q}_i\Delta_i^{-1}\tilde{Q}_i^t$ et la distance euclidienne $\frac{1}{b_i}\|x - P_i(x)\|^2$ pondérée par la variance en dehors du sous-espace de la classe. Ainsi, la fonction de coût K_i favorisera l'affectation d'une observation x à la classe dont l'observation est à la fois proche du sous-espace $\mathbb{E}_i$ de la classe et dont sa projection sur ce sous-espace est proche du centre

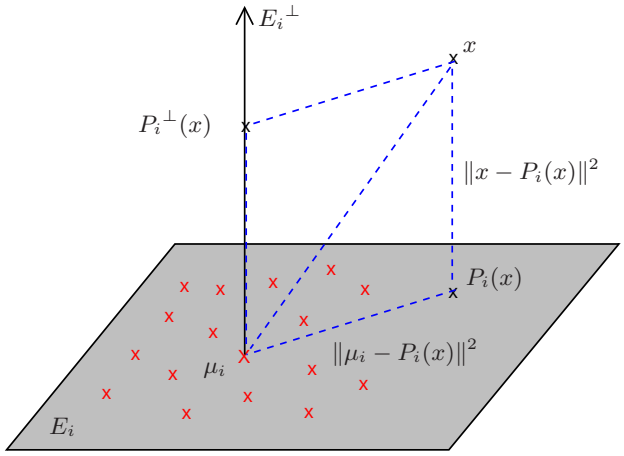


FIG. 3.3 – Les sous-espaces $\mathbb{E}_i$ et $\mathbb{E}_i^\perp$ associés à la $i$ème composante du mélange.

de la classe. Evidemment, Les variances dans les sous-espaces $\mathbb{E}_i$ et $\mathbb{E}_i^\perp$ entrent aussi en compte pour pondérer l'importance relative de ces deux distances. Par exemple, si les données sont très bruitées, *i.e.* b_i est grand, il est naturel de pondérer la distance $\|x - P_i(x)\|^2$ du point au sous-espace $\mathbb{E}_i$ par $1/b_i$ afin de prendre en compte la grande variance en dehors du sous-espace de la classe. L'action de la fonction de coût K_i est finalement assez naturelle et elle réalise ce que nous ferions intuitivement.

3.2.3 Complexité du modèle $[a_{ij}b_iQ_id_i]$

Le modèle $[a_{ij}b_iQ_id_i]$, né de notre re-paramétrisation du modèle de mélange gaussien, nécessite donc uniquement l'estimation de k sous-espaces de dimensions $d_1, ..., d_k$. En effet, comme il a été noté à la remarque 3.2.1, la fonction de coût K_i ne nécessite pas d'estimer les $(p - d_i)$ dernières colonnes des matrices Q_i, $i = 1, ..., k$, et la complexité du modèle est de ce fait directement liée aux dimensions d_i. Ainsi, plus les dimensions d_i sont petites comparées à la dimension de l'espace, plus le modèle $[a_{ij}b_iQ_id_i]$ est parcimonieux. Nous allons à présent calculer le nombre total de paramètres à estimer pour ce modèle. Cette information sera en particulier très utile pour la sélection du modèle le plus approprié aux données dans le cadre de la classification non supervisée.

Nombre de paramètres à estimer

Le nombre de paramètres nécessaires à l'estimation des moyennes et des proportions des k classes est égal à $\rho = kp + k - 1$. Les matrices Q_i requièrent quant à elles chacune l'estimation de $\tau_i = d_i[p - (d_i + 1)/2]$ paramètres puisque nous n'avons que les d_i premières colonnes à estimer et que les colonnes de Q_i sont orthonormées. Il reste encore à estimer les paramètres a_{ij}, b_i et d_i, pour tout $j = 1, ..., d_i$ et $i = 1, ..., k$, ce qui représente $\sum_{i=1}^{k} d_i + 2k$ paramètres. Finalement, le nombre de

paramètres à estimer pour le modèle $[a_{ij}b_iQ_id_i]$ est égal à :

$$\nu = k(p+3) - 1 + \sum_{i=1}^{k} d_i[p + \frac{1-d_i}{2}].$$

On remarque que le nombre de paramètres à estimer pour le modèle $[a_{ij}b_iQ_id_i]$ est linéaire en p au contraire des modèles gaussiens classiques dont le nombre de paramètres à estimer croît avec p^2.

Ordre asymptotique et cas particulier

En notant $\bar{d} = \frac{1}{k}\sum_{i=1}^{k} d_i$ la dimension intrinsèque moyenne et en supposant que $k \ll d_i \ll p$, pour tout $i = 1, ..., k$, l'ordre asymptotique du nombre de paramètres à estimer est $kp\bar{d}$. Si l'on considère le cas de données composées de $k = 4$ classes de dimension intrinsèque moyenne $\bar{d} = 10$ et dont la dimension de l'espace original est $p = 100$, alors le modèle $[a_{ij}b_iQ_id_i]$ nécessite l'estimation de 4321 paramètres. Rappelons que dans ce même cas, le modèle gaussien classique full-GMM requiert l'estimation de 20603 paramètres et le modèle com-GMM nécessite lui l'estimation de 5453 paramètres. Il est donc intéressant de remarquer que le modèle $[a_{ij}b_iQ_id_i]$ requiert l'estimation en grande dimension d'un plus petit nombre de paramètres qu'un modèle donnant naissance à une règle linéaire alors que nous verrons au chapitre suivant qu'il engendre une règle de décision quadratique.

3.3 Les sous-modèles de $[a_{ij}b_iQ_id_i]$

En suivant l'approche de Celeux et Govaert [21], il est possible de générer un ensemble de sous-modèles du modèle $[a_{ij}b_iQ_id_i]$ en contraignant certains paramètres à être communs à l'intérieur d'une classe ou entre les classes. Par exemple, si l'on fixe les dimensions d_i à être communes entre les classes, on obtient alors un sous-modèle noté $[a_{ij}b_iQ_id]$ de notre modèle général $[a_{ij}b_iQ_id_i]$. Nous verrons d'ailleurs au paragraphe 3.4.2 que ce modèle est lié aux méthodes basées sur les mélanges de *factor analyzers*. Dans la suite du mémoire, « Q_i libres » signifiera que chaque classe C_i a une matrice Q_i spécifique et « Q_i communes » traduira le fait que pour tout $i = 1, ..., k$, $Q_i = Q$ et donc que l'orientation des classes est la même. La famille du modèle $[a_{ij}b_iQ_id_i]$, qui compte 28 modèles, peut ainsi être divisée en trois catégories de modèles : les modèles à orientations libres (Q_i libres), les modèles à orientations communes (Q_i communes) et les modèles à matrices de covariance communes. Le tableau 3.1 recense l'ensemble des modèles issus de la re-paramétrisation du modèle de mélange gaussien, proposée au paragraphe 3.2, ainsi que leurs principales propriétés.

3.3.1 Modèles à orientations libres

Cette catégorie de modèles suppose que les groupes vivent dans des sous-espaces dont les orientations sont différentes, *i.e.* les matrices Q_i sont spécifiques à chaque classe. Naturellement, le modèle principal $[a_{ij}b_iQ_id_i]$ appartient à cette catégorie qui en compte 14 au total.

Modèle	Nombre de paramètres	Ordre asymptotique	Nb de prms $k = 4$, $\bar{d} = 10$, $p = 100$	Estimation par MV
$[a_{ij}b_iQ_id_i]$	$\rho + k(\bar{\tau} + 2 + \bar{d})$	$kp\bar{d}$	4231	CF
$[a_{ij}bQ_id_i]$	$\rho + k(\bar{\tau} + \bar{d} + 1) + 1$	$kp\bar{d}$	4228	CF
$[a_ib_iQ_id_i]$	$\rho + k(\bar{\tau} + 3)$	$kp\bar{d}$	4195	CF
$[ab_iQ_id_i]$	$\rho + k(\bar{\tau} + 2) + 1$	$kp\bar{d}$	4192	CF
$[a_ibQ_id_i]$	$\rho + k(\bar{\tau} + 2) + 1$	$kp\bar{d}$	4192	CF
$[abQ_id_i]$	$\rho + k(\bar{\tau} + 1) + 2$	$kp\bar{d}$	4189	CF
$[a_{ij}b_iQ_id]$	$\rho + k(\tau + d + 1) + 1$	$kp\bar{d}$	4228	CF
$[a_jb_iQ_id]$	$\rho + k(\tau + 1) + d + 1$	$kp\bar{d}$	4198	CF
$[a_{ij}bQ_id]$	$\rho + k(\tau + d) + 2$	$kp\bar{d}$	4225	CF
$[a_jbQ_id]$	$\rho + k\tau + d + 2$	$kp\bar{d}$	4195	CF
$[a_ib_iQ_id]$	$\rho + k(\tau + 2) + 1$	$kp\bar{d}$	4192	CF
$[ab_iQ_id]$	$\rho + k(\tau + 1) + 2$	$kp\bar{d}$	4189	CF
$[a_ibQ_id]$	$\rho + k(\tau + 1) + 2$	$kp\bar{d}$	4189	CF
$[abQ_id]$	$\rho + k\tau + 3$	$kp\bar{d}$	4186	CF
$[a_{ij}b_iQd_i]$	$\rho + \tau + k(\bar{d} + 2)$	pd	1396	FG
$[a_{ij}bQd_i]$	$\rho + \tau + k(\bar{d} + 1) + 1$	pd	1393	FG
$[a_ib_iQd_i]$	$\rho + \tau + 3k$	pd	1360	FG
$[a_ibQd_i]$	$\rho + \tau + 2k + 1$	pd	1357	FG
$[ab_iQd_i]$	$\rho + \tau + 2k + 1$	pd	1357	FG
$[abQd_i]$	$\rho + \tau + k + 2$	pd	1354	FG
$[a_{ij}b_iQd]$	$\rho + \tau + kd + k + 1$	pd	1393	FG
$[a_jb_iQd]$	$\rho + \tau + k + d + 1$	pd	1363	FG
$[a_{ij}bQd]$	$\rho + \tau + kd + 2$	pd	1390	FG
$[a_ib_iQd]$	$\rho + \tau + 2k + 1$	pd	1357	IP
$[ab_iQd]$	$\rho + \tau + k + 2$	pd	1354	IP
$[a_ibQd]$	$\rho + \tau + k + 2$	pd	1354	IP
$[a_jbQd]$	$\rho + \tau + d + 2$	pd	1360	CF
$[abQd]$	$\rho + \tau + 3$	pd	1351	CF
Full-GMM	$\rho + kp(p+1)/2$	$kp^2/2$	20603	CF
Com-GMM	$\rho + p(p+1)/2$	$p^2/2$	5453	CF
Diag-GMM	$\rho + kp$	$2kp$	803	CF
Sphe-GMM	$\rho + k$	kp	407	CF

TAB. 3.1 – Propriétés du modèle $[a_{ij}b_iQ_id_i]$ et de ses sous-modèles : $\rho = kp + k - 1$ est le nombre de paramètres nécessaire à l'estimation des moyennes et proportions, $\bar{\tau} = \frac{1}{k}\sum_{i=1}^{k} d_i[p - (d_i + 1)/2]$ est le nombre moyen de paramètres nécessaire à l'estimation d'une matrice $\tilde{Q}_i$ et $\tau = d[p - (d + 1)/2]$ est le nombre de paramètres nécessaires à l'estimation de la matrice $\tilde{Q}$. Nous notons également $\bar{d} = \frac{1}{k}\sum_{i=1}^{k} d_i$ et supposons que $k \ll d_i \ll p$, pour tout $i = 1, ..., k$, pour le calcul des ordres asymptotiques. CF signifie que les estimateurs du MV sont explicites, IP signifie qu'ils nécessitent le recours à une procédure itérative, FG signifie qu'ils nécessitent l'utilisation de l'algorithme FG.

Caractéristiques des modèles à orientations libres

Les modèles de cette catégorie sont obtenus à partir du modèle principal en contraignant sa paramétrisation par le biais des paramètres a_{ij}, b_i ou d_i. Tout d'abord, il est possible de fixer les d_i premières valeurs propres à être communes et ce dans chacune des classes, *i.e.* $a_{i1} = ... = a_{id_i} = a_i$, pour tout $i = 1, ..., k$. Cela revient à régulariser l'estimation des d_i premiers coefficients de chaque matrice Δ_i. En suivant le système de notation introduit au paragraphe 3.2, ce modèle sera noté $[a_ib_iQ_id_i]$. Nous verrons dans les chapitres 5 et 6 que ce modèle donnera des résultats très satisfaisants dans la pratique. Cela permet de penser que l'hypothèse selon laquelle chaque matrice Δ_i ne contient que 2 valeurs propres différentes est un moyen efficace de régulariser l'estimation de Δ_i. Notons que cela revient à supposer que les densités des classes sont de forme sphérique à la fois dans les sous-espaces $\mathbb{E}_i$ et dans les sous-espaces $\mathbb{E}_i^{\perp}$. Un autre type de régularisation possible est de fixer les paramètres b_i à être communs entre les classes. Nous obtenons alors le modèle $[a_{ij}bQ_id_i]$ qui suppose donc que la variance en dehors du sous-espace spécifique de chaque classe est commune. Cela peut être vu comme une façon de modéliser le fait que le bruit est commun aux classes, ce qui est assez naturel si les données ont été obtenues selon le même protocole d'acquisition. Cette catégorie de modèles contient également les modèles $[a_ibQ_id_i]$, $[ab_iQ_id_i]$, $[abQ_id_i]$ et tous les modèles à orientations libres et dimensions communes. La figure 3.4 permet d'observer l'influence de ces contraintes sur la forme des densités des classes. La représentation étant faite en dimension 2, les dimensions intrinsèques d_i des classes sont fixées et égales à 1. Les modèles de la première ligne font l'hypothèse que les variances dans et en dehors de leur sous-espace spécifique sont propres pour chacune des deux classes. A la seconde ligne, les modèles supposent que les variances dans les sous-espaces spécifiques sont égales. A l'inverse, les modèles de la troisième ligne supposent que la variance en dehors des sous-espaces des classes est commune. Enfin, la dernière ligne de la figure présente les modèles faisant l'hypothèse que les variances dans et en dehors des sous-espaces sont communes entre les classes.

Fonction de coût K_i associée aux modèles à orientations libres

Les fonctions de coût K_i associées aux modèles à orientations libres s'expriment facilement à partir de la proposition 3.2.1 qui donne l'expression de K_i pour le modèle général $[a_{ij}b_iQ_id_i]$. Nous allons donner dans ce paragraphe l'expression de K_i pour deux modèles à orientations libres dont la fonction de coût est interprétable de façon géométrique.

Cas du modèle $[a_ib_iQ_id_i]$ La fonction de coût K_i associée au modèle $[a_ib_iQ_id_i]$ s'exprime de la façon suivante :

$$K_i(x) = \frac{1}{a_i}\|\mu_i - P_i(x)\|^2 + \frac{1}{b_i}\|x - P_i(x)\|^2 + d_i\log(a_i) + (p - d_i)\log(b_i) - 2\log(\pi_i) + C^{te}.$$

Avec les hypothèses de ce modèle, la distance $\|\mu_i - P_i(x)\|^2_{\mathcal{A}_i}$ entre la projection de x sur le sous-espace $\mathbb{E}_i$ et la moyenne de la classe est alors une distance euclidienne pondérée simplement par

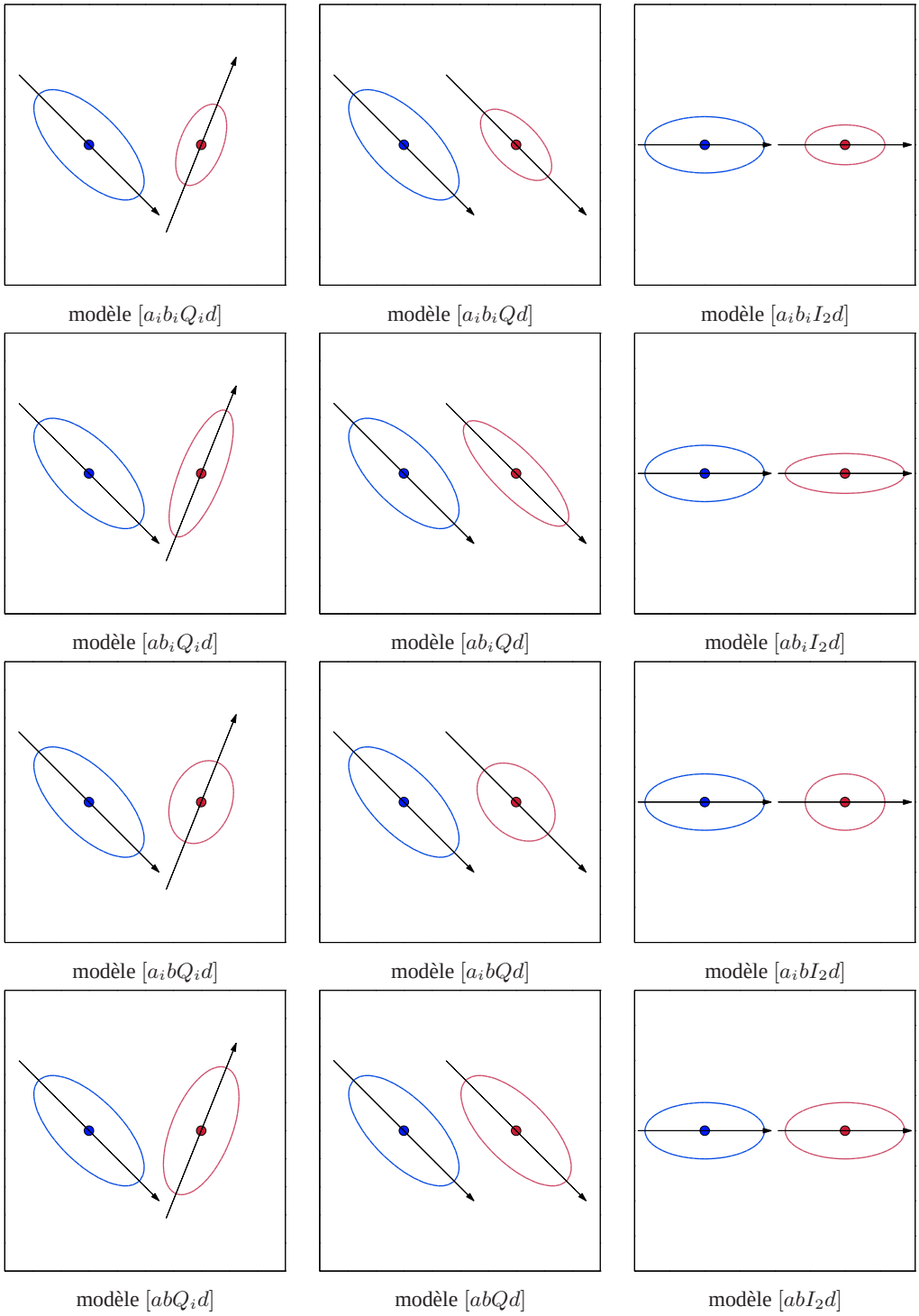


FIG. 3.4 – Influence des paramètres a_{ij}, b_i et Q_i sur les densités des classes. La représentation étant faite en dimension 2, les dimensions intrinsèques d_i des classes sont fixées et égales à 1.

l'unique paramètre a_i modélisant la variance dans le sous-espace de la classe C_i. Ainsi, si la variance dans le sous-espace de la classe est importante, *i.e.* a_i est grand, il est naturel de pondérer cette distance par $1/a_i$ afin de ne pas affecter trop facilement à cette classe des observations dont les projections sur $\mathbb{E}_i$ sont certes relativement proches de μ_i mais dont les distances au sous-espace de la classe sont très grandes.

Cas du modèle $[abQ_id]$ L'expression de la fonction de coût K_i associée au modèle $[abQ_id]$ s'exprime de la façon suivante :

$$K_i(x) = \frac{1}{a}\|\mu_i - P_i(x)\|^2 + \frac{1}{b}\|x - P_i(x)\|^2 + d\log(a) + (p-d)\log(b) - 2\log(\pi_i) + C^{te}.$$

Afin de simplifier l'écriture de la fonction K_i associée au modèle $[abQ_id]$, nous introduisons les notations suivantes :

$$a = \frac{\sigma^2}{\alpha} \text{ et } b = \frac{\sigma^2}{(1-\alpha)},$$

avec $\alpha \in]0,1[$. En supprimant les valeurs qui ne dépendent plus des classes et en supposant de plus que les proportions π_i sont égales, on peut ré-écrire la fonction de coût K_i associée au modèle $[abQ_id]$ sous la forme suivante :

$$K_i(x) = \alpha\|\mu_i - P_i(x)\|^2 + (1-\alpha)\|x - P_i(x)\|^2 + C^{te}.$$

Il apparaît alors trois cas de figure :

(i) α est proche de 0 : la fonction de coût K_i entraînera généralement l'affectation de l'observation x à la classe dont elle est la plus proche du sous-espace de la classe.

(ii) α est égal à 0.5 : dans ce cas, $K_i(x) = \|\mu_i - x\|^2 + C^{te}$ et la règle de décision affecte alors l'observation x à la classe dont elle est la plus proche du centre.

(iii) α est proche de 1 : dans ce cas, la fonction de coût K_i entraînera généralement l'affectation de l'observation x à la classe dont la projection sur le sous-espace de la classe est la plus proche du centre.

Complexité des modèles à orientations libres

Le nombre de paramètres à estimer pour les modèles de cette catégorie est naturellement du même ordre que le modèle principal $[a_{ij}b_iQ_id_i]$. La seconde colonne du tableau 3.1 donne le nombre de paramètres à estimer pour chacun de ces modèles. La troisième colonne fournit l'ordre asymptotique du nombre de paramètres à estimer (en supposant $k \ll d_i \ll p$, pour $i = 1, ..., k$) et la quatrième colonne donne ce nombre pour le cas particulier de données composées de 4 classes de dimension intrinsèque moyenne égale à 10 et dans un espace original de dimension 100. Ce tableau nous indique que le nombre total de paramètres (incluant proportions, moyennes et matrices de covariances des k

classes) est asymptotiquement de l'ordre de $kp\bar{d}$ en supposant que $k \ll d_i \ll p$, pour $i = 1, ..., k$, et que le nombre de paramètres à estimer pour les modèles à orientations libres est de l'ordre de 4200 pour le cas particulier $k = 4$, $p = 100$ et $\bar{d} = 10$. Il apparaît clairement que les modèles de cette catégorie sont, comme le modèle $[a_{ij}b_iQ_id_i]$, beaucoup moins pénalisés en grande dimension que les modèles gaussiens classiques dont le nombre de paramètres à estimer est proportionnel au carré de la dimension. Enfin, en anticipant sur le prochain chapitre, la dernière colonne du tableau 3.1 indique que les estimateurs du *maximum* de vraisemblance des paramètres des modèles à orientations libres sont explicites.

3.3.2 Modèles à orientations communes

Les 12 modèles de cette catégorie supposent que les classes ont même orientation en forçant les matrices Q_i à être communes, *i.e.* $Q_i = Q$ pour $i = 1, ..., k$.

Caractéristiques des modèles à orientations communes

Il faut tout d'abord noter que supposer les orientations communes ne signifie pas nécessairement que les données des différentes classes vivent dans le même sous-espace. En effet, si les dimensions d_i et les moyennes μ_i ne sont pas contraintes à être égales, alors les sous-espaces sont différents et utilisent seulement le même système de coordonnées. La figure 3.4 permet d'observer l'influence du paramètre matriciel Q_i sur les orientations des classes. Les modèles de cette catégorie sont destinés à modéliser des groupes ayant à la fois des propriétés communes et des caractéristiques spécifiques. Par exemple, les données « visages », présentées au chapitre 2, ont été obtenues en faisant varier les mêmes traits d'expression chez chacun des 13 sujets. Pour l'analyse de ces données, un modèle à orientations communes pourrait être tout à fait adapté car il prendrait en compte à la fois le fait que la nature de la variation est commune pour tous les sujets et que l'expression de cette variation est spécifique à chacun des sujets.

Fonction de coût K_i associée aux modèles à orientations communes

L'expression de la fonction de coût K_i associée aux modèles à orientations communes s'obtient facilement à partir de de la proposition 3.2.1 qui donne l'expression de K_i pour le modèle général $[a_{ij}b_iQ_id_i]$. La fonction de coût K_i associée au modèle $[a_{ij}b_iQd_i]$, le plus général de cette catégorie, est de la forme suivante :

$$K_i(x) = \|\mu_i - P_i(x)\|^2_{\mathcal{A}_i} + \frac{1}{b_i}\|x - P_i(x)\|^2 + \sum_{j=1}^{d_i} \log(a_{ij}) + (p - d_i)\log(b_i) - 2\log(\pi_i) + C^{te},$$

où $\|.\|_{\mathcal{A}_i}$ est une norme sur $\mathbb{E}_i$ telle que $\|x\|^2_{\mathcal{A}_i} = x^t\mathcal{A}_ix$ avec $\mathcal{A}_i = \tilde{Q}\Delta_i^{-1}\tilde{Q}^t$ et où $P_i(x) = \tilde{Q}\tilde{Q}^t(x - \mu_i) + \mu_i$.

Complexité des modèles à orientations communes

Le nombre de paramètres à estimer pour cette catégorie de modèle est globalement k fois moins important que pour les modèles à orientations libres. En effet, les matrices Q_i, $i = 1, ..., k$, étant supposées égales, il est alors nécessaire d'estimer qu'une seule matrice Q. La complexité asymptotique des modèles à orientations communes est donc égale à $p\bar{d}$. Ces modèles sont ainsi particulièrement parcimonieux par rapport aux modèles full-GMM et com-GMM. Le tableau 3.1 indique que le nombre de paramètres à estimer pour ces modèles est inférieur à 1400 pour le cas particulier de données composées de 4 classes de dimension intrinsèque moyenne égale à 10 et dans un espace original de dimension 100. Parmi ces modèles à orientations communes, plusieurs nécessitent, pour estimer leurs paramètres, l'utilisation d'une méthode numériquement coûteuse basée sur l'algorithme FG [32] et qui rend leur utilisation difficile (voir chapitre 4).

3.3.3 Modèles à matrices de covariance communes

Cette catégorie de modèles ne contient que les deux modèles $[a_jbQd]$ et $[abQd]$ qui supposent que les classes ont même matrice de covariance $\Sigma = Q\Delta Q^t$.

Caractéristiques des modèles à matrices de covariance communes

On peut tout d'abord remarquer que, si $d = (p-1)$, le modèle $[a_jbQd]$ est équivalent au modèle gaussien classique com-GMM qui, rappelons-le donne naissance à la très populaire méthode LDA dans le cadre de l'analyse discriminante (voir chapitre 2). Ainsi, si $d < (p-1)$, le modèle $[a_jbQd]$ peut être vu comme la combinaison d'une réduction de dimension avec un modèle gaussien sous l'hypothèse d'égalité des matrices de covariance, mais tout cela sans perte d'information. En effet, l'information portée par les dimensions associées aux plus petites valeurs propres est conservée. Le modèle $[abQd]$ est quant à lui très proche du modèle du classifieur *nearest-mean* qui suppose que les matrices de covariance sont égales et proportionnelles à la matrice identité I_p. En effet, les seules différences sont que le modèle $[abQd]$ ne suppose pas l'indépendance conditionnelle des variables et réalise la classification en tenant compte des sous-espaces spécifiques des classes. Il est enfin possible de contraindre l'unique matrice d'orientation Q à être égale à l'identité et de contraindre les valeurs propres a et b à être égales. Le modèle ainsi obtenu est le modèle gaussien du classifieur *nearest-mean*.

Fonction de coût K_i associée aux modèles à matrices de covariance communes

L'expression de la fonction de coût K_i associée aux modèles à matrices de covariance communes s'obtient facilement à partir de la proposition 3.2.1 qui donne l'expression de K_i pour le modèle général $[a_{ij}b_iQ_id_i]$.

Cas du modèle $[a_j bQd]$ La fonction de coût K_i associée au modèle $[a_j bQd]$ s'exprime de la façon suivante :

$$K_i(x) = \|\mu_i - P_i(x)\|^2_{\mathcal{A}} + \frac{1}{b}\|x - P_i(x)\|^2 - 2\log(\pi_i) + C^{te},$$

où $\|.\|_{\mathcal{A}}$ est une norme sur $\mathbb{E}_i$ telle que $\|x\|^2_{\mathcal{A}} = x^t\mathcal{A}x$ avec $\mathcal{A} = \tilde{Q}\Delta^{-1}\tilde{Q}^t$ et $P_i(x) = \tilde{Q}\tilde{Q}^t(x - \mu_i) + \mu_i$. En effet, les termes $\sum_{j=1}^{d}\log(a_j)$ et $(p-d)\log(b)$ ne dépendant plus des classes, ils n'interviennent pas dans la fonction de coût.

Cas du modèle $[abQd]$ La fonction de coût K_i associée au modèle $[abQd]$, le plus parcimonieux de la famille du modèle $[a_{ij}b_iQ_id_i]$, a la forme suivante :

$$K_i(x) = \frac{1}{a}\|\mu_i - P_i(x)\|^2 + \frac{1}{b}\|x - P_i(x)\|^2 - 2\log(\pi_i) + C^{te},$$

où $P_i(x) = \tilde{Q}\tilde{Q}^t(x - \mu_i) + \mu_i$. En utilisant à nouveau les notations $a = \frac{\sigma^2}{\alpha}$ et $b = \frac{\sigma^2}{1-\alpha}$, il possible d'écrire la fonction de coût K_i associée au modèle $[abQd]$ sous la forme simplifiée :

$$K_i(x) = \alpha\|\mu_i - P_i(x)\|^2 + (1-\alpha)\|x - P_i(x)\|^2 - 2\log(\pi_i) + C^{te},$$

avec toujours $P_i(x) = \tilde{Q}\tilde{Q}^t(x - \mu_i) + \mu_i$. On observe alors que la règle de décision associée au modèle $[abQd]$, déterminée par la fonction de coût K_i, ne dépend alors plus que des moyennes et proportions des classes ainsi que du rapport de a et b et de l'unique matrice Q.

Complexité des modèles à matrices de covariance communes

Ces modèles supposant implicitement que les matrices d'orientation Q_i sont communes entre les classes, leur complexité est du même ordre que la complexité des modèles à orientations communes. Le nombre de paramètre à estimer est donc asymptotiquement égal à pd. Si l'on considère enfin le cas particulier $k = 4$, $p = 100$ et $\bar{d} = 10$ alors les modèles $[a_j bQd]$ et $[abQd]$ nécessitent respectivement l'estimation de 1360 et 1351 paramètres ce qui confirme leur proximité avec les modèles diag-GMM et sphe-GMM.

3.4 Liens avec les modèles gaussiens existants

Nous allons à présent montrer que la re-paramétrisation du modèle de mélange que nous avons proposée dans ce chapitre permet d'unifier les différentes approches existantes de classification des données de grande dimension. Nous allons pour cela démontrer que les modèles gaussiens classiques et les modèles de classification dans des sous-espaces appartiennent à la famille du modèle $[a_{ij}b_iQ_id_i]$.

3.4.1 Liens avec les modèles gaussiens classiques

Intéressons-nous tout d'abord aux modèles gaussiens classiques qui, comme nous l'avons vu au chapitre 2, sont très utilisés en pratique en classification.

Modèle gaussien full-GMM

Si l'on suppose que $d_i = p-1$, et ce pour $i = 1, ..., k$, alors le modèle $[a_{ij}b_iQ_id]$ est équivalent au modèle de mélange gaussien classique full-GMM. La matrice de covariance Δ_i ayant alors p valeurs propres distinctes, la fonction de coût K_i associée au modèle $[a_{ij}b_iQ_id]$ s'exprime de la façon suivante :

$$\begin{aligned} K_i(x) &= \|\mu_i - P_i(x)\|^2_{\mathcal{A}_i} + \frac{1}{b_i}\|x - P_i(x)\|^2 + \sum_{j=1}^{p-1}\log(a_{ij}) + \log(b_i) - 2\log(\pi_i) + C^{te} \\ &= \|\mu_i - x\|^2_{Q_i\Delta_i^{-1}Q_i^t} + \log(\det\Delta_i) - 2\log(\pi_i) + C^{te}. \end{aligned}$$

Or $\Sigma_i = Q_i\Delta_iQ_i^t$ et $\det\Delta_i = \det\Sigma_i$, par conséquent on a :

$$K_i(x) = (\mu_i - x)^t\Sigma_i^{-1}(\mu_i - x) + \log(\det\Sigma_i) - 2\log(\pi_i) + C^{te},$$

qui est en effet la fonction de coût associée au modèle full-GMM (*cf.* paragraphe 2.2.2).

Modèle gaussien com-GMM

En supposant toujours que $d_i = p-1$ et ce pour $i = 1, ..., k$, le modèle $[a_jbQd]$ est équivalent au modèle com-GMM qui suppose l'égalité des matrices de covariance. En effet, la fonction de coût K_i associée au modèle $[a_jbQd]$ s'exprime, sous la contrainte $d = p-1$, de la façon suivante :

$$\begin{aligned} K_i(x) &= \|\mu_i - P_i(x)\|^2_{\mathcal{A}} + \frac{1}{b}\|x - P_i(x)\|^2 + \sum_{j=1}^{p-1}\log(a_j) + \log(b) - 2\log(\pi_i) + C^{te} \\ &= \|\mu_i - x\|^2_{Q^t\Delta^{-1}Q^t} + \log(\det\Delta) - 2\log(\pi_i) + C^{te}. \end{aligned}$$

Or $\Sigma = Q\Delta Q^t$ et $\log(\det\Delta)$ étant une constante indépendante de l'indice de la classe, on peut alors écrire K_i sous la forme suivante :

$$\begin{aligned} K_i(x) &= \|\mu_i - x\|^2_{\Sigma^{-1}} - 2\log(\pi_i) + C^{te} \\ &= \mu_i^t\Sigma^{-1}\mu_i - 2\mu_i^t\Sigma^{-1}x - 2\log(\pi_i) + C^{te}, \end{aligned}$$

qui est en effet la fonction de coût associée au modèle com-GMM (*cf.* paragraphe 2.2.2).

Autres modèles gaussiens classiques

De même, si $d_i = p-1$, pour tout $i = 1, ..., k$, le modèle $[a_{ij}b_iI_pd]$ est alors équivalent au modèle diag-GMM supposant l'indépendance conditionnelle des variables et, si de plus $a_{ij} = b_i$, pour tout $j = 1, ..., p-1$, alors le modèle $[a_{ij}b_iI_pd]$ est équivalent au modèle sphe-GMM. Enfin, si $d_i = p-1$ et $\pi_i = 1/k$, pour tout $i = 1, ..., k$, et $a_j = b$, pour tout $j = 1, ..., p-1$, alors le modèle $[a_jbI_pd]$ est le modèle du classifieur *nearest-mean*. En effet, la fonction de coût K_i associée au modèle $[a_jbI_pd]$ s'exprime, sous les contraintes $d = p-1$ et $\pi_i = 1/k$, pour tout $i = 1, ..., k$, de la façon suivante :

$$\begin{aligned} K_i(x) &= \|\mu_i - P_i(x)\|^2_{\mathcal{A}} + \frac{1}{b}\|x - P_i(x)\|^2 + \sum_{j=1}^{p-1}\log(a_j) + \log(b) - 2\log(\pi_i) + C^{te} \\ &= \|\mu_i - x\|^2_{I_p\Delta^{-1}I_p^t} + \log(\det\Delta) - 2\log(\pi_i) + C^{te}. \end{aligned}$$

En notant $\sigma^2 = a_j = b$, on a $I_p\Delta I_p^t = \sigma^2 I_p$ et comme les quantités $\log(\det\Delta)$ et $-2\log(\pi_i)$ sont indépendantes de l'indice de la classe, on obtient :

$$K_i(x) = \|\mu_i - x\|^2_{\sigma^{-2}I_p} = \frac{1}{\sigma^2}\|\mu_i - x\|^2 + C^{te}.$$

Cette fonction de coût est bien celle du classifieur *nearest-mean* puisqu'elle entraîne l'affectation de l'observation x à la classe dont elle est la plus proche du centre.

3.4.2 Liens avec les modèles de classification dans des sous-espaces

Nous allons également montrer que les modèles de mélange gaussien des approches récentes de classification dans des sous-espaces appartiennent également à la famille du modèle $[a_{ij}b_iQ_id_i]$.

Mélange de *probabilistic principal component analyzers*

Les mélanges de *probabilistic principal component analyzers* (PPCA) [77, 88] supposent que la densité de chacun des groupes est celle d'une loi normale dont la matrice de covariance à la forme suivante :

$$\Sigma_i = H_iH_i^t + \sigma_i^2 I_p.$$

Les colonnes des matrices H_i, de taille $p \times d$, étant orthogonales deux à deux, on peut reformuler la matrice de covariance du modèle de mélange de PPCA dans notre formalisme :

$$\Sigma_i = Q_i\Delta_iQ_i^t,$$

où, pour $j = 1, ..., d$, q_{ij}, la $j$ème colonne de Q_i, est égale à h_{ij}/δ_{ij}, h_{ij} étant la $j$ème colonne de H_i et δ_{ij} sa norme. Le reste de la matrice Q_i étant complété de sorte que $Q_iQ_i^t = I_p$. La matrice Δ_i est alors une matrice diagonale qui contient sur ses d premières lignes les valeurs $a_{ij} = \delta_{ij} + \sigma_i^2$ et sur ses $(p-d)$ dernières lignes l'unique valeur $b_i = \sigma_i^2$. Nous avons ainsi montré qu'en écrivant

le modèle de mélange de PPCA dans notre formalisme, il est équivalent au modèle $[a_{ij}b_iQ_id]$ et appartient à la famille du modèle $[a_{ij}b_iQ_id_i]$. Il est intéressant de noter que ce modèle, ayant été proposé uniquement dans le cadre non supervisé, est désormais utilisable dans le cadre supervisé grâce à notre re-paramétrisation.

Modèle DSM de Flury *et al.*

Le modèle parcimonieux *discrimination subspace model* (DSM), proposé par Flury *et al.* [34], suppose que toutes les différences entre les populations, supposées gaussiennes, se produisent dans un sous-espace de faible dimension. Il est alors assez évident qu'il existe une relation entre notre paramétrisation du modèle de mélange gaussien et le modèle DSM. Le théorème 2.2 de l'article [34] présentant le modèle DSM pour le cas de 2 classes peut être ré-écrit, avec nos conventions, de la façon suivante. Le vecteur aléatoire X de $\mathbb{R}^p$, distribué selon $\mathcal{N}(\mu_i, \Sigma_i)$, $i = 1, 2$, satisfait à un modèle d-DSM si et seulement si il existe une matrice $Q = [\tilde{Q} : \bar{Q}]$ non singulière, de taille $p \times p$, où $\tilde{Q}$ est une matrice $p \times d$ telle que les trois conditions suivantes soient satisfaites :

(i) $Q^t\Sigma_iQ$ soit diagonale pour $i = 1, 2$,

(ii) $\bar{Q}^t\Sigma_1\bar{Q} = \bar{Q}^t\Sigma_2\bar{Q}$,

(iii) $\bar{Q}^t\mu_1 = \bar{Q}^t\mu_2$.

Ces hypothèses impliquent alors que les matrices de covariances des classes ont la forme suivante :

$$\Sigma_i = Q \begin{pmatrix} \Lambda_i & 0 \\ 0 & \Lambda \end{pmatrix} Q^t,\ i = 1, 2,$$

où Λ_i, $i = 1, 2$, est une matrice diagonale comportant les valeurs $a_{i1}, ..., a_{id}$ sur sa diagonale. Nous pouvons alors établir que le modèle DSM assorti des contraintes $\Lambda = b\, I_{p-d}$ et $a_{ij} \geq b$, pour $j = 1, ..., d$, est équivalent au modèle $[a_{ij}bQd]$ de notre paramétrisation avec la contrainte $\bar{Q}^t\mu_1 = \bar{Q}^t\mu_2$.

Chapitre 4

Classification des données de grande dimension

Au cours du chapitre 2, nous avons mis en évidence que la construction de la règle de décision d'un classifieur basé sur un modèle de mélange paramétrique (dont le mélange gaussien fait partie) revient à estimer les paramètres de ce mélange. En effet, la règle de décision de Bayes affecte chaque nouvelle observation à la classe la plus probable *a posteriori* et la formule de Bayes permet d'obtenir cette probabilité directement à partir des paramètres du mélange. Ce chapitre va donc traiter de la construction des classifieurs associés aux modèles présentés au chapitre précédent au travers de l'estimation des paramètres de ces modèles. Ces classifieurs seront respectivement nommés HDDA (*High Dimensional Discriminant Analysis*) et HDDC (*High Dimensional Data Clustering*) dans les cadres supervisé et non supervisé. Pour ce faire, nous donnerons tout d'abord l'expression de la vraisemblance associée au modèle $[a_{ij}b_iQ_id_i]$ et à ses sous-modèles au paragraphe 4.1. La construction des classifieurs HDDA et HDDC fera l'objet du paragraphe 4.2. Le paragraphe 4.3 traitera de l'estimation des paramètres du modèle $[a_{ij}b_iQ_id_i]$ et de ses sous-modèles. L'estimation des dimensions intrinsèques des classes et du nombre de composantes du mélange sera abordée au paragraphe 4.4. Enfin, le choix du modèle fera l'objet du paragraphe 4.5.

4.1 Vraisemblance du modèle $[a_{ij}b_iQ_id_i]$ et de ses sous-modèles

Afin de simplifier le calcul des estimateurs du *maximum* de vraisemblance des paramètres du modèle $[a_{ij}b_iQ_id_i]$ et de ses sous-modèles, nous allons introduire dans ce paragraphe des expressions de la log-vraisemblance associées au modèles à orientations libres, à orientations communes et à matrices de covariance communes.

4.1.1 Vraisemblance des modèles à orientations libres

Le lemme ci-dessous fournit l'expression de la log-vraisemblance complète, notée $l(\theta)$, pour le modèle $[a_{ij}b_iQ_id_i]$ qui est, rappelons-le, le modèle le plus général parmi les modèles à orientations libres.

Lemme 4.1.1. *La log-vraisemblance complète du modèle* $[a_{ij}b_iQ_id_i]$ *vaut :*

$$l(\theta) = -\frac{1}{2}\sum_{i=1}^{k} n_i \left[\sum_{\ell=1}^{d_i}\left(\log(a_{i\ell}) + \frac{q_{i\ell}^t W_i q_{i\ell}}{a_{i\ell}}\right) + \sum_{\ell=d_i+1}^{p}\left(\log(b_i) + \frac{q_{i\ell}^t W_i q_{i\ell}}{b_i}\right) - 2\log(\pi_i)\right] + C^{te},$$

où $W_i = \frac{1}{n_i}\sum_{j=1}^{n} s_{ij}(x_j - \mu_i)^t(x_j - \mu_i)$ *et* $n_i = \sum_{j=1}^{n} s_{ij}$.

Démonstration. La log-vraisemblance complète, notée $l(\theta)$, vaut :

$$l(\theta) = \sum_{j=1}^{n}\sum_{i=1}^{k} s_{ij}\log(\pi_i f(x_j, \theta_i)),$$

où $s_{ij} = 1$ si x_j provient de la classe C_i et $s_{ij} = 0$ sinon. En remplaçant $f(x_j, \theta_i)$ par son expression en fonction des paramètres $\theta_i = \{\mu_i, \Sigma_i\}$, on obtient :

$$\begin{aligned} l(\theta) &= \sum_{j=1}^{n}\sum_{i=1}^{k} s_{ij}\log\left[\pi_i \frac{1}{(2\pi)^{p/2}\det(\Sigma_i)^{1/2}}\exp\left(-\frac{1}{2}(x_j-\mu_i)^t\Sigma_i^{-1}(x_j-\mu_i)\right)\right] \\ &= \sum_{j=1}^{n}\sum_{i=1}^{k} s_{ij}\left[\log(\pi_i) - \frac{1}{2}\log(\det\Sigma_i) - \frac{1}{2}(x_j-\mu_i)^t\Sigma_i^{-1}(x_j-\mu_i)\right] + C^{te}, \end{aligned}$$

où $C^{te} = -\frac{p}{2}\log(2\pi)$. En utilisant la paramétrisation du modèle $[a_{ij}b_iQ_id_i]$, on obtient :

$$\begin{aligned} l(\theta) &= -\frac{1}{2}\sum_{j=1}^{n}\sum_{i=1}^{k} s_{ij}\left[-2\log(\pi_i) + \log(\prod_{\ell=1}^{d_i} a_{i\ell}\prod_{\ell=d_i+1}^{p} b_i)\right. \\ &\qquad \left. +(x_j-\mu_i)^t Q_i\Delta_i^{-1}Q_i^t(x_j-\mu_i)\right] + C^{te} \\ &= -\frac{1}{2}\sum_{j=1}^{n}\sum_{i=1}^{k} s_{ij}\left[-2\log(\pi_i) + \sum_{\ell=1}^{d_i}\log(a_{i\ell}) + \sum_{\ell=d_i+1}^{p}\log(b_i)\right. \\ &\qquad \left. +(x_j-\mu_i)^t Q_i\Delta_i^{-1}Q_i^t(x_j-\mu_i)\right] + C^{te}. \end{aligned}$$

En notant $n_i = \sum_{j=1}^{n} s_{ij}$ le nombre d'éléments de la classe C_i, la log-vraisemblance s'écrit :

$$l(\theta) = -\frac{1}{2}\sum_{i=1}^{k} n_i \left[-2\log(\pi_i) + \sum_{\ell=1}^{d_i}\log(a_{i\ell}) + \sum_{\ell=d_i+1}^{p}\log(b_i) + \frac{1}{n_i}\sum_{j=1}^{n} s_{ij}(x_j-\mu_i)^t Q_i \Delta_i^{-1} Q_i^t (x_j-\mu_i) \right] + C^{te}. \tag{4.1}$$

La quantité $(x_j-\mu_i)^t Q_i \Delta_i^{-1} Q_i^t (x_j-\mu_i)$ étant un scalaire, elle est égale à sa trace et il est alors possible d'écrire :

$$\frac{1}{n_i}\sum_{j=1}^{n} s_{ij}(x_j-\mu_i)^t Q_i \Delta_i^{-1} Q_i^t (x_j-\mu_i) = \frac{1}{n_i}\sum_{j=1}^{n} s_{ij}\,\mathrm{tr}\left((x_j-\mu_i)^t Q_i \Delta_i^{-1} Q_i^t (x_j-\mu_i)\right).$$

Or $\mathrm{tr}\left(\left[(x_j-\mu_i)^t Q_i\right] \times \left[\Delta_i^{-1} Q_i^t (x_j-\mu_i)\right]\right) = \mathrm{tr}\left(\left[\Delta_i^{-1} Q_i^t (x_j-\mu_i)\right] \times \left[(x_j-\mu_i)^t Q_i\right]\right)$ et par conséquent :

$$\begin{aligned}
\frac{1}{n_i}\sum_{j=1}^{n} s_{ij}(x_j-\mu_i)^t Q_i \Delta_i^{-1} Q_i^t (x_j-\mu_i) &= \frac{1}{n_i}\sum_{j=1}^{n} s_{ij}\,\mathrm{tr}\left(\Delta_i^{-1} Q_i^t (x_j-\mu_i)(x_j-\mu_i)^t Q_i\right). \\
&= \mathrm{tr}\left(\Delta_i^{-1} Q_i^t \left[\frac{1}{n_i}\sum_{j=1}^{n} s_{ij}(x_j-\mu_i)^t (x_j-\mu_i)\right] Q_i\right) \\
&= \mathrm{tr}\left(\Delta_i^{-1} Q_i^t W_i Q_i\right),
\end{aligned}$$

où $W_i = \frac{1}{n_i}\sum_{j=1}^{n} s_{ij}(x_j-\mu_i)^t(x_j-\mu_i)$ est la matrice de covariance empirique à μ_i connue de la $i$ème composante du mélange. La matrice Δ_i étant diagonale, on peut encore écrire :

$$\frac{1}{n_i}\sum_{j=1}^{n} s_{ij}(x_j-\mu_i)^t Q_i \Delta_i^{-1} Q_i^t (x_j-\mu_i) = \sum_{\ell=1}^{d_i}\frac{q_{i\ell}^t W_i q_{i\ell}}{a_{i\ell}} + \sum_{\ell=d_i+1}^{p}\frac{q_{i\ell}^t W_i q_{i\ell}}{b_i},$$

où $q_{i\ell}$ est la ℓème colonne de Q_i. Enfin, en remplaçant dans (4.1), on obtient l'expression finale de la log-vraisemblance $l(\theta)$. □

L'expression de la log-vraisemblance des autres modèles à orientations libres se déduit facilement de l'expression du logarithme de la vraisemblance du modèle $[a_{ij}b_iQ_id_i]$ donnée au lemme 4.1.1. Pour cela, il suffit de remplacer les paramètres fixés dans une classe ou entre les classes par un unique paramètre. Ainsi, le logarithme de la vraisemblance du modèle $[a_ib_iQ_id_i]$ s'écrit plus simplement :

$$l(\theta) = -\frac{1}{2}\sum_{i=1}^{k} n_i \left[d_i\log(a_i) + (p-d_i)\log(b_i) + \sum_{\ell=1}^{d_i}\frac{q_{i\ell}^t W_i q_{i\ell}}{a_i} + \sum_{\ell=d_i+1}^{p}\frac{q_{i\ell}^t W_i q_{i\ell}}{b_i} - 2\log(\pi_i)\right] + C^{te}.$$

4.1.2 Vraisemblance des modèles à orientations communes

L'expression du logarithme de la vraisemblance des modèles à orientations communes s'obtient également à partir du lemme 4.1.1. Le logarithme de la vraisemblance du modèle $[a_{ij}b_iQd_i]$, le plus général de cette catégorie, s'écrit :

$$l(\theta) = -\frac{1}{2}\sum_{i=1}^{k} n_i \left[\sum_{\ell=1}^{d_i}\left(\log(a_{i\ell}) + \frac{q_\ell^t W_i q_\ell}{a_{i\ell}}\right) + \sum_{\ell=d_i+1}^{p}\left(\log(b_i) + \frac{q_\ell^t W_i q_\ell}{b_i}\right) - 2\log(\pi_i)\right] + C^{te}.$$

Si les dimensions d_i sont supposées différentes entre les classes ou que les matrices de covariances sont supposées avoir plus de deux valeurs propres différentes, la maximisation de $l(\theta)$ requiert alors l'utilisation de l'algorithme itératif FG [32]. En revanche, le logarithme de la vraisemblance du modèle $[a_ib_iQd]$ prend une forme qui permet de s'affranchir de l'utilisation de l'algorithme FG.

Lemme 4.1.2. *La log-vraisemblance complète du modèle* $[a_ib_iQd]$ *vaut :*

$$l(\theta) = -\frac{1}{2}\left[\sum_{i=1}^{k} n_i\left(d\log(a_i) + (p-d)\log(b_i) - 2\log(\pi_i) + \frac{\mathrm{tr}(W_i)}{b_i}\right) - \sum_{\ell=1}^{d} q_\ell^t M q_\ell\right] + C^{te},$$

où $M = \sum_{i=1}^{k} n_i\left(\frac{1}{b_i} - \frac{1}{a_i}\right)W_i$.

Démonstration. En partant de l'expression de $l(\theta)$ donnée au lemme 4.1.1, on peut écrire que, dans le cas du modèle $[a_ib_iQd]$, $l(\theta)$ prend la forme suivante :

$$l(\theta) = -\frac{1}{2}\sum_{i=1}^{k} n_i\left[\sum_{\ell=1}^{d}\left(\log(a_i) + \frac{q_\ell^t W_i q_\ell}{a_i}\right) + \sum_{\ell=d+1}^{p}\left(\log(b_i) + \frac{q_\ell^t W_i q_\ell}{b_i}\right) - 2\log(\pi_i)\right] + C^{te},$$

En inversant les sommes sur i et sur ℓ, on peut écrire $l(\theta)$ sous la forme :

$$l(\theta) = -\frac{1}{2}\left[\sum_{i=1}^{k} n_i\left(d\log(a_i) + (p-d)\log(b_i) - 2\log(\pi_i)\right) + \sum_{\ell=1}^{d} q_\ell^t A q_\ell + \sum_{\ell=d+1}^{p} q_\ell^t B q_\ell\right] + C^{te},$$

où $A = \sum_{i=1}^{k}\frac{n_i}{a_i}W_i$ et $B = \sum_{i=1}^{k}\frac{n_i}{b_i}W_i$. En remarquant de plus que $\sum_{\ell=d+1}^{p} q_\ell^t B q_\ell$ peut s'exprimer en fonction de la trace de B :

$$\sum_{\ell=d+1}^{p} q_\ell^t B q_\ell = \mathrm{tr}(B) - \sum_{\ell=1}^{d} q_\ell^t B q_\ell,$$

on peut écrire :

$$l(\theta) = -\frac{1}{2}\left[\sum_{i=1}^{k} n_i\left(d\log(a_i) + (p-d)\log(b_i) - 2\log(\pi_i)\right) - \sum_{\ell=1}^{d} q_\ell^t(B-A)q_\ell + \mathrm{tr}(B)\right] + C^{te}.$$

La relation $\mathrm{tr}(B) = \sum_{i=1}^{k} \frac{n_i}{b_i} \mathrm{tr}(W_i)$ conclut la preuve. □

Le logarithme de la vraisemblance des modèles $[ab_iQd]$ et $[a_ibQd]$ s'exprime facilement à partir de l'expression de $l(\theta)$ donnée au lemme 4.1.2.

4.1.3 Vraisemblance des modèles à matrices de covariance communes

L'expression du logarithme de la vraisemblance des modèles à matrices de covariance communes s'obtient également à partir de l'expression de $l(\theta)$ donnée au lemme 4.1.1.

Lemme 4.1.3. *Le logarithme de la vraisemblance du modèle $[a_jbQd]$, le plus général de cette catégorie, s'écrit :*

$$l(\theta) = -\frac{1}{2}n\left[\sum_{\ell=1}^{d}\log(a_\ell) + (p-d)\log(b) + \frac{\mathrm{tr}(W)}{b} + \sum_{\ell=1}^{d}\left(\frac{1}{a_\ell} - \frac{1}{b}\right)q_\ell^t W q_\ell\right] + \sum_{i=1}^{k} n_i \log(\pi_i) + C^{te},$$

où $W = \frac{1}{n}\sum_{i=1}^{k} n_i W_i$ est la matrice de covariance intra-classe empirique.

Démonstration. En partant de l'expression de $l(\theta)$ donnée au lemme 4.1.1, on peut écrire la log-vraisemblance du modèle $[a_jbQd]$ de la façon suivante :

$$\begin{aligned} l(\theta) &= -\frac{1}{2}\sum_{i=1}^{k} n_i\left[\sum_{\ell=1}^{d}\left(\log(a_\ell) + \frac{q_\ell^t W_i q_\ell}{a_\ell}\right) + \sum_{\ell=d+1}^{p}\left(\log(b) + \frac{q_\ell^t W_i q_\ell}{b}\right) - 2\log(\pi_i)\right] + C^{te} \\ &= -\frac{1}{2}\sum_{i=1}^{k} n_i\left[\sum_{\ell=1}^{d}\log(a_\ell) + (p-d)\log(b) + \sum_{\ell=1}^{d}\frac{q_\ell^t W_i q_\ell}{a_\ell} + \sum_{\ell=d+1}^{p}\frac{q_\ell^t W_i q_\ell}{b} - 2\log(\pi_i)\right] + C^{te}. \end{aligned}$$

En inversant les sommes sur i et sur ℓ, on peut encore écrire $l(\theta)$ sous la forme :

$$l(\theta) = -\frac{1}{2}n\left[\sum_{\ell=1}^{d}\log(a_\ell) + (p-d)\log(b) + \sum_{\ell=1}^{d}\frac{1}{a_\ell}q_\ell^t W q_\ell + \sum_{\ell=d+1}^{p}\frac{1}{b}q_\ell^t W q_\ell\right] + \sum_{i=1}^{k} n_i \log(\pi_i) + C^{te},$$

où $W = \frac{1}{n}\sum_{i=1}^{k} n_i W_i$ est la matrice de covariance intra-classe empirique. De plus, le fait de remarquer que $\sum_{\ell=d+1}^{p} q_\ell^t W q_\ell$ peut s'exprimer en fonction de la trace de W :

$$\sum_{\ell=d+1}^{p} q_\ell^t W q_\ell = \mathrm{tr}(W) - \sum_{\ell=1}^{d} q_\ell^t W q_\ell,$$

permet de conclure la démonstration. □

4.2 Construction des classifieurs HDDA et HDDC

Au cours du chapitre 2, nous avons vu que la construction d'un classifieur génératif, qu'il soit supervisé ou non, passait par deux étapes : l'estimation des paramètres du modèle et le calcul des probabilités conditionnelles. Nous proposons d'utiliser cette même approche pour la construction du classifieur supervisé HDDA et du classifieur non supervisé HDDC, associés au modèle $[a_{ij}b_iQ_id_i]$ et à ses sous-modèles. De la même manière que pour les modèles de mélange gaussien classiques, la technique d'estimation des paramètres des modèles de notre famille est différente selon que l'on soit dans le cadre de l'analyse discriminante ou dans le cadre de la classification automatique. Nous allons présenter dans ce paragraphe les approches générales de construction des classifieurs associés aux modèles de notre famille dans les cadres supervisé et non supervisé. Dans le cadre supervisé, la technique d'estimation des paramètres consistera à maximiser directement la vraisemblance complète alors que dans le cas supervisé l'estimation des paramètres nessecitera l'utilisation de l'algorithme itératif EM. Toutefois, les expressions des estimateurs des paramètres du modèle $[a_{ij}b_iQ_id_i]$ et de ses sous-modèles étant communs dans les cas supervisé et non supervisé, ils seront présentés au paragraphe 4.3.

4.2.1 Construction du classifieur HDDA

Dans le cadre supervisé et de la même façon que les autres classifieurs génératifs, la construction du classifieur HDDA associé à notre famille de modèles nécessite l'estimation des paramètres du modèle et le calcul des probabilités conditionnelles.

Expression des probabilités conditionnelles

Nous allons tout d'abord introduire l'expression de la probabilité conditionnelles $P(Z = i|X = x, \theta)$ qu'une observation x appartienne à la classe C_i sachant les paramètres du mélange et ce, en fonction de la fonction de coût K_i introduite au chapitre 3. La formule de Bayes permet tout d'abord d'écrire :

$$P(Z = i|X = x, \theta) = \frac{\pi_i f(x, \theta_i)}{\sum_{\ell=1}^{k} \pi_\ell f(x, \theta_\ell)},$$

et comme $K_i(x) = -2\log(\pi_i f(x, \theta_i))$, on obtient l'expression suivante de $P(Z = i|X = x, \theta)$:

$$P(Z = i|X = x, \theta) = 1/\sum_{\ell=1}^{k} \exp\left(\frac{1}{2}(K_i(x) - K_\ell(x))\right).$$

La probabilité conditionnelle $P(Z = i|X = x, \theta)$ est donc entièrement déterminée par la fonction de coût K_i. Nous rappelons que lors du chapitre 3, nous avons donné l'expression de K_i en fonction des paramètres $\{\pi_i, \mu_i, a_{ij}, b_i, Q_i, d_i\}$, $i = 1, ..., k$, $j = 1, ..., d_i$, pour chacune des trois catégories de modèles de notre famille.

Estimation des paramètres

Il apparaît donc que l'affectation d'une observation x à une des k classes dépend directement de l'estimation des paramètres du modèle utilisé. Dans le cadre supervisé, les données d'apprentissage étant complètes, *i.e.* composées à la fois des observations x_j et de leurs labels z_j, la maximisation de la vraisemblance peut être faite directement. La proposition suivante donne les estimateurs du *maximum* de vraisemblance des paramètres π_i et μ_i, $i = 1, ..., k$.

Proposition 4.2.1. *Les estimateurs du* maximum *de vraisemblance des proportions π_i et moyennes μ_i des classes sont :*

$$\begin{aligned} \hat{\pi}_i &= \frac{n_i}{n}, \\ \hat{\mu}_i &= \frac{1}{n_i} \sum_{j=1}^{n} s_{ij} x_j, \end{aligned}$$

où $n_i = \sum_{j=1}^{n} s_{ij}$ avec, rappelons-le, $s_{ij} = 1_{\{z_j = i\}}$.

Démonstration. La maximisation de la log-vraisemblance sous la contrainte $\sum_{i=1}^{k} \pi_i = 1$ est équivalente à rechercher un point selle de la fonction de Lagrange suivante :

$$\mathcal{L} = -2l(\theta) - \omega \left(\sum_{i=1}^{k} \pi_i - 1 \right),$$

où ω est le multiplicateur de Lagrange. En utilisant l'expression de la log-vraisemblance $l(\theta)$ donnée au lemme 4.1.1, on obtient :

$$\frac{\partial \mathcal{L}}{\partial \pi_i} = 0 \quad \Leftrightarrow \quad 2n_i + \omega \pi_i = 0.$$

Cette relation étant vraie pour $i = 1, ..., k$, on obtient en sommant sur i que :

$$\omega = -2n.$$

En remplaçant ω dans l'expression de la dérivée partielle de $\mathcal{L}$ par rapport à π_i, on obtient l'estimateur de π_i :

$$\hat{\pi}_i = \frac{n_i}{n}.$$

De même, on a :

$$\frac{\partial \mathcal{L}}{\partial \mu_i} = 0 \quad \Leftrightarrow \quad \sum_{j=1}^{n} s_{ij} x_j = \sum_{j=1}^{n} s_{ij} \mu_i,$$

ce qui permet de conclure que $\hat{\mu}_i = \frac{1}{n_i} \sum_{j=1}^{n} s_{ij} x_j$. □

Il ne reste alors plus qu'à estimer les paramètres a_{ij}, b_i, Q_i et d_i des matrices de covariance des classes. Les expressions des estimateurs de ces paramètres étant similaires dans les cas supervisé et non supervisé, ces estimateurs seront présentés au paragraphe 4.3. Dans le cadre supervisé, l'estimation des paramètres a_{ij}, b_i, Q_i s'appuiera sur l'expression suivante des matrices de covariance empiriques W_i, pour $i = 1, ..., k$:

$$W_i = \frac{1}{n_i} \sum_{j=1}^{n} s_{ij}(x_j - \hat{\mu}_i)(x_j - \hat{\mu}_i)^t,$$

où $n_i = \sum_{j=1}^{n} s_{ij}$. La matrice de covariance intra-classe empirique sera définie quant à elle de manière usuelle en fonction des matrices W_i par $W = \frac{1}{n} \sum_{i=1}^{k} n_i W_i$.

4.2.2 Construction du classifieur HDDC

Dans le cas de la classification automatique, les données disponibles pour apprendre le classifieur n'étant pas complètes, *i.e.* elles ne comportent que les observations x_j et leurs labels z_j sont manquants, il est nécessaire d'utiliser l'algorithme itératif EM pour estimer les paramètres du mélange et les probabilités conditionnelles. La procédure de construction du classifieur HDDC alterne donc entre les deux étapes suivantes :

Etape E de l'algorithme

Cette étape calcule à l'itération q les probabilités *a posteriori* $t_{ij}^{(q)} = P(Z = i|X = x_j, \theta^{(q-1)})$ que l'observation x_j appartienne à la classe C_i :

$$t_{ij}^{(q)} = 1 / \sum_{\ell=1}^{k} \exp\left(\frac{1}{2}(K_i^{(q-1)}(x) - K_\ell^{(q-1)}(x))\right),$$

où $K_i^{(q-1)}$ est la fonction de coût associée au paramètre $\theta^{(q-1)}$ de l'itération $q - 1$.

Etape M de l'algorithme

L'étape M calcule quant à elle, à l'itération q, les estimateurs des paramètres $\theta_i^{(q)}$ par maximisation de l'espérance de la vraisemblance complète conditionnellement aux $t_{ij}^{(q)}$ (*cf.* chapitre 2.3.2). La proposition suivante donne les estimateurs du *maximum* de vraisemblance des paramètres π_i et μ_i, $i = 1, ..., k$.

Proposition 4.2.2. *Les proportions* π_i *et les moyennes* μ_i *sont estimées, à l'étape* q*, par :*

$$\begin{aligned}\hat{\pi}_i^{(q)} &= \frac{n_i^{(q)}}{n},\\ \hat{\mu}_i^{(q)} &= \frac{1}{n_i^{(q)}}\sum_{j=1}^{n} t_{ij}^{(q)} x_j,\end{aligned}$$

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

Démonstration. A chaque étape q, la maximisation de l'espérance de la vraisemblance conditionnellement aux t_{ij} est similaire à la maximisation de la vraisemblance dans le cadre supervisé (*cf.* démonstration de la proposition 4.2.1) et conduit à des équations similaires où les s_{ij} sont remplacés par les t_{ij}, pour $i = 1, ..., k$ et $j = 1, ..., n$. □

Les expressions des estimateurs des paramètres propres à notre paramétrisation, à savoir a_{ij}, b_i, Q_i et d_i, seront présentées au paragraphe suivant dans une approche commune avec le cadre supervisé. A l'itération q, l'estimation de ces paramètres s'appuiera sur l'expression suivante des matrices de covariance empiriques $W_i^{(q)}$, pour $i = 1, ..., k$, qui seront alors des matrices de covariance floues :

$$W_i^{(q)} = \frac{1}{n_i^{(q)}}\sum_{j=1}^{n} t_{ij}^{(q)}(x_j - \hat{\mu}_i^{(q)})(x_j - \hat{\mu}_i^{(q)})^t,$$

avec $n_i^{(q)} = \sum_{j=1}^{n} t_{ij}^{(q)}$. La matrice de covariance intra-classe empirique sera également définie de manière usuelle en fonction des matrices $W_i^{(q)}$ par $W^{(q)} = \frac{1}{n}\sum_{i=1}^{k} n_i^{(q)} W_i^{(q)}$.

4.3 Estimation des paramètres de la famille du modèle $[a_{ij}b_iQ_id_i]$

Nous allons maintenant présenter dans ce paragraphe les estimateurs du *maximum* de vraisemblance du modèle $[a_{ij}b_iQ_id_i]$ et de ses sous-modèles. Les estimateurs donnés dans ce paragraphe sont communs aux cadres supervisés et non supervisés sous réserve de choisir les expressions adaptées de W_i et de n_i, pour $i = 1, ..., k$ (voir paragraphe précédent).

4.3.1 Estimateurs des modèles à orientations libres

Nous allons donner dans ce paragraphe les estimateurs du *maximum* de vraisemblance des paramètres des modèles à orientations libres. Tous les estimateurs présentés dans ce paragraphe sont explicites.

Proposition 4.3.1. *Sous-espace* $\mathbb{E}_i$ *: les* d_i *premières colonnes de la matrice* Q_i *sont estimées par les vecteurs propres associés aux* d_i *plus grandes valeurs propres* λ_{ij} *de* W_i*.*

Démonstration. Nous avons donc à maximiser la vraisemblance sous la contrainte $q_{ij}^t q_{ij} = 1$, ce qui est équivalent à rechercher un point selle de la fonction de Lagrange :

$$\mathcal{L} = -2l(\theta) - \sum_{j=1}^{p} \omega_{ij}(q_{ij}^t q_{ij} - 1),$$

où ω_{ij} sont les multiplicateurs de Lagrange. En utilisant l'expression de la log-vraisemblance $l(\theta)$ du lemme 4.1.1, on obtient :

$$\begin{aligned}\mathcal{L} &= \sum_{i=1}^{k} n_i \left[-2\log(\pi_i) + \sum_{j=1}^{d_i} \left(\log(a_{ij}) + \frac{q_{ij}^t W_i q_{ij}}{a_{ij}} \right) + \sum_{j=d_i+1}^{p} \left(\log(b_i) + \frac{q_{ij}^t W_i q_{ij}}{b_i} \right) \right] \\ &\quad - \sum_{j=1}^{p} \omega_{ij}(q_{ij}^t q_{ij} - 1) + C^{te}.\end{aligned}$$

Ainsi, le gradient de $\mathcal{L}$ par rapport à q_{ij} vaut :

$$\nabla_{q_{ij}} \mathcal{L} = 2\frac{n_i}{\delta_{ij}} W_i q_{ij} - 2\omega_{ij} q_{ij},$$

où δ_{ij} est le $j$ème terme diagonal de la matrice Δ_i. En multipliant cette quantité à gauche par q_{ij}^t, on obtient :

$$q_{ij}^t \nabla_{q_{ij}} \mathcal{L} = 0 \Leftrightarrow \omega_{ij} = \frac{n_i}{\delta_{ij}} q_{ij}^t W_i q_{ij}.$$

Par conséquent, on obtient la relation :

$$W_i q_{ij} = \frac{\omega_{ij}\delta_{ij}}{n_i} q_{ij},$$

et on peut conclure que q_{ij} est le vecteur propre de W_i associé à la valeur propre $\lambda_{ij} = \frac{\omega_{ij}\delta_{ij}}{n_i} = q_{ij}^t W_i q_{ij}$. Etant donné que les vecteurs $q_{ij}, j = 1, ..., p$, sont vecteurs propres de la matrice symétrique W_i, cela implique que $q_{ij}^t q_{i\ell} = 0$ si $j \neq \ell$. La log-vraisemblance peut alors être écrite sous la forme suivante :

$$-2l(\theta) = \sum_{i=1}^{k} n_i \left(\sum_{j=1}^{d_i} \left(\log(a_{ij}) + \frac{\lambda_{ij}}{a_{ij}} \right) + \sum_{j=d_i+1}^{p} \left(\log(b_i) + \frac{\lambda_{ij}}{b_i} \right) \right) + C^{te},$$

et, en utilisant la relation $\sum_{j=d_i+1}^{p} \lambda_{ij} = \mathrm{tr}(W_i) - \sum_{j=1}^{d_i} \lambda_{ij}$, on obtient :

$$-2l(\theta) = \sum_{i=1}^{k} n_i \left(\sum_{j=1}^{d_i} \log(a_{ij}) + (p - d_i)\log(b_i) + \frac{\mathrm{tr}(W_i)}{b_i} + \sum_{j=1}^{d_i} \left(\frac{1}{a_{ij}} - \frac{1}{b_i} \right) \lambda_{ij} \right) + C^{te}. \quad (4.2)$$

Ainsi, la minimisation de la quantité $-2l(\theta)$ par rapport à q_{ij} est équivalente à la minimisation de la quantité $\sum_{i=1}^{k} n_i \sum_{j=1}^{d_i}(\frac{1}{a_{ij}} - \frac{1}{b_i})\lambda_{ij}$ par rapport à λ_{ij}. Etant donné que $(\frac{1}{a_{ij}} - \frac{1}{b_i}) \leq 0, \forall j = 1, ..., d_i$, les valeurs λ_{ij} doivent être aussi grandes que possible. Ainsi, la $j$ème colonne q_{ij} de la matrice Q, $\forall j = 1, ..., d_i$, est estimée par le vecteur propre associé à la $j$ème plus grande valeur propre de W_i. □

Nous allons à présent donner les estimateurs du *maximum* de vraisemblance des paramètres des modèles à matrices d'orientations et dimensions libres. Les estimateurs des paramètres des modèles à matrices d'orientations libres et dimensions communes seront présentés dans un second temps.

Proposition 4.3.2. *Modèle* $[a_{ij}b_iQ_id_i]$ *: l'estimateur de* a_{ij}, $j = 1, ..., d_i$, *est* $\hat{a}_{ij} = \lambda_{ij}$ *et l'estimateur de* b_i *est la moyenne des* $(p - d_i)$ *plus petites valeurs propres de* W_i *et peut s'écrire de la façon suivante :*

$$\hat{b}_i = \frac{1}{(p - d_i)} \left(\text{tr}(W_i) - \sum_{j=1}^{d_i} \lambda_{ij} \right). \tag{4.3}$$

Modèle $[a_{ij}bQ_id_i]$ *: l'estimateur de* a_{ij}, $j = 1, ..., d_i$, *est* $\hat{a}_{ij} = \lambda_{ij}$ *et l'estimateur de* b *est :*

$$\hat{b} = \frac{1}{(p - \xi)} \left(\text{tr}(W) - \sum_{i=1}^{k} \hat{\pi}_i \sum_{j=1}^{d_i} \lambda_{ij} \right), \tag{4.4}$$

où $\xi = \sum_{i=1}^{k} \hat{\pi}_i d_i$ *et* $W = \sum_{i=1}^{k} \hat{\pi}_i W_i$ *est la matrice de covariance intra-classe empirique.*
Modèle $[a_ib_iQ_id_i]$ *: l'estimateur de* b_i *est donné par (4.3) et l'estimateur de* a_i *est :*

$$\hat{a}_i = \frac{1}{d_i} \sum_{j=1}^{d_i} \lambda_{ij}. \tag{4.5}$$

Modèle $[ab_iQ_id_i]$ *: l'estimateur de* b_i *est donné par (4.3) et l'estimateur de* a *est :*

$$\hat{a} = \frac{1}{\xi} \sum_{i=1}^{k} \hat{\pi}_i \sum_{j=1}^{d_i} \lambda_{ij}. \tag{4.6}$$

Modèle $[a_ibQ_id_i]$ *: les estimateurs de* a_i *et* b *sont respectivement donnés par (4.5) et (4.4).*
Modèle $[abQ_id_i]$ *: les estimateurs de* a *et* b *sont respectivement donnés par (4.6) et (4.4).*

Démonstration. Modèle $[a_{ij}b_iQ_id_i]$: en partant de l'expression de la log-vraisemblance du lemme 4.1.1, la dérivée partielle de $l(\theta)$ par rapport à a_{ij} est :

$$-2\frac{\partial l(\theta)}{\partial a_{ij}} = n_i \left(\frac{1}{a_{ij}} - \frac{\lambda_{ij}}{a_{ij}^2} \right)$$

et celle par rapport à b_i est :

$$-2\frac{\partial l(\theta)}{\partial b_i} = \frac{n_i(p-d_i)}{b_i} - \frac{n_i}{b_i^2}\left(\mathrm{tr}(W_i) - \sum_{j=1}^{d_i}\lambda_{ij}\right).$$

La condition $\frac{\partial l(\theta)}{\partial a_{ij}} = 0$ implique que $\hat{a}_{ij} = \lambda_{ij}$, pour $j = 1, ..., d_i$. De même, $\frac{\partial l(\theta)}{\partial b_i} = 0$ implique que :

$$\hat{b}_i = \frac{1}{(p-d_i)}\left(\mathrm{tr}(W_i) - \sum_{j=1}^{d_i}\lambda_{ij}\right).$$

Modèle $[a_{ij}bQ_id_i]$: la dérivée partielle de $l(\theta)$ par rapport à b est :

$$-2\frac{\partial l(\theta)}{\partial b} = \frac{n(p-\xi)}{b} - \frac{1}{b^2}\sum_{i=1}^{k} n_i\left(\mathrm{tr}(W_i) - \sum_{j=1}^{d_i}\lambda_{ij}\right),$$

où $\xi = \sum_{i=1}^{k}\hat{\pi}_i d_i$. La condition $\frac{\partial l(\theta)}{\partial b} = 0$ permet d'établir que :

$$\hat{b} = \frac{1}{(p-\xi)}\left(\mathrm{tr}(W) - \sum_{i=1}^{k}\hat{\pi}_i\sum_{j=1}^{d_i}\lambda_{ij}\right).$$

Modèle $[a_ib_iQ_id_i]$: la dérivée partielle de $l(\theta)$ par rapport à a_i est :

$$-2\frac{\partial l(\theta)}{\partial a_i} = \frac{n_i d_i}{a_i} - \frac{n_i}{a_i^2}\sum_{j=1}^{d_i}\lambda_{ij},$$

et la condition $\frac{\partial l(\theta)}{\partial a_i} = 0$ implique que :

$$\hat{a}_i = \frac{1}{d_i}\sum_{j=1}^{d_i}\lambda_{ij}.$$

Modèle $[ab_iQ_id_i]$: la dérivée partielle de $l(\theta)$ par rapport à a est :

$$-2\frac{\partial l(\theta)}{\partial a} = \frac{n\xi}{a} - \frac{1}{a^2}\sum_{i=1}^{k} n_i\sum_{j=1}^{d_i}\lambda_{ij},$$

et la condition $\frac{\partial l(\theta)}{\partial a} = 0$ donne l'estimateur de a :

$$\hat{a} = \frac{1}{\xi}\sum_{i=1}^{k}\hat{\pi}_i\sum_{j=1}^{d_i}\lambda_{ij},$$

et conclut ainsi la démonstration. □

Nous allons à présent donner les estimateurs du *maximum* de vraisemblance des paramètres des modèles à matrices d'orientations libres et dimensions communes.

Proposition 4.3.3. *Modèles à dimensions communes : les estimateurs des modèles à dimensions* d_i *communes peuvent être obtenus à partir des estimateurs des modèles précédents en remplaçant* d_i *par* d *pour tout* $i = 1, ..., k$*. Dans ce cas, les équations (4.4) et (4.6) peuvent s'écrire respectivement sous les formes simplifiées suivantes :*

$$\hat{a} = \frac{1}{d}\sum_{j=1}^{d}\lambda_j, \tag{4.7}$$

$$\hat{b} = \frac{1}{(p-d)}\left(\mathrm{tr}(W) - \sum_{j=1}^{d}\lambda_j\right), \tag{4.8}$$

où λ_j *est la* j*ème plus grande valeur propre de* W*.*
Modèle $[a_jb_iQ_id]$ *: l'estimateur de* a_j *est* $\hat{a}_j = \lambda_j$ *et l'estimateur de* b_i *est donné par (4.3).*
Modèle $[a_jbQ_id]$ *: l'estimateur de* a_j *est* $\hat{a}_j = \lambda_j$ *et l'estimateur de* b *est donné par (4.8).*

Démonstration. Modèle $[a_jb_iQ_id]$: en partant de l'expression de la log-vraisemblance du lemme 4.1.1, la dérivée partielle de $l(\theta)$ par rapport à a_j est :

$$-2\frac{\partial l(\theta)}{\partial a_j} = \frac{n}{a_j} - \frac{1}{a_j^2}\sum_{i=1}^{k} n_i\lambda_{ij}.$$

La condition $\frac{\partial l(\theta)}{\partial a_j} = 0$ et la relation $\sum_{i=1}^{k} n_i\lambda_{ij} = n\lambda_j$ implique que $\hat{a}_j = \lambda_j$. □

4.3.2 Estimateurs des modèles à orientations communes

Nous allons donner dans ce paragraphe les estimateurs du *maximum* de vraisemblance des paramètres des modèles à orientations communes qui ne requièrent pas l'utilisation de l'algorithme FG [32]. Les estimateurs présentés dans ce paragraphe nécessitent tout de même l'utilisation d'une procédure d'estimation itérative mais très simple comparée à l'algorithme FG.

Proposition 4.3.4. *Sous-espace* $\mathbb{E}_i$ *: pour les modèles* $[a_ib_iQd]$*,* $[a_ibQd]$ *et* $[ab_iQd]$*, les* d *premières colonnes de* Q *sont estimées, sachant* a_i *et* b_i*, par les vecteurs propres associés aux* d *plus grandes valeurs propres de la matrice* M *définie par :*

$$M(a_1, ..., a_k, b_1, ..., b_k) = \sum_{i=1}^{k} n_i(\frac{1}{b_i} - \frac{1}{a_i})W_i. \tag{4.9}$$

Démonstration. Etant donné l'expression de la log-vraisemblance $l(\theta)$ des modèles à orientations communes du lemme 4.1.2, on peut écrire le gradient de la fonction de Lagrange $\mathcal{L} = -2l(\theta) - \sum_{j=1}^{p} \omega_j(q_j^t q_j - 1)$ par rapport à q_j comme suit :

$$\nabla_{q_j}\mathcal{L} = -2Mq_j - 2\omega_j q_j,$$

où $M = \sum_{i=1}^{k} n_i(\frac{1}{b_i} - \frac{1}{a_i})W_i$, et ω_j est le $j$ème multiplicateur de Lagrange. La relation $\nabla_{q_j}\mathcal{L} = 0$ implique :

$$Mq_j = -\omega_j q_j.$$

Cela signifie que q_j est vecteur propre de la matrice M. Par conséquent, la minimisation de la quantité $-2l(\theta)$ par rapport à q_j est équivalente à la minimisation de la quantité $\sum_{j=1}^{d} q_j^t M q_j$ (*cf.* expression de $-2l(\theta)$ donnée au lemme 4.1.2). Or M est une matrice symétrique définie positive et par conséquent les d premières colonnes de Q doivent être les vecteurs propres associés aux d plus grandes valeurs propres de M. □

Proposition 4.3.5. *Modèle $[a_i b_i Q d]$: sachant Q, les estimateurs de a_i et b_i sont :*

$$\hat{a}_i(Q) = \frac{1}{d}\sum_{j=1}^{d} q_j^t W_i q_j, \tag{4.10}$$

$$\hat{b}_i(Q) = \frac{1}{(p-d)}\left(\mathrm{tr}(W_i) - \sum_{j=1}^{d} q_j^t W_i q_j\right). \tag{4.11}$$

Modèle $[a_i b Q d]$: sachant Q, l'estimateur de a_i est donné par (4.10) et l'estimateur de b est :

$$\hat{b}(Q) = \frac{1}{(p-d)}\left(\mathrm{tr}(W) - \sum_{j=1}^{d} q_j^t W q_j\right). \tag{4.12}$$

Modèle $[a b_i Q d]$: sachant Q, l'estimateur de b_i est donné par (4.11) et l'estimateur de a est :

$$\hat{a}(Q) = \frac{1}{d}\sum_{j=1}^{d} q_j^t W q_j. \tag{4.13}$$

Démonstration. Modèle $[a_i b_i Q d]$: En partant de l'expression de la log-vraisemblance $l(\theta)$ du lemme 4.1.2, les dérivées partielles de $l(\theta)$ par rapport à a_i et b_i sont :

$$\begin{aligned} -2\frac{\partial l(\theta)}{\partial a_i} &= \frac{n_i d}{a_i} - \frac{n_i}{a_i^2}\sum_{j=1}^{d} q_j^t W_i q_j, \\ 2\frac{\partial l(\theta)}{\partial b_i} &= \frac{n_i(p-d)}{b_i} - \frac{n_i}{b_i^2}\left(\mathrm{tr}(W_i) - \sum_{j=1}^{d} q_j^t W_i q_j\right). \end{aligned}$$

Les conditions $\frac{\partial l(\theta)}{\partial a_i} = 0$ et $\frac{\partial l(\theta)}{\partial b_i} = 0$ donnent respectivement :

$$\begin{aligned}\hat{a}_i(Q) &= \frac{1}{d}\sum_{j=1}^{d} q_j^t W_i q_j, \\ \hat{b}_i(Q) &= \frac{1}{(p-d)}\left(\mathrm{tr}(W_i) - \sum_{j=1}^{d} q_j^t W_i q_j\right).\end{aligned}$$

Modèle $[a_i bQd]$: la dérivée partielle de $l(\theta)$ par rapport à b est :

$$-2\frac{\partial \log(L)}{\partial b} = \frac{n(p-d)}{b} - \frac{n}{b^2}\left(\mathrm{tr}(W) - \sum_{j=1}^{d} q_j^t W q_j\right),$$

et la condition $\frac{\partial l(\theta)}{\partial b} = 0$ implique que :

$$\hat{b}(Q) = \frac{1}{(p-d)}\left(\mathrm{tr}(W) - \sum_{j=1}^{d} q_j^t W q_j\right).$$

Modèle $[ab_iQd]$: la dérivée partielle de $l(\theta)$ par rapport à a est :

$$-2\frac{\partial \log(L)}{\partial a} = \frac{nd}{a} - \frac{n}{a^2}\sum_{j=1}^{d} q_j^t W q_j,$$

et la condition $\frac{\partial l(\theta)}{\partial a} = 0$ permet d'établir que :

$$\hat{a}(Q) = \frac{1}{d}\sum_{j=1}^{d} q_j^t W q_j,$$

et conclut ainsi la démonstration. □

Il est par exemple possible d'utiliser la procédure itérative suivante pour estimer les paramètres du modèle $[a_i b_i Qd]$:

(i) Initialisation : les d premières colonnes de $Q^{(0)}$ sont les vecteurs propres associés aux d plus grandes valeurs propres de W.

(ii) Jusqu'à convergence :

(a) estimation de $a_i^{(\ell)}$ et $b_i^{(\ell)}$:

$$\begin{aligned}a_i^{(\ell)} &= \hat{a}_i(Q^{(\ell-1)}), \\ b_i^{(\ell)} &= \hat{b}_i(Q^{(\ell-1)}).\end{aligned}$$

(b) estimation de $Q^{(\ell)}$: les d premières colonnes de $Q^{(0)}$ sont les vecteurs propres associés aux d plus grandes valeurs propres de $M(a_1^{(\ell)}, ..., a_k^{(\ell)}, b_1^{(\ell)}, ..., b_k^{(\ell)})$.

4.3.3 Estimateurs des modèles à matrices de covariance communes

Nous allons donner dans ce paragraphe les estimateurs du *maximum* de vraisemblance des paramètres des modèles à matrices de covariance communes. Les estimateurs des deux modèles présentés ici sont explicites.

Proposition 4.3.6. *Sous-espace $\mathbb{E}_i$: les d premières colonnes de Q sont les vecteurs propres associés aux d plus grandes valeurs propres de W.*

Démonstration. En partant de l'expression de la log-vraisemblance $l(\theta)$ des modèles à matrices de covariance communes donnée au lemme 4.1.3, on peut écrire le gradient de la fonction de Lagrange $\mathcal{L} = -2l(\theta) - \sum_{j=1}^{p} \omega_j(q_j^t q_j - 1)$ par rapport à q_j de la façon suivante :

$$\nabla_{q_j}\mathcal{L} = 2n(\frac{1}{a_j} - \frac{1}{b})Wq_j - 2\omega_j q_j,$$

où ω_j est le $j$ème multiplicateur de Lagrange. La relation $\nabla_{q_j}\mathcal{L} = 0$ implique que q_j est vecteur propre de W. Pour minimiser la quantité $-2l(\theta)$, il est nécessaire que les premières colonnes de Q soient les vecteurs propres associés aux d plus grandes valeurs propres de W (*cf.* expression de $-2l(\theta)$ donnée au lemme 4.1.3). □

Proposition 4.3.7. *Modèle $[a_j bQd]$: l'estimateur de a_j, $j = 1, ..., d$, est $\hat{a}_j = \lambda_j$ et l'estimateur de b est donné par (4.8).*
Modèle $[abQd]$: les estimateurs de a et b sont respectivement donnés par (4.7) et (4.8).

Démonstration. Modèle $[a_j bQd]$: en partant de l'expression de la log-vraisemblance $l(\theta)$ du lemme 4.1.3, les dérivées partielles de $l(\theta)$ par rapport à a_j et b s'écrivent respectivement :

$$-2\frac{\partial l(\theta)}{\partial a_j} = \frac{n}{a_j} - \frac{n}{a_j^2} q_j^t W q_j$$

et

$$-2\frac{\partial l(\theta)}{\partial b} = \frac{n(p-d)}{b} - \frac{n}{b^2}\sum_{j=d+1}^{p} q_j^t W q_j.$$

La condition $\frac{\partial l(\theta)}{\partial a_i} = 0$ implique que $\hat{a}_j = \lambda_j$. De même, la condition $\frac{\partial l(\theta)}{\partial b} = 0$ et la relation $\sum_{j=d+1}^{p} \lambda_j = \mathrm{tr}(W) - \sum_{j=1}^{d} \lambda_j$ donnent l'estimateur de b :

$$\hat{b} = \frac{1}{(p-d)}\left(\mathrm{tr}(W) - \sum_{j=1}^{d} \lambda_j\right).$$

Modèle $[abQd]$: la dérivée partielle de $l(\theta)$ par rapport à a est :

$$-2\frac{\partial \log(L)}{\partial a} = \frac{nd}{a} - \frac{n}{a^2}\sum_{j=1}^{d} q_j^t W q_j,$$

et la condition $\frac{\partial l(\theta)}{\partial a} = 0$ démontre que l'estimateur de a est :

$$\hat{a} = \frac{1}{d}\sum_{j=1}^{d} \lambda_j,$$

et conclut ainsi la démonstration. □

4.3.4 Considérations numériques

Nous allons montrer dans ce paragraphe que la re-paramétrisation du modèle gaussien, que nous avons proposée et dont l'estimation des paramètres a fait l'objet des paragraphes précédents, permet aux classifieurs HDDA et HDDC d'être à la fois stables numériquement et efficaces en temps de calcul. D'autre part, nous allons voir que, dans le cas où le nombre d'observations est inférieur à la dimension de l'espace, notre paramétrisation permet d'utiliser une astuce de calcul matriciel portant sur la détermination des vecteurs propres des matrices de covariance et l'algorithme de classification est alors extrêmement rapide.

Stabilité numérique

Au paragraphe 3.2.2, nous avions fait remarquer au lecteur que la fonction de coût K_i associées aux modèles de notre famille n'utilise pas la projection sur le sous-espace $\mathbb{E}_i^{\perp}$ et de ce fait ne requiert que la détermination des d_i premières colonnes de la matrice Q_i, $i = 1, .., k$. Au cours des paragraphes précédents, nous avons montré que les estimateurs du maximum de vraisemblance des d_i premières colonnes de la matrice Q_i sont, dans le cas des modèles à orientations libres, les vecteurs propres associés aux d_i plus grandes valeurs propres de la matrice de covariance empirique W_i. Par conséquent, la règle de décision des classifieurs HDDA et HDDC ne dépend pas des vecteurs propres associés aux plus petites valeurs propres de W_i dont la détermination est numériquement instable quand le nombre d'observations est petit devant le nombre de paramètres à estimer. Ainsi, les méthodes de classification HDDA et HDDC ne sont pas perturbées par le mauvais conditionnement ou la singularité des matrices de covariance empiriques des classes.

Réduction de la durée de calcul

Nous avons pu remarquer que l'estimation des paramètres des modèles de notre famille nécessite uniquement la détermination des d_i plus grandes valeurs propres ainsi que leur vecteur propre associé, et ce pour $i = 1, ..., k$. Cette remarque peut en particulier être mise à profit pour réduire la durée de

calcul. En effet, certains logiciels de statistique comportent des procédures spécifiques permettant de ne déterminer que les vecteurs propres associés aux plus grandes valeurs propres d'une matrice. Par exemple, la fonction *« eigs »* du logiciel Matlab permet cela et se base sur la méthode de Arnoldi [54]. L'économie de temps réalisée est bien sûr d'autant plus importante que la dimension d_i est petite devant p. Lors de nos expérimentations sur les données « visages », qui ont été présentées au chapitre 2 et dont la dimension de l'espace original est 1024, nous avons observé une diminution d'un facteur 60 du temps de calcul des vecteurs propres associés aux 4 plus grandes valeurs propres de la matrice de covariance empirique de chacune des classes avec la méthode efficace *« eigs »* par rapport à la méthode classique *« eig »*.

Cas où le nombre d'observations est inférieur à la dimension de l'espace

En outre, le fait de n'avoir qu'à déterminer les d_i plus grandes valeurs propres ainsi que leur vecteur propre associé se révèle être d'un intérêt crucial quand le nombre d'observations est plus petit que la dimension de l'espace original. Dans ce cas, il est préférable d'un point de vue numérique de calculer les valeurs propres et les vecteurs propres de la matrice $\Upsilon_i\Upsilon_i^t$ au lieu de calculer ceux de la matrice de covariance empirique $W_i = \Upsilon_i^t\Upsilon_i$, où Υ_i est la matrice $n_i \times p$ contenant les n_i observations de la $i$ème classe vivant dans $\mathbb{R}^p$ centrées. En effet, la matrice W_i étant de dimensions $p \times p$, la détermination des vecteurs propres associés aux d_i plus grandes valeurs propres de W_i est beaucoup plus longue et instable numériquement que la détermination des vecteurs propres associés aux d_i plus grandes valeurs propres de la matrice $\Upsilon_i\Upsilon_i^t$ si $n_i < p$. Soit v_{ij} le vecteur propre associé à la $j$ème plus grande valeur propre λ_{ij} de la matrice $\Upsilon_i\Upsilon_i^t$, nous avons alors, pour $j = 1, ..., d_i$:

$$\Upsilon_i\Upsilon_i^t v_{ij} = \lambda_{ij} v_{ij}.$$

En multipliant à gauche par Υ_i^t, nous obtenons :

$$\left(\Upsilon_i^t\Upsilon_i\right)\Upsilon_i^t v_{ij} = \lambda_{ij}\Upsilon_i^t v_{ij}.$$

Ainsi, le vecteur propre de la matrice $W_i = \Upsilon_i^t\Upsilon_i$ associé à la valeur propre λ_{ij} s'obtient à partir du vecteur propre v_{ij} de $\Upsilon_i\Upsilon_i^t$ associé à λ_{ij} en le multipliant à gauche par Υ_i^t. Typiquement, dans le cas des données « visages », où chacune des classes est représentée par 13 observations en dimension 1024 dans le jeu d'apprentissage, la détermination des vecteurs propres associés aux d_i plus grandes valeurs propres de $\Upsilon_i\Upsilon_i^t$ est 500 fois plus rapide que la détermination des vecteurs propres associés aux d_i plus grandes valeurs propres de W_i.

4.4 Estimation des paramètres discrets

Après l'estimation par *maximum* de vraisemblance des différents paramètres du mélange, il reste encore à estimer les dimensions intrinsèques des classes et le nombre de composantes du mélange.

Toutefois, l'estimation de ces paramètres discrets ne peut pas être faite en utilisant la méthode du *maximum* de vraisemblance. Dans ce paragraphe, nous allons donc traiter de l'estimation des dimensions intrinsèques d_i et du nombre de composantes du mélange k.

4.4.1 Estimation des dimensions intrinsèques d_i

La démarche adoptée dans ce mémoire fait l'hypothèse que les données de chaque groupes vivent dans des sous-espaces dont les dimensions intrinsèques d_i, $i = 1, ..., k$, sont plus petites que la dimension de l'espace total. Cependant, l'estimation de la dimension intrinsèque d'un groupe de données ne possède pas de solution explicite. Nous allons proposer ici une stratégie d'estimation des paramètres $d_1, ..., d_k$.

Notre approche de l'estimation des d_i

Le problème dont il est question ici est similaire au problème du choix du nombre de composantes à retenir en analyse en composantes principales (voir [79, chap. 8]). Par conséquent, notre approche s'appuiera sur les valeurs propres de la matrice de covariance de chacune des classes sur lesquelles sont basés des critères théoriques et empiriques de détermination de la dimension intrinsèque. Nous avons détaillé ces critères au paragraphe 2.4.2 de ce mémoire et notre choix s'est porté sur l'utilisation d'un critère empirique. Ainsi, dans le cas où les dimensions d_i sont supposées différentes entre les k classes, nous proposons d'utiliser le scree-test de Cattell [17] pour estimer la dimension intrinsèque du sous-espace de chaque classe. La méthode du scree-test de Cattell permet de déterminer un coude dans l'éboulis des valeurs propres de la matrice de covariance de chaque classe en retenant la dimension pour laquelle les différences entre valeurs propres consécutives sont toutes plus petites qu'un certain seuil s_i. Par souci de simplicité, nous supposons que les seuils s_i, $i = 1, ..., k$, sont égaux à s et l'estimation des k dimensions intrinsèques se résume alors à l'estimation de l'unique paramètre s. Dans le cas où les dimensions d_i sont supposées égales à d entre les k classes, l'estimation des dimensions intrinsèques se réduit de même à l'estimation d'un unique paramètre, le paramètre d.

Estimation de s ou d dans le cadre supervisé

Dans le cadre supervisé, il est possible d'estimer s, ou d, par validation croisée sur le jeu d'apprentissage (voir annexe A) puisque l'on connaît les appartenances aux classes des observations du jeu d'apprentissage. L'estimateur de s, ou d, sera la valeur pour laquelle le taux de classification correcte par validation croisée du jeu d'apprentissage sera le meilleur.

Estimation de s ou d dans le cadre non supervisé

Dans le cadre non supervisé, il n'est en revanche pas possible de faire de la validation croisée sur les données disponibles puisque seules les observations x_j, $j = 1, ..., n$, sont disponibles. Le problème de l'estimation du paramètre s, ou d, peut être vu comme un problème de choix de modèle. En effet,

le paramètre s, ou d, représentant les k dimensions intrinsèques d_i, contrôle la complexité du modèle et l'estimation de ce paramètre revient à choisir entre plusieurs modèles plus ou moins complexes. Il existe de nombreux critères permettant de choisir entre différents modèles. Dans notre cadre d'étude, les critères bayésiens qui choisissent le modèle pour lequel la probabilité des observations est la plus grande sont largement utilisés. Etant donné que ce problème de choix se pose aussi entre les divers modèles de notre famille, les différents critères seront présentés au paragraphe 4.5. Nous dirons ici juste que, dans le cadre non supervisé, l'estimation de s, ou d, sera faite par minimisation du critère BIC [81] :

$$\hat{s} = \underset{s\in]0,1[}{\operatorname{argmin}} BIC(s) \quad \text{ou} \quad \hat{d} = \underset{d=1,\ldots,p-1}{\operatorname{argmin}} BIC(d).$$

4.4.2 Estimation du nombre de classes k

Le dernier paramètre permettant la modélisation des données en vue de leur classification n'est pas le moins important. Il est en effet très important de connaître en combien de groupes doivent être organisées les données. Dans le cadre de l'analyse discriminante, le problème de l'estimation de k ne se pose pas car le nombre de classes est supposé être connu. En revanche, dans le cadre de la classification automatique, les données disponibles pour l'estimation des paramètres du mélange n'étant pas complètes, il n'est pas possible d'en déduire directement une estimation de k. En dehors des cas bien particuliers où l'on sait par avance en combien de classes on souhaite organiser les données (par exemple, mâles et femelles, ...), nous allons tout de même devoir estimer k à partir des données disponibles. L'estimation de ce paramètre peut également être vue comme un problème de choix de modèle : il faut choisir entre modéliser les données avec 1 classe, 2 classes, ..., n classes. Nous dirons juste que l'estimation du nombre de classes dans le cadre non supervisé sera faite par minimisation du critère BIC [81] :

$$\hat{k} = \underset{k=1,\ldots,n}{\operatorname{argmin}} BIC(k).$$

4.5 Choix du modèle

Nous avons présenté dans ce mémoire un ensemble de modèles adaptés aux données de grande dimension, plus ou moins parcimonieux, et traité de l'estimation de leurs paramètres. Il se pose alors le problème de choisir quel modèle utiliser pour modéliser et classer au mieux un jeu de données spécifique. Le but est donc de trouver un modèle parmi l'ensemble des modèles disponibles dont on puisse estimer correctement les paramètres avec les observations disponibles et qui soit suffisamment flexible pour modéliser correctement les données. Par conséquent, il s'agit de trouver le modèle ayant le juste degré de complexité, qui permette d'obtenir le meilleur compromis entre le biais et la variance des estimateurs de ses paramètres. Nous allons donc voir dans ce paragraphe quelles techniques sont à notre disposition pour choisir le « meilleur » modèle parmi les 28 à notre disposition et ce dans les cadres de l'analyse discriminante et de la classification automatique.

4.5.1 Choix du modèle dans le cadre supervisé

En classification supervisée, l'approche la plus simple est de choisir le modèle le mieux adapté aux données par validation croisée sur le jeu d'apprentissage. Le modèle retenu pour classer ensuite de nouvelles observations sera le modèle qui aura fourni le classifieur le plus performant sur les données complètes d'apprentissage. Il est également possible de sélectionner ce « meilleur » modèle sur un critère probabiliste qui prenne à la fois en compte l'objectif de discrimination et celui de modélisation probabiliste. Un tel critère a été proposé récemment par Bouchard et Celeux [14] et est nommé *Bayesian Entropy Criterion* (BEC). Enfin, il est aussi possible de faire comme si les données d'apprentissage n'étaient pas complètes et de choisir le modèle le plus adapté aux données sur un critère qui prenne uniquement en compte l'aspect de modélisation probabiliste. On pourra alors utiliser une des techniques proposées dans le cadre non supervisé et qui seront présentées au paragraphe suivant. Toutefois, sur la base de notre expérience, nous conseillons d'utiliser la technique de ré-échantillonage de la validation croisée qui donne le plus souvent des résultats satisfaisants lors du classement des données de validation.

4.5.2 Choix du modèle dans le cadre non supervisé

Comme nous l'avons déjà dit brièvement dans ce chapitre, de nombreux critères ont été proposés pour choisir entre différents modèles dans le cadre non-supervisé. Parmi ces modèles, les critères bayésiens qui choisissent le modèle pour lequel la probabilité des observations est la plus grande sont largement utilisés dans notre cadre d'étude. Le critère *Bayesian Information Criterion* (BIC) [81] est certainement l'un des plus connus et des plus utilisés. Le critère BIC est constitué de deux termes : le terme de vraisemblance qui favorise la sélection d'un modèle complexe et un terme de pénalité, fonction croissante du nombre de paramètres, qui favorise la sélection d'un modèle parcimonieux. Le principe du critère BIC est donc de choisir le modèle qui minimise la quantité suivante :

$$BIC(m) = -2\log(L) + \nu(m)\log(n),$$

où $\nu(m)$ est le nombre de paramètres du modèle m, L est la vraisemblance et n est le nombre d'observations. Pour les modèles adaptés aux données de grande dimension, présentés au chapitre précédent, le nombre ν est donné dans le tableau 3.1. En pratique, le critère BIC est à préférer au critère *Akaike Information Criterion* (AIC) qui ne pénalise pas suffisamment la complexité des modèles et a, de fait, tendance à surestimer le nombre de paramètres à estimer. Dans les expérimentations qui seront proposées dans le prochain chapitre, nous utiliserons le critère BIC pour le choix du modèle. Notons qu'il est également possible d'utiliser un critère probabiliste qui prenne en compte l'objectif de discrimination si le *clustering* est effectué dans cet objectif. Le critère *Integrated Classification Likelihood* (ICL) proposé par Biernacki *et al.* dans [9] réalise cela en maximisant la vraisemblance classifiante intégrée.

Chapitre 5

Validation expérimentale

Dans ce chapitre, nous allons mettre en œuvre et évaluer l'efficacité des méthodes de classification HDDA et HDDC basées sur le modèles gaussiens introduits au chapitre 3 et dont l'estimation des paramètres a fait l'objet du chapitre 4. Nous mettrons tout d'abord en œuvre, au paragraphe 5.1, les méthodes supervisées et non supervisées HDDA et HDDC sur des données réelles et simulées afin de permettre au lecteur de mieux appréhender le fonctionnement de ces deux méthodes. Le paragraphe 5.2 sera consacré à la validation expérimentale de l'analyse discriminante de grande dimension (HDDA). Au cours de ce paragraphe, nous verrons que l'HDDA fournit des résultats très satisfaisants en grande dimension et avec des jeux d'apprentissage de petites tailles. La validation expérimentale dans la classification automatique des données de grande dimension (HDDC) fera l'objet du paragraphe 5.3. Nous pourrons y observer également le bon comportement de l'HDDC en grande dimension. Dans les paragraphes 5.2 et 5.3, les expérimentations seront menées sur des données simulées et réelles. Une comparaison de grande ampleur du modèle $[a_{ij}b_iQ_id_i]$ et de ses sous-modèles avec les modèles gaussiens classiques sera faite au chapitre suivant dans le cadre de l'application à la reconnaissance d'objets dans des images.

5.1 Mise en œuvre des méthodes HDDA et HDDC

Dans ce premier paragraphe, nous allons mettre en œuvre les méthodes HDDA et HDDC sur des données réelles de grande dimension afin de donner au lecteur une première idée du fonctionnement et des performances de ces deux méthodes.

5.1.1 Mise en œuvre de l'HDDA sur données simulées

Nous proposons tout d'abord d'appliquer l'HDDA à un jeu de données simulées afin d'observer la forme de la règle de décision en fonction des différents modèles de mélange gaussien introduits au chapitre 3. Nous comparerons également la forme de la règle de décision de l'HDDA avec celles associées aux méthodes classiques de discrimination qui ont été présentées au paragraphe 2.2. Afin

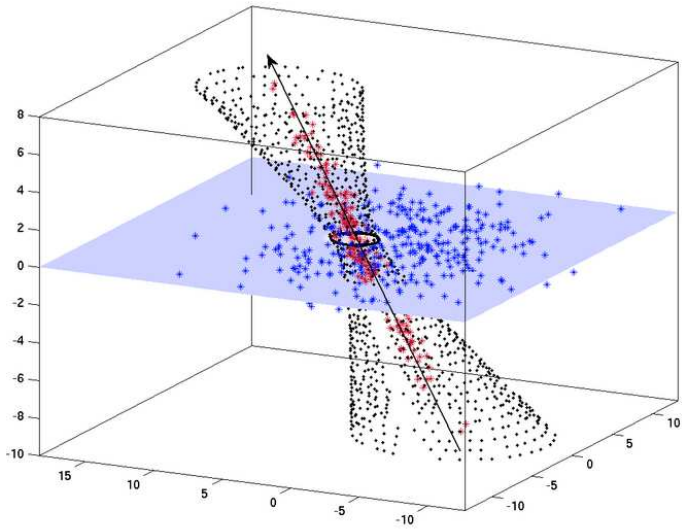


FIG. 5.1 – Données simulées dans $\mathbb{R}^3$ et composées de deux classes de dimensions intrinsèques respectives 2 (classe bleue) et 1 (classe rouge). La frontière de décision dessinée en pointillés est celle de l'HDDA avec le modèle $[a_i b_i Q_i d_i]$.

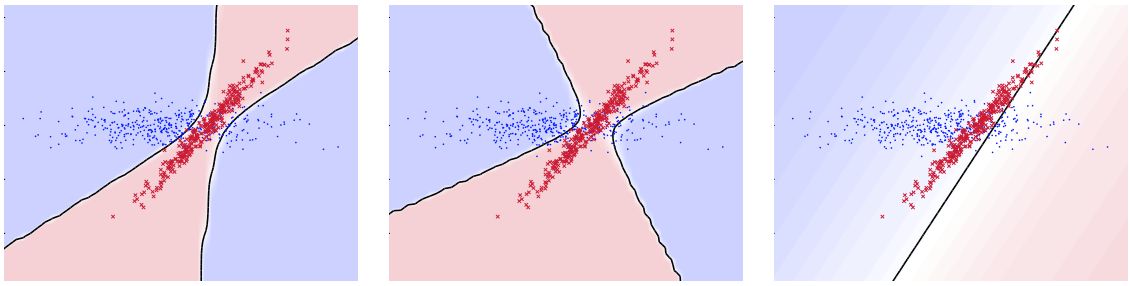
de faciliter la visualisation des règles de décision associées aux différentes méthodes, nous utiliserons dans ce paragraphe des données dans $\mathbb{R}^3$ dont les dimensions intrinsèques des classes seront inférieures à la dimension de l'espace original.

Données et protocole d'étude

Nous avons donc simulé pour les besoins de cette expérimentation 1500 données réparties en 2 groupes dans $\mathbb{R}^3$ selon le modèle $[a_i b_i Q_i d_i]$. Les paramètres de la première classe, qui sera représentée en bleue par la suite, sont : $\pi_1 = 0.6$, $\mu_1 = (0, 0, 0)$, $a_1 = 20$, $b_1 = 3$ et $d_1 = 2$. Les paramètres de la seconde classe, qui sera quant à elle représentée en rouge, sont : $\pi_2 = 0.4$, $\mu_2 = (0, 3, 0)$, $a_2 = 30$, $b_2 = 1$, $d_2 = 1$. Enfin, les matrices d'orientations des classes sont :

$$Q_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \text{et} \quad Q_2 = \begin{pmatrix} 0.6 & 0.4 & 0 \\ 0.4 & 1 & 0.45 \\ 0 & 0.45 & 0.55 \end{pmatrix}.$$

Ainsi, le sous-espace de la première classe est un plan parallèle aux axes de l'espace original alors que le sous-espace de la seconde classe est une droite qui n'est parallèle à aucun des axes originaux. La figure 5.1 montre les données qui ont été simulées pour cette expérimentation. Le plan représenté en bleu et l'axe oblique sont respectivement les sous-espaces spécifiques de la première et de la seconde classe. Nous avons également observé, sur ce jeu de données, les frontières de décision générées par les méthodes d'analyse discriminante basées sur les modèles gaussiens full-GMM (QDA), com-



$[a_{ij}b_iQ_id_i]$ $[a_{ij}bQ_id_i]$ $[a_jbQd]$

FIG. 5.2 – Frontières de décision de l'HDDA avec les modèles $[a_{ij}b_iQ_id_i]$, $[a_{ij}bQ_id_i]$ et $[a_jbQd]$ sur les données simulées. Les données sont projetées sur deux des axes originaux pour la représentation uniquement.

GMM (LDA), diag-GMM et sphe-GMM. Nous avons également utilisé la méthode discriminative SVM assortie du noyau classique RBF, *i.e.* $K(x, x') = \exp(-\gamma\|x - x'\|^2)$ avec $\gamma = 1$.

Résultats expérimentaux

Tout d'abord, la figure 5.1 montre la frontière de décision, dessinée en pointillés noirs, générée par l'HDDA dans l'espace original à 3 dimensions. On observe que le modèle $[a_{ij}b_iQ_id_i]$ de l'HDDA a correctement identifié le sous-espace spécifique de chacune des 2 classes et engendre une séparation quadratique entre elles. La figure 5.2 présente les frontières de décision de l'HDDA associées aux modèles $[a_{ij}b_iQ_id_i]$, $[a_{ij}bQ_id_i]$ et $[a_jbQd]$ sur les données simulées. Les données ont été projetées sur le premier et le troisième axe de l'espace original pour la représentation uniquement. Pour le modèle $[a_{ij}b_iQ_id_i]$, on retrouve la séparation quadratique entre les classes observée sur la figure 5.1. On remarque également que la règle de décision associée à ce modèle donne l'avantage à la première classe (en bleue sur la figure) car sa dimension intrinsèque est égale à 2. On remarque en outre que l'HDDA a parfaitement identifié l'orientation de la seconde classe qui n'est pas celle des axes originaux. On peut également observer que la règle de décision associée au modèle $[a_{ij}bQ_id_i]$ est également quadratique mais favorise moins la première classe que précédemment car ce modèle fait l'hypothèse que le bruit est commun aux classes. Enfin, la frontière de décision dessinée par l'HDDA associée au modèle $[a_jbQd]$ est linéaire et cela confirme l'analogie que nous avions fait entre ce modèle et celui de l'analyse discriminante linéaire (LDA). Nous avons choisi de ne représenter ici que les frontières de décision associées à ces trois modèles de notre famille car les autres modèles ont des comportements similaires dans cet espace de faible dimension. La figure 5.3 présente quant à elle les frontières de décision des méthodes classiques et permet de les comparer à celle de l'HDDA. On peut constater que le modèle $[a_{ij}b_iQ_id_i]$ de l'HDDA fournit un classifieur très proche de QDA, qui est basé sur le modèle full-GMM, et qui est très proche du classifieur optimal dans cet espace de faible dimension. La méthode LDA, basée sur le modèle com-GMM, fournit une règle de décision linéaire qui ne parvient naturellement pas à séparer les deux classes. L'hypothèse d'égalité des matrices

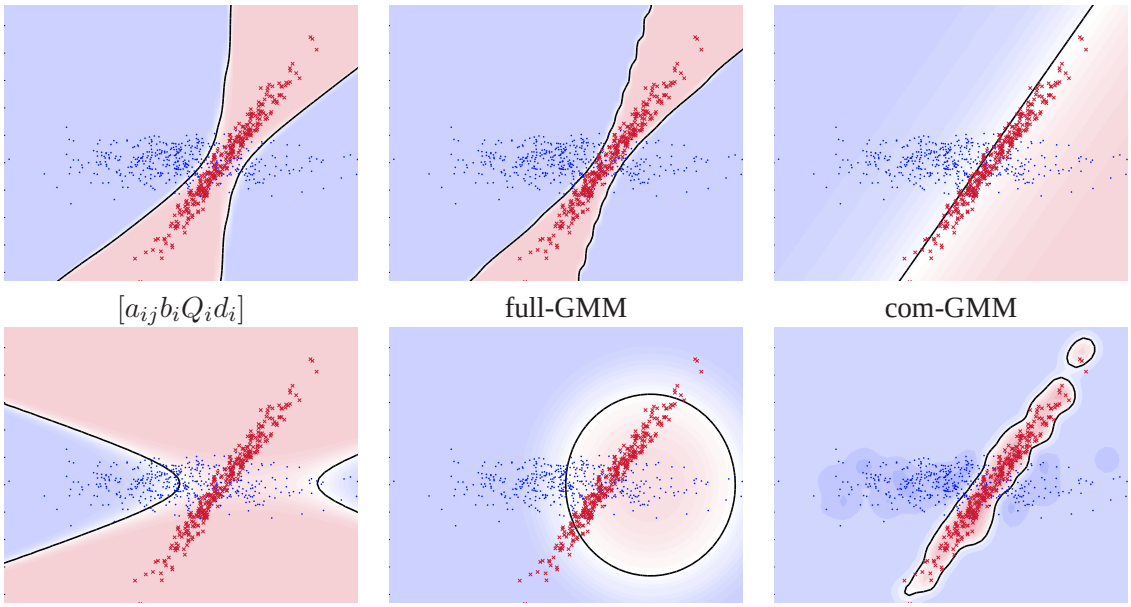


FIG. 5.3 – Frontières de décision de l'HDDA et des méthodes d'analyse discriminante classiques sur les données simulées. Les données sont projetées sur deux des axes originaux pour la représentation uniquement.

de covariance faite par LDA explique l'inaptitude de ce classifieur à modéliser des données complexes et ce même dans un espace de faible dimension. Il est d'autre part particulièrement intéressant d'observer le comportement du classifieur associé au modèle parcimonieux diag-GMM qui suppose l'indépendance conditionnelle des variables. Cette hypothèse empêche le classifieur à déterminer la bonne orientation de chacune des classes et le contraint à fournir une règle de décision très éloignée de la vérité. La frontière de décision que l'on observe pour ce classifieur est orientée selon les axes originaux en accord avec son hypothèse initiale. Le classifieur associé au modèle gaussien sphe-GMM, fidèle à son hypothèse de sphéricité des classes dans le système des axes originaux, fournit une règle de décision sphérique autour de la seconde classe. Enfin, la méthode discriminative SVM produit une frontière de décision extrêmement complexe et difficilement interprétable, au contraire de l'HDDA, mais qui s'avère généralement très performante sous réserve que les données d'apprentissage soient bien représentatives des populations étudiées.

5.1.2 Mise en œuvre de l'HDDA sur les données « visages »

Nous proposons à présent d'appliquer l'HDDA à un problème réel de reconnaissance de visages où les données sont de très grande dimension. Le fait que les données utilisées dans ce paragraphe soient des images, nous permettra de visualiser facilement l'action de l'HDDA.

FIG. 5.4 – Quelques exemples de la base de données « visages » qui contient 13 sujets.

Sujet	1	2	3	4	5	6
μ_i						
a_i	36000	50900	37300	49000	24300	50600
b_i	6.2	20.5	14.7	3.1	2.8	16.3
d_i	3	5	4	4	4	3

FIG. 5.5 – Valeur des paramètres des classes associées aux 6 premiers sujets.

Données et protocole d'étude

La base de données « visages »[1] a été collectée par le *Advanced Multimedia Processing Laboratory* et est composée de 975 images des visages de 13 sujets. L'intérêt de cette base est qu'il existe une variation d'expressions faciales importante parmi les images d'un même sujet. Les images ont été acquises dans les mêmes conditions d'illumination et ont été recadrées en se basant sur la position des yeux des sujets. Chaque observation de la base est une image en échelle de gris de taille 32×32 pixels et représentée par un vecteur de dimension 1024. Les jeux d'apprentissage et de validation comprennent respectivement 10 et 65 images par sujet. La figure 5.4 présente quelques images tirées de la base de données « visages ». Notre but est donc de faire de la reconnaissance de visages, *i.e.* reconnaître le sujet présent sur les images de validation en se basant sur la connaissance de 10 images de chaque sujet. Pour discriminer ces données, nous avons choisi d'utiliser le modèle $[a_i b_i Q_i d_i]$ de l'HDDA et le seuil s de la méthode d'estimation des dimensions a été déterminé par validation croisée sur le jeu d'apprentissage. La méthode d'analyse discriminante HDDA étant programmée en langage Matlab, le temps de calcul nécessaire à l'apprentissage du classifieur est de 22 secondes sur un ordinateur récent (Pentium IV, 3Ghz).

FIG. 5.6 – Images reconstruites correspondantes aux points le long des axes spécifiques du sujet 1, *i.e.* les axes correspondants aux 3 premières colonnes de la matrice Q_1.

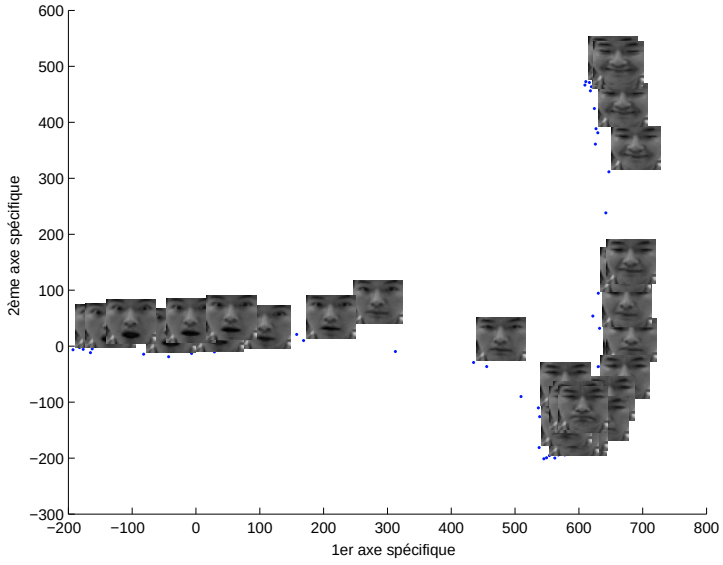


FIG. 5.7 – Projection des données de validation sur les deux premiers axes spécifiques du sujet 1.

Résultats expérimentaux

Tout d'abord, notons que cette base de données, qui est certes en très grande dimension, est relativement facile à discriminer puisque notre méthode obtient un taux de classification correcte égal à 1 sur le jeu de validation. La figure 5.5 donne les valeurs des paramètres μ_i, a_i, b_i et d_i pour les 6 premières classes. Il est tout particulièrement intéressant de remarquer que l'HDDA estime que la dimension intrinsèque moyenne des sous-espaces où vivent les données est proche de 4. Ce chiffre est particulièrement petit quand on le compare à la dimension originale des données qui est 1024. Par conséquent, le modèle gaussien appris par l'HDDA est extrêmement parcimonieux car sa complexité est déterminée essentiellement par les dimensions d_i. Il est assez naturel de trouver des dimensions intrinsèques de cet ordre de grandeur car le nombre de degré de liberté d'un visage n'est pas très élevé. Notons également que le rapport entre les valeurs des paramètres a_i et b_i est très grand et cela traduit le fait que les données ne sont que très peu bruitées. Intéressons nous à présent à l'interprétation du sens physique de chacun des axes spécifiques. Dans ce but, nous avons reconstruit les images associées aux points de chacun des axes situés autour de la moyenne. La figure 5.6 présente ces images reconstruites pour les 3 axes spécifiques de la classe du sujet 1. On peut observer que chacun des 3 axes représente une expression faciale : l'axe 1 est associé à l'étonnement, l'axe 2 est associé à la gaieté et l'axe 3 est associé à la tristesse. Cela correspond en effet aux expressions faciales que chacun des 13 sujets devaient mimer durant l'enregistrement des images. La figure 5.7 présente la projection des images de la base de validation sur les deux premiers axes spécifiques de la classe du sujet 1. On observe également qu'en se déplaçant vers la gauche de l'image et parallèlement au premier axe, le visage du sujet 1 traduit l'étonnement. De même, en se déplaçant parallèlement au second axe spécifique de la classe du sujet 1 et vers le haut de l'image, le visage du sujet apparaît comme de plus en plus joyeux. On peut remarquer enfin un certain « écrasement » des données en bas à droite de la figure qui correspond à la projection des données évoluant le long du troisième axe spécifique qui n'est pas représenté sur cette figure.

5.1.3 Mise en œuvre de l'HDDC sur les données « crabes »

Nous allons à présent mettre en œuvre la méthode de classification non supervisée HDDC associée au modèle $[a_{ij}b_iQ_id_i]$ et à ses sous-modèles. De la même façon que dans le cas supervisé, nous allons considérer la classification de données issues du monde réel. L'objectif de cette expérimentation est donc d'organiser en k groupes homogènes les données d'étude.

Données et protocole

Les données « crabes », utilisées dans cette mise en pratique de l'algorithme de type EM qu'est l'HDDC, sont composées de 5 mesures faites sur 200 individus équi-répartis dans 4 classes : les crabes mâles et femelles à carapace orange, les crabes mâles et femelles à carapace bleue. Pour chacun

[1] disponible à l'adresse *http ://amp.ece.cmu.edu/downloads.htm*.

des sujets, 5 variables ont été observées : largeur de la lèvre frontale, largeur arrière, longueur de la carapace, largeur maximale de la carapace et profondeur du corps de l'animal. Toutes ces observations sont mesurées en millimètres. Les données que nous considérons ici ne sont donc pas, à proprement parler, des données de grande dimension mais présentent l'intérêt que les dimensions intrinsèques des sous-espaces spécifiques des groupes sont égales à 1. Cette spécificité des données permettra donc une visualisation aisée de la recherche des sous-espaces des classes par l'HDDC. Le seuil s du *scree-test* de Cattell permettant de déterminer dans l'HDDC les dimensions intrinsèques des groupes a été choisi grâce au critère BIC et l'algorithme HDDC a été initialisé de façon aléatoire.

Résultats expérimentaux

La figure 5.8 montre les 12 étapes de l'algorithme d'estimation des paramètres du modèle $[a_{ij}b_iQ_id_i]$ sur les données « crabes ». Les données sont projetées sur les deux premiers axes principaux pour la visualisation uniquement. Nous rappelons que l'HDDC, comme l'HDDA, ne réduit jamais la dimension des données mais le modèle gaussien sous-jacent tient compte du fait que la dimension intrinsèque des données de chaque classe est plus petite que p. Les sous-espaces spécifiques des composantes du mélange sont représentés sur la figure par des lignes bleues et les moyennes des classes floues sont symbolisées par des disques de couleurs. On peut observer tout d'abord que le seuil s sélectionné par le critère BIC a conduit à estimer les dimensions intrinsèques des classes comme étant égales à 1. A la première étape, on observe que les données sont encore réparties de façon quasi-aléatoire. Au fur et à mesure des itérations, on découvre que l'estimation des moyennes et des sous-espaces des classes s'affine jusqu'à convergence. A la 12ème et dernière étape, le sous-espace de chacune des classes apparaît être parfaitement estimé. Etant donné que nous avons la connaissance des labels réels des observations, nous avons pu vérifier que le *clustering* fournit par l'HDDC est très proche de la réalité. Le taux de classification correcte est en effet proche de 0.95. Nous reviendrons au paragraphe 5.3.4 sur la performance de l'HDDC sur ces données et nous la comparerons aux résultats obtenus avec une technique de sélection de variables dans le cadre du modèle de mélange gaussien.

5.2 Validation dans le cadre supervisé

Nous étudierons tout d'abord l'influence des différents facteurs étant à l'origine du fléau de la dimension en classification sur les performances de l'HDDA associée aux différents modèles proposés au chapitre 3. Ces différentes expérimentations nous permettrons en outre de comparer l'HDDA aux méthodes de référence en analyse discriminante. Nous mènerons ensuite une étude comparative sur données réelles entre l'HDDA et les principales méthodes d'analyse discriminante. Les méthodes auxquelles nous comparerons l'HDDA sont l'analyse discriminante quadratique (QDA), l'analyse discriminante linéaire (LDA), l'analyse discriminante à décomposition spectrale (EDDA) et la combinaison d'une réduction de dimension globale par analyse en composantes principales et de LDA (PCA+LDA). Pour EDDA [7], qui est la méthode d'analyse discriminante basée sur les modèles par-

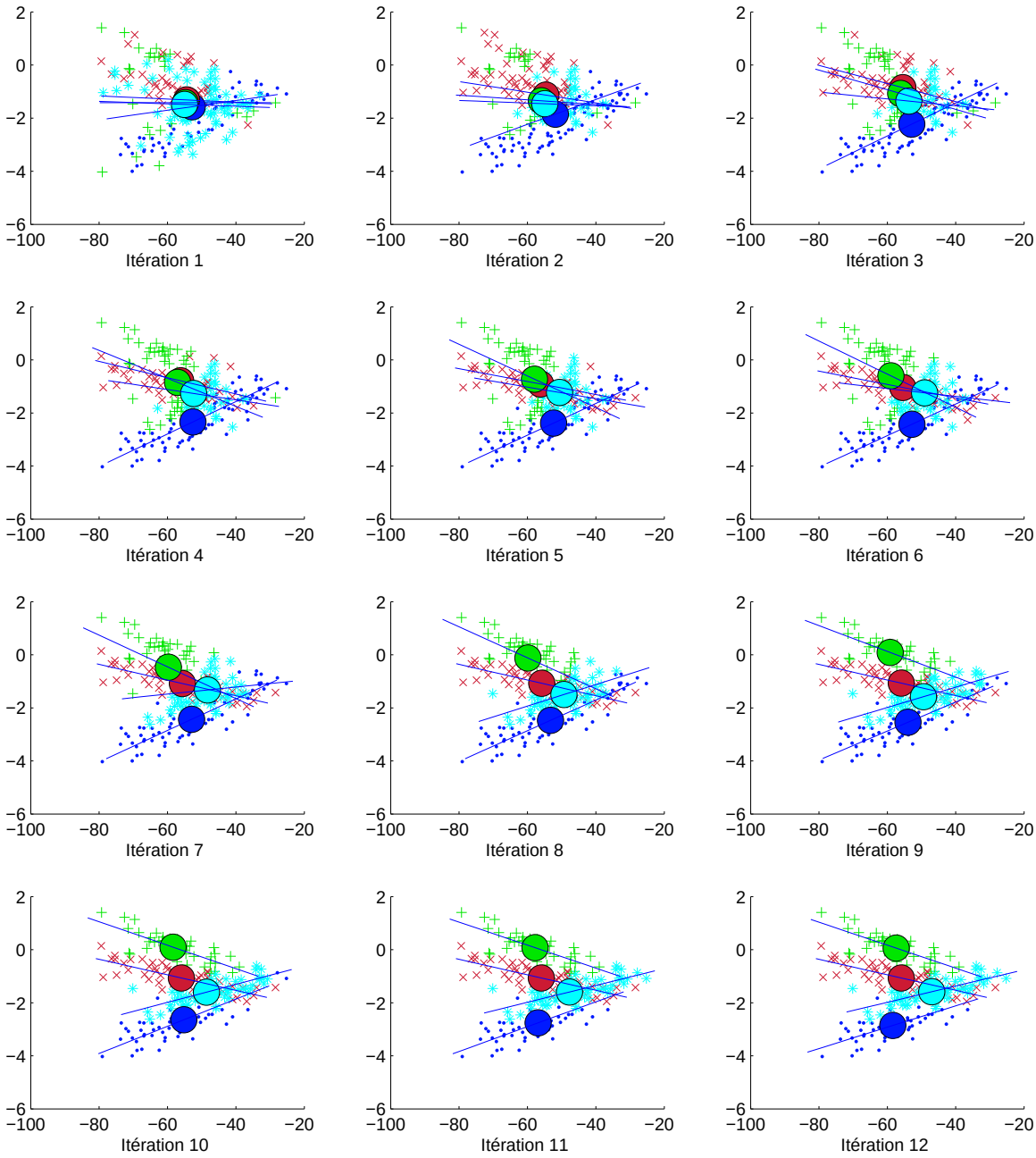


FIG. 5.8 – Les étapes de l'algorithme HDDC, de type EM, sur le jeu de données « crabes » et les sous-espaces spécifiques des composantes du mélange (représentés par les lignes bleues).

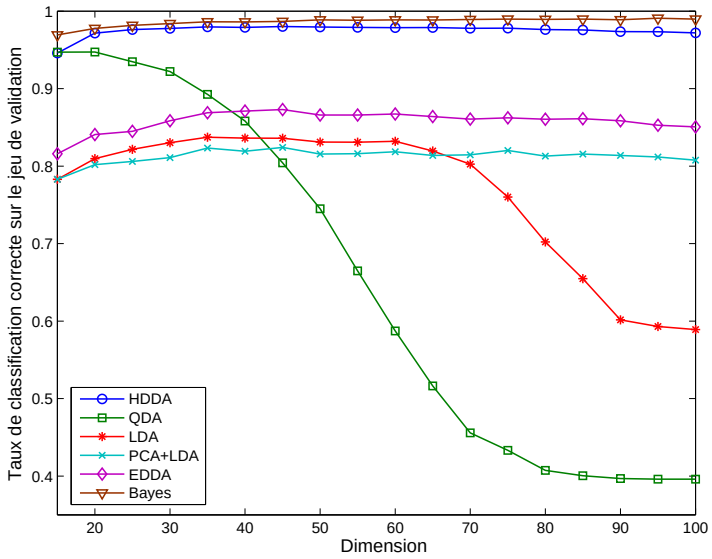


FIG. 5.9 – Influence de la dimension sur les résultats de classification obtenus avec l'HDDA et les méthodes classiques sur données simulées (voir le texte pour les détails de la simulation des données).

cimonieux de Celeux et Govaert [21], nous utiliserons le sous-modèle $[\lambda_k B_k]$ qui semble être le modèle le plus adapté pour ces données. Notons qu'une régularisation de type *ridge* a été nécessaire afin d'inverser les matrices de covariance pour les méthodes QDA, LDA et EDDA. Ces expérimentations, menées sur données simulées et réelles, nous permettrons de mettre en évidence les principales caractéristiques de l'HDDA.

5.2.1 Influence de la dimension

Nous avons mis en évidence au paragraphe 2.4.1 que la dimension de l'espace d'origine est la cause principale du mauvais comportement des méthodes de classification des données de grande dimension. Nous allons donc étudier l'influence de la dimension sur les résultats de classification obtenus avec l'HDDA et les méthodes classiques étudiées.

Données et protocole

Pour cette expérimentation, nous avons simulé des données réparties en 3 groupes dans $\mathbb{R}^p$, $p = 15, ..., 100$, selon le modèle $[a_i b Q_i d_i]$ avec les paramètres suivants : $\{\pi_1, \pi_2, \pi_3\} = \{0.4, 0.3, 0.3\}$, $\{a_1, a_2, a_3\} = \{150, 75, 50\}$, $b = 10$, $\{d_1, d_2, d_3\} = \{2, 5, 10\}$, avec des moyennes proches ($\|\mu_i - \mu_\ell\| \simeq 10$, pour tout $i, \ell \in \{1, 2, 3\}$ tels que $i \neq \ell$) et des matrices d'orientation Q_i aléatoires. Les matrices d'orientation Q_i ont été obtenues en effectuant la décomposition QR d'une matrice symétrique aléatoire. Les jeux d'apprentissage et de test sont respectivement composés de 250 et

1000 observations. La performance de chacune des méthodes étudiées est mesurée par le taux moyen de classification correcte calculé sur 50 répétitions. En outre, les données étant simulées, il a été possible de calculer le taux optimal de classification correcte obtenu par le classifieur de Bayes basé directement sur les densités réelles. Le nombre d'axes retenus pour la méthode PCA+LDA est égal à 15 pour cette étude et correspond à un coude dans l'ébouli des valeurs propres de la matrice de variance totale. Enfin, nous avons choisi d'utiliser le modèle gaussien $[a_i b_i Q_i d_i]$ dans l'HDDA car il nous semble réaliser un bon compromis entre complexité et parcimonie.

Résultats expérimentaux

La figure 5.9 met en évidence l'influence de la dimension sur les résultats de classification obtenus avec l'HDDA et les méthodes classiques. Tout d'abord, cette figure montre que la dimension des données n'a pas d'influence sur la performance de la méthode HDDA associée au modèle $[a_i b_i Q_i d_i]$. En effet, le taux de classification correcte est constant et égal à 97% de la dimension 20 à la dimension 100. Il est intéressant de remarquer que dans les premières dimensions, *i.e.* $d = 15, 20, 25$, le taux de classification correcte de HDDA augmente. Cela valide l'idée qu'il est plus aisé de discriminer dans des espaces de grande dimension. On retrouve d'ailleurs ce comportement pour le classifieur de Bayes (construit à partir des densités réelles) dont le taux de classification augmente légèrement au fur et à mesure que la dimension des données augmente. Remarquons aussi que la performance de l'HDDA est très proche du classifieur optimal de Bayes et que l'HDDA fournit des résultats similaires à ceux de QDA en dimension 15. L'analyse discriminante quadratique (QDA), qui est connue pour être très sensible à la dimension, voit en effet ses performances être sévèrement affectées dès que la dimension augmente. Le classifieur linéaire LDA est connu pour être moins sensible à la dimension des données et l'on remarque en effet que sa performance n'est affectée que pour des dimensions supérieures à 60. En revanche, l'hypothèse que les matrices de covariance des classes sont égales semble trop restrictive pour ces données car les résultats montrent que LDA n'arrive pas à discriminer correctement les classes, et ce même dans des espaces de petite dimension. L'étape de réduction de dimension de PCA+LDA permet au classifieur linéaire d'avoir des résultats constants en fonction de la dimension mais sans donner lieu à une augmentation des performances de LDA. La réduction de dimension est donc bien une technique permettant aux classifieurs d'être appliqués à des données de grande dimension mais qui n'améliore pas les résultats de classification. Enfin, le modèle $[\lambda_k B_k]$ de l'EDDA ne semble pas être sensible à la dimension des données mais est beaucoup moins performant que l'HDDA. Cela est certainement dû au fait que le modèle $[\lambda_k B_k]$ est un modèle très parcimonieux et il n'est donc pas capable de modéliser correctement de telles données. Pour récapituler, l'HDDA s'est révélée ne pas être sensible à la dimension des données en fournissant de très bons résultats aussi bien en faible dimension qu'en grande dimension.

FIG. 5.10 – Quelques caractères manuscrits issus du jeu de données USPS.

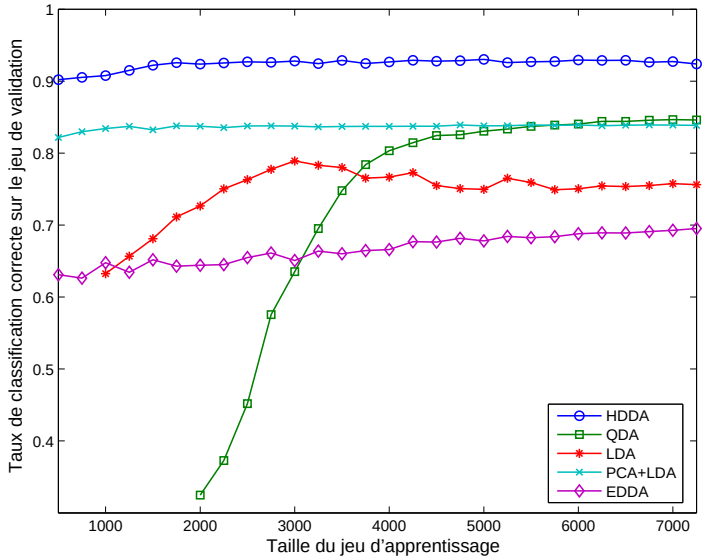


FIG. 5.11 – Influence de la taille du jeu d'apprentissage sur les résultats de classification obtenus avec l'HDDA et les méthodes classiques sur données réelles (données USPS).

5.2.2 Influence de la taille de l'échantillon

Nous avons également vu au paragraphe 2.4.1 que si la taille du jeu d'apprentissage est petite devant le nombre de paramètres à estimer alors les méthodes d'analyse discriminante classiques rencontrent des difficultés. Ces difficultés sont liées au problème posé par la nécessité d'inverser les matrices de covariance des classes. Nous allons étudier dans ce paragraphe le comportement de l'HDDA vis à vis de la taille de l'échantillon.

Données et protocole

Pour cette expérimentation, nous avons choisi d'utiliser le jeu de données réelles USPS[2] qui est constitué de 9298 caractères manuscrits recueillis par le *United States Postal Service* de la ville de Buffalo. Les données sont divisées en 10 classes correspondant aux caractères 0, 1, ..., 9. Chaque caractère de la base est une image en échelle de gris de taille 16×16 pixels et représentée par un vecteur de 256 dimensions. Les jeux d'apprentissage et de test sont respectivement composés de 7291

[2] disponible à l'adresse *www.kernel-machines.org.*

et 2007 observations en dimension 256. La figure 5.10 donne un aperçu des données USPS utilisées dans ce paragraphe. Afin de mettre en évidence l'influence de la taille de l'échantillon d'apprentissage sur les résultats de classification de l'HDDA et des méthodes classiques, nous avons utilisé successivement une part croissante et aléatoirement déterminée du jeu d'apprentissage initial pour construire le classifieur. Le nombre d'observations qui composaient les différents jeux d'apprentissage utilisés lors de cette expérience variait entre 250 et 7250. La performance des méthodes étudiées est mesurée par le taux moyen, calculé sur 50 répétitions, de classification correcte du jeu de validation. Le modèle utilisé pour l'HDDA est le modèle $[a_i b_i Q_i d_i]$ puisqu'il s'est avéré être un modèle efficace lors de l'étude précédente. Le seuil s permettant de déterminer les dimensions intrinsèques des classes a été déterminée par validation croisée sur le jeu d'apprentissage initial et a été fixée à 18. Rappelons aussi que nous avons ajouté une régularisation de type *ridge* aux méthodes QDA, LDA et EDDA afin qu'elles puissent traiter des données de grande dimension. Enfin, le nombre d'axes retenus pour la méthode PCA+LDA est aussi égal à 15 pour cette étude et à été déterminé de la même manière que précédemment.

Résultats expérimentaux

La figure 5.11 présente les résultats de classification correcte des méthodes étudiées en fonction de la taille du jeu utilisé pour apprendre le classifieur. Il apparaît que l'HDDA n'est pas sensible à la taille de l'apprentissage et fournit des résultats très satisfaisants même avec un jeu d'apprentissage composé de 500 observations. Il est important de rappeler que les données sont en dimension 256 et que la tâche est de discriminer 10 classes. On peut également remarquer que QDA ne parvient pas à fournir un résultat si le nombre d'observations du jeu d'apprentissage est plus petit que 2000, et ce même avec la régularisation numérique. De plus, même avec un grand jeu d'apprentissage, QDA ne parvient pas à être performante. Les méthodes LDA et EDDA sont également affectées par le facteur « taille du jeu d'apprentissage », mais dans des proportions moindres, et ne parviennent pas à fournir une classification satisfaisante des données de test. L'origine de ces performances décevantes est certainement l'inadéquation entre les hypothèses faites par les modèles sur lesquels ces deux méthodes sont basées et la complexité des données utilisées dans cette étude. Il est intéressant de noter que la réduction de dimension par ACP a permis d'améliorer la performance générale de LDA. Cela est certainement dû au fait que l'hypothèse d'égalité des matrices de covariance est trop éloignée de la situation réelle en dimension 256 mais l'est beaucoup moins en dimension 15. D'autre part, l'étape de réduction de dimension a également affranchi LDA de sa dépendance vis à vis de la taille de l'échantillon en réduisant le nombre de paramètres à estimer. Nous avons en outre étudié le comportement des méthodes HDDA et PCA+LDA avec des jeux d'apprentissage de tailles inférieures à 500. La méthode HDDA a continué à donner des résultats satisfaisants ($\simeq 0.79$) et meilleurs que ceux de PCA+LDA ($\simeq 0.77$) pour des jeux d'apprentissage composés de 200 observations. En deça de ce cas limite, les résultats de l'HDDA se sont avérés être très instables. Cette étude a, en tout cas, montré que l'HDDA fournit des résultats de classification très satisfaisants en grande dimension et avec un jeu d'appren-

Modèle de l'HDDA	Dimension moy. $\bar{d}$	Taux de classif. correcte
$[a_{ij}b_iQ_id_i]$	17.6	0.926
$[a_{ij}bQ_id_i]$	18.5	0.936
$[a_ib_iQ_id_i]$	18.5	0.928
$[a_ibQ_id_i]$	17.6	**0.937**
$[abQ_id_i]$	16.7	0.932
$[a_{ij}b_iQ_id]$	23	0.928
$[a_{ij}bQ_id]$	20	**0.948**
$[a_ib_iQ_id]$	22	0.928
$[a_ibQ_id]$	20	0.946
$[abQ_id]$	20	0.945

TAB. 5.1 – Résultats de classification obtenus avec l'HDDA sur les données USPS.

Méthode	Taux de classif. correcte	Temps d'apprentissage (sec.)
HDDA $[a_{ij}bQ_id]$	**0.948**	~ 1
QDA (full-GMM)	0.846	~ 1
LDA (com-GMM)	0.757	~ 1
EDDA $[\lambda_k B_k]$	0.696	~ 1
SVM (linéaire)	0.926	~ 12

TAB. 5.2 – Résultats de classification obtenus avec l'HDDA et les méthodes classiques de discrimination sur les données USPS.

tissage de taille très limitée. En outre, l'HDDA s'est révélée être une méthode bien plus performante que les autres méthodes de discrimination sur ce jeu réel de données de grande dimension. Une étude comparative approfondie menée sur ce jeu de données est proposée au paragraphe suivant.

5.2.3 Comparaison avec les méthodes classiques

Nous nous proposons à présent de comparer l'HDDA aux méthodes classiques sur le jeu de données réelles USPS utilisé dans l'étude précédente. Les méthodes de référence auxquelles nous allons nous comparer dans ce paragraphe sont les 4 méthodes classiques étudiées précédemment et la méthode discriminative *support vector machines* (SVM). Rappelons que SVM, que nous avons présentée au paragraphe 2.2.3, est une technique de discrimination qui doit sa grande popularité à ses excellentes performances.

Données et protocole

Cette étude a été menée sur le jeu de données réelles USPS que nous avons présenté au paragraphe précédent. Nous avons cette fois ci utilisé les jeux d'apprentissage et de validation classiques qui sont respectivement composés de 7291 et 2007 observations de dimension 256. Pour chaque modèle de l'HDDA, nous avons recherché par validation croisée sur le jeu d'apprentissage les dimensions intrin-

sèques d_i, $i = 1, ..., k$, des classes. Une fois ces dimensions estimées, nous avons appris les classifieurs associés au modèle $[a_{ij}b_iQ_id_i]$ et à ses sous-modèles sur le jeu d'apprentissage. La performance des différents modèles a été ensuite mesurée par le taux de classification correcte sur le jeu de validation. Nous avons également mis en compétition dans cette étude la méthode discriminative SVM associée au noyau linéaire, *i.e.* $K(x, x') = \langle x, x' \rangle$. Nous avons choisi d'utiliser le noyau linéaire car il est le seul à ne pas nécessiter de paramétrage. En effet, une des limitations des SVM, comme le fait remarquer Burges dans [15], est la difficulté pour l'utilisateur de choisir le noyau le mieux adapté aux données qu'il a à traiter.

Résultats expérimentaux

Le tableau 5.1 présente les résultats de classification obtenus par l'HDDA sur le jeu de données USPS et en fonction du modèle gaussien utilisé. Ce tableau indique en outre, pour le modèle $[a_{ij}b_iQ_id_i]$ et ses sous-modèles, la dimension intrinsèque moyenne $\bar{d}$ associée au paramètre s qui a été choisi par validation croisée sur le jeu d'apprentissage. On remarque tout d'abord que les dimensions intrinsèques moyennes, choisies par validation croisée, sont du même ordre de grandeur. Cela signifie, que pour les modèles supposant que les dimensions d_i sont différentes, la procédure d'estimation basée sur le *scree-test* de Cattell fonctionne correctement. Notons que, sur les 256 variables mesurées, l'HDDA ne retient en moyenne que 20 dimensions pour décrire chacune des classes. On observe également que les modèles faisant l'hypothèse que les dimensions d_i et les paramètres b sont communs sont les plus performants. Les données ayant été acquises de la même façon, il est assez vraisemblable que le bruit soit commun entre les classes et que par conséquent les modèles à b_i communs soient les plus efficaces. Le fait que soit un modèle à dimensions communes qui soit le plus performant de tous nos modèles traduit que la modélisation de chacune des 10 classes de la base, *i.e.* les 10 chiffres, est du même ordre de complexité. Le tableau 5.2 permet de comparer les performances du meilleur résultat de l'HDDA et des méthodes classiques de discrimination. Nous y avons ajouté le résultat de classification obtenu sur le jeu de validation par la méthode discriminative SVM assortie du noyau linéaire. Nous avons ajouté à ce tableau le temps de calcul, en secondes, nécessaire à l'apprentissage du classifieur sur un ordinateur récent (Pentium IV, 3Ghz). L'HDDA s'avère être la méthode la plus performante devant la méthode SVM qui est connue pour son excellent pouvoir de prévision. L'observation du temps nécessaire à l'apprentissage révèle que l'HDDA nécessite un temps de calcul de l'ordre des autres méthodes génératives alors que la méthode SVM nécessite approximativement 10 fois plus de temps. D'ailleurs, dans [15], Burges retient également le temps de calcul nécessaire à l'apprentissage du classifieur SVM comme l'une de ses principales limitations. L'HDDA est par conséquent une méthode d'analyse discriminante à la fois performante et rapide. Cette qualité de l'HDDA pourra par exemple s'avérer utile dans des situations d'apprentissage en temps réel, *i.e.* mise à jour du classifieur en fonction de nouvelles informations.

5.3 Validation dans le cadre non supervisé

De la même façon que dans le cadre supervisé, nous allons à présent évaluer et comparer entre eux les différents modèles proposés au chapitre 3 dans le cadre non supervisé. Nous nous intéresserons ensuite à l'estimation des hyper-paramètres que sont le nombre k de composantes du mélange et les dimensions intrinsèques d_i, $i = 1, ..., k$, des groupes. Nous étudierons également l'influence de la dimension sur les performances de l'HDDC et de la valeur BIC associée. Enfin, nous mettrons en concurrence notre méthode de *clustering* HDDC et une technique proposée récemment par Raftery *et al.* [74] qui combine sélection de variables et modèle de mélange gaussien dans le cadre de la classification non supervisée. Cette dernière étude sera menée sur données réelles, les précédentes étant faites sur données simulées.

5.3.1 Sélection de modèles

Notre méthode de *clustering* HDDC étant basée sur un modèle de mélange gaussien, il est donc possible de sélectionner le modèle le plus adapté aux données en utilisant le critère BIC que nous avons présenté au chapitre 4. Nous allons donc utiliser ce critère en combinaison avec le taux de classification correcte pour évaluer et comparer les modèles de notre famille. Rappelons que le taux de classification correcte peut être calculé dans ce cas car les données sont simulées et par conséquent nous avons connaissance des labels corrects des observations. Cette information est naturellement omise lors de l'utilisation de l'algorithme HDDC. De même que dans le cas supervisé, nous ne considérerons dans cette étude qu'une partie des modèles de notre famille qui en compte 28. Les modèles étudiés sont les modèles à orientations et dimensions libres dont les estimateurs sont explicites et dont nous pensons qu'ils sont propres à modéliser une large gamme de situations. Il est facile de conjecturer le comportement des modèles à orientations libres et dimensions communes à partir du comportement des 6 modèles étudiés puisque ces deux types de modèles sont très proches. Le modèles à orientations communes ne sont pas étudiés ici car l'estimation de leurs paramètres requière le plus souvent l'utilisation d'une méthode itérative.

Données et protocole

Nous avons réalisé un grand nombre de simulations (50 répétitions pour chacun des 6 modèles de simulation) et utilisé ensuite les 6 différents modèles à orientations libres dans l'HDDC pour classer les différents jeux de données simulés. Pour chacun des jeux simulés, nous avons simulé des données réparties en 3 groupes dans $\mathbb{R}^{100}$ selon chacun des 6 modèles à orientations libres de notre famille. Les paramètres qui ont été utilisés pour la simulation sont les suivants : $a_{1j} = 150 \pm 50$, $a_{2j} = 75 \pm 25$, $a_{3j} = 50 \pm 25$ pour $j = 1, ..., d_i$, $b_i = 10 \pm 5$ pour $i = 1, ..., k$, $\{\pi_1, \pi_2, \pi_3\} = \{0.4, 0.3, 0.3\}$ et $\{d_1, d_2, d_3\} = \{2, 5, 10\}$. Les moyennes des groupes étaient relativement proches ($\|\mu_i - \mu_\ell\| \simeq 10$, pour tout $i, \ell \in \{1, 2, 3\}$ tels que $i \neq \ell$) et les matrices d'orientation Q_i ont été obtenues de la même façon que dans le cas supervisé en effectuant la décomposition QR d'une matrice symétrique

Modèle de simulation	Modèle de classification					
	$[a_{ij}b_iQ_id_i]$	$[a_{ij}bQ_id_i]$	$[a_ib_iQ_id_i]$	$[a_ibQ_id_i]$	$[ab_iQ_id_i]$	$[abQ_id_i]$
$[a_{ij}b_iQ_id_i]$	357	373	**349**	359	**349**	360
$[a_{ij}bQ_id_i]$	403	404	397	**396**	397	397
$[a_ib_iQ_id_i]$	389	419	**377**	391	**377**	394
$[a_ibQ_id_i]$	438	440	**419**	**419**	420	420
$[ab_iQ_id_i]$	399	433	**380**	402	384	403
$[abQ_id_i]$	456	451	428	**427**	434	433

TAB. 5.3 – Valeur moyenne du critère BIC pour les modèles de l'HDDC pour différents jeux de données simulés. La valeur du critère associée au modèle sélectionné par BIC pour chacune des lignes est indiquée en gras.

Modèle de simulation	Modèle de classification					
	$[a_{ij}b_iQ_id_i]$	$[a_{ij}bQ_id_i]$	$[a_ib_iQ_id_i]$	$[a_ibQ_id_i]$	$[ab_iQ_id_i]$	$[abQ_id_i]$
$[a_{ij}b_iQ_id_i]$	0.967	0.828	0.973*	0.919	**0.975***	0.903
$[a_{ij}bQ_id_i]$	0.730	0.727	0.779	**0.782***	0.758	0.751
$[a_ib_iQ_id_i]$	0.979	0.871	0.983*	0.929	**0.986***	0.917
$[a_ibQ_id_i]$	0.826	0.800	**0.882***	0.863*	0.875	0.865
$[ab_iQ_id_i]$	0.965	0.825	**0.980***	0.844	0.952	0.822
$[abQ_id_i]$	0.712	0.752	**0.797**	0.793*	0.711	0.707

TAB. 5.4 – Taux de classification correcte moyen pour les modèles de l'HDDC pour différents jeux de données simulés. Pour chacune des lignes, le meilleur résultat de classification est indiqué en gras et le modèle sélectionné par le critère BIC est indiqué par une étoile.

aléatoire. Chacun des jeux de données était composé de 1000 observations. Rappelons enfin que le modèle choisi comme le mieux adapté aux données est celui qui est associé à la valeur du critère BIC la plus petite. A l'inverse, le meilleur taux de classification correcte est celui qui est le plus proche de 1. Lors de chacune des 50 répétitions, l'algorithme HDDC a été initialisé grâce à une partition obtenue aléatoirement.

Résultats expérimentaux

Le tableau 5.3 présente la valeur moyenne sur les 50 répétitions du critère BIC correspondant à chacun des 6 modèles et en fonction du modèle utilisé pour simuler les données. Le tableau 5.4 présente quant à lui le taux moyen de classification correcte obtenu avec chacun des 6 modèles de l'HDDC et en fonction des différents jeux de données. Dans les deux tableaux, le meilleur résultat de chaque ligne est indiqué en gras. Dans le tableau 5.4 nous avons noté d'une étoile le modèle sélectionné par BIC pour chaque jeu de données. On observe tout d'abord que le critère BIC sélectionne le plus souvent le modèle fournissant le meilleur taux de classification correcte. Cela confirme que le critère BIC est un outil efficace de sélection de modèle. Il n'est pas surprenant non plus de voir que le modèle ayant servi à simuler les données obtient une valeur du critère BIC petite et que sont taux de classification correcte est satisfaisant. L'observation du tableau 5.3 fait néanmoins apparaître que le modèle $[a_i b_i Q_i d_i]$ est le plus souvent élu par le critère BIC comme étant le modèle le plus adapté aux données. De plus, le tableau 5.4 nous indique que ce modèle obtient de très bons résultats de classification. Cette étude montre donc que dans un cas comme celui-ci, où le nombre d'observations par classe est petit devant le nombre de paramètres à estimer, ce modèle intermédiaire est efficace même dans une situation qui lui est *a priori* défavorable. Cela est certainement dû au fait que, dans cette situation, les grandes valeurs propres ont tendance à être surestimées et que par conséquent l'estimation de chacune des plus grandes valeurs propres est globalement moins bonne que l'estimation de la moyenne des plus grandes valeurs propres. Le modèle $[a_i b_i Q_i d_i]$ semble donc avoir le bon degré de complexité et l'hypothèse que Δ_i ne possède que deux valeurs propres différentes apparaît être un moyen efficace de régulariser l'estimation de la matrice de covariance de chacune des classes. Notons également que les modèles $[a_i b Q_i d_i]$ et $[a b_i Q_i d_i]$, qui sont très proches en complexité du modèle $[a_i b_i Q_i d_i]$, sont également plébiscités par le critère BIC et le taux de classification correcte.

5.3.2 Estimation des paramètres discrets

Dans le cadre qui est le notre ici, l'estimation du nombre de composantes du mélange et des dimensions intrinsèques de groupes ne peut pas être faite par validation croisée puisque les labels des observations ne sont pas connus. Nous allons donc envisager le problème de l'estimation de ces paramètres comme un problème de sélection de modèle. En effet, les paramètres k, $d_1, ..., d_k$ contrôlant la complexité du modèle, le critère BIC peut donc être utilisé pour sélectionner ces paramètres.

Nb de classes k	Seuil choisi s	Dimensions d_i	Valeur BIC
2	0.18	2,16	414
3	0.21	**2,5,10**	**407**
4	0.25	2,2,5,10	414
5	0.28	2,5,5,10,12	416
6	0.28	2,5,6,10,10,12	424

TAB. 5.5 – Sélection du nombre de classes et des dimensions grâce au critère BIC sur un jeu de données artificielles composées de 3 groupes dont les dimensions intrinsèques respectives sont 2, 5 et 10.

Données et protocole

Nous avons simulé, pour cette expérience, des données réparties en 3 groupes dans $\mathbb{R}^{100}$ selon le modèle $[a_i b Q_i d_i]$ avec les paramètres suivants : $\{\pi_1, \pi_2, \pi_3\} = \{0.4, 0.3, 0.3\}$, $\{a_1, a_2, a_3\} = \{150, 75, 50\}$, $b = 10$, $\{d_1, d_2, d_3\} = \{2, 5, 10\}$, avec des moyennes proches ($\|\mu_i - \mu_\ell\| \simeq 10$, pour tout i, $\ell \in \{1, 2, 3\}$ tels que $i \neq \ell$) et des matrices d'orientation Q_i aléatoires. Enfin, le jeu de données était composé de 1000 observations. Nous avons ensuite calculé la valeur du critère BIC pour chacun des couples (k, s), $k = 1, ..., 6$ et s variant entre 0.1 et 0.4. Rappelons que s est le paramètre du scree-test de Cattell permettant d'unifier l'estimation des k dimensions d_i. L'expérience a été répétée à 50 reprises afin de moyenner les effets liés à l'initialisation aléatoire de l'algorithme.

Résultats expérimentaux

Le tableau 5.5 donne, pour chaque valeur de k, les valeurs moyennes du seuil s ayant la valeur BIC la plus petite ainsi que les estimations des dimensions associées à ce choix de seuil. On observe tout d'abord que le critère BIC parvient à estimer correctement les valeurs de k et des dimensions d_i, $i = 1, ..., k$. Le critère BIC sélectionne en effet le modèle $[a_i b_i Q_i d_i]$ ayant les paramètres $\hat{k} = 3$ et $\{\hat{d}_1, \hat{d}_2, \hat{d}_3\} = \{2, 5, 10\}$ ce qui correspond aux paramètres du modèle de simulation. Il est d'autre part intéressant d'observer l'évolution de l'estimation des dimensions en fonction de k. Si l'on considère le cas d'un mélange de seulement 2 composantes, l'HDDC semble correctement modéliser le premier groupe, *i.e.* $\hat{d}_1 = d_1$, et créer un second groupe hybride composé du second et du troisième groupe réel, *i.e.* $\hat{d}_2 \simeq d_2 + d_3$. De même, si l'on considère le cas d'un mélange de 4 composantes, l'HDDC semble avoir divisé l'effectif du premier groupe dans deux sous-groupes de même dimension intrinsèque. Cette étude a donc permis de valider l'approche que nous avons proposée pour estimer le paramètre k et les dimensions d_i, $i = 1, ..., k$. D'autre part, nous n'avons pas remarquer lors de cette étude une instabilité particulière de l'HDDC qui soit due au choix des dimensions intrinsèques. Cela est certainement dû au fait que le choix de la dimension des sous-espaces des classes n'est pas aussi crucial que dans le cas des méthodes de réduction de dimension (comme l'ACP). Rappelons, en effet, que notre paramétrisation du modèle gaussien conserve toutes les dimensions, au contraire

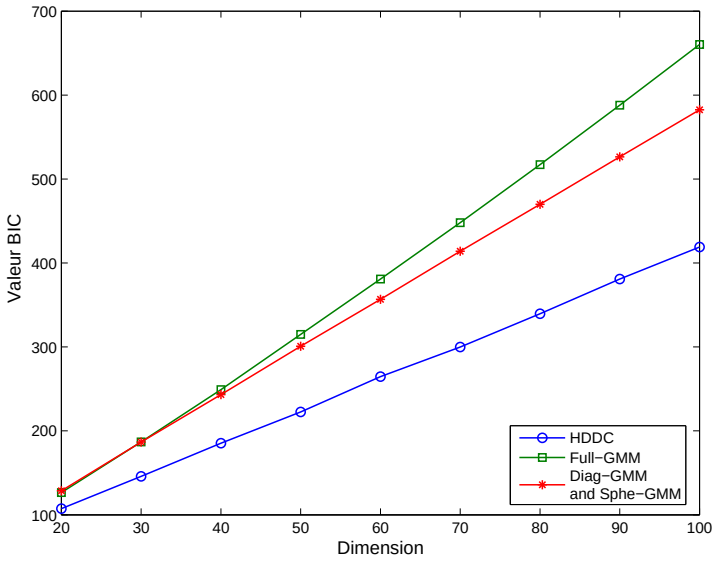


FIG. 5.12 – Influence de la dimension sur la valeur du critère BIC associé au modèle $[a_{ij}b_iQ_id_i]$ de l'HDDC et aux méthodes classiques sur données simulées (voir le texte pour les détails de la simulation des données).

des méthodes de réduction de dimension, et ne fait qu'assigner un poids plus faible aux sous-espaces supplémentaires des sous-espaces où vivent les données des classes.

5.3.3 Influence de la dimension

Nous allons maintenant mettre en évidence le comportement de l'HDDC vis à vis de la dimension des données. Nous étudierons également les comportements des méthodes classiques de classification non supervisée afin de comparer notre méthode à ces méthodes de référence.

Données et protocole

Pour cette étude, nous avons simulé des données réparties en 3 groupes dans $\mathbb{R}^p$, $p = 20, ..., 100$, selon le modèle $[a_ibQ_id_i]$ avec les même paramètres que dans l'expérience correspondante dans le cas supervisé (*cf.* paragraphe 5.2.1). Le jeu de données était composé de 1000 observations. La performance de chacune des méthodes étudiées est mesurée par le taux moyen de classification correcte calculé sur 50 répétitions. Les données étant simulées, il a également été possible de calculer le taux optimal de classification correcte obtenu par le classifieur de Bayes basé sur les densités réelles. Le modèle de l'HDDC utilisé ici est le modèle $[a_ib_iQ_id_i]$ et nous le comparerons aux 3 modèles classiques suivant : full-GMM, diag-GMM et sphe-GMM. Les 4 méthodes de classification automatique

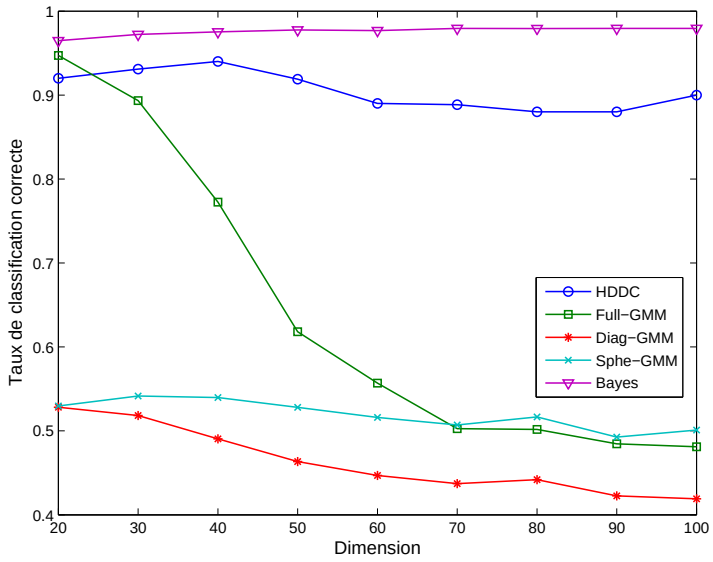


FIG. 5.13 – Influence de la dimension sur les résultats de classification obtenus avec l'HDDC (modèle $[a_{ij}b_iQ_id_i]$) et les méthodes classiques sur données simulées (voir le texte pour les détails de la simulation des données).

utilisées lors de cette étude ont été initialisées à chaque répétition avec la même partition obtenue aléatoirement.

Résultats expérimentaux

La figure 5.12 montre le comportement du critère BIC associé à chacun des modèles sur lesquels sont basés les méthodes étudiées en fonction de la dimension des données. Il n'est pas surprenant de remarquer que le critère BIC sélectionne toujours le modèle $[a_ib_iQ_id_i]$ comme étant le modèle le plus adapté aux données puisque elles ont été simulées selon ce modèle. En revanche, il est intéressant de noter que, plus la dimension augmente, plus la différence entre les valeurs du critère BIC des modèles classiques et du modèle $[a_ib_iQ_id_i]$ s'accroît et cela en faveur du modèle $[a_ib_iQ_id_i]$. Rappelons que le critère BIC sélectionne le modèle qui réalise le meilleur compromis entre complexité et adéquation aux données. L'augmentation de la dimension de l'espace semble aider à trouver le bon modèle. La figure 5.13, quant à elle, met en évidence l'évolution du taux de classification correcte de chacun des modèles en fonction de la dimension. On retrouve, pour l'HDDC, un comportement vis à vis de la dimension similaire à celui de l'HDDA dans le cadre supervisé. En effet, le taux de classification correcte de l'HDDC ne semble pas particulièrement affecté par la dimension des données. On note toutefois une plus grande irrégularité de la courbe des résultats par rapport au cas supervisé mais cet effet est certainement du à l'initialisation aléatoire de l'algorithme. Les résultats de la méthode de classification automatique associée au modèle full-GMM sont très rapidement pénalisés par la dimen-

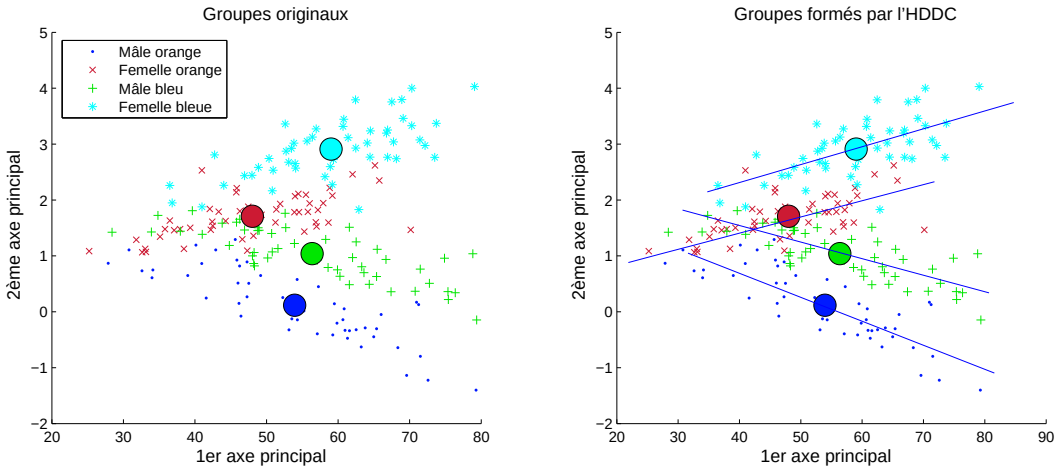


FIG. 5.14 – Données « crabes » : à gauche, projection des données sur les deux premiers axes principaux et, à droite, classification obtenue avec l'HDDC et le modèle $[a_i b_i Q_i d_i]$. Les lignes bleues représentent les sous-espaces spécifiques de chaque groupe.

sion de l'espace d'étude. Nous avions d'ailleurs observé un comportement similaire pour ce modèle dans le cas supervisé. Les modèles diag-GMM et sphe-GMM, quant à eux, ne semblent pas subir le fléau de la dimension mais fournissent des résultats très médiocres quelle que soit la dimension de l'espace. Le comportement de ces deux derniers modèles s'explique par le fait que ce sont des modèles très parcimonieux. Cette étude montre que notre modélisation des données de grande dimension semble posséder la qualité d'invariance à la dimension comme les modèles parcimonieux et la qualité de produire de résultats satisfaisants comme les modèles complexes dans les espaces de faible dimension.

5.3.4 Comparaison avec la sélection de variables

Nous nous proposons dans ce paragraphe de comparer notre approche à la sélection de variables. Une récente approche proposée par Raftery et Dean [74] permet de combiner la sélection de variables à l'étape de classification dans le cadre du modèle de mélange gaussien. Pour ce faire, les auteurs considèrent le problème de la sélection de variables comme un problème de choix de modèles. Dans leur approche, la sélection est réalisée en utilisant le critère BIC en combinaison avec un algorithme efficace de recherche. Les auteurs font remarquer de plus qu'il est possible d'effectuer cette sélection de variables sur les variables originales ou sur les composantes principales. Cette méthode de sélection de variables pour la classification automatique sera notée VS-GMM dans la suite de ce paragraphe.

Données et protocole

Afin de comparer notre méthode de clustering HDDC à cette technique, nous avons choisi d'utiliser le jeu de données « crabes » utilisé dans [74] et que nous avons déjà présenté au paragraphe 5.1.3.

Modèle	TCC sur var. orig.	TCC sur comp. princ.	TCC avec ACP
Sphe-GMM	0.340	0.340	0.340
Diag-GMM	0.355	0.355	0.535
Com-GMM	0.625	0.625	0.635
Full-GMM	0.640	0.640	0.845
VS-GMM [74]	0.925	0.935	/
HDDC $[a_i b_i Q_i d_i]$	**0.950**	**0.950**	/

TAB. 5.6 – Taux de classification correcte (TCC) obtenus avec l'HDDC, les modèles gaussiens classiques et la méthode de sélection de variable VS-GMM [74] sur un jeu de données réelles (données Crabes).

Rappelons tout de même que ces données sont composées de 200 observations en dimension 5 et réparties équitablement en 4 classes. La figure 5.14 présente l'organisation des données « crabes » projetées sur les deux premiers axes principaux. Les disques de couleurs représentent les moyennes empiriques des groupes. Le modèle sélectionné par le critère BIC pour être utilisé par l'HDDC est le modèle $[a_i b_i Q_i d_i]$. Nous avons également utilisé les méthodes de clustering associées aux modèles classiques full-GMM, com-GMM, diag-GMM et sphe-GMM. Le *clustering* a été réalisé 50 fois pour chaque méthode afin de moyenner les résultats de classification et, à chaque répétition, les algorithmes ont été initialisés à partir de la même partition aléatoire des données. Pour les méthodes associées aux modèles classiques, le *clustering* a été également été effectué sur les composantes principales et en combinaison avec une réduction à 3 dimensions par ACP. Le nombre de composantes principales retenues pour l'étape de réduction de dimension a été déterminé à partir de l'ébouli des valeurs propres. Notons de plus que le nombre de variables sélectionnées par VS-GMM est aussi égal à 3 et que le modèle gaussien utilisé dans cette méthode est le modèle full-GMM.

Résultats expérimentaux

Le tableau 5.6 présente les résultats de classification obtenus avec l'HDDC et les méthodes de *clustering* classiques. Nous y avons également reporté le résultat de VS-GMM donné par [74]. Ces résultats sont disponibles pour la classification sur variables originales, sur composantes principales et sur données réduites par ACP. Il apparaît tout d'abord que ces données, qui ne sont certes pas de grande dimension, s'avèrent très difficiles à classer avec les méthodes classiques. En effet, le meilleur résultat obtenu avec une méthode classique est 0.64 en utilisant le modèle gaussien général full-GMM. On remarque que la classification sur composantes principales n'améliore pas les résultats des méthodes classiques. Le fait que la classification sur composantes principales n'améliore pas les résultats des modèles diag-GMM et sphe-GMM confirme notre hypothèse que les groupes vivent dans des sous-espaces d'orientations différentes. En revanche, on peut noter que la réduction de dimension permet d'améliorer significativement le résultat de classification du modèle full-GMM. La méthode de sélection de variables VS-GMM obtient un taux de classification correcte égal à 0.925 sur variables ori-

ginales et à 0.935 sur composantes principales et devance ainsi les méthodes classiques. Sachant que la réduction de dimension par ACP a permis à la méthode associée au modèle full-GMM d'améliorer ses résultats, il est assez naturel que les résultats de VS-GMM soient aussi meilleurs sur composantes principales. On peut également conclure de cette observation que VS-GMM n'a certainement pas sélectionné les 3 mêmes variables que l'ACP puisque les résultats de VS-GMM sont bien meilleurs que ceux de full-GMM sur données réduites par ACP. Cela confirme une autre de nos hypothèses initiales qui suppose que la réduction de dimension par ACP n'est pas une approche optimale dans le but de classer des données. Enfin, l'HDDC domine cette étude en obtenant un taux de classification correcte égal à 0.95 sur variables originales et sur composantes principales. Rappelons que le fait de travailler sur les composantes originales ne change rien aux résultats de l'HDDC puisqu'elle cherche le sous-espace de chaque groupe dans son espace propre. On peut déduire de la comparaison des résultats de l'HDDC avec ceux de VS-GMM que le fait de ne pas supprimer de variables et de modéliser chaque groupe dans son espace propre est la meilleure approche pour la classification. Notons au passage que l'HDDC est également capable de surclasser les méthodes existantes sur des jeux de données dont la dimension est plutôt faible.

Chapitre 6

Application à la reconnaissance d'objets

Dans ce chapitre, nous allons considérer le problème de la reconnaissance d'objets dans des images qui est un des problèmes les plus difficiles à l'heure actuelle en vision par ordinateur. Outre la difficulté intrinsèque du problème, celui-ci présente plusieurs intérêts vis à vis de notre sujet d'étude : les données sont en grande dimension et l'approche classique combine apprentissage non supervisé et supervisé. Pour ces raisons, nous avons choisi d'appliquer les méthodes de classification proposées dans ce mémoire à la reconnaissance d'objets. Nous proposerons une approche de la reconnaissance d'objets basée sur le modèle de mélange permettant de localiser de manière probabiliste l'objet étudié. Nous présenterons tout d'abord au paragraphe 6.1 le problème de la reconnaissance d'objets et les approches existantes. Au cours du paragraphe 6.2, nous proposerons une approche probabiliste de la reconnaissance d'objets adaptée aux cadres supervisés et faiblement supervisés. Enfin, le paragraphe 6.3 sera consacré à l'évaluation expérimentale de notre approche. Ces expérimentations seront menées sur deux bases de données récentes de reconnaissance d'objets et mettront en évidence que notre approche est plus efficace que les méthodes existantes.

6.1 La reconnaissance d'objets

Ce premier paragraphe va donc être consacré à introduire le problème de la reconnaissance d'objets et présenter l'état de l'art dans ce domaine qui connaît actuellement un très fort développement. La grande activité qui règne dans ce domaine est en partie due aux intérêts liés aux applications de la reconnaissance d'objets. En effet, la reconnaissance d'objets est au cœur d'un grand nombre de progrès technologiques parmi lesquels on peut citer la télé-surveillance, la sécurité, l'indexation des images sur le web et le pilotage autonome de véhicules.

6.1.1 Le problème de la reconnaissance d'objets

Le problème de la reconnaissance d'objets est d'identifier les images comportant une instance d'un objet donné et, le cas échéant, de localiser cet objet dans l'image. Ce problème est rendue difficile par

FIG. 6.1 – Quelques exemples de la catégorie d'objets « vélo ».

la très grande variabilité qui existe dans une catégorie d'objets. La figure illustre cette caractéristique des classes d'objets en présentant différentes instances de la classe « vélo ».

Localisation d'objets et classification d'images

Dans ce travail, nous nous considérerons les deux tâches de la reconnaissance d'objets : la classification d'images et la localisation d'objets. La classification d'images consiste à décider pour une nouvelle image si elle contient une instance de l'objet étudié, que nous noterons O, ou si elle ne contient aucune instance de cet objet et qu'elle ne contient que du fond, que nous noterons B. Cette tâche est par conséquent un problème de classification binaire. La localisation d'objets consiste quant à elle à circonscrire l'objet dans une nouvelle image dont on sait qu'elle contient au moins une instance de l'objet. Cette localisation de l'objet O à l'intérieur d'une image peut être réalisée soit par segmentation, *i.e.* pour chaque pixel de l'image on décide s'il représente ou non l'objet, soit en délimitant la zone de l'image contenant l'objet grâce à un cadre, également appelé *bounding box*. Il va de soi que la localisation d'objets requiert une plus haut degré de précision que la classification d'images et est, de ce fait, considérée comme plus difficile.

Apprentissage supervisé et faiblement supervisé

La reconnaissance d'objets nécessite bien entendu une phase d'apprentissage permettant de prendre connaissance des caractéristiques de la catégorie d'objets étudiée. Pour cela, on dispose généralement d'une base d'images pour lesquelles on a une information, plus ou moins précise, sur la présence ou l'absence de l'objet étudié. Le type de supervision de l'apprentissage peut en effet être de deux sortes : supervisé ou faiblement supervisé. Dans le cas de l'apprentissage supervisé, les objets sont segmentés dans chacune des images d'apprentissage, *i.e.* on sait pour chaque point d'une image s'il appartient ou non à l'objet. Un point sera alors référencé comme positif s'il appartient à l'objet O et négatif dans le cas contraire. Dans le cas faiblement supervisé, les points des images d'apprentissage qui contiennent au moins une instance de l'objet O sont référencés comme positifs. Les points des images qui ne contiennent aucune instance de l'objet O sont, quant à eux, référencés comme négatifs. Il est important de noter que, dans le cas d'une supervision faible, les points référencés comme positifs peuvent appartenir à l'objet O mais également au fond B. En revanche, les points référencés comme négatifs ne peuvent appartenir qu'au fond B. Naturellement, le second type de supervision est moins coûteux en temps d'annotation et les recherches actuelles se tournent de plus en plus vers cette approche.

6.1.2 Etat de l'art

Nous allons à présent donner un aperçu de l'état de l'art en reconnaissance d'objets. Il faut tout d'abord noter que la plupart des méthodes récentes de reconnaissance d'objets utilisent une description locale de l'image, *i.e.* l'image est représentée par un ensemble de descripteurs décrivant les zones d'intérêts de l'image. La description locale de l'image sera abordée en détail au paragraphe suivant.

Les méthodes *part-based*

La plupart des approches récentes organisent tout d'abord les descripteurs locaux en groupes homogènes en utilisant des techniques de *clustering* classiques (k-means ou les méthodes génératives basées sur les modèles de mélange gaussien). Agarwal et Roth [1] déterminent ensuite les relations spatiales entre les groupes précédemment formés et utilisent le classifieur *Sparse Network Classifier* pour l'étape de reconnaissance. Dorko et Schmid [28] sélectionnent les groupes discriminants de l'objet étudié à partir de leur *ratio* de vraisemblance et se servent uniquement des groupes les plus discriminants pour la phase de reconnaissance. Leibe et Schiele [55] apprennent la distribution spatiale des groupes formés et mettent en œuvre ensuite un système de vote.

Les méthodes de *bag-of-keypoints*

D'autre part, les méthodes de *bag-of-keypoints* [92, 94] représentent chaque image par un histogramme des groupes appris et utilisent ensuite un classifieur SVM dans l'étape de reconnaissance. Sivic *et al.* [85] combinent une représentation de type *bag-of-keypoints* avec la méthode *Probabilistic Latent Semantic Analysis* (PLSA) [48] pour découvrir un « vocabulaire » de parties d'objets. Enfin, Opelt *et al.* [67] utilisent une méthode de sélection de variables pour identifier les caractéristiques discriminantes de l'objet.

6.1.3 Description locale de l'image

Les premières méthodes proposées pour reconnaître des objets dans des images caractérisaient les objets par leur apparence globale. La principale limite de cette approche est qu'elle n'est pas robuste aux occlusions, aux changements de luminosité et aux transformations géométriques. Pour éviter ces problèmes, les méthodes récentes utilisent des descripteurs locaux qui sont par construction robustes à l'ensemble de ces perturbations. Ainsi, dans le cadre d'une approche utilisant des descripteurs locaux, chaque image est associée à un nombre plus ou moins grand de descripteurs locaux.

Détection des points d'intérêt

L'extraction de ces descripteurs comporte deux phases : la détection de points d'intérêt et la description de la zone de l'image située autour de chacun des points d'intérêt. La détection des points d'intérêt est réalisée grâce à un opérateur de détection qui parcourt l'image et identifie les zones

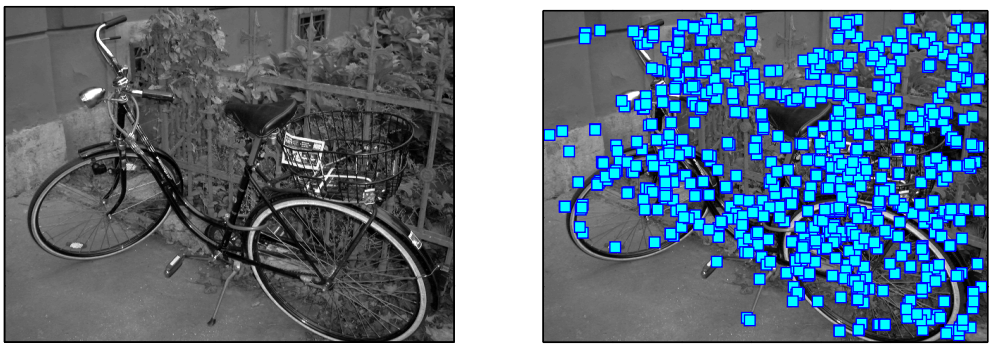


FIG. 6.2 – Extraction des points d'intérêt : à gauche, l'image originale et, à droite, les points d'intérêt détectés sur l'image et qui seront ensuite convertis en descripteurs locaux.

de l'image ayant des caractéristiques particulières. de fort gradient dans toutes les directions. Par exemple, le détecteur de Harris-Laplace (HL) [61], l'un des les plus utilisés, recherche les zones de fort gradient dans toutes les directions. Pour cela, il calcule sur un voisinage de chacun des pixels de l'image la matrice A définie par :

$$A(x,y) = \begin{pmatrix} \nabla_x^2(x,y) & \nabla_{xy}(x,y) \\ \\ \nabla_{yx}(x,y) & \nabla_y^2(x,y) \end{pmatrix},$$

où ∇_u est le gradient de l'image dans la direction u. Les pixels (x,y) pour lesquels la matrice $A(x,y)$ à deux valeurs propres fortes sont des points d'intérêt. En pratique, ce procédé détecte principalement les parties saillantes (angles, bords, ...) des objets de l'image. Le détecteur de Harris-Laplace associe à chacun des points d'intérêt détectés une échelle caractéristique qui servira à déterminer la taille du voisinage du point qui sera transformée en descripteur local. Cette échelle caractéristique est déterminée, pour chacun des points d'intérêt, en maximisant le Laplacien dans l'espace multi-échelles. Parmi les autres détecteurs existants, on peut citer le détecteur DoG (*Difference of Gaussian*) [56] et le détecteur de Kadir *et al.* [51] qui, tous deux, extraient des zones homogènes (*blobs* en anglais) de l'image.

Description des points d'intérêts

La zone de l'image avoisinant chacun des points d'intérêt détectés est ensuite décrite et transformée en un vecteur de grande dimension grâce à un descripteur. Le voisinage de chacun des points d'intérêt est défini et normalisé en fonction de l'échelle caractéristique qui lui est associée. Le descripteur SIFT [56], qui s'est avéré être très efficace dans le domaine de la reconnaissance d'objets, comme en témoignent les travaux [28, 67, 94], divise le voisinage du point d'intérêt courant en 4×4 zones et calcule les gradients dans les 8 directions de l'image à l'intérieur des ces 16 zones. Ainsi,

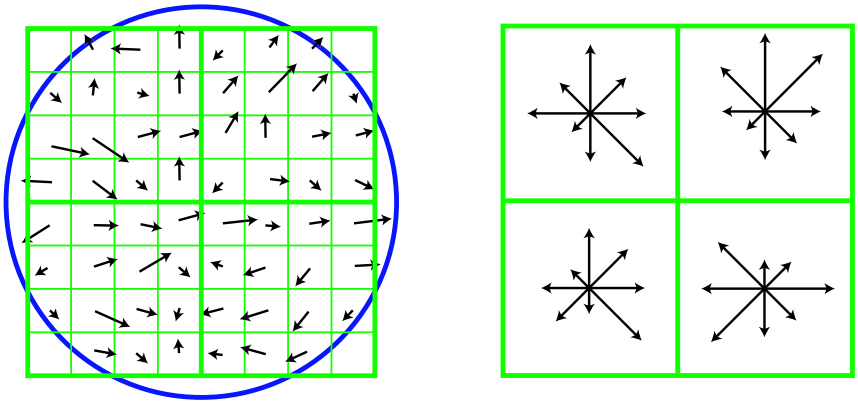


FIG. 6.3 – Description locale des points d'intérêt : à gauche, gradient du voisinage d'un point d'intérêt et, à droite, quantification du gradient selon les 8 directions cardinales pour 2×2 zones du voisinage considéré. Le descripteur utilisé dans ce mémoire considère une division en 4×4 zones du voisinage d'un point d'intérêt.

chaque voisinage d'un point d'intérêt est décrit par un vecteur de taille $128 = 4 \times 4 \times 8$. La figure 6.3 illustre le procédé de description associé à l'opérateur SIFT. Cette description invariante à l'échelle de chacun des points d'intérêt peut en outre être rendue invariante aux transformations géométriques de l'image. Au terme de ce procédé d'extraction, une image est associée à m descripteurs locaux en dimension 128 si l'on considère le cas du détecteur-descripteur HL+SIFT.

6.2 Approche probabiliste de la reconnaissance d'objets

La plupart des méthodes existantes de reconnaissance d'objets ne sont que partiellement probabilistes et nécessitent le réglage de nombreux paramètres de manière strictement empirique. De plus, peu de méthodes sont adaptées à la fois au cas supervisé et au cas faiblement supervisé. Nous nous plaçons dans le cadre classique de la reconnaissance d'objets et considérons donc que les observations, notées x_j dans la suite, sont des descripteurs locaux d'images. Nous proposons ici une approche probabiliste de la reconnaissance d'objets qui intègre de manière naturelle les méthodes de classification proposées dans ce mémoire et qui permette de connaître pour chaque point d'intérêt la probabilité qu'il appartienne à l'objet étudié. L'apprentissage pourra en outre être soit supervisé, *i.e.* les objets sont segmentés dans les images d'apprentissage, soit faiblement supervisé, *i.e.* les objets ne sont pas segmentés dans les images d'apprentissage.

6.2.1 Le modèle

Nous nous plaçons dans le cadre du modèle de mélange gaussien pour modéliser la distribution des descripteurs d'une image. D'un point de vue formel, nous supposerons que les descripteurs x_j, $j = 1, ..., n$, sont des réalisations indépendantes d'une vecteur aléatoire à valeur dans $\mathbb{R}^p$ dont la

densité s'écrit sous la forme suivante :

$$f(x) = \sum_{i=1}^{k} \pi_i f(x, \theta_i),$$

où π_i est la probabilité a priori de la $i$ème composante du mélange et $f(x, \theta_i)$ est la densité d'une loi normale $\mathcal{N}(\mu_i, \Sigma_i)$ de moyenne μ_i et de matrice de covariance Σ_i, et ce pour $i = 1, ..., k$. Nous proposons en outre de re-formuler la densité f comme suit :

$$f(x) = \tau f^O(x) + (1 - \tau) f^B(x),$$

où τ est la probabilité *a priori* de l'objet O et où f^O et f^B sont les densités respectives des distributions de l'objet O et du fond B qui s'expriment de la façon suivante :

$$\begin{aligned} \tau f^O(x) &= \sum_{i=1}^{k} R_i \pi_i f(x, \theta_i), \\ (1 - \tau) f^B(x) &= \sum_{i=1}^{k} (1 - R_i) \pi_i f(x, \theta_i), \end{aligned}$$

où $R_i = P(C_i \in O)$, $i = 1, ..., k$, est le pouvoir discriminant de la classe C_i par rapport à O. Par intégration des densités, on obtient la relation $\tau = \sum_{i=1}^{k} R_i \pi_i$ qui lit la probabilité *a priori* τ à R_i et π_i. La densité f peut finalement s'exprimer comme suit :

$$f(x) = \underbrace{\sum_{i=1}^{k} R_i \pi_i f(x, \theta_i)}_{\text{Objet}} + \underbrace{\sum_{i=1}^{k} (1 - R_i) \pi_i f(x, \theta_i)}_{\text{Fond}}.$$

Il est alors naturel de penser qu'une classe ayant une valeur de R_i proche de 1 représente une partie de l'objet étudié et en est discriminante. De même, on peut supposer qu'une classe ayant une valeur de R_i proche de 0 représente le fond B et en est discriminante. En revanche, les classes ayant des valeurs de R_i proche de 0.5 ne sont discriminantes ni de l'objet O ni du fond B.

6.2.2 Apprentissage

Etant donnée la modélisation introduite au paragraphe précédent, la phase d'apprentissage consiste à estimer les paramètres du modèle à partir des descripteurs des images d'apprentissage. Cette phase d'apprentissage, qui peut être supervisée ou faiblement supervisée, comporte deux étapes : la première est l'estimation des paramètres π_i et θ_i du mélange, la seconde est l'estimation des paramètres R_i. Nous verrons que la seconde étape uniquement dépend du type de supervision.

Estimation des paramètres π_i et θ_i

Le but de cette première étape est l'estimation des paramètres π_i et θ_i du mélange à partir des données d'apprentissage. Pour ce faire, les descripteurs positifs et négatifs sont organisés en k groupes grâce à une méthode de classification automatique basée sur le modèle de mélange gaussien. Par exemple et comme les descripteurs sont en grande dimension, la méthode HDDC basée sur le modèle $[a_{ij}b_iQ_id_i]$ ou un de ses sous-modèles peut être utilisée pour regrouper ces descripteurs en k groupes homogènes. L'HDDC fournira en outre les estimations des paramètres du mélange π_i et θ_i. Mentionnons également que cette classification peut être effectuée grâce à toute méthode générative de classification non supervisée basée sur le modèle de mélange gaussien. Par exemple, il est possible d'estimer les paramètres π_i et θ_i en utilisant la méthode classique de *clustering* basée sur le modèle gaussien diag-GMM et l'algorithme d'estimation EM.

Estimation des paramètres R_i

Cette seconde étape consiste à estimer le vecteur de paramètres $R = (R_1, ..., R_k)$ du mélange. Nous proposons d'estimer ces paramètres par la méthode du *maximum* de vraisemblance sur les données d'apprentissage.

Proposition 6.2.1. *L'estimateur du* maximum *de vraisemblance conditionnelle du vecteur de paramètres $R = (R_1, ..., R_k)$ satisfait la relation suivante :*

$$\hat{R}_{MV} = \underset{R}{\operatorname{argmax}}\{J(R)\},$$

où la quantité à maximiser J est définie en fonction de R par :

$$J(R) = \sum_{x_j \in O} \log\left(\langle R, \Psi_j \rangle\right) + \sum_{x_j \in B} \log\left(1 - \langle R, \Psi_j \rangle\right) - n_O \log(\langle R, \pi \rangle) - n_B \log(1 - \langle R, \pi \rangle),$$

avec $\pi = (\pi_1, ..., \pi_k)$ et $\Psi_{ji} = P(x_j \in C_i | x_j)$ pour $i = 1, ..., k$.

Démonstration. Nous allons chercher à maximiser la vraisemblance conditionnellement à l'appartenance des points à O et B, que nous noterons Q :

$$Q = \prod_{x_j \in O} \frac{f^O(x_j)}{f(x_j)} \prod_{x_j \in B} \frac{f^B(x_j)}{f(x_j)},$$

qui, d'après la modélisation présentée au paragraphe précédent, peut s'exprimer de la façon suivante :

$$Q = \prod_{x_j \in O} \frac{1}{\tau} \sum_{i=1}^{k} R_i \frac{\pi_i f(x_j, \theta_i)}{f(x_j)} \prod_{x_j \in B} \frac{1}{(1-\tau)} \sum_{i=1}^{k} (1 - R_i) \frac{\pi_i f(x_j, \theta_i)}{f(x_j)}.$$

En notant $\Psi_{ji} = P(x_j \in C_i | x_j)$ et grâce à la formule de Bayes, on a la relation $\Psi_{ji} = \pi_i f(x_j, \theta_i)/f(x_j)$, pour $i = 1, ..., k$, et l'on peut écrire :

$$Q = \tau^{-n_0}(1-\tau)^{-n_B} \prod_{x_j \in O} \sum_{i=1}^{k} R_i \Psi_{ji} \prod_{x_j \in B} \sum_{i=1}^{k} (1 - R_i)\Psi_{ji},$$

où n_0 et n_B sont le nombre de points appartenant respectivement à l'objet O et au fond B. En passant au logarithme, on obtient :

$$\log(Q) = \sum_{x_j \in O} \log\left(\sum_{i=1}^{k} R_i \Psi_{ji}\right) + \sum_{x_j \in B} \log\left(\sum_{i=1}^{k} (1 - R_i)\Psi_{ji}\right) - n_O \log(\tau) - n_B \log(1-\tau),$$

et en adoptant une notation vectorielle, on peut écrire :

$$\log(Q) = \sum_{x_j \in O} \log\left(\langle R, \Psi_j \rangle\right) + \sum_{x_j \in B} \log\left(\langle 1 - R, \Psi_j \rangle\right) - n_O \log(\langle R, \pi \rangle) - n_B \log(1 - \langle R, \pi \rangle).$$

En remarquant que $< 1 - R_i, \Psi_j >= 1 - < R_i, \Psi_j >$, nous obtenons la relation :

$$\log(Q) = \sum_{x_j \in O} \log\left(\langle R, \Psi_j \rangle\right) + \sum_{x_j \in B} \log\left(1 - \langle R, \Psi_j \rangle\right) - n_O \log(\langle R, \pi \rangle) - n_B \log(1 - \langle R, \pi \rangle).$$

La maximisation de la vraisemblance conditionnelle Q est donc équivalente à la maximisation de la quantité $J(R) = \sum_{x_j \in O} \log\left(\langle R, \Psi_j \rangle\right) + \sum_{x_j \in B} \log\left(1 - \langle R, \Psi_j \rangle\right) - n_O \log(\langle R, \pi \rangle) - n_B \log(1 - \langle R, \pi \rangle)$. □

Malheureusement, l'estimateur $\hat{R}_{MV}$ n'a pas d'expression explicite et son calcul requiert l'utilisation d'une méthode d'optimisation itérative. Nous avons donc mis en œuvre une méthode de descente de gradient pour estimer le vecteur de paramètres R. L'expression du gradient de la quantité J à maximiser en fonction de R est :

$$\nabla_R J = \sum_{x_j \in O} \frac{\Psi_j}{< R, \Psi_j >} - \sum_{x_j \in B} \frac{\Psi_j}{1 - < R, \Psi_j >} - n_O \frac{\pi}{\langle R, \pi \rangle} + n_B \frac{\pi}{1 - \langle R, \pi \rangle}.$$

6.2.3 Reconnaissance d'objets

La phase de reconnaissance consiste à décider de manière probabiliste si les descripteurs d'une nouvelle image appartiennent ou non à l'objet O étudié. A partir de cette connaissance, deux tâches sont possibles : la localisation de l'objet et la classification de l'image.

Localisation d'objets

La localisation d'objets consiste à classer chacun des descripteurs d'une image I, différente des images d'apprentissage, comme appartenant à l'objet O ou comme appartenant au fond B. Cette tâche est particulièrement difficile car elle requiert une classification très précise de chacun des descripteurs de l'image I. La modélisation de la densité des descripteurs d'une image, introduite au paragraphe précédent, va nous permettre pour chaque descripteur d'une nouvelle image I de calculer sa probabilité *a posteriori* d'appartenir à l'objet O. Il sera ensuite aisé de décider pour chaque descripteur s'il appartient ou non à l'objet étudié. Etant donnée la modélisation de la densité des descripteurs d'une image, présentée au paragraphe précédent, il est possible de calculer pour chaque descripteur d'une nouvelle image I la probabilité d'appartenir à l'objet étudié O.

Proposition 6.2.2. *La probabilité* a posteriori *que le descripteur x_j appartienne à l'objet O est égale à :*

$$P(x_j \in O|x_j) = \sum_{i=1}^{k} R_i \Psi_{ji}, \tag{6.1}$$

où $\Psi_{ji} = P(x_j \in C_i|x_j)$.

Démonstration. La modélisation présentée au paragraphe précédent permet d'écrire :

$$\begin{aligned} P(x_j \in O|x_j) &= \frac{\tau f^O(x_j)}{f(x_j)} \\ &= \sum_{i=1}^{k} R_i \frac{\pi_i f(x_j, \theta_i)}{f(x_j)}. \end{aligned}$$

Or, la formule de Bayes implique que $P(x_j \in C_i|x_j, \theta) = \pi_i f(x_j, \theta_i)/f(x_j)$ et cela permet de conclure la démonstration. □

Remarque 6.2.1. Nous avons observé que l'algorithme d'optimisation permettant d'estimer le paramètre R convergeait vers une solution très proche de l'estimateur des moindres carrés $\hat{R}_{MC}$ déduit de (6.1) et donné par :

$$\hat{R}_{MC} = \left(\Psi^t \Psi\right)^{-1} \Psi^t \Phi, \tag{6.2}$$

où $\Phi_j = P(x_j \in O|x_j)$. Dans les expérimentations présentées au paragraphe 6.3, nous utiliserons l'estimateur des moindres carrés du vecteur de paramètre R afin de réduire les temps de calculs.

Remarque 6.2.2. On peut alors remarquer que le calcul de l'estimateur des moindres carrés $\hat{R}_{MC}$ dépend de la quantité $\Phi_j = P(x_j \in O|x_j)$ et que celle-ci s'exprime également en fonction de R *via* (6.1). Il nous a alors semblé intéressant de mettre en place une stratégie récursive d'estimation de R et des Φ_j mais cet algorithme a fourni des estimations stables de ces quantités dès la première itération.

La dépendance du processus d'estimation de R vis à vis du type de supervision de l'apprentissage apparaît dans l'expression de l'estimateur $\hat{R}_{MC}$ donnée par l'équation (6.2) au travers de la probabilité

$\Phi_j = P(x_j \in O|x_j)$. On supposera que $P(x_j \in O|x_j) = 1$ pour tout descripteur x_j positif et $P(x_j \in O|x_j) = 0$ sinon. Le choix du type de supervision se répercutera alors au travers des labels positifs et négatifs des descripteurs. En pratique, la probabilité *a posteriori* que le descripteur x appartienne à l'objet O est estimée par $\sum_{i=1}^{k} \hat{R}_i P(x \in C_i|x, \hat{\theta}_i)$ où $\hat{R}_i$ et $\hat{\theta}_i$, pour $i = 1, ..., k$, ont été estimés durant la phase d'apprentissage. L'objet O peut ensuite être localisé précisément dans l'image I en se basant uniquement sur les points d'intérêts ayant les plus grandes probabilités d'appartenir à l'objet. Une des manières les plus simples de préciser la localisation de l'objet dans une image est de dessiner un cadre (on retrouvera dans la littérature le terme de *bounding box*) autour de l'objet. Dans les expériences qui seront présentées au paragraphe 6.3, des *bounding boxes* seront utilisées pour localiser l'objet d'intérêt dans les images de validation[1] et cela nous permettra de comparer nos résultats à ceux de la littérature. Pour déterminer ce cadre pour une image I, nous aurons besoin de calculer la moyenne et la variance des coordonnées de points d'intérêts de l'image pondérées par les probabilités *a posteriori* données par la proposition 6.2.2. La moyenne sera alors le centre de la *bounding box* et sa taille sera réglée par rapport à la variance. Notons que cette approche a l'avantage de tenir compte de la probabilité d'appartenir à l'objet de chacun des points d'intérêt de l'image mais ne permet pas de détecter plusieurs instances d'un même objet dans une image.

Classification d'images

La classification d'images consiste à décider pour une nouvelle image I, *i.e.* différente des images d'apprentissage, si elle contient ou non l'objet O étudié. Le cadre probabiliste que nous avons introduit au cours de ce chapitre ne permet pas de fournir pour l'image I sa probabilité de contenir l'objet O. Néanmoins, nous proposons d'utiliser les probabilités d'appartenir à l'objet des descripteurs de I pour établir un score $S(I)$ qui permettra de décider si l'image I contient l'objet O ou non. Le score $S(I) \in [0, 1]$ que l'image I contienne l'objet O est défini par :

$$S(I) = \frac{1}{m} \sum_{x \in I} P(x \in O|x),$$

où m est le nombre de descripteurs extraits dans l'image I. On décidera ensuite que l'image I contient au moins une instance de l'objet O si son score $S(I)$ est supérieur à un seuil donné.

6.3 Résultats expérimentaux

Les expérimentations présentées dans ce chapitre, menées sur données réelles, permettront d'une part de valider l'approche probabiliste proposée et d'autre part de comparer les performances de nos méthodes de classification des données de grande dimension aux méthodes classiques dans le cadre d'une application réelle. Après avoir décrit les deux jeux d'images utilisés et le protocole d'étude adopté, nous présenterons et commenterons les résultats de localisation d'objets et de classification

[1] en vision par ordinateur, le jeu d'images permettant d'évaluer les méthodes est généralement appelé « jeu de test ».

Apprentissage Validation

FIG. 6.4 – Echantillon d'images d'apprentissage et de validation de la base d'images *Graz* [67].

Vélo Humain Voiture Moto

FIG. 6.5 – Echantillon d'images d'apprentissage et de validation de la base d'images *Pascal* [67] .

d'images. Nous comparerons également notre approche aux méthodes ayant pris part au *Challenge Pascal* [22], qui peuvent être considérées comme les méthodes de reconnaissance d'objets les plus performantes à l'heure actuelle.

6.3.1 Bases de données utilisées

Pour nos expérimentations, nous avons utilisé les bases d'images *Graz* [67] et *Pascal* [22] qui ont été proposées récemment. Un échantillon des images contenues dans ces bases est présenté par les figures 6.4 et 6.5.

Base d'images *Graz*

Le jeu d'images *Graz*[2] a été proposé en 2004 et contient deux catégories d'objets : l'objet « humain » et l'objet « vélo ». Elle est composée de 200 images d'apprentissage et de 100 images de validation. La résolution de chaque image de la base est 640×480 et les images sont converties en échelle de gris afin de leur appliquer les algorithmes d'extraction des descripteurs. La segmentation des images de cette base est disponible uniquement pour la catégorie « vélo ». Par conséquent, l'évaluation de la localisation d'objet n'est possible que pour cette catégorie et nous ne considérerons que cette tâche dans nos expérimentations sur ce jeu de données.

Base d'images *Pascal*

Le jeu d'images *Pascal* a été proposé en 2005 à l'occasion d'un concours de reconnaissance d'objets proposé par le réseau d'excellence Pascal[3]. La base d'images *Pascal* comporte quatre catégories d'objets : « moto », « vélo », « humain » et « voiture ». Elle est composée de 684 images d'apprentissage et de deux jeux de validation : le jeu *test1* qui comporte 689 images et le jeu *test2* qui en comporte 956. La résolution de chaque image de la base est 640×480 et les images sont converties en échelle de gris pour l'extraction des descripteurs. Les images du jeu *test1* sont du même type que les images d'apprentissage, *i.e.* les objets sont de même taille et dans des poses similaires. Par conséquent, ce jeu de validation est considéré comme un jeu « facile ». En revanche, les images du jeu de validation *test2* sont issues du moteur de recherche « Google Image » et sont par conséquent très différentes des images utilisées par l'apprentissage. La figure 6.5 met en évidence la différence de nature des images entre le jeu d'apprentissage et le jeu de validation *test2*. La reconnaissance des objets de ce second jeu de validation peut donc être considérée comme une tâche bien plus difficile. Une difficulté supplémentaire vient du fait que bon nombre des images de *test2* contiennent des instances de plusieurs catégories d'objets. Pour cette base d'images, les *bouding boxes* sont disponibles pour toutes les catégories d'objets et nous avons donc pu évaluer notre approche pour les quatre catégories d'objets. Ainsi, dans le cadre supervisé, les points d'intérêts situés dans la *bouding box* ont été référencés comme positifs.

6.3.2 Protocole expérimental

Nous avons mis en pratique notre approche probabiliste de la reconnaissance d'objets sur les deux bases d'images que nous venons de décrire et selon le même protocole expérimental. Nous allons maintenant présenter les protocoles expérimentaux relatifs à l'extraction des descripteurs et à la comparaison des différentes méthodes de classification.

[2] disponible à l'adresse *http ://www.emt.Graz.at/~pinz/data/*.
[3] site web : *http ://www.pascal-network.org*.

Extraction des descripteurs

Les descripteurs locaux ont été extraits de chaque image de la base en utilisant le détecteur Harris-Laplace (HL) [61], détectant les points d'intérêt de l'image, et le descripteur SIFT [56], fournissant une représentation invariante à l'échelle du voisinage de chacun des points d'intérêt de l'image. Nous avons choisi d'utiliser le descripteur SIFT car il a été plébiscité par l'étude comparative [60], menée par Mikolajczyk et Schmid [60], comme étant particulièrement adapté à la reconnaissance d'objets et est depuis utilisé dans de nombreux travaux [28, 67, 94]. L'utilisation de ce protocole d'extraction des données (HL+SIFT) conduit à l'obtention de, en moyenne, 200 descripteurs locaux de dimension 128 pour chaque image de la base.

Comparaison des méthodes de classification

Notre approche probabiliste autorisant l'utilisation de toute méthode de classification basée sur le modèle de mélange gaussien, cela nous a permis d'utiliser à la fois la méthode de classification basée sur le modèle $[a_{ij}b_iQ_id_i]$ et ses sous-modèles, mais aussi les méthodes basées sur les modèles gaussiens classiques. Les méthodes de classification étudiées sont basées sur les modèles gaussiens suivants : diag-GMM, sphe-GMM et PCA+diag-GMM. Le modèle PCA+diag-GMM est en fait la combinaison d'une étape de réduction de dimension avec une étape de classification basée sur le modèle diag-GMM. Pour toutes ces méthodes, les paramètres ont été estimés, de façon classique, grâce à l'algorithme EM qui a été initialisé à partir de la même partition. L'HDDC s'est d'ailleurs révélée ne pas être trop sensible à l'initialisation puisque nous avons noté pas plus d'un pourcent de variabilité des résultats de classification pour 50 initialisations différentes. D'autre part, le seuil permettant d'estimer les dimensions intrinsèques des classes pour l'HDDC a été déterminé par validation croisée sur le jeu d'apprentissage de deux bases d'images. La valeur qui a été retenue est 0.2 et cela a conduit à une estimation de la dimension moyenne $\bar{d} = \frac{1}{k}\sum_{i=1}^{k} d_i$ égale à 10. Enfin, le nombre de groupes formés par les méthodes de classification a été fixé à 40 et 50 pour respectivement les bases d'images *Graz* et *Pascal*. Ainsi, nous avons pu comparer l'HDDC et les méthodes classiques de classification dans un même cadre d'étude, celui de notre approche probabiliste de la reconnaissance d'objets, et sur des bases de données réelles de grande taille.

Mesures de performance

Pour la classification d'images, la mesure qui a été retenue est l'*equal error rate* (EER) qui est la valeur de la courbe ROC pour laquelle les taux de vrais positifs et de vrais négatifs sont égaux. Rappelons que la courbe ROC (*Receiver Operating Characteristic*) permet d'évaluer la performance d'un classifieur en mesurant sa sensibilité et sa spécificité (voir annexe A). Nous utiliserons également l'aire sous cette courbe (AUC), qui en est une bonne statistique, pour comparer les différentes méthodes. Pour la localisation d'objets, nous utiliserons la mesure *Average Precision* (AP) introduite dans [22]

qui est la moyenne arithmétique de 11 mesures le long de la courbe « précision-rappel » calculée à partir des *bounding-boxes* prédites (voir l'annexe A pour une définition de la courbe « précision-rappel »).

6.3.3 Résultats de localisation

Dans ce paragraphe, nous allons présenter les résultats de localisation d'objets obtenus sur les bases d'images *Graz* et *Pascal*. Les expérimentations menées sur la la base d'image *Graz* sont toutes faiblement supervisées.

Résultats de localisation d'objets sur la base *Graz*

Nous allons tout d'abord vérifier la validité de notre approche qui calcule pour chacun des points d'intérêt d'une image sa probabilité d'appartenir à l'objet étudié et ensuite utilise les points les plus probables d'appartenir à l'objet pour le localiser. La figure 6.6 montre la localisation de l'objet « vélo » sur la version segmentée d'une image de la base de validation et ce en fonction du seuil sur les probabilités d'appartenir à l'objet étudié. La vignette de gauche montre l'ensemble des descripteurs de l'image et l'on observe que la majorité d'entre eux ne sont pas localisés sur le vélo. La seconde et la troisième vignette montrent les descripteurs de l'image ayant respectivement des probabilités d'appartenir à l'objet supérieures à 0.5 et 0.65. On peut observer sur ces deux vignettes que le fait de ne retenir que les descripteurs ayant les probabilités les plus fortes permet d'affiner la localisation du vélo. Enfin, la vignette de droite présente la localisation de l'objet obtenue en n'utilisant que les descripteurs ayant des probabilités d'appartenir à l'objet supérieures à 0.7 et cette localisation est sans erreur. Cela signifie d'une part que notre approche probabiliste est efficace et, d'autre part, que l'étape d'identification des parties discriminantes de l'objet, basée sur les paramètres R_i, est efficace et permet de fournir une localisation très satisfaisante de l'objet étudié.

La figure 6.7 présente les résultats de la localisation de l'objet « vélo » en fonction du seuil sur les probabilités $P(x \in O|x)$ pour l'HDDC et les méthodes de classification basées sur les modèles gaussiens classiques. La mesure utilisée ici est la précision, *i.e.* le pourcentage de descripteurs locaux localisés sur l'objet dont la probabilité d'appartenir à l'objet « vélo » est plus grande qu'un certain seuil. A gauche de l'image, l'ensemble des descripteurs de chaque image est utilisé pour calculer la précision de la localisation et, par conséquent, toutes les méthodes de classification obtiennent le même résultat. A l'inverse, à droite de l'image, uniquement les points d'intérêt dont la probabilité d'appartenir à l'objet est supérieure à 0.95 sont classés comme appartenant à l'objet. On observe que, dans le cadre de notre approche probabiliste, l'HDDC, basée respectivement sur les modèles $[a_{ij}b_iQ_id_i]$, $[a_{ij}b_iQ_id_i]$, $[a_{ij}b_iQ_id_i]$ et $[a_{ij}b_iQ_id_i]$, permet de localiser efficacement l'objet étudié en se basant sur les descripteurs les plus probables. En effet, la croissance des courbes de précision associées à l'HDDC confirme ce que nous avons observé sur la figure 6.6. Cela signifie que l'HDDC a réussi à regrouper les données de grande dimension en groupes homogènes et discriminants de l'objet et du fond. En particulier, le modèle $[a_ib_iQ_id_i]$ permet à notre approche d'obtenir une précision de localisation égale à 92% en considérant uniquement les descripteurs tels que $P(x_j \in O|x_j) \geq 0.9$ (ce

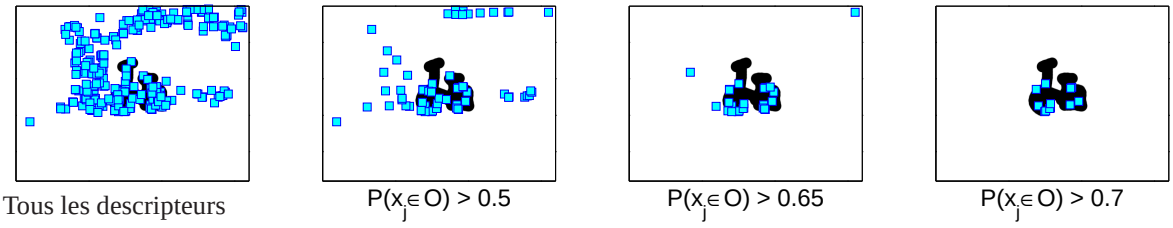


FIG. 6.6 – Localisation faiblement supervisée sur la base *Graz* : localisation de l'objet « vélo » par seuillage sur les probabilités $P(x_j \in O|x_j)$ avec la méthode de classification HDDC (modèle $[a_i b_i Q_i d_i]$).

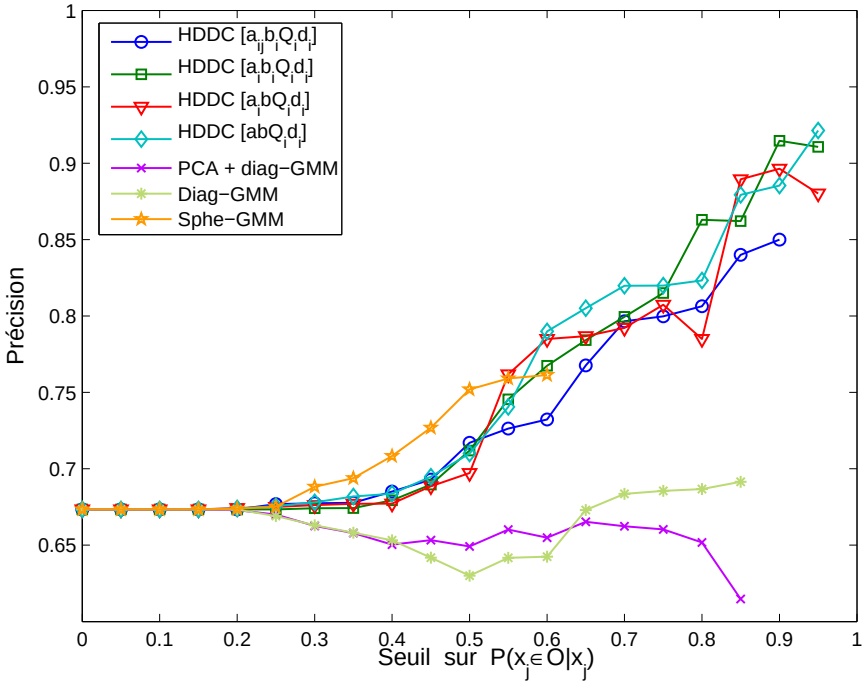


FIG. 6.7 – Localisation faiblement supervisée sur la base *Graz* : résultats de localisation de l'objet « vélo » en fonction du seuil sur les probabilités $P(x_j \in O|x_j)$ pour l'HDDC et les méthodes de classification basées sur les modèles gaussiens classiques.

Méthode	Précision	AP
HDDC $[a_{ij}b_iQ_id_i]$	0.85	0.796
HDDC $[a_{ij}bQ_id_i]$	0.83	0.748
HDDC $[a_ib_iQ_id_i]$	**0.92**	**0.833**
HDDC $[a_ibQ_id_i]$	0.89	0.758
HDDC $[abQ_id_i]$	0.88	0.785
PCA+diag-GMM	0.63	0.755
Diag-GMM	0.70	0.777
Sphe-GMM	0.76	0.754
Résultat de [28]	0.62	/

TAB. 6.1 – Localisation faiblement supervisée sur la base *Graz* : précision de la localisation calculée sur les 10 descripteurs de chaque image ayant les plus grandes probabilités d'appartenir à l'objet « vélo ». La mesure AP est calculée à partir des *bounding boxes* prédites (voir le texte pour plus de détails).

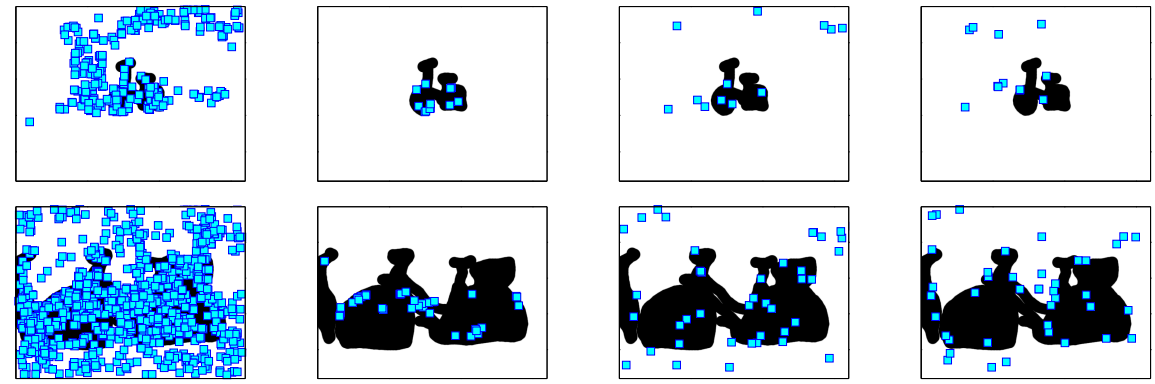


FIG. 6.8 – Localisation faiblement supervisée sur la base *Graz* : résultats de localisation affichés sur les versions segmentées des images de validation. Les points d'intérêts affichés sont ceux ayant les probabilités d'appartenir à l'objet « vélo » les plus grandes. Le même nombre de points est affiché pour chacun des modèles (5% du nombre total de détections par image).

qui correspond à une sélection d'environ 10 points par image). D'autre part, les méthodes basées sur les modèles gaussiens classiques sphe-GMM, diag-GMM et PCA+diag-GMM ne permettent pas de localiser aussi efficacement le vélo que l'HDDC. En effet, notre approche probabiliste combinée au modèle sphe-GMM n'attribue pas de valeur de $P(x_j \in O|x_j)$ plus grande que 0.7. Cela est certainement du au fait que ce modèle gaussien est particulièrement parcimonieux et ne peut donc modéliser correctement les données considérées ici qui sont relativement complexes. La stationnarité globale des courbes de précision associées aux méthodes basées sur les modèles diag-GMM et PCA+diag-GMM traduit le fait que ces deux méthodes n'ont pas fourni une partition des données en groupes homogènes et discriminants. Par conséquent, la localisation basée sur les points d'intérêts ayants les plus forte probabilité d'appartenir à l'objet n'est pas meilleure que la localisation utilisant tous les descripteurs.

Le tableau 6.1 donne les résultats de la localisation de l'objet « vélo » basée sur les 10 descripteurs de chaque image les plus probables obtenus sur cette base d'images avec les différentes méthodes de classification étudiées et dans le cadre de notre approche probabiliste. Nous y avons ajouté les résultats de localisation obtenus sur cette même base d'images par Dorko et Schmid [28] dont l'approche est non probabiliste mais identifie également les groupes discriminants et utilise le modèle gaussien diag-GMM pour l'étape de clustering. On remarque tout d'abord que notre approche probabiliste permet une localisation bien plus efficace que l'approche de Dorko et Schmid. D'autre part, le modèle $[a_{ij}b_iQ_id_i]$ s'avère être le plus efficace pour reconnaître l'objet « vélo » parmi les modèles gaussiens dans le cadre de notre approche probabiliste. La figure 6.8 présente les résultats de la localisation de l'objet « vélo » sur quelques images de validation de la base *Graz* et ce pour les différentes méthodes de classification. Enfin, la figure 6.9 montre la localisation finale obtenue avec notre approche probabiliste et l'HDDC sur deux images de validation. La *bounding box* tracée en rouge a été calculée en pondérant chacun des points détectés par sa probabilité d'appartenir à l'objet. Ces deux figures mettent en évidence l'aptitude de notre approche, combinée à l'HDDC, à localiser des objets de taille différentes dans des images.

Résultats de localisation d'objets sur la base *Pascal*

Le tableau 6.2 présente un résumé des résultats de localisation supervisée et faiblement supervisée. Les valeurs reportées dans ce tableau sont les moyennes des mesures AP sur les 4 catégories d'objets (« moto », « vélo », « humain » et « voiture ») et ce pour le deux jeux de validation de la base *Pascal*. Les résultats détaillés sont présentés à l'annexe B. Les résultats sont donnés en fonction de la méthode de clustering utilisée dans le cadre de notre approche probabiliste. Nous avons également reporté dans le tableau 6.2 les résultats de la meilleure méthode du *Challenge Pascal* [22]. Notons que, lors du *Challenge Pascal*, la localisation faiblement supervisée n'a pas été envisagée. L'ordre de grandeur des valeurs du tableau met en évidence la difficulté de localiser les objets dans les images de cette base. On observe tout d'abord que notre approche probabiliste combinée à l'utilisation de l'HDDC fournit en moyenne de meilleurs résultats que ceux de la méthode ayant remporté le *Chal-*

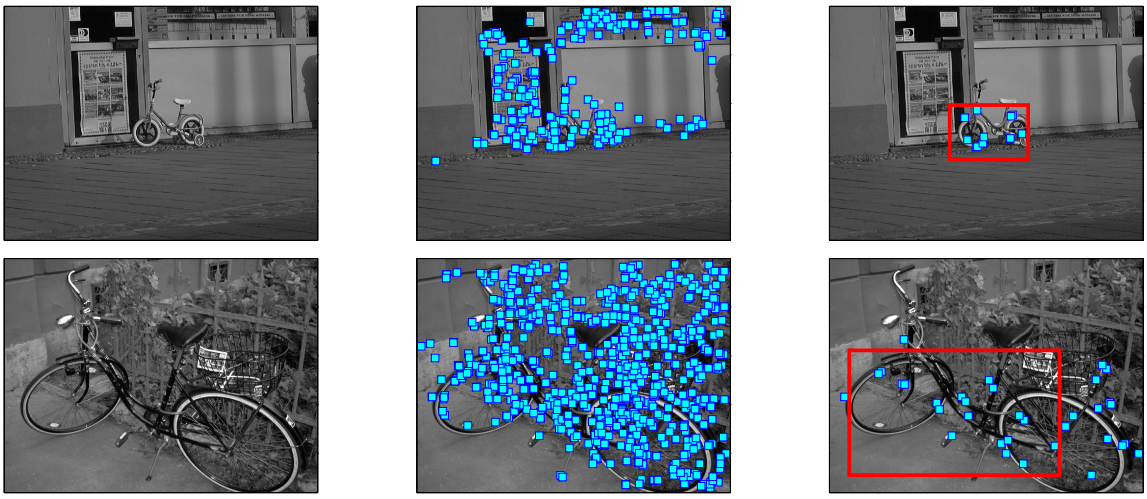


Image originale | Tous les descripteurs | Localisation finale

FIG. 6.9 – Localisation faiblement supervisée sur la base *Graz* : résultats de localisation de l'objet « vélo » obtenus avec l'HDDC (modèle $[a_i b_i Q_i d_i]$) dans le cadre de notre approche probabiliste et ce sur deux images de validation.

Jeu de validation	*Pascal test1*		*Pascal test2*	
Supervision	supervisé	faibl. sup.	supervisé	faibl. sup.
HDDC $[a_{ij} b_i Q_i d_i]$	0.302	0.273	0.172	0.145
HDDC $[a_{ij} b Q_i d_i]$	**0.318**	**0.287**	0.181	0.147
HDDC $[a_i b_i Q_i d_i]$	0.313	0.285	**0.183**	0.142
HDDC $[a_i b Q_i d_i]$	**0.318**	0.283	0.176	**0.148**
HDDC $[a b Q_i d_i]$	0.317	0.282	0.172	0.134
HDDC $[a_i b_i Q_i d]$	0.314	**0.287**	0.179	0.130
HDDC $[a b Q_i d]$	0.300	0.285	0.185	0.139
HDDC $[a b Q d]$	0.256	0.229	0.159	0.106
PCA+diag-GMM	0.257	0.215	0.177	0.120
Diag-GMM	0.271	0.216	0.149	0.106
Sphe-GMM	0.276	0.227	0.161	0.110
Com-GMM	0.267	0.246	0.164	0.116
K-means	0.261	0.204	0.160	0.099
Meilleure méthode de [22]	0.279	/	0.112	/

TAB. 6.2 – Localisation supervisée et faiblement supervisée sur la base *Pascal* : les résultats présentés ici sont les moyennes sur les 4 catégories des mesures AP (voir le texte pour plus de détails).

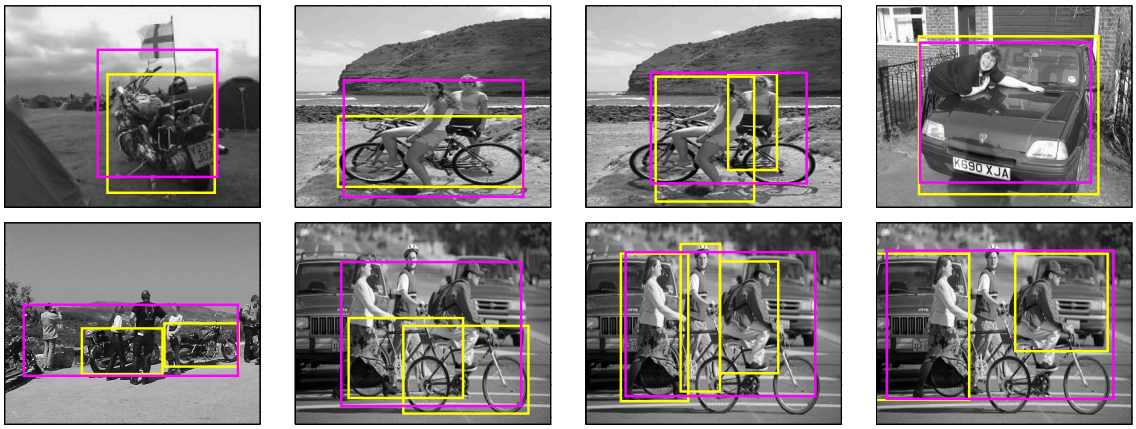


FIG. 6.10 – Localisation supervisée sur le jeu *test2* de la base *Pascal* : les *bounding boxes* prédites sont tracées en rouge et les *bounding boxes* réelles sont en jaune.

lenge Pascal et ce sur les deux jeux de validation de la base. L'observation des résultats détaillés nous indique que notre méthode domine les compétitions portant sur les objets « vélo » et « humain » pour le jeu de validation *test1* et celles portant sur les objets « vélo », « humain » et « moto » pour le jeu de validation *test2*. Ces résultats sont d'autant plus remarquables que notre approche ne permet ni de détecter plusieurs instances d'un même objet dans une image, ni de modéliser les relations spatiales des descripteurs, alors que certaines méthodes ayant participé au *Challenge Pascal* le peuvent. On observe également que, dans le cadre de notre approche probabiliste, la méthode de clustering HDDC, basée sur le modèle $[a_{ij}b_iQ_id_i]$ et ses sous-modèles, améliore significativement les résultats par rapport aux méthodes basées sur le modèles gaussiens classiques et à la méthode k-means. En particulier, les modèles $[a_{ij}bQ_id_i]$ et $[a_ibQ_id_i]$ apparaissent comme particulièrement adaptés aux données de grande dimension de cette base. Le fait que ce soient des modèles à b_i communs qui fournissent les meilleurs résultats peut s'expliquer par le fait que les données d'apprentissage ont été acquises de la même façon et, par conséquent, il est raisonnable de penser que le bruit, modélisé par le paramètre b_i, est commun. Enfin, il est tout particulièrement intéressant de remarquer que, dans le cadre de notre approche probabiliste, les résultats de localisation supervisée et faiblement supervisée sont relativement proches. Cela est très encourageant puisque l'annotation manuelle des bases d'images est très coûteuse. En outre, les résultats de localisation obtenus par notre approche probabiliste dans le cadre faiblement supervisé sont meilleurs en moyenne que la méthode la plus performante du *Challenge Pascal* dans le cadre supervisé.

La figure 6.10 présente des résultats de localisation supervisée sur des images du jeu de validation *test2*. Le cadre prédit par notre approche probabiliste combinée à l'utilisation de l'HDDC est tracé en rouge et le cadre réel, fourni par la supervision manuelle, est dessiné en jaune. On observe que pour les images du haut de la figure, qui présentent une difficulté de localisation moyenne, la localisation fournie par notre méthode est particulièrement efficace. D'autre part, pour les images du bas de la

figure, qui contiennent plusieurs instances d'un même objet et qui présentent de ce fait une difficulté de localisation élevée, la localisation fournie par notre méthode englobe les différentes instances de l'objet mais ne parvient pas à les discerner.

6.3.4 Résultats de classification d'images

Nous allons à présent évaluer notre approche probabiliste pour la tâche de la classification d'images et comparer les différentes méthodes de *clustering* dans ce cadre. Nous allons présenter, dans ce paragraphe, les résultats de classification d'images obtenus sur les bases d'images *Graz* et *Pascal*.

Résultats expérimentaux obtenus sur la base *Graz*

Nous rappelons que les expérimentations menées sur la la base d'image *Graz* sont toutes faiblement supervisées. Le tableau 6.3 présente les résultats de classification d'images pour différentes méthodes de *clustering*, dans le cadre de notre approche probabiliste, et d'autres méthodes de classification d'images issues de la littérature. On observe tout d'abord que le modèle $[a_i b_i Q_i d_i]$ de l'HDDC permet à notre approche probabiliste d'obtenir des résultats de classification meilleurs à la fois que les modèles gaussiens classiques dans le cadre de notre approche et que les méthodes de Dorko et Schmid [28], Opelt *et al.* [67] et Zhang *et al.* [94]. On remarque à nouveau que l'approche probabiliste que nous avons proposée permet d'améliorer significativement les résultats de classification d'image par rapport à l'approche non probabiliste de Dorko et Schmid. La base de validation contenant 50 images de vélo, cela signifie que notre méthode probabiliste (modèle $[a_i b_i Q_i d_i]$) ne fait que 3 erreurs. Les images du bas de la figure 6.11 sont les 3 images contenant l'objet « vélo » qui ont été catégorisées comme ne contenant pas de vélo par notre approche (modèle $[a_i b_i Q_i d_i]$). Cette figure présente également les 3 images de la base de validation ayant le plus grand score S (voir paragraphe 6.2.3). Il est intéressant de noter que la première image mal classée est l'image qui contient le petit vélo et sur laquelle la localisation de l'objet était pourtant particulièrement efficace. Cela indique que le score probabiliste par image, que nous avons proposé pour décider si une image contient ou non l'objet étudié, est certes globalement efficace mais n'est pas suffisamment précis pour détecter des objets petits à l'intérieur de l'image.

Résultats expérimentaux obtenus sur la base *Pascal*

Le tableau 6.4 présente les résultats de classification d'images obtenus sur les deux jeux de validation de la base *Pascal* dans les cadres supervisés et faiblement supervisés. Les valeurs du tableau correspondent aux moyennes des mesures EER sur les 4 catégories d'objets et sont données pour chacune des méthodes de *clustering* utilisées dans le cadre de notre approche probabiliste. Nous y avons également reporté les résultats obtenus par la méthode de Zhang *et al.* [94] qui est la meilleure méthode du *Challenge Pascal* [22]. Les résultats détaillés, catégorie par catégorie, sont donnés à l'annexe B. L'observation du tableau 6.4 met en évidence que notre approche n'est pas globalement aussi efficace que

Méthode	EER	AUC
HDDC $[a_{ij}b_iQ_id_i]$	0.88	0.945
HDDC $[a_{ij}bQ_id_i]$	0.90	0.950
HDDC $[a_ib_iQ_id_i]$	**0.94**	**0.976**
HDDC $[a_ibQ_id_i]$	0.92	0.959
HDDC $[abQ_id_i]$	0.88	0.956
PCA+diag-GMM	0.89	0.959
Diag-GMM	0.88	0.955
Sphe-GMM	0.88	0.944
Résultat de [94]	0.92	/
Résultat de [67]	0.87	/
Résultat de [28]	0.84	/

TAB. 6.3 – Classification d'image faiblement supervisée sur la base *Graz* : *equal-error rates* (EER) et aire sous la courbe (AUC) pour la catégorie d'objet « vélo ».

FIG. 6.11 – Classification d'image faiblement supervisée sur la base *Graz* : en haut, les 3 images de validation ayant le plus grand score S de classification et, en bas, les 3 images de validation qui ont été classées comme ne contenant pas l'objet « vélo ». Résultats obtenus avec notre score probabiliste S et le modèle $[a_ib_iQ_id_i]$ de l'HDDC.

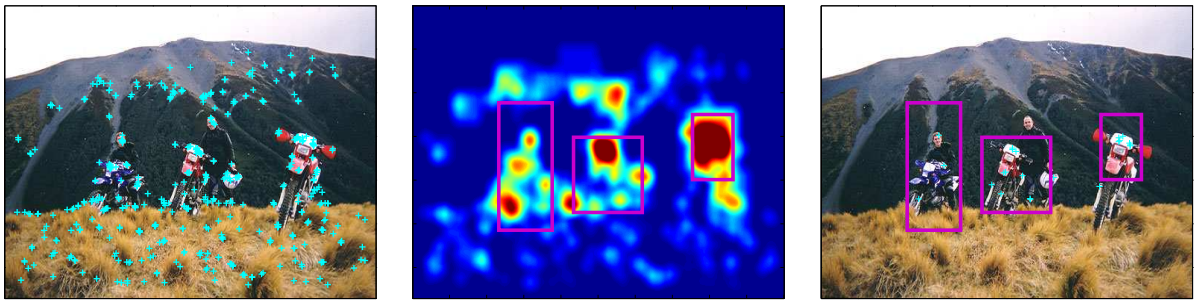
Jeu de validation	*Pascal test1*		*Pascal test2*	
Supervision	supervisé	faibl. sup.	supervisé	faibl. sup.
HDDC $[a_{ij}b_iQ_id_i]$	0.822	0.865	0.714	0.708
HDDC $[a_{ij}bQ_id_i]$	0.841	0.870	0.716	0.712
HDDC $[a_ib_iQ_id_i]$	0.826	0.871	0.720	0.709
HDDC $[a_ibQ_id_i]$	0.839	0.874	0.720	0.705
HDDC $[abQ_id_i]$	0.845	0.863	0.727	0.714
HDDC $[a_ib_iQ_id]$	0.853	0.887	0.727	0.712
HDDC $[abQ_id]$	**0.860**	0.885	**0.735**	0.719
HDDC $[abQd]$	0.787	0.809	0.689	0.682
PCA+diag. GMM	0.780	0.843	0.695	0.692
Diag-GMM	0.775	0.835	0.690	0.687
Sphe-GMM	0.797	0.852	0.682	0.682
Com-GMM	0.818	0.843	0.707	0.699
K-means	0.778	0.827	0.687	0.676
Meilleure méthode de [22]	/	**0.937**	/	**0.741**

TAB. 6.4 – Classification d'images supervisée et faiblement supervisée sur la base *Pascal* : les résultats présentés ici sont les moyennes sur les 4 catégories des mesures EER (voir le texte pour plus de détails).

la meilleure méthode du *Challenge Pascal* dans le cadre faiblement supervisé. En menant une analyse plus détaillée des résultats, on s'aperçoit tout de même que notre approche remporte deux compétitions (« vélo » et « humain ») sur le jeu de validation *test2*. La raison de cette contre-performance de notre approche est que notre score probabiliste de classification n'est pas suffisamment précis. Nous l'avons d'ailleurs déjà noté lors de l'étude des résultats obtenus sur la base *Graz*. En outre, la meilleure méthode du *Challenge Pascal* utilise une combinaison de deux descripteurs d'images alors que nous n'en utilisons qu'un. En revanche, dans le cadre de notre approche probabiliste, l'HDDC domine les méthodes de *clustering* basées sur les modèles gaussiens classiques et la méthode non paramétrique k-means.

6.4 Discussion

Nous avons présenté dans ce chapitre une approche probabiliste de la reconnaissance d'objets qui permet, notamment, d'exploiter au mieux les résultats de la méthode de clustering HDDC. Cette approche a montré son efficacité à la fois en localisation d'objets et en classification d'images. Les résultats de localisation d'objets sont, en particulier, très prometteurs car notre approche fournit des résultats meilleurs que les méthodes actuelles les plus performantes. En outre, ces résultats ont été obtenus en utilisant un seul type de détecteur / descripteur et sans prendre en compte les relations spatiales existantes entre descripteurs d'une même image. D'autre part, la mise en œuvre de notre approche probabiliste a mis en évidence que sa principale limite est le fait de ne pouvoir localiser



Toutes les détections | Carte de probabilité | Localisation finale

FIG. 6.12 – Localisation supervisée de plusieurs instances d'un même objet sur le jeu *test2* de la base *Pascal*.

qu'une unique instance d'un même objet dans une image. Cela a notamment pénalisé notre approche pour la localisation d'objets et la classification d'image sur la base *test2* dont les images contiennent le plus souvent plusieurs instances d'un même objet. Une extension naturelle de ce travail serait d'utiliser les probabilités *a posteriori* d'appartenir à l'objet de chacun des descripteurs d'une image pour localiser les différentes instances de cet objet. A partir de ces informations, il est par exemple possible de construire une carte de probabilité qui mette en évidence la localisation probable des instances de l'objet. La figure 6.12 présente une telle carte obtenue en combinant les informations de probabilité d'appartenir à l'objet, de localisation et d'échelle qui sont disponibles *a posteriori* pour chaque point d'intérêt de l'image. On observe que cette carte donne déjà une idée de la localisation des différentes instances de l'objet. Il est ensuite possible de réaliser un *clustering* spatial sur les points les plus probables pour localiser les K instances de l'objet.

Chapitre 7

Conclusion et perspectives

Dans ce dernier chapitre, nous produirons tout d'abord une synthèse des travaux qui ont été présentés dans ce mémoire. Le paragraphe 7.1 récapitulera les principaux résultats et contributions de ce travail. Nous présenterons ensuite au paragraphe 7.2 les travaux actuels basés sur les résultats présentés dans ce mémoire. Enfin, le paragraphe 7.3 annoncera les thèmes de recherche connexes aux thèmes abordés au cours de ce travail que nous envisageons d'explorer.

7.1 Synthèse des travaux présentés dans ce mémoire

Ce paragraphe s'organise autour des deux principaux thèmes qui ont été abordés au cours de ce travail : la modélisation et la classification des données de grande dimension, d'une part, et l'application à la reconnaissance d'objets, d'autre part.

7.1.1 Modélisation et classification des données de grande dimension

Le principal thème d'étude de ce mémoire a été la modélisation et la classification des données de grande dimension. Nous allons à présent rappeler les principales contributions de cette thèse à ce thème d'étude.

Un panorama de la classification des données de grande dimension

Nous avons tout d'abord dressé un panorama des différentes approches existantes pour modéliser et classer les données de grande dimension et ce, à la fois dans le cadre supervisé et dans le cadre non supervisé. Il s'est avéré que l'approche probabiliste basée sur le modèle de mélange gaussien, dont la renommée n'est plus à faire, présente l'avantage de pouvoir être utilisée dans les cadres supervisés et non supervisés. Nous avons également pu observer que les espaces de grande dimension posent certaines difficultés aux méthodes de classification et que l'ensemble de ces difficultés peuvent être

réunies sous le terme générique de « fléau de la dimension ». Nous avons ensuite recensé les différentes solutions existantes pour permettre aux méthodes classiques de traiter des données de grande dimension. En outre, nous avons mis en évidence les « bienfaits de la dimension » : d'une part, les données de grande dimension vivent dans des sous-espaces de dimensions intrinsèques inférieures à la dimension de l'espace original et, d'autre part, il est plus facile de discriminer avec un classifieur adapté dans un espace de grande dimension que dans un espace de faible dimension.

Une famille de modèles gaussiens adaptés aux données de grande dimension

Après avoir dressé une analyse critique des approches existantes de classification des données de grande dimension, nous avons proposé une re-paramétrisation du modèle de mélange gaussien prenant en compte le fait que les données de grande dimension vivent dans des sous-espaces de dimensions intrinsèques inférieures à la dimension de l'espace original. Cette re-paramétrisation a donné naissance au modèle gaussien $[a_{ij}b_iQ_id_i]$, baptisé du nom de ses paramètres. De plus, en forçant certains paramètres à être communs dans une même classe ou entre les classes, nous avons mis à jour une famille de 28 modèles gaussiens adaptés aux données de grande dimension allant du modèle le plus général au modèle le plus parcimonieux. Enfin, les règles de décision associées à certains modèles de notre famille peuvent être interprétés d'un point de vue géométrique.

Les méthodes de classification HDDA et HDDC

Nous avons ensuite utilisé ces modèles gaussiens pour la discrimination et la classification automatique de données de grande dimension. Les classifieurs supervisés et non supervisés associés aux modèles gaussiens proposés dans ce mémoire ont été baptisés respectivement *High Dimensional Discriminant Analysis* (HDDA) et *High Dimensional Data Clustering* (HDDC). La construction du classifieur supervisé HDDA et du classifieur non supervisé HDDC a nécessité l'estimation par la méthode du *maximum* de vraisemblance des paramètres des différents modèles issus de notre re-paramétrisation du modèle de mélange gaussien. En outre, la nature de notre re-paramétrisation permet aux méthodes HDDA et HDDC de ne pas être perturbées par le mauvais conditionnement où la singularité des matrices de covariance empiriques des classes et d'être efficaces en terme de temps de calcul. Le problème de l'estimation des dimensions intrinsèques des classes a également été abordé et nous avons proposé d'utiliser la méthode du *scree-test* de Cattell pour les estimer. Ces méthodes de classification dédiées aux données de grande dimension on ensuite été mises en œuvre et évaluées sur des jeux de données réelles et simulées. Ces expérimentations ont montré que les méthodes de classification HDDA et HDDC présentent l'avantage de ne pas subir le « fléau de la dimension » et d'être performantes aussi bien dans des espaces de grande dimension que de faible dimension. Enfin, la comparaison sur données réelles avec les méthodes classiques de classification a montré l'efficacité de notre approche pour la modélisation et la classification des données de grande dimension.

7.1.2 Application à la reconnaissance d'objets

Nous nous sommes également intéressé au problème de la reconnaissance d'objets dans des images qui est un problème particulièrement intéressant et difficile. L'approche statistique s'est révélée être une bonne solution et a permis d'améliorer significativement les résultats expérimentaux. Nous allons à présent rappeler les principales contributions de ce mémoire à ce second thème d'étude.

Une approche probabiliste de la reconnaissance d'objets

Nous avons tout d'abord proposé une approche basée sur le modèle de mélange gaussien qui permette de localiser de manière probabiliste un objet dans une image inconnue. Cette approche modélise la densité de points d'intérêts d'une image par un mélange de parties dont certaines sont discriminantes de l'objet et d'autres du fond. Le pouvoir discriminant de chacune des parties apprises sur un jeu d'apprentissage est estimé par la méthode du *maximum* de vraisemblance. Cette approche probabiliste présente en outre l'avantage de pouvoir exploiter au mieux les résultats des méthodes génératives de classification. Il est donc en particulier possible de combiner cette approche aux méthodes de classification des données de grande dimension qui ont également été proposées dans ce mémoire. Cette approche utilisant nos modèles gaussiens adaptés aux données de grande dimension a ensuite été mise en œuvre sur deux bases d'images récentes et comparée aux meilleures méthodes de reconnaissance d'objets. Ces expérimentations ont montré d'une part que notre approche probabiliste est plus efficace que les méthodes existantes. D'autre part, elles ont aussi mis en évidence que, dans le cadre de notre approche, les modèles gaussiens adaptés aux données de grande dimension forment des groupes homogènes et discriminants et par conséquent permettent une meilleure localisation des objets dans les images que les modèles gaussiens classiques.

Une approche adaptée au cadre faiblement supervisé

La seconde contribution majeure de ce travail est la possibilité pour l'approche probabiliste proposée d'être utilisée soit dans un cadre supervisé, *i.e.* les objets sont segmentés dans les images d'apprentissage, soit dans un cadre faiblement supervisé, *i.e.* les objets ne sont pas segmentés dans les images d'apprentissage et l'on sait uniquement quelles images contiennent au moins une instance de l'objet. Les expérimentations menées sur les deux bases d'images ont mis en évidence que les résultats de reconnaissance d'objets obtenus dans le cadre faiblement supervisé sont presque aussi bons que ceux obtenus dans le cadre supervisé. Outre l'intérêt de ce résultat pour la reconnaissance d'objets, il ouvre la voie de manière plus générale à la classification faiblement supervisée et, de ce point de vue, notre approche a introduit un cadre formel pour la classification faiblement supervisée. Ce type de supervision pourrait par exemple être utilisée pour la classification de documents textuels.

Modèle	Brique	Moquette	Tissu	Sol 1	Sol 2	Marbre	Bois
Indep. + diag-GMM	0.776	0.316	0.583	0.283	0.588	0.339	0.586
Indep. + $[a_i b_i Q_i d_i]$	0.812	0.569	0.625	0.356	0.674	0.371	0.650
Markov + diag-GMM	0.964	0.810	0.797	0.830	0.834	0.461	0.958
Markov + $[a_i b_i Q_i d_i]$	1	0.939	0.982	0.853	0.996	0.480	0.999

TAB. 7.1 – Résultats de reconnaissance de textures : taux de classification correcte en fonction des différentes textures présentes dans la base.

7.2 Travaux en cours

Les modèles gaussiens proposés dans ce mémoire peuvent naturellement être utilisés dans des contextes très différents et même combinés à d'autres modélisations afin d'acquérir des caractéristiques supplémentaires. Dans ce paragraphe, nous allons présenter les travaux en cours où nos modèles gaussiens adaptés aux données de grande dimension entrent en jeu. Nous illustrerons notre propos avec quelques résultats préliminaires.

7.2.1 Classification de données de grande dimension spatialement corrélées

Dans de nombreux problèmes, par exemple en analyse d'images, les données sont à la fois de grande dimension et spatialement corrélées. Nous nous sommes intéressés à ce thème dans le contexte de la reconnaissance de textures. La reconnaissance de texture est un problème difficile pour les raisons suivantes : les observations sont en grande dimension, il existe une relation spatiale entre les observations et les textures étudiées ne sont pas homogènes. En effet, les points d'intérêts détectés sur chaque image de la base sont décrits par de vecteur de dimension 128 (*cf.* chapitre 6) et les méthodes de classification classiques ont des difficultés à traiter de telles données. D'autre part, il est naturel de penser qu'il existe une relation spatiale entre deux points d'intérêts voisins. Enfin, il n'est pas raisonnable de modéliser une texture par une unique densité car les textures sont généralement hétérogènes. Ce travail est réalisé en collaboration avec J. Blanchet (sous la direction de thèse de C. Schmid et de F. Forbes).

Données et protocole

Nous avons donc proposé de combiner les modèles gaussiens adaptés aux données de grande dimension, présentés dans ce mémoire, à une modélisation des relations spatiales des textures basée sur les champs de Markov cachés proposée dans [12]. En outre, nous avons considéré que chaque texture était composée de 10 parties homogènes et nous avons utilisé le graphe de Delaunay, dual du diagramme de Voronoï, comme système de voisinage. Pour nos expérimentations, nous avons utilisé une base composée de 7 textures (brique, moquette, tissu, sol 1, sol 2, marbre et bois) et qui comporte 10 images d'apprentissage pour chaque texture. La base de validation comporte quant à elle 250 images.

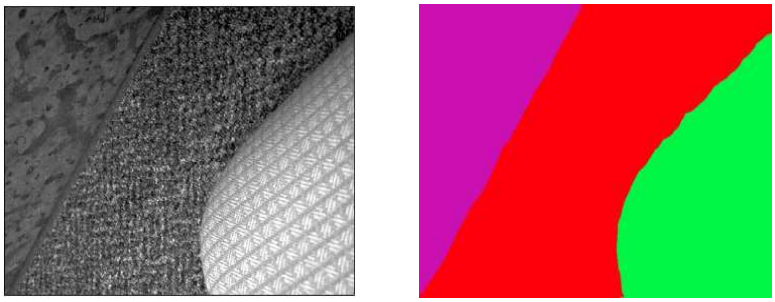


FIG. 7.1 – Segmentation d'une image composée de 3 textures différentes (moquette, tissu et sol 2) : à gauche, image originale et, à droite, image segmentée.

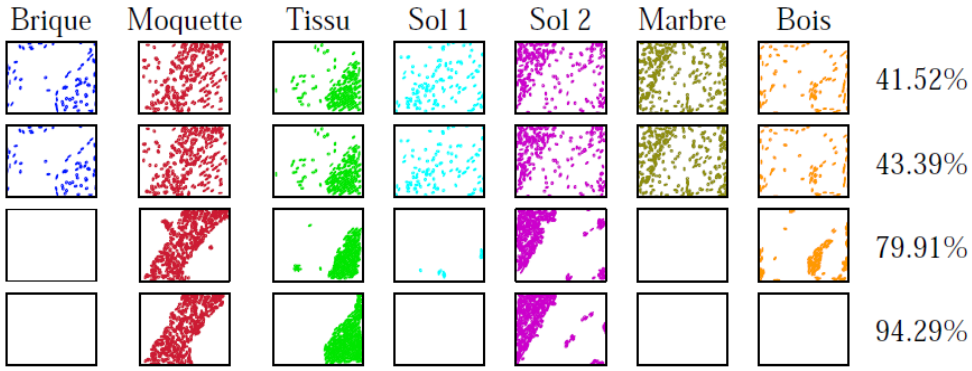


FIG. 7.2 – Segmentation d'une image composée de 3 textures différentes (moquette, tissu et sol 2) avec, de haut en bas : modèle indépendant et diag-GMM, modèle indépendant et $[a_i b_i Q_i d_i]$, champ de Markov caché et diag-GMM, champ de Markov caché et $[a_i b_i Q_i d_i]$.

Nous avons comparé sur cette base notre approche, qui sera notée « Markov + $[a_i b_i Q_i d_i]$ », aux approches « Indep. + diag-GMM », « Indep. + $[a_i b_i Q_i d_i]$ » et « Markov + diag-GMM ». La notation « Indep. » traduit le fait que l'approche ne prend pas en compte les relations spatiales et la notation « diag-GMM » indique que le modèle parcimonieux diag-GMM a été utilisé.

Résultats expérimentaux

Le tableau 7.1 présente les taux de classification correcte obtenus par les 4 approches étudiées. On observe tout d'abord que notre approche permet une reconnaissance particulièrement bonne et, en tout cas, nettement meilleure que les autres approches. On peut également remarquer que, même si la modélisation spatiale par champs de Markov cachés permet une grande amélioration des résultats par rapport au modèle indépendant, l'utilisation du modèle gaussien $[a_i b_i Q_i d_i]$ améliore également et de manière significative les résultats de reconnaissance de textures. La figure 7.1 montre, à gauche, une image comportant trois textures différentes et, à droite, la segmentation idéale. La figure 7.2 présente les segmentations obtenues sur cette image de validation avec les 4 approches étudiées. Il apparaît que

notre approche « Markov + $[a_i b_i Q_i d_i]$ » fournit une segmentation très satisfaisante et bien meilleure que les autres approches.

7.2.2 Catégorisation automatique du sol de la planète Mars

Nous nous sommes récemment intéressés au problème de la catégorisation d'images hyper-spectrales du sol de la planète Mars pour lequel les données sont à la fois de grande dimension et en très grand nombre. L'imagerie hyper-spectrale visible et infrarouge est une technique de télé-détection clef pour l'étude et le suivi des planètes du système solaire. Les spectromètres imageurs intégrés dans un nombre croissant de satellites génèrent des images hyper-spectrales à trois composantes (deux composantes spatiales et une spectrale). Au mois de mars 2004 l'instrument OMEGA (*Mars Express*, ESA) [8] avait déjà collecté 310 giga-octets de données brutes. Une nouvelle génération de spectromètres imageurs est en train d'émerger et est dotée d'une composante supplémentaire de mesure (angulaire) pour une meilleure caractérisation des matériaux planétaires et pour mieux séparer les signaux venant de l'atmosphère et de la surface. Les sites planétaires seront maintenant observés non seulement à la verticale mais aussi selon différents points de vue le long de la trajectoire du satellite. Le spectromètre imageur CRISM de l'orbiteur *Mars Reconnaissance Orbiter* sera la première caméra hyper-spectrale multi-angulaire à opérer depuis l'espace. Ces nouveaux instruments accentueront encore plus la taille des données qui devrait atteindre plusieurs tera-octets pour une dimension de l'ordre de 4000 variables. Il est donc crucial pour les scientifiques et les agences qui devront traiter ces nouvelles données de disposer d'outils d'analyse performants.

Données et protocole

Les données, mises à notre disposition par le laboratoire de Planétologie de Grenoble, ont été acquises par l'imageur OMEGA. Cet imageur a observé le sol de la planète Mars avec une résolution spatiale variant entre 300 et 3000 mètres en fonction de l'altitude du satellite. Il a acquis pour chaque pixel observé les spectres dont les longueurs d'ondes vont de 0.36 à 5.2 μm et stocké ces informations dans un vecteur de 256 dimensions. Le but de cette étude préliminaire est de caractériser la composition de la surface du sol martien en affectant chacun des pixels observés à une des 5 classes minéralogiques indiquées par les experts. Pour cette expérimentation, visant à vérifier l'aptitude de nos méthodes de classification à traiter de telles données, nous avons considéré une image de taille 300×128 pixels de la surface de la planète Mars dont chacun des 38 400 pixels est décrit par 256 variables. L'image de gauche de la figure 7.3 représente la zone étudiée.

Résultats expérimentaux

L'image de droite de la figure 7.3 montre la segmentation obtenue avec le modèle $[a_i b_i Q_i d_i]$ de l'HDDC. On peut tout d'abord observer que la segmentation fournie par l'HDDC est très satisfaisante sur une grande partie de l'image. Les résultats insuffisants de la partie supérieure droite de l'image

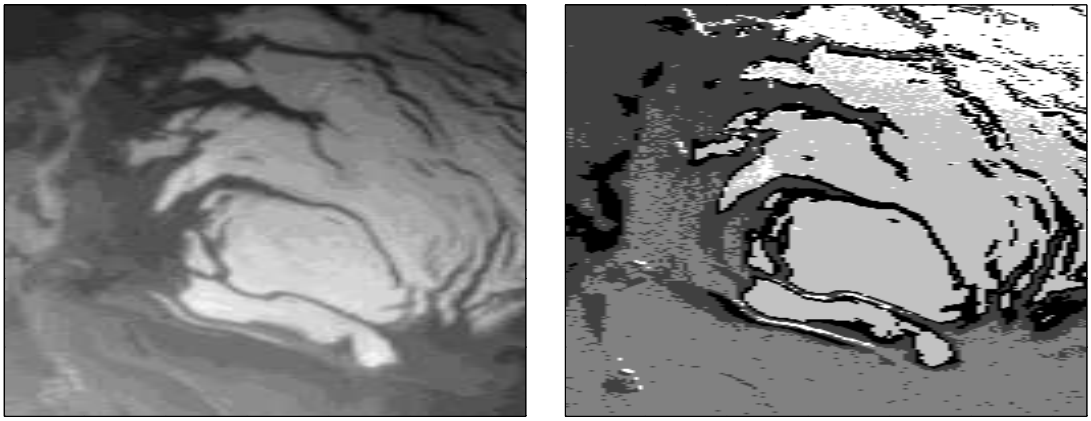


FIG. 7.3 – Catégorisation de la composition de la surface de la planète Mars avec l'HDDC : à gauche, image de la zone étudiée et, à droite, segmentation obtenue avec l'HDDC sur les données de dimension 256 associées à chaque pixel de l'image de gauche.

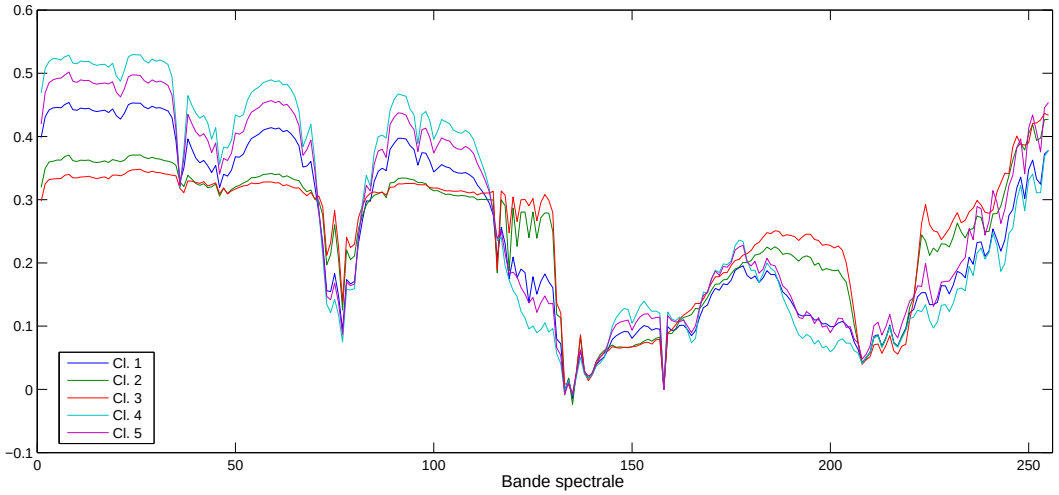


FIG. 7.4 – Moyennes spectrales obtenues avec l'HDDC des 5 classes minéralogiques recherchées.

sont dus à la courbure de la planète et peuvent être corrigés. Les experts du laboratoire de Planétologie de Grenoble ont particulièrement apprécié que notre méthode soit capable de détecter le mélange de glace et de carbonate (liseré noir) présent autour des zone de glaces (zones claires de l'image). La figure 7.4 présente les moyennes spectrales des 5 classes. A partir de cette information, les experts peuvent déterminer avec précision la composition minéralogique de chacune des classes. Cette étude a démontré que notre méthode de *clustering* HDDC est capable de traiter efficacement des bases de données réelles de grande dimension et de grande taille. De plus, cette étude préliminaire a été réalisé sans prendre en compte les relations spatiales existantes entre les pixels et gageons que la prise en compte de ces relations améliore encore la segmentation. Nous envisageons de prendre en compte ces relations spatiales en utilisant l'approche qui combine l'HDDC à la modélisation par champs de Markov cachés et qui a donné des résultats prometteurs en reconnaissance de textures.

7.2.3 Incorporation de nos modèles dans le logiciel MixMod

Très récemment, les responsables du logiciel MixMod [10][1] nous ont donné l'opportunité d'inclure dans leur logiciel les modèles gaussiens adaptés aux données de grande dimension présentés dans ce mémoire. Ainsi, dans un futur proche, il sera possible de choisir le modèle gaussien $[a_{ij}b_iQ_id_i]$ ou l'un de ses 27 sous-modèles pour modéliser et classer des données dans le logiciel MixMod. Cela nous permettra en outre de comparer nos modèles adaptés aux données de grande dimension à l'ensemble des modèles parcimonieux proposés par Celeux et Govaert dans [21] et d'utiliser les différents critères bayésiens de sélection de modèle disponibles (notamment le critère ICL [9]). Enfin, les modèles à matrices d'orientations communes pourront certainement être mis en œuvre grâce à la présence de l'algorithme FG dans le logiciel Mixmod et des temps de calculs optimisés.

7.3 Perspectives de recherche

Dans ce dernier paragraphe, nous allons présenter brièvement les perspectives de recherche qui s'offrent à nous dans les cadres théoriques et applicatifs.

7.3.1 Perspectives théoriques

Nos perspectives de recherche dans le cadre théorique s'organisent autour de l'amélioration de notre approche et de son extension aux données qualitatives de grande dimension, de la classification faiblement supervisée et du couplage Statistique–Optimisation pour la régularisation des matrices de covariance.

[1] disponible à l'adresse *http ://www-math.univ-fcomte.fr/mixmod/*.

Estimation des dimensions intrinsèques des classes

Nous avons proposé dans ce mémoire d'estimer les dimensions intrinsèques des classes grâce à la méthode du scree-test de Cattell qui a donné des résultats satisfaisants mais qui reste une méthode simple et empirique. Nous pensons que l'estimation des dimensions intrinsèques des classes mérite d'être améliorée tant du point de vue théorique que pratique. On pourrait par exemple considérer le choix des dimensions intrinsèques comme un problème de choix de modèles et utiliser alors le critère BIC pour cette tâche. En outre, une étude approfondie de la sensibilité de la méthode HDDC vis-à-vis du choix des dimensions intrinsèques serait très intéressante.

Extension de notre approche aux données qualitatives de grande dimension

Nous envisageons également d'étendre aux données qualitatives l'approche qui nous a amené à proposer la re-paramétrisation du modèle de mélange gaussien et qui a été présentée dans ce mémoire dans le cadre des données quantitatives. Pour cela, il faudra ne pas se placer dans le cadre habituel de la classification des données qualitatives où l'hypothèse d'indépendance conditionnelle des variables est faite.

Classification faiblement supervisée

Un autre thème principal de recherche que nous aimerions développer dans le futur est celui de la classification faiblement supervisée que nous avons abordé au cours du chapitre 6. Nous pensons en effet que les approches supervisées seront de plus en plus coûteuse en temps d'annotation dans le futur étant donné que la taille des base de données croît chaque jour. Il paraît alors raisonnable d'envisager d'autres types de supervision dont l'approche faiblement supervisée fait partie. L'approche probabiliste que nous avons introduite au cours du chapitre 6 propose déjà un cadre théorique sain pour la discrimination de deux classes dans un cadre faiblement supervisé. Cependant, cette approche ne permet pas nativement de discriminer plusieurs classes d'objets et nous envisageons de porter nos efforts sur cette limitation de notre approche.

Régularisation des matrices de covariance

Enfin, nous aimerions faire le lien entre la statistique et l'optimisation autour des problèmes posés par le mauvais conditionnement des matrices de covariance. Dans [57], Malick a exprimé le problème de la régularisation des matrices de covariance sous la forme d'un problème d'optimisation appelé « moindres carrés semi-définis ». Nous pensons que l'association des deux disciplines serait un moyen élégant d'aborder ce problème que l'on retrouve dans un grand nombre d'applications.

7.3.2 Perspectives applicatives

Du point de vue applicatif, nous envisageons de porter nos efforts sur l'approche faiblement supervisée en reconnaissance d'objets et l'utilisation des relations spatiales pour améliorer la localisation des objets dans les images.

Approche faiblement supervisée en reconnaissance d'objets

L'extension de l'approche probabiliste de la classification faiblement supervisée au cas de plusieurs classes permettra, dans le cadre de l'application à la reconnaissance d'objets, de localiser plusieurs objets dans une même image de façon automatique. C'est en effet la principale limitation de méthodes actuelles de reconnaissance d'objets. Il est en effet essentiel de pouvoir localiser dans une même image des instances d'objets différents car les images réelles comportent un nombre très grand de catégories d'objets.

Utilisation des relations spatiales pour la localisation d'objets

Nous envisageons également de combiner l'approche probabiliste que nous avons proposé pour la reconnaissance d'objets avec une modélisation des relations spatiales des parties des objets. Cette modélisation pourra être faite par en modélisant la densité de chacune des parties dans l'espace ou par l'utilisation d'un champ de Markov caché (comme dans l'application de reconnaissance de textures).

Annexe A

Evaluation des performances d'un classifieur

Nous avons vu au cours de ce mémoire que l'évaluation de la performance d'un classifieur est un problème récurrent et qui dépend du type de supervision. Nous allons considérer, dans cette annexe, l'évaluation de la performance d'un classifieur dans le cas supervisé. Nous présenterons tout d'abord les techniques de ré-échantillonage permettant en particulier de régler les paramètres des méthodes de classification. Nous nous intéresserons ensuite aux mesures qu'il est possible d'utiliser pour comparer deux classifieurs.

A.1 Techniques de ré-échantillonage

De nombreuses méthodes d'analyse discriminante possèdent un ou plusieurs paramètres qu'il n'est pas possible d'estimer par *maximum* de vraisemblance à partir du jeu d'apprentissage. On peut alors déterminer empiriquement, sur ce même jeu d'apprentissage, quelle valeur est la plus proche de la valeur réelle de ce paramètre. Pour cela, il suffit de parcourir l'espace des valeurs possibles de ce paramètre et de calculer le taux d'erreur obtenu par le classifieur associé sur le jeu d'apprentissage. Cependant, une telle approche conduit à une estimation très optimiste du taux d'erreur réel du classifieur et le choix du paramètre est donc biaisé. Il est donc nécessaire de mettre en place une technique de ré-échantillonage pour réduire ce biais.

A.1.1 La validation croisée

Afin d'obtenir une estimation réaliste du taux d'erreur d'un classifieur, nous allons nous ramener à la situation classique, *i.e.* échantillon d'apprentissage / échantillon de validation, en ré-échantillonnant le jeu d'apprentissage initial $\mathcal{A}$.

Approche classique

L'approche classique de la validation croisée, également dénommée *leave-one-out*, revient à construire le classifieur à partir du jeu d'apprentissage $\mathcal{A}$ privé de son $i$ème élément, pour $i = 1, ..., n$, et d'affecter ce dernier à l'une des k classes. On peut alors estimer le taux d'erreur du classifieur par le nombre moyen de points mal classés selon cette procédure. Cette approche, qui fournit une estimation fiable du taux d'erreur du classifieur, est néanmoins coûteuse si le nombre d'observations du jeu d'apprentissage initial $\mathcal{A}$ est grand.

Version améliorée

En pratique, la règle de décision construite à chaque étape par l'approche précédente ne varie pas beaucoup et il est donc possible de ne diviser qu'en L parties l'échantillon d'apprentissage initial $\mathcal{A}$. Le jeu $\mathcal{A}$ est alors divisé en L parties égales et l'on construit ensuite le classifieur à partir du jeu d'apprentissage $\mathcal{A}$ privé de sa ℓème partie, pour $\ell = 1, ..., L$. A chaque étape, la partie mise à l'écart est affectée aux groupes définis *a priori* et le taux d'erreur du classifieur est à nouveau estimé par le nombre moyen de points mal classés selon cette procédure. Remarquons que si $L = n$, on retrouve alors la procédure classique du *leave-one-out*. D'autre part, si $L = 2$, la procédure porte le nom de *half-sampling*. Pour les expériences présentées dans ce mémoire, nous avons utilisé cette technique de ré-échantillonage avec $L = 10$.

A.1.2 Le *bootstrap*

La technique de ré-échantillonage du *bootstrap*[1] reprend l'idée de la validation croisée mais introduit de l'aléa dans la construction des jeux d'apprentissage et de validation. La méthode du *bootstrap* construit, à chaque itération $q = 1, ..., Q$, un jeu de validation par tirage aléatoire de n_v observations dans le jeu d'apprentissage initial $\mathcal{A}$ et utilise le reste pour apprendre le classifieur. Une fois le classifieur appris, les n_v observations du jeu de validation sont classées dans les k groupes connus *a priori*. Le taux d'erreur du classifieur est également estimé par le nombre moyen de points mal classés sur les Q répétitions de la procédure. Le nombre de répétitions de la procédure est généralement fixé à 100 et, de ce fait, cette procédure est coûteuse en temps de calcul. Pour cette raison, on lui préférera le plus souvent la validation croisée. On pourra lire [47, chap. 7] pour plus de détails sur cette technique de ré-échantillonage.

[1] En anglais, le *bootstrapping* fait référence aux aventures du baron de Münchhausen qui s'est sorti d'un marécage, où il était embourbé, uniquement en tirant à de nombreuses reprises sur ses bottes pour se propulser vers le haut. Les *bootstraps* sont les anneaux, en cuir ou en tissu, cousus sur le rebord des bottes et dans lesquels on passe les doigts pour s'aider à les enfiler.

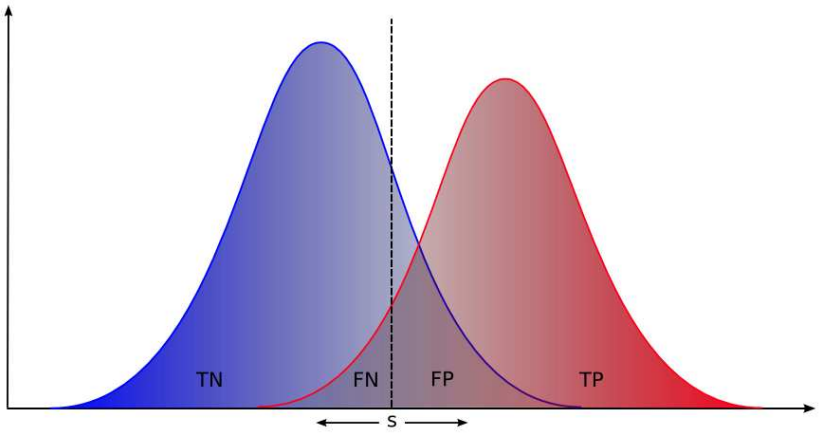


FIG. A.1 – Décision d'un classifieur binaire en fonction du seuil de discrimination s : la courbe bleue représente la probabilité *a posteriori* de la classe « négative » et la courbe rouge représente la probabilité *a posteriori* de la classe « posititive ».

A.2 Courbes ROC et « rappel-précision »

Le second point que nous souhaitions aborder est celui de la comparaison de deux classifieurs. La mesure la plus couramment utilisée pour comparer deux classifieurs est le taux d'erreur (ou de façon équivalente, le taux de classification correcte) obtenue sur le jeu de validation. Toutefois, cette mesure peut être dépendante du jeu de validation et ne donne pas d'indication sur l'ensemble des caractéristiques des classifieurs. Dans le cas de la classification binaire, des outils permettent à la fois de décrire les caractéristiques des classifieurs étudiés et de les comparer entre eux. Les courbes ROC et « rappel-précision » remplissent cette tâche. On pourra consulter [23] et [29] pour de plus amples détails sur ces mesures.

A.2.1 Définitions préalables

Avant d'entrer dans le détail de la construction des courbes ROC et « rappel-précision », il est nécessaire de définir certaines quantités entrant en jeu dans les deux cas. Un classifieur binaire, dont on souhaite évaluer la performance, va affecter chacune des observations à classer au groupe des positifs ou au groupe des négatifs. Cependant, l'origine des observations de l'échantillon de validation étant connues, il nous est possible de diviser le groupe des points positifs de la façon suivante :

$$\#\,\text{positifs} = TP + FN,$$

où TP est le nombre de vrais positifs, *i.e.* points classés positifs à juste titre, et FN est le nombre de faux négatifs, *i.e.* points classés négatifs à tort. De même, le groupe des points négatifs peut être

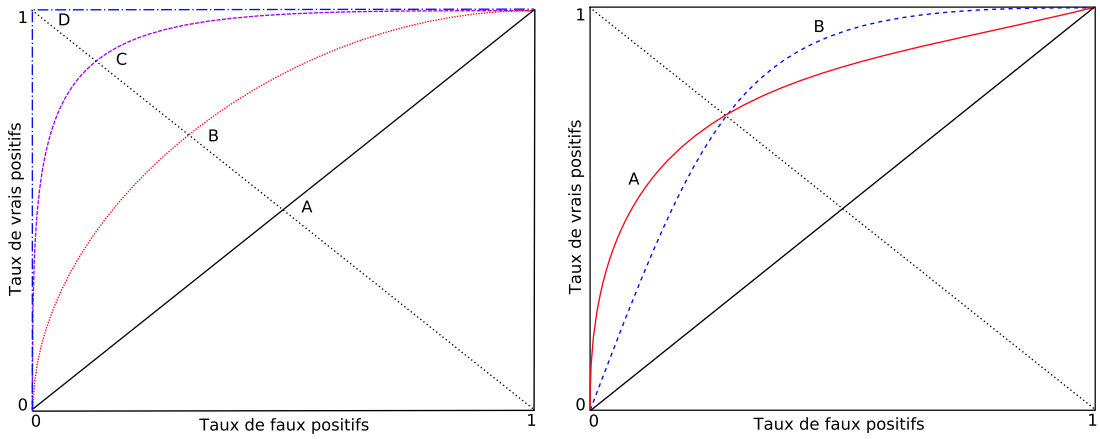


FIG. A.2 – Courbes ROC : à gauche, les courbes A, B, C et D sont respectivement associées à des classifieurs de performances croissantes. A droite, les courbes A et B ont la même aire (AUC) et la même valeur EER et sont pourtant associées à des classifieurs qui n'ont pas les mêmes qualités.

divisé de la façon suivante :

$$\#\text{ négatifs} = TN + FP,$$

où TN est le nombre de vrais négatifs, *i.e.* points classés négatifs à juste titre, et FP est le nombre de faux positifs, *i.e.* points classés positifs à tort. La figure A.1 permet de visualiser les différentes quantités entrant en jeu dans la construction de la courbe ROC d'un classifieur binaire.

A.2.2 La courbe ROC

La courbe *Receiver Operating Characteristic* (ROC) fut inventée durant la seconde guerre mondiale pour aider les opérateurs radar à décider si les *spots* apparaissant sur leur écran représentaient un avion ennemi, un avion ami ou juste du bruit. De nos jours, la courbe ROC est par exemple très utilisée en médecine pour évaluer les risques associés à une décision thérapeutique.

Construction de la courbe ROC

La courbe ROC exprime la sensitivité en fonction de (1 – spécificité) pour différentes valeurs du seuil de discrimination s (qui est normalement fixé à 0.5 pour la règle du MAP dans le cas binaire). La sensitivité, qui est la probabilité de détecter correctement un positif, se définit comme le taux de vrais positifs :

$$\text{sensitivité} = \frac{TP}{\#\text{ positifs}}.$$

La spécificité est quant à elle la probabilité de détecter correctement un négatif et elle se définit de part la quantité (1 – spécificité) qui est égale au taux de faux positifs :

$$1 - \text{spécificité} = \frac{FP}{\#\text{ négatifs}},$$

La figure A.2 présente des exemples de courbes ROC. Les courbes A, B, C et D, représentées sur l'image de gauche, sont respectivement associées à des classifieurs de performances croissantes. En particulier, la courbe A est associée au classifieur aléatoire, qui affecte chacune des observations au hasard, alors que la courbe D est associée au classifieur optimal. Les courbes ROC permettent donc de visualiser les caractéristiques d'un classifieur. Ainsi, le classifieur associé à la courbe A de l'image de droite de la figure A.2 est un classifieur peu performant mais relativement précis. A l'inverse, le classifieur associé à la courbe B est un classifieur performant mais au prix d'un grand nombre de fausses détections.

Les mesures AUC et EER

Il existe également des moyens de résumer l'information contenue dans une courbe ROC afin de comparer simplement plusieurs classifieurs. Le premier est le calcul de l'aire sous la courbe ROC, notée AUC pour *Area Under the Curve*. En effet, si l'AUC d'un classifieur est plus grande que celle d'un autre classifieur, on peut alors dire que le premier classifieur est globalement plus performant que le second. Il est également possible de relever la valeur à l'*Equal Error Rate*, notée EER, qui fournit la performance d'un classifieur quand le taux de vrais positifs est égale à 1 moins le taux de faux positifs. Cette valeur peut être lue simplement sur le graphique en relevant l'ordonnée de l'intersection de la courbe ROC du classifieur et de la diagonale opposée (droite oblique tracée en pointillés sur les images de la figure A.2). Cette mesure est notamment très utilisée en vision par ordinateur. Cependant, ces valeurs ne permettent pas de comparer deux classifieurs ayant des valeurs AUC et ERR égales. Sur l'image de droite de la figure A.2, les courbes A et B ont la même aire (AUC) et la même valeur EER mais les classifieurs auxquels elles sont associées n'ont pas les mêmes qualités.

Extension au cas multi-classes

Ces dernières années, les recherches dans le domaine de l'analyse ROC se sont naturellement tournées vers le cas multi-classes. En effet, les classifieurs qui considèrent nativement plusieurs classes sont très courants et il était donc temps de fournir un outil d'évaluation des performances de ce type de classifieur. Mossman [65] a ouvert la voie en considérant le cas particulier de trois classes et a proposé de construire une surface ROC à la place de la traditionnelle courbe ROC. De plus, dans ce cadre, Ferry *et al.* [30] ont proposé de substituer la mesure AUC par la mesure VUS qui est l'acronyme de *Volume Under the Surface*. Toutefois, l'extension au cas réel multi-classes reste encore un problème ouvert.

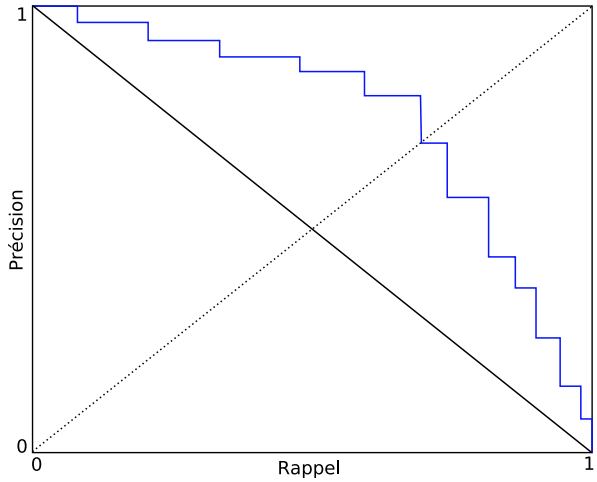


FIG. A.3 – Courbe « rappel-précision » : cette courbe est obtenue en traçant le rappel en fonction de la précision au fur et à mesure de la classification des observations.

A.2.3 La courbe « rappel-précision »

La courbe « rappel-précision » est plutôt utilisée en informatique mais a les mêmes fonctions que la courbe ROC. La courbe « rappel-précision » exprime, comme son nom l'indique, le rappel en fonction de la précision au fur et à mesure de l'augmentation du seuil de classification. Le rappel, qui est strictement la sensitivité de la courbe ROC, est défini comme étant le taux de vrais positifs, *i.e.* le rapport entre le nombre de positifs retrouvés et le nombre total de positifs du jeu de validation :

$$\text{rappel} = \frac{TP}{\#\text{ positifs}},$$

La précision est quant à elle définie comme étant le rapport du nombre de vrais positifs sur le nombre de points classés comme positifs par le classifieur :

$$\text{précision} = \frac{TP}{\#\text{ négatifs}},$$

La figure A.3 présente un exemple de courbe « rappel-précision ».

Annexe B

Détails des résultats expérimentaux de reconnaissance d'objets

Dans cette annexe, nous donnons le détails des résultats expérimentaux de reconnaissance d'objets obtenus sur la base d'images *Pascal* [22]. Les résultats donnés sont les suivants :

Localisation d'objets sur la base *Pascal test1* Le paragraphe B.1 présente les résultats de localisation obtenus sur le jeu *Pascal test1*. On peut remarquer que notre approche combinée à la méthode de *clustering* HDCC est plus efficace que les autres méthodes sur 2 catégories dans le cadre supervisé et sur 3 catégories dans le cadre faiblement supervisé. Notre méthode est de plus toujours plus efficiente en moyenne que les autres méthodes.

Localisation d'objets sur la base *Pascal test2* Le paragraphe B.2 présente les résultats de localisation obtenus sur le jeu *Pascal test*2. Notre méthode de localisation d'objets distance ses concurrentes sur 2 des catégories d'objets dans le cadre supervisé et sur l'ensemble des catégories dans le cadre faiblement supervisé. Notre méthode est encore une fois toujours plus efficiente en moyenne que les autres méthodes.

Classification d'images sur la base *Pascal test1* Le paragraphe B.3 présente les résultats de classification d'images obtenus sur le jeu *Pascal test1*. On observe que notre approche fournit des résultats meilleurs que les autres approches dans le cadre supervisé et proches de ceux de la meilleure méthodes du challenge *Pascal* dans le cadre faiblement supervisé.

Classification d'images sur la base *Pascal test2* Enfin, le paragraphe B.4 présente les résultats de classification d'images obtenus sur le jeu *Pascal test*2. Notre méthode de classification d'images est plus performante que les autres méthodes sur l'ensemble des catégories dans le cadre supervisé et remporte deux compétitions dans le cadre faiblement supervisé.

B.1 Résultats de localisation d'objets sur la base *Pascal test1*

Model	Moto	Vélo	Humain	Voiture	Moyenne
$[a_{ij}b_iQ_id_i]$	0.665	0.403	0.047	0.095	0.302
$[a_{ij}bQ_id_i]$	0.680	**0.439**	0.032	0.123	**0.318**
$[a_ib_iQ_id_i]$	0.664	0.404	0.062	0.120	0.313
$[a_ibQ_id_i]$	0.671	0.437	0.035	0.128	**0.318**
$[abQ_id_i]$	0.673	0.430	0.040	0.125	0.317
$[a_ib_iQ_id]$	0.665	0.432	0.065	0.093	0.314
$[abQ_id]$	0.657	0.416	**0.053**	0.076	0.300
$[abQd]$	0.595	0.317	0.047	0.066	0.256
PCA+diag-GMM	0.572	0.339	0.052	0.067	0.257
sphe-GMM	0.572	0.349	0.042	0.118	0.271
diag-GMM	0.587	0.344	0.052	0.122	0.276
com-GMM	0.640	0.341	0.041	0.049	0.267
kmeans	0.581	0.359	0.033	0.070	0.261
Pascal	**0.886**	0.119	0.013	**0.613**	0.279

TAB. B.1 – Localisation supervisée, test1, AP

Model	Moto	Vélo	Humain	Voiture	Moyenne
$[a_{ij}b_iQ_id_i]$	0.658	0.384	0.020	0.027	0.273
$[a_{ij}bQ_id_i]$	**0.686**	0.404	0.038	0.020	**0.287**
$[a_ib_iQ_id_i]$	0.674	0.399	0.027	**0.042**	0.285
$[a_ibQ_id_i]$	0.671	0.403	0.022	0.034	0.283
$[abQ_id_i]$	0.677	0.406	0.026	0.019	0.282
$[a_ib_iQ_id]$	0.677	0.410	0.035	0.026	**0.287**
$[abQ_id]$	0.669	**0.411**	0.037	0.023	0.285
$[abQd]$	0.611	0.261	0.024	0.018	0.229
PCA+diag-GMM	0.585	0.247	0.018	0.009	0.215
sphe-GMM	0.592	0.228	0.021	0.025	0.216
diag-GMM	0.603	0.234	**0.044**	0.027	0.227
com-GMM	0.655	0.293	0.015	0.021	0.246
kmeans	0.573	0.212	0.022	0.010	0.204

TAB. B.2 – Localisation faiblement supervisée, test1, AP

B.2 Résultats de localisation d'objets sur la base *Pascal test2*

Model	Moto	Vélo	Humain	Voiture	Moyenne
$[a_{ij}b_iQ_id_i]$	0.305	0.169	0.061	0.154	0.172
$[a_{ij}bQ_id_i]$	0.316	0.164	**0.091**	0.151	0.181
$[a_ib_iQ_id_i]$	0.315	0.172	**0.091**	0.155	0.183
$[a_ibQ_id_i]$	0.307	0.169	**0.091**	0.136	0.176
$[abQ_id_i]$	0.316	0.168	0.045	0.158	0.172
$[a_ib_iQ_id]$	0.351	0.164	0.061	0.141	0.179
$[abQ_id]$	0.324	**0.180**	**0.091**	0.145	**0.185**
$[abQd]$	0.275	0.124	**0.091**	0.146	0.159
PCA+diag-GMM	0.296	0.173	0.091	0.148	0.177
sphe-GMM	0.261	0.142	0.045	0.149	0.149
diag-GMM	0.245	0.153	0.091	0.156	0.161
com-GMM	0.301	0.163	0.045	0.147	0.164
kmeans	0.252	0.149	0.091	0.148	0.160
Pascal	**0.341**	0.113	0.021	**0.304**	0.112

TAB. B.3 – Localisation supervisée, test2, AP

Model	Moto	Vélo	Humain	Voiture	Moyenne
$[a_{ij}b_iQ_id_i]$	0.304	0.141	0.021	0.115	0.145
$[a_{ij}bQ_id_i]$	0.312	0.141	0.018	0.115	0.147
$[a_ib_iQ_id_i]$	0.311	**0.161**	**0.045**	0.049	0.142
$[a_ibQ_id_i]$	0.298	0.153	0.026	**0.116**	**0.148**
$[abQ_id_i]$	0.312	0.156	0.023	0.046	0.134
$[a_ib_iQ_id]$	0.322	0.141	0.023	0.034	0.130
$[abQ_id]$	**0.324**	0.156	0.030	0.044	0.139
$[abQd]$	0.273	0.091	0.023	0.038	0.106
PCA+diag-GMM	0.268	0.136	0.024	0.050	0.120
sph-GMM	0.254	0.111	0.023	0.037	0.106
diag-GMM	0.239	0.120	0.011	0.069	0.110
com-GMM	0.276	0.142	0.008	0.036	0.116
kmeans	0.242	0.093	0.017	0.042	0.099

TAB. B.4 – Localisation faiblement supervisée, test2, AP

B.3 Résultats de Classification d'images sur la base *Pascal test1*

Model	Moto	Vélo	Humain	Voiture	Moyenne
$[a_{ij}b_iQ_id_i]$	0.903	0.763	0.810	0.811	0.822
$[a_{ij}bQ_id_i]$	0.907	0.781	**0.845**	0.829	0.841
$[a_ib_iQ_id_i]$	0.894	0.763	0.821	0.825	0.826
$[a_ibQ_id_i]$	0.903	0.798	0.821	0.833	0.839
$[abQ_id_i]$	0.907	0.798	0.833	0.840	0.845
$[a_ib_iQ_id]$	0.921	**0.816**	0.833	0.840	0.853
$[abQ_id]$	**0.926**	**0.816**	**0.845**	**0.855**	**0.860**
$[abQd]$	0.875	0.719	0.798	0.756	0.787
PCA+diag-GMM	0.866	0.737	0.774	0.742	0.780
sphe-GMM	0.870	0.728	0.786	0.716	0.775
diag-GMM	0.889	0.763	0.786	0.749	0.797
com-GMM	0.898	0.781	0.810	0.782	0.818
kmeans	0.861	0.737	0.786	0.727	0.778

TAB. B.5 – Classification supervisée, test1, EER

Model	Moto	Vélo	Humain	Voiture	Moyenne
$[a_{ij}b_iQ_id_i]$	0.898	0.851	0.845	0.865	0.865
$[a_{ij}bQ_id_i]$	0.907	0.851	0.857	0.865	0.870
$[a_ib_iQ_id_i]$	0.898	0.860	0.845	0.880	0.871
$[a_ibQ_id_i]$	0.907	0.860	0.845	0.884	0.874
$[abQ_id_i]$	0.917	0.851	0.821	0.862	0.863
$[a_ib_iQ_id]$	0.917	0.877	0.869	0.884	0.887
$[abQ_id]$	0.921	0.868	0.869	0.880	0.885
$[abQd]$	0.866	0.772	0.774	0.825	0.809
PCA+diag-GMM	0.875	0.807	0.845	0.844	0.843
sphe-GMM	0.861	0.798	0.833	0.847	0.835
diag-GMM	0.894	0.816	0.845	0.855	0.852
com-GMM	0.894	0.807	0.821	0.851	0.843
kmeans	0.852	0.789	0.833	0.833	0.827
Pascal	**0.977**	**0.930**	**0.917**	**0.961**	**0.937**

TAB. B.6 – Classification faiblement supervisée, test1, EER

B.4 Résultats de classification d'images sur la base *Pascal test2*

Model	Moto	Vélo	Humain	Voiture	Moyenne
$[a_{ij}b_iQ_id_i]$	0.738	0.713	0.686	0.720	0.714
$[a_{ij}bQ_id_i]$	0.723	0.735	0.698	0.709	0.716
$[a_ib_iQ_id_i]$	0.738	0.728	0.698	0.716	0.720
$[a_ibQ_id_i]$	0.733	0.724	0.698	0.724	0.720
$[abQ_id_i]$	0.738	0.731	0.698	**0.742**	0.727
$[a_ib_iQ_id]$	0.743	0.742	0.698	0.727	0.727
$[abQ_id]$	**0.752**	**0.749**	**0.702**	0.738	**0.735**
$[abQd]$	0.708	0.685	0.665	0.698	0.689
PCA+diag-GMM	0.703	0.742	0.639	0.695	0.695
sphe-GMM	0.698	0.724	0.656	0.684	0.690
diag-GMM	0.688	0.713	0.646	0.680	0.682
com-GMM	0.723	0.724	0.675	0.705	0.707
kmeans	0.693	0.720	0.641	0.695	0.687

TAB. B.7 – Classification supervisée, test2, EER

Model	Moto	Vélo	Humain	Voiture	Moyenne
$[a_{ij}b_iQ_id_i]$	0.738	0.731	0.696	0.669	0.708
$[a_{ij}bQ_id_i]$	0.733	0.713	**0.728**	0.673	0.712
$[a_ib_iQ_id_i]$	0.743	0.717	0.715	0.662	0.709
$[a_ibQ_id_i]$	0.738	0.710	0.707	0.665	0.705
$[abQ_id_i]$	0.733	**0.728**	0.711	0.684	0.714
$[a_ib_iQ_id]$	0.748	**0.728**	0.703	0.669	0.712
$[abQ_id]$	0.752	**0.728**	0.719	0.676	0.719
$[abQd]$	0.698	0.677	0.707	0.647	0.682
PCA+diag-GMM	0.713	0.699	0.705	0.651	0.692
sphe-GMM	0.703	0.703	0.700	0.644	0.687
diag-GMM	0.698	0.688	0.686	0.655	0.682
com-GMM	0.723	0.724	0.705	0.644	0.699
kmeans	0.693	0.695	0.688	0.629	0.676
Pascal	**0.798**	0.728	0.719	**0.720**	**0.741**

TAB. B.8 – Classification faiblement supervisée, test2, EER

Annexe C

Publications liées à la thèse

Articles et chapitres de livres

– C. Bouveyron, S. Girard and C. Schmid. High Dimensional Discriminant Analysis. *Communications in Statistics : Theory and Methods*, to appear, 2006.

– C. Bouveyron, S. Girard and C. Schmid. *Class-Specific Subspace Discriminant Analysis for High-Dimensional Data*. Lecture Notes in Computer Science, volume 3940, pages 139-150, 2006.

Rapports de recherche

– C. Bouveyron, S. Girard and C. Schmid. High Dimensional Data Clustering. Technical Report 1083M, LMC-IMAG, Université J. Fourier Grenoble 1, 2006.

– C. Bouveyron, S. Girard and C. Schmid. Analyse Discriminante de Haute Dimension. Technical Report 5470, INRIA, 2005.

Conférences internationales

– C. Bouveyron, J. Kannala, C. Schmid and S. Girard. Object Localization by Subspace clustering of Local Descriptors. *5th Indian Conference on Computer Vision, Graphics and Image Processing*, Madurai, India, 2006.

– C. Bouveyron, S. Girard and C. Schmid. High Dimensional Data Clustering. *17th International Conference on Computational Statistics*, pages 812-820, Rome, Italy, 2006.

– C. Bouveyron, S. Girard and C. Schmid. High Dimensional Discriminant Analysis. *11th International Conference on Applied Stochastic Models and Data Analysis*, pages 526-534, Brest, France, 2005.

– C. Bouveyron, S. Girard and C. Schmid. Classification of High Dimensional Data : High Dimensional Discriminant Analysis. *Workshop on Subspace, Latent Structure and Feature Selection techniques*, Bohinj, Slovenia, 2005.

Conférences nationales

– J. Blanchet and C. Bouveyron. Modèle markovien caché pour la classification supervisée de données de grande dimension spatialement corrélées. *38èmes Journées de Statistique de la Société Française de Statistique*, Clamart, France, 2006.

– C. Bouveyron, S. Girard and C. Schmid. Classification des données de grande dimension : application à la vision par ordinateur. *2èmes Rencontres Inter-Associations sur la classification et ses applications*, pages 24-25, Lyon, France, 2006.

– C. Bouveyron, S. Girard and C. Schmid. Une nouvelle méthode de classification pour la reconnaissance de formes. *20ème colloque GRETSI sur le traitement du signal et des images*, pages 711-714, Louvain-la-Neuve, Belgique, 2005.

– C. Bouveyron, S. Girard and C. Schmid. Une méthode de classification des données de grande dimension. *37ème Journées de Statistique de la Société Française de Statistique*, Pau, France, 2005.

– C. Bouveyron, S. Girard and C. Schmid. Dimension Reduction and Classification Methods for Object Recognition in Vision. *5th French-Danish Workshop on Spatial Statistics and Image Analysis in Biology*, pages 109-113, St-Pierre de Chartreuse, France, 2004.

Bibliographie

[1] S. Agarwal and D. Roth. Learning a sparse representation for object detection. In *7th European Conference on Computer Vision*, volume 4, pages 113–130, 2002.

[2] C. Aggarwal, J. Wolf, P. Yu, and J. Park. Fast algorithms for projected clustering. In *International Conference on Management of Data*, pages 61–72. ACM Press, 1999.

[3] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high-dimensional data for data mining application. In *ACM SIGMOD International Conference on Management of Data*, pages 94–105, 1998.

[4] A. Banerjee, C. Krumpelman, J. Ghosh, S. Basu, and R. Mooney. Model-based overlapping clustering. In *Conference on Knowledge Discovery in Data*, pages 532–537, Chicago, USA, 2005.

[5] J. Banfield and A. Raftery. Model-based Gaussian and non-Gaussian clustering. *Biometrics*, 49 :803–821, 1993.

[6] R. Bellman. *Dynamic programming*. Princeton University Press, 1957.

[7] H. Bensmail and G. Celeux. Regularized Gaussian discriminant analysis through eigenvalue decomposition. *Journal of the American Statistical Association*, 91 :1743–1748, 1996.

[8] J.-P. Bibring and 42 co authors. *OMEGA : Observatoire pour la Minéralogie, l'Eau, les Glaces et l'Activité*, page 37 49. ESA SP-1240 : Mars Express : the Scientific Payload, 2004.

[9] C. Biernacki, G. Celeux, and G. Govaert. Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(7) :719–725, 2000.

[10] C. Biernacki, G. Celeux, G. Govaert, and F. Langrognet. Model-based cluster analysis and discriminant analysis with the MIXMOD software. *Computational Statistics and Data Analysis*, to appear, 2006.

[11] C. Bishop and M. Tipping. A hierarchical latent variable model for data visualization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3) :281–293, 1998.

[12] J. Blanchet, F. Forbes, and C. Schmid. Markov random fields for textures recognition with local invariant regions and their geometric relationships. In *British Machine Vision Conference*, Oxford, UK, September 2005.

[13] L. Bocci, D. Vicari, and M. Vichi. A mixture model for the classification of three-way proximity data. *Computational Statistics and Data Analysis*, 50(7) :1625–1654, 2006.

[14] G. Bouchard and G. Celeux. Model selection in supervised classification. *Transactions on Pattern Analysis and Machine Intelligence*, 28(4) :544–554, 2005.

[15] C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2) :121–167, 1998.

[16] M. Carreira-Perpinan. A review of dimension reduction techniques. Technical Report CS-96-09, Dept. of Computer Science, University of Sheffield, 1997.

[17] R. Cattell. The scree test for the number of factors. *Multivariate Behavioral Research*, 1(2) :245–276, 1966.

[18] G. Celeux and J. Diebolt. The SEM algorithm : a probabilistic teacher algorithm from the EM algorithm for the mixture problem. *Computational Statistics Quarterly*, 2(1) :73 92, 1985.

[19] G. Celeux and J. Diebolt. Une version de type recuit simul de l'algorithme EM. *Notes aux Comptes Rendus de l'Acad mie des Sciences*, 310 :119 124, 1990.

[20] G. Celeux and G. Govaert. A classification EM algorithm for clustering and two stochastic versions. *Computational Statistics and Data Analysis*, 14 :315–332, 1992.

[21] G. Celeux and G. Govaert. Gaussian parsimonious clustering models. *International Journal of Pattern Recognition*, 28(5) :781–793, 1995.

[22] F. d'Alche Buc, I. Dagan, and J. Quinonero, editors. *The 2005 Pascal visual object classes challenge*. Proceedings of the first PASCAL Challenges Workshop. Springer, 2006.

[23] J. Davis and M. Goadrich. The relationship between precision-recall and ROC curves. In *Proceedings of the Twenty-Third International Conference on Machine Learning*, Pittsburgh, USA, 2006.

[24] P. Delicado. Another look at principal curves and surfaces. *Journal of Multivariate Analysis*, 77(1) :84–116, 2001.

[25] P. Demartines and J. Hérault. Curvilinear component analysis : a self-organizing neural network for non linear mapping of data sets. *IEEE Transactions on Neural Networks*, 8(1) :148–154, 1997.

[26] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39(1) :1–38, 1977.

[27] D. Donoho. High-dimensional data analysis : the curses and blessings of dimensionality. In *Math Challenges of the 21st Century*. American Mathematical Society, 2000.

[28] G. Dorko and C. Schmid. Object class recognition using discriminative local features. Technical Report 5497, INRIA, 2004.

[29] T. Fawcett. ROC graphs : notes and practical considerations for researchers. Technical Report HPL-2003-4, HP Laboratories, Palo Alto, USA, 2004.

[30] C. Ferry and J. Hernandez-Orallo. Volume under the ROC surface for multi-class problems. In *14th European Conference on Machine Learning*, pages 108–120, 2003.

[31] E. Fix and J. Hodges. Discriminatory analysis and nonparametric discrimination : consistency properties . Technical Report 21-49-004, USAF, School of Aviation Medecine, Texas, 1951.

[32] B. Flury and W. Gautschi. An algorithm for simultaneous orthogonal transformation of several positive definite symmetric matrices to nearly diagonal form. *SIAM Journal on Scientific and Statistical Computing*, 7 :169–184, 1986.

[33] B. Flury, M. Schmid, and A. Narayanan. Error rates in quadratic discrimination with constraints on the covariance matrices. *Journal of Classification*, 11 :101–120, 1994.

[34] L. Flury, B. Boukai, and B. Flury. The discrimination subspace model. *Journal of American Statistical Association*, 92(438) :758–766, 1997.

[35] I. Fodor. A survey of dimension reduction techniques. Technical report, Center for Applied Scientific Computing, Livermore, Canada, 2002.

[36] C. Fraley and A. Raftery. Model-based clustering, discriminant analysis and density estimation. *Journal of American Statistical Association*, 97 :611–631, 2002.

[37] Y. Freund and R. Shapire. A decision-theoretic generalization of online learning and an application to boosting. *Journal of Computer and System Science*, 55 :119–139, 1997.

[38] J. Friedman. Regularized discriminant analysis. *Journal of American Statistical Association*, 84(405) :165–175, 1989.

[39] J. Friedman. Another approach to polychotomous classification. Technical report, Department of Statistics, Stanford University, 1996.

[40] Z. Ghahramani and G. Hinton. The EM algorithm for mixtures of factor analyzers. Technical Report CRG-TR-96-1, Dept. of Computer Science, University of Toronto, 1996.

[41] S. Girard and S. Iovleff. Auto-associative models and generalized principal component analysis. *Journal of Multivariate Analysis*, 93(1) :21–39, 2005.

[42] G. Govaert, editor. *Analyse des données*. Traitement du signal et de l'image. Hermes Science, Paris, 2003.

[43] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3 :1157–1182, 2003.

[44] H. Harman. *Modern factor analysis*. University of Chicago Press, Chicago, 3rd edition, 1976.

[45] T. Hastie, A. Buja, and R. Tibshirani. Penalized discriminant analysis. *Annals of Statistics*, 23 :73–102, 1995.

[46] T. Hastie and W. Stuetzle. Principal curves. *Journal of the American Statistical Association*, 84 :502–516, 1989.

[47] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning*. Springer, New York, 2001.

[48] T. Hofmann. Probabilistic latent semantic analysis. In *Proc. of Uncertainty in Artificial Intelligence, UAI'99*, Stockholm, 1999.

[49] P. Huber. Projection pursuit. *The Annals of Statistics*, 13(2) :435–525, 1985.

[50] I. Jolliffe. *Principal component analysis*. Springer-Verlag, New York, 1986.

[51] T. Kadir, A. Zisserman, and M. Brady. An affine invariant salient region detector. In *8th European Conference on Computer Vision*, 2004.

[52] W. Krzanowski, P. Jonathan, W. McCarthy, and M. Thomas. Discriminant analysis with singular covariance matrices : methods and applications in spectroscopic data. *Journal of Applied Statistics*, 44 :101–115, 1995.

[53] J. W. Lee, J. B. Lee, M. Park, and S. Song. An extensive comparison of recent classification tools applied to microarray data. *Computational Statistics and Data Analysis*, 48 :869–885, 2005.

[54] R. Lehoucq, D. Sorensen, and C. Yang. *ARPACK users' guide : solution of large-scale eigenvalue problems with implicitly restarted Arnoldi methods*. SIAM Publications, Philadelphia, 1998.

[55] B. Leibe and B. Schiele. Interleaved object categorization and segmentation. In *British Machine Vision Conference*, Norwich, England, 2003.

[56] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2) :91–110, 2004.

[57] J. Malick. A dual approach to semi-definite least-squares problems. *SIAM Journal of Matrix Analysis and Applications*, 26(1) :272–284, 2004.

[58] G. McLachlan and D. Peel. *Finite Mixture Models*. Wiley Interscience, New York, 2000.

[59] G. McLachlan, D. Peel, and R. Bean. Modelling high-dimensional data by mixtures of factor analyzers. *Computational Statistics and Data Analysis*, 41 :379–388, 2001.

[60] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 257–263, 2003.

[61] K. Mikolajczyk and C. Schmid. Scale and affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1) :63–86, 2004.

[62] A. Mkhadri, G. Celeux, and A. Nasrollah. Regularization in discriminant analysis : a survey. *Computational Statistics and Data Analysis*, 23 :403–423, 1997.

[63] P. Moerland. A comparison of mixture models for density estimation. *Artificial Neural Networks*, 7 :25–30, 1999.

[64] B. Moghaddam. Principal manifolds and probabilistic subspaces for visual recognition. *Pattern Analysis and Machine Intelligence*, 26(6) :780–788, 2002.

[65] D. Mossman. Three-way ROC. *Medical Decision Making*, 19 :78–89, 1999.

[66] T. O'Neill. Error rate of non-Bayesian classification rules and robustness of Fisher's linear discriminant function. *Biometrika*, 79 :177–184, 1992.

[67] A. Opelt, M. Fussenegger, A. Pinz, and P. Auer. Weak hypotheses and boosting for generic object detection and recognition. In *European Conference on Computer Vision*, volume 2, pages 71–84, 2004.

[68] L. Parsons, E. Haque, and H. Liu. Subspace clustering for high dimensional data : a review. *SIGKDD Explor. Newsl.*, 6(1) :90–105, 2004.

[69] T. Pavlenko. On feature selection, curse of dimensionality and error probability in discriminant analysis. *Journal of Statistical Planning and Inference*, 115 :565–584, 2003.

[70] T. Pavlenko and D. Von Rosen. Effect of dimensionality on discrimination. *Statistics*, 35(3) :191–213, 2001.

[71] K. Pearson. Contributions to the mathematical theory of evolution. *Philosophical Transactions of the Royal Society*, 185 :71–110, 1894.

[72] K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 6(2) :559–572, 1901.

[73] I. Pima and M. Aladjem. Regularized discriminant analysis for face recognition. *Pattern Recognition*, 37(9) :1945–1948, 2004.

[74] A. Raftery and N. Dean. Variable selection for model-based clustering. *Journal of American Statistical Association*, 101(473) :168–178, 2006.

[75] S. Roweis and Z. Ghahramani. A Unifying review of linear Gaussian models . *Neural Computation*, 11 :305–345, 1999.

[76] S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500) :2323–2326, 2000.

[77] Sam Roweis. EM algorithms for PCA and SPCA. In Michael I. Jordan, Michael J. Kearns, and Sara A. Solla, editors, *Advances in Neural Information Processing Systems*, volume 10. The MIT Press, 1998.

[78] D. Rubin and D. Thayer. EM algorithms for ML factor analysis. *Psychometrika*, 47(1) :69–76, 1982.

[79] G. Saporta. *Probabilités, analyse de données et statistique*. Editions Technip, Paris, 1990.

[80] B. Schölkopf, A. Smola, and K. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10 :1299–1319, 1998.

[81] G. Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6 :461–464, 1978.

[82] D. Scott. *Multivariate density estimation*. Wiley & Sons, New York, 1992.

[83] D. Scott and J. Thompson. Probability density estimation in higher dimensions. In *Fifteenth Symposium in the Interface*, pages 173–179, 1983.

[84] B. Silverman. *Density estimation for Statistics and data analysis*. Chapman & Hall, New York, 1986.

[85] J. Sivic, B. Russell, A. Efros, A. Zisserman, and W. Freeman. Discovering objects and their location in images. In *International Conference on Computer Vision*, 2005.

[86] J. Tenenbaum, V. De Silva, and J. Langford. A global geometric framework for non linear dimensionality reduction. *Science*, 290(5500) :2319–2323, 2000.

[87] M. Tipping and C. Bishop. Probabilistic principal component analysis. Technical Report NCRG-97-010, Neural Computing Research Group, Aston University, 1997.

[88] M. Tipping and C. Bishop. Mixtures of probabilistic principal component analysers. *Neural Computation*, 11(2) :443–482, 1999.

[89] V. Vapnik. *The nature of statistical learning theory*. Springer-Verlag, New York, 1996.

[90] M. Verleysen. *Learning high-dimensional data*, pages 141–162. Limitations and Future Trends in Neural Computations. IOS Press, 2003.

[91] A. Webb. *Statistical pattern recognition*. Wiley, New York, 2 edition, 2002.

[92] J. Willamowski, D. Arregui, G. Csurka, C. Dance, and L. Fan. Coategorizing nine visual classes using local appareance descriptors. In *International Workshop on Learning for Adaptable Visual Systems*, Cambridge, UK, 2004.

[93] C. Wu. On the convergence properties of the EM algorithm. *Annals of Statistics*, 11 :95–103, 1983.

[94] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories. Technical report, INRIA, 2005.

[95] W. Zhong, P. Zeng, P. Ma, J. Liu, and Y. Zhu. Regularized sliced inverse regression for motif discovery. *Bioinformatics*, 21(22) :4169–4175, 2005.

Résumé

Le thème principal d'étude de cette thèse est la modélisation et la classification des données de grande dimension. Partant du postulat que les données de grande dimension vivent dans des sous-espaces de dimensions intrinsèques inférieures à la dimension de l'espace original et que les données de classes différentes vivent dans des sous-espaces différents dont les dimensions intrinsèques peuvent être aussi différentes, nous proposons une re-paramétrisation du modèle de mélange gaussien. En forçant certains paramètres à être communs dans une même classe ou entre les classes, nous exhibons une famille de 28 modèles gaussiens adaptés aux données de grande dimension, allant du modèle le plus général au modèle le plus parcimonieux. Ces modèles gaussiens sont ensuite utilisés pour la discrimination et la classification automatique de données de grande dimension. Les classifieurs associés à ces modèles sont baptisés respectivement *High Dimensional Discriminant Analysis* (HDDA) et *High Dimensional Data Clustering* (HDDC) et leur construction se base sur l'estimation par la méthode du *maximum* de vraisemblance des paramètres du modèle. La nature de notre re-paramétrisation permet aux méthodes HDDA et HDDC de ne pas être perturbées par le mauvais conditionnement ou la singularité des matrices de covariance empiriques des classes et d'être efficaces en terme de temps de calcul. Les méthodes HDDA et HDDC sont ensuite mises en œuvre dans le cadre d'une approche probabiliste de la reconnaissance d'objets dans des images. Cette approche, qui peut être supervisée ou faiblement supervisée, permet de localiser de manière probabiliste un objet dans une nouvelle image. Notre approche est validée sur des bases d'images récentes et comparée aux meilleures méthodes actuelles de reconnaissance d'objets.

Mots-clefs : Classification, données de grande dimension, modèle de mélange gaussien, réduction de dimension, modèles parcimonieux, reconnaissance d'objets faiblement supervisée.

Abstract

The main topic of this thesis is modeling and classification of high-dimensional data. Based on the assumption that high-dimensional data live in subspaces with intrinsic dimensions smaller than the dimension of the original space and that the data of different classes live in different subspaces with different intrinsic dimensions, we propose a re-parametrization of the Gaussian mixture model. By forcing some parameters to be common within or between classes, we show a family of 28 Gaussian models appropriated for high-dimensional data, from the most general model to the most parsimonious one. These models are then used for discrimination and clustering of high-dimensional data. The classifiers associated with these models are called respectively *High Dimensional Discriminant Analysis* (HDDA) and *High Dimensional Data Clustering* (HDDC) and their construction is based on the maximum likelihood estimation of model parameters. The nature of our re-parametrization allows HDDA and HDDC not to be disturbed by the ill-conditioning or the singularity of empirical covariance matrices and to be efficient in terms of computing time. The methods HDDA and HDDC are then used in a probabilistic framework to object recognition in images. This approach, which can be supervised or weakly-supervised, allows to locate in a probabilistic way an object in a new image. Our approach is validated on two recent image databases and compared to the most efficient object recognition methods.

Keywords : Classification, high-dimensional data, Gaussian mixture model, dimension reduction, parsimonious models, weakly-supervised object recognition.