Les modèles génératifs en classification supervisée et applications à la catégorisation d'images et à la fiabilité industrielle

Guillaume Bouchard

▶ **To cite this version:**

Guillaume Bouchard. Les modèles génératifs en classification supervisée et applications à la catégorisation d'images et à la fiabilité industrielle. Interface homme-machine [cs.HC]. Université Joseph-Fourier - Grenoble I, 2005. Français. NNT : . tel-00541059

HAL Id: tel-00541059

https://theses.hal.science/tel-00541059

Submitted on 29 Nov 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Université Joseph Fourier – Grenoble 1

Les modèles génératifs en classification supervisée et applications à la catégorisation d'images et à la fiabilité industrielle.

THÈSE

Soutenance en 2005

pour l'obtention du

Doctorat de l'université Joseph Fourier – Grenoble 1

(spécialité mathématiques appliquées)

par

Guillaume Bouchard

Composition du jury

Directeur de thèse :	Gilles Celeux	INRIA
Co-directeur de thèse :	William Triggs	CNRS

Institut National Recherche en Informatique et Automatique

à Amandine

Remerciements

Je tiens à remercier les rapporteurs David J. Hand et Gérard Govaert pour la lecture attentive de la thèse et les remarques constructives qu'ils ont faites. Merci aussi au jury de thèse et en particulier à Claudine Robert pour avoir accepté de venir le jour de la soutenance.

La période passée à l'INRIA m'a beaucoup apporté. J'ai rencontré des personnes remarquables, d'un point de vue personnel et professionnel. Gilles, mon directeur de thèse m'a conseillé, soutenu et écouté sur ma thèse, avec franchise une objectivité. Dans les périodes difficiles, il a su prendre du temps pour m'aider à avancer. Je lui exprime ma profonde reconnaissance.

Bill, qui m'a aussi conseillé dans différentes directions de recherche, a passé beaucoup de temps à comprendre et améliorer mes propres recherches. Je l'admire pour sa qualité d'écoute et sa volonté sans faille de faire avancer la Science.

Cette thèse est avant tout dédiée à celle qui est devenue ma femme. L'amour et la patience dont a fait preuve Amandine tout au long de ces trois années de thèse ont été remarquables.

Je remercie en notamment :

– Stéphane pour son soutien scientifique et ses bons conseils de lecture,
– la team 111 : Julien pour ses cours de cyclisme et Gérard pour sa vision parallèle et originale du monde qui nous entoure.[1],
– Olivier pour les discussions scientifiques et les "repas" au Quick, jusqu'à deux heures du mat lorsqu'il finissait sa thèse en squattant avenue de Valmy,
– Aris et son monde de théorèmes,
– Jérome, le chercheur cherchant à être un bon chercheur. Un seul regret : ne pas lui avoir trouvé un problème d'optimisation original qu'il soit en mesure de résoudre,
– Les lapins : Mathieu, Maryline et Tristan pour leur amitié et leur barbecue à l'improviste,
– tous les autres membres de l'équipe IS2/MISTIS avec qui j'ai partagé de nombreux repas : Florence, Christian, Henry, Françoise, Elodie, Jean-Baptiste, Emilie, , Franck et Mohamed,
– les chercheurs INRIA dans leur ensemble. Claude et l'escalade, Sabine et les BX, Eric et la peinture, Alain

[1] J'espère tout de même qu'il passera un jour le permis de conduire

et le vin, Frédéric et la Clairette, Cordelia et Niels, Radu et la montagne, Peter et son sourire, Edmond et le sport, Thierry et le Tai-Shi.

- Chantal pour l'aide qu'elle m'a apporté à la Documentation, et Ben pour les cours de Tae Kwan Do.
- l'équipe du LMC, le groupe de travail FIMA pour l'organisation de séminaires intéressants,
- Navneet et Ankur pour les bons moments et les nuits blanches partagés dans le même bureau,
- Gyorgy pour ses discussions, ses jeux de société et son vin,
- le reste de l'équipe Lear qui tient bien sa réputation de *groupe le plus travailleur de l'INRIA*[2] : Frédéric, Peter, Eric, Jango, Salil, Michael,
- Gilbert, Eric, Jean-Marc et Marc : les personnes à Xerox, qui m'ont fait confiance avant que je soutienne ma thèse. Je les remercie aujourd'hui de m'avoir embauché,
- plein d'autre personnes que j'ai pu rencontrés, comme Léo, Carine, David, Timon, Pär, Nicolas, Guillaume Saint-Pierre, etc.
- mes beaux-parents : Nicole pour les corrections orthographiques indispensables et à Jean claude pour ses "Alors Guillaume, quand est-ce que vous finissez votre thèse ?".

Enfin, ma famille, même si elle est éparpillée aux quatre coins du monde a toujours été là lorsque j'en avais besoin et je lui suis grandement reconnaissant.

[2] Il suffit de venir le week-end pour s'en convaincre.

Table des matières

Chapitre 1

Introduction

1.1 L'apprentissage statistique

À l'intersection entre l'intelligence artificielle et les statistiques, l'*apprentissage statistique*, aussi appelé *machine learning*[3] a pour but la résolution automatique de problèmes complexes à partir d'exemples.

La croissance actuelle de la puissance des ordinateurs et des réseaux de télécommunications crée de nouveaux besoins, et permet à des applications innovantes d'apparaître. Ainsi, le développement d'internet à l'échelle mondiale a rendu indispensable les moteurs de recherche efficaces et les filtres anti-spam du courrier électronique. Parallèlement, l'utilisation montante des images numériques donne lieu à une multitude de nouvelles possibilités, comme la catégorisation automatique des photos, la reconnaissance de visage, ou l'aide au diagnostic médical. Ces problèmes doivent par essence être gérés informatiquement, mais peuvent être aisément résolus par un opérateur humain s'il dispose du temps nécessaire. Il est aujourd'hui possible d'effectuer ces tâches de manière automatique, après une phase *d'apprentissage* basée sur des observations passées.

Mais l'apprentissage statistique ne se limite pas à une simple imitation du comportement humain, et n'est pas seulement lié aux nouvelles technologies. Il intervient aussi dans des contextes industriels complexes. Par exemple, déterminer une stratégie de maintenance préventive est d'un grand intérêt industriel. Dans ce type d'application,

[3] En réalité, le *machine learning* englobe aussi des méthodes d'apprentissage non statistiques, mais elle ne seront pas abordées dans ce document.

les outils d'apprentissage statistiques permettent de prendre en compte à la fois :

- la complexité du contexte : facteurs humains, études en laboratoire, coûts des contrôles, mesures physiques, *etc*.
- l'observation effective des défaillances passées, c'est-à-dire le *retour d'expérience* (REX).

La reconnaissance des caractères manuscrits pour le classement du courrier et le traitement automatique de la parole pour la redirection téléphonique sont des exemples d'applications très répandues des techniques de classification automatique. Elle permettent de s'affranchir de l'intervention humaine pour effectuer des tâches répétitives. Cependant, lors de la phase d'apprentissage précédant toute utilisation du classifieur, un grand nombre d'exemples de classifications correctes sont nécessaires pour obtenir des résultats acceptables. En général, la construction de ces bases de données d'apprentissage requiert un travail « manuel » de classification qui peut être fastidieux. Ce type d'approche est appelé *apprentissage supervisé*, car il est possible de confronter les réponses du système aux réponses du « tuteur » humain.

1.1.1 Contexte

Les méthodes d'apprentissage statistique analysent des *entrées* (images, sons, mesures, symptômes, *etc*.) pour en déduire des sorties (catégorie d'objet, phrase, probabilité de défaillance, diagnostic, *etc*.). Nous sommes dans un cadre *supervisé* si les sorties sont observées, *non supervisé* si elles ne le sont pas et *semi-supervisé* si elles ne sont que partiellement observées. On peut classer ces méthodes en deux catégories :

- les approches *discriminatives* ou de *régression* consistent à modéliser les relations qui lient les entrées aux sorties du système, avec un minimum d'hypothèses sur la structure des données d'entrée. Ces méthodes répondent directement à l'objectif de l'utilisateur en se focalisant sur la règle de décision plutôt que sur l'interprétation de cette décision. Ces méthodes ne sont possibles que dans le cadre supervisé, ou semi-supervisé.
- les approches *génératives* modélisent en premier lieu la structure du système. La réponse à l'objectif de l'utilisateur est déduite de ce modèle. Le terme « génératif » vient du fait que la structure des données est généralement obtenue en modélisant le processus de création des données. Par rapport aux méthodes discriminatives qui ne s'intéressent qu'aux relations entrées-sorties, ces approches modélisent en plus les

liens qui existent au sein des variables d'entrée. Ainsi, le travail de modélisation est plus important.

Ces deux approches sont distinctes et complémentaires. Lorsqu'un grand nombre de données d'apprentissage sont disponibles, certaines méthodes discriminatives non linéaires sont très performantes. A l'inverse, les méthodes génératives offrent des outils de modélisation élaborés qui permettent de limiter la quantité de données nécessaires à l'apprentissage.

1.1.2 Problématique

En apprentissage supervisé, les méthodes génératives sont parfois considérées comme sous-optimales. Une citation à propos de ces méthodes résume bien cette pensée : « One should solve the [classification] problem directly and never solve a more general problem as an intermediate step [comme modéliser la distribution de toutes les données] » (Vapnik, 1998 [158]). Cette remarque a probablement freiné le développement des méthodes génératives au profit des méthodes discriminatives qui résolvent le problème « directement ».

Mais les problèmes auxquels s'intéresse aujourd'hui la communauté de l'apprentissage statistique sont de plus en plus complexes et la seule utilisation des données d'apprentissage limite les performances. La prise en compte d'informations extérieures et notamment la structure des données devient indispensable. La modélisation de cette structure peut se faire grâce aux outils des méthodes non supervisées à travers l'approche générative. On pense notamment à l'efficacité des modèles graphiques pour modéliser des systèmes complexes.

Il semble donc nécessaire de proposer des outils efficaces qui manquent aux approches supervisées basées sur des modèles génératifs. Les questions suivantes n'ont aujourd'hui pas de de réponse claire :

- Est-ce que l'estimateur classique du maximum de vraisemblance pour les modèles génératifs est adapté à un cadre de la discrimination ou de la régression ?
- J'hésite entre plusieurs modèles génératifs. Comment choisir celui qui me donne le meilleur taux de classification sur des données de test ? Y a-t-il des alternatives à la validation croisée ?
- L'estimation ponctuelle des paramètres d'un modèle génératif se fait par minimisation d'une fonction de coût. Comment choisir cette fonction ?
- Y a-t-il des modèles génératifs de référence, c'est-à-dire s'adaptant à toute sorte de données ? Quel est l'intérêt des modèles à classes latentes pour la classification supervisée ?

– J'ai très peu d'observations, mais une forte connaissance *a priori* du système. Comment construire un modéle bayésien cohérent et estimer ses paramètres ?

Nous allons, au cours de cette thèse, tenter d'apporter des éléments de réponse à ces questions.

Dans la suite de ce chapitre, nous introduirons les domaines d'applications abordés dans la thèse puis nous récapitulerons les développements récents de l'apprentissage statistique. Enfin, nous détaillerons la structure de la thèse.

1.2 Les domaines applicatifs

La plupart des méthodes proposées dans cette thèse proposent des applications sur des données réelles. En particulier, deux chapitres ont été dédiés à des domaines spécifiques : la vision par ordinateur et la fiabilité industrielle. Nous donnons ici un bref aperçu de ces disciplines.

1.2.1 La vision par ordinateur

La vision par ordinateur est un domaine scientifique à part entière, mais une partie de ses applications est en lien étroit avec l'apprentissage statistique. Elle est d'ailleurs parfois nommée *reconnaissance des formes*.

Aujourd'hui, les techniques de vision artificielle sont appliquées principalement en inspection et contrôle de qualité, en télédétection et en vision 3D. La reconnaissance de documents manuscrits permet d'authentifier le signataire d'un chèque ou encore d'automatiser le traitement des enveloppes postales ou de tout autre formulaire administratif. Dans le domaine médical, le traitement des images permet de réhausser le contraste d'informations pertinentes, afin de faciliter l'interprétation. Cependant, ces applications commerciales se limitent à des cadres très spécifiques, où les images proviennent du même appareil et contiennent un objet généralement entier et situé au centre de l'image.

La vision par ordinateur a aussi un intérêt pour les chercheurs d'autres disciplines scientifiques, dont la météorologie, la cosmologie, les neurosciences, la robotique, etc. En effet, les caméras numériques se généralisent et leurs fonctionnalités augmentent. La compréhension automatique des images environnantes est une étape indispensable pour le traitement d'un grand nombre d'images, et, dans le cas de la robotique, pour la construction

d'agents autonomes. Les résultats existants sont très limités par la grande complexité du problème à traiter et la plupart des méthodes nécessitent des images de très bonne qualité pour être effectives.

Dans cette thèse le problème de la catégorisation d'objet sera considéré, pour lequel l'objectif recherché est d'affecter une catégorie à une nouvelle image. Une des difficultés réside dans le type d'objet que l'on cherche à classer. Ainsi, un objet dont la forme globale varie peu, comme une voiture ou un avion, sera relativement facile à classer, en comparaison avec des objets définis par un attribut non visuel, tels que les chaises. En outre, un problème inhérent aux images réelles est que l'information pertinente n'est pas séparée du fond et qu'on ne connaît pas *a priori* la position et la taille de l'objet dans l'image, ce qui rend l'apprentissage automatique extrêmement difficile.

1.2.2 La fiabilité industrielle

La sûreté de fonctionnement des systèmes complexes est un enjeu majeur pour l'industrie. Les pannes et défaillances sont des éléments imprévisibles qui peuvent avoir un coût exhorbitant en termes économiques et humains. Inversement, la mise en place d'une politique sévère de sûreté, par exemple en multipliant les maintenances et en achetant des machines de qualité optimale peut avoir un coût très élévé. Toute entreprise industrielle susceptible d'être victime de défaillances cherche donc à minimiser le coût global de ce risque, c'est-à-dire la somme du coût moyen (ou espéré) des pannes et du coût de la maintenance.

On comprend que la fiabilité contient intrinsèquement un aspect probabiliste, et de nombreux modèles statistiques ont été proposés pour évaluer le taux de défaillance de matériels. Cependant, ces modèles sont souvent réduits à un seul type de machine et prennent rarement les informations extérieures en compte, comme les conditions d'utilisation. La raison est très simple : les défaillances étant en général très rares, il est très difficile en statistique classique d'utiliser plus de variables que le nombre d'observations de défaillance.

Parallèlement à ces outils portant sur des éléments isolés de leur contexte, il existe des stratégies de maintenances de systèmes industriels complexes. Par exemple, la Gestion de la MAintenance par Ordinateur (GMAO ou CMMS en anglais) propose des outils informatiques élaborés associant de nombreux facteurs, tels que les observations passées des défaillances, les plannings des équipes de maintenance, les mesures issues de sondes thermiques, etc. Ces méthodes manquent cependant d'outils probabilistes performants à cause de la spécificité de chaque application, et se limitent en général à des statistiques descriptives. Dans ce cadre, les *réseaux bayésiens*

sont parfaitement adaptés à la modélisation probabiliste des systèmes complexes, grâce à leur conception modulaire. De plus en plus de chercheurs et d'ingénieurs utilisent de telles approches pour résoudre les problèmes de maintenance.

Pour répondre au problème du faible nombre d'observations de défaillance, la statistique bayésienne, associée aux réseaux bayésiens, prend tout son sens : à travers des lois *a priori*, des informations extérieures sont prises en compte de manière naturelle lors de l'estimation des paramètres. Grâce à ces deux outils, le choix d'une politique globale de maintenance préventive se ramène à l'estimation d'un modèle probabiliste.

Nous considèrerons dans le chapitre 7 la fiabilité d'éléments métalliques dont le modèle de dégradation est connu, avec un nombre réduit d'observations de défaillance et une très forte connaissance *a priori*, notamment la prise en compte des études en laboratoire et les mesures effectuées sur les sites d'exploitation. L'alliance des réseaux bayésiens et de la statistique bayésienne porte ses fruits et une politique de maintenance est déduite à partir des probabilités de défaillance.

1.3 État de l'art

On distingue deux grandes catégories d'apprentissage statistique :

- les méthodes *supervisées*, ont un rôle *prédictif* : elle permettent d'évaluer la distribution d'une quantité (e.g. la taille d'un individu) sans la mesurer directement, mais en se basant sur des valeurs qui lui sont liées (e.g. le poids de la personne),
- les méthodes *non-supervisées*, dont le rôle est principalement *descriptif*, s'attachent à isoler l'information utile au sein d'un jeu de données.

Pour répondre à ces deux problèmes, (*régression* et *estimation de densité*), la différence fondamentale entre la statistique classique et le *machine learning* est que ce dernier ne se base pas nécessairement sur la théorie des probabilités.

Cette thèse se focalise sur l'apprentissage supervisé, qui répond à un objectif clair : minimiser un coût ou une erreur de prédiction sur des données de test. Cependant, les différents chapitres montreront que les méthodes non-supervisées ont un grand intérêt pour la construction de méthodes supervisées.

1.3.1 L'apprentissage supervisé

L'apprentissage supervisé fait intervenir deux types de variables :

– les variables d'entrée, notées X. En statistique, ces variables sont appelées covariables ou régresseurs,

– les variables de sortie, notées Y. Ces variables doivent être prédites à partir de la valeur de X associé.

La modélisation du lien entre X et Y est donc primordiale. Lorsque la variable Y est discrète, on peut associer chaque valeur possible à une catégorie. Ce type de problème est appelé *classification supervisée*.

Y a-t-il une différence réelle entre la régression et la classification supervisée ? D'un point de vue décisionnel, il y a une différence fondamentale entre ces deux approches puisqu'au final on ne souhaite pas une loi de probabilité mais une réponse binaire (dans le cas de deux classes). Certain auteurs insistent sur cette différence en précisant que la classification est un problème plus simple que la régression [158, 100]. En réalité ces considérations sont purement théoriques, car l'utilisation finale n'est pas toujours décisionnelle et nécessite au contraire une probabilité. C'est notamment le cas en fiabilité statistique où la probabilité d'observer Y est extrèmement faible, mais est d'un grand intérêt applicatif. De plus, les applications de l'apprentissage statistique sont souvent inclues dans des contextes globaux où une décision n'est pas prise uniquement à partir du résultat du classificateur, mais aussi à partir d'informations extérieures. On pense par exemple au diagnostic médical, qui ne peut pas être restreint aux variables mesurées, comme les résultats d'analyses médicales, mais doit aussi tenir compte d'un contexte difficilement mesurable, tel que le passé ou l'état psychologique d'un patient. Avoir une probabilité au lieu d'une réponse binaire prend donc là aussi tout son sens.

On peut différencier deux types d'approche pour résoudre un problème d'apprentissage supervisé :

– l'approche *discriminative* modélise directement la règle de classification $P(Y|X)$,

– l'approche *générative* (parfois appelée informative [142]) cherche à modéliser la distribution jointe $P(X, Y)$ des entrées et des sorties. On en déduit ensuite la règle de classification par application de la loi de Bayes.

La figure 1.1 regroupe les principales méthodes de classification génératives et discriminatives.

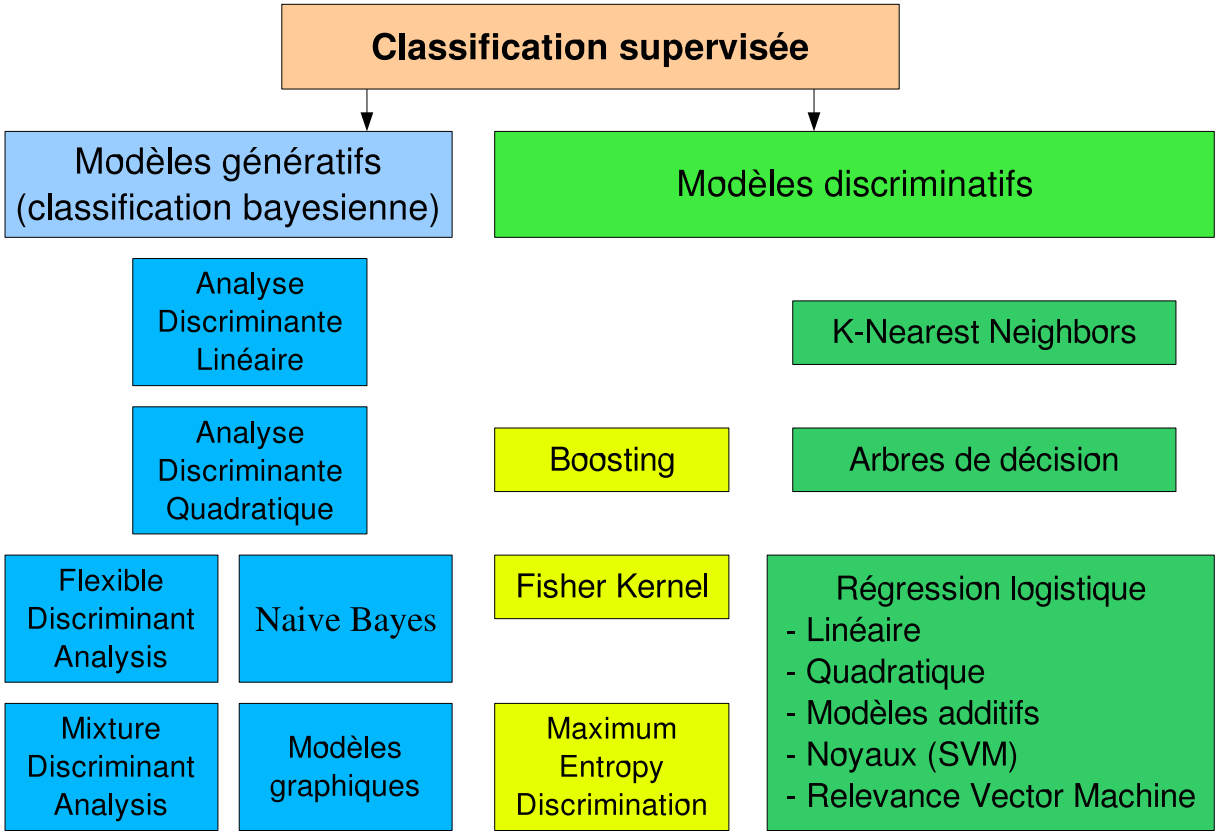


FIG. 1.1 – Les principales méthodes de classification, d'un côté génératives, de l'autre discriminatives. Les méthodes indiquées par une couleur claire sont des méthodes discriminatives qui peuvent être liées à un modèle génératif.

1.3.2 L'approche discriminative ou conditionnelle

Les méthodes discriminatives proposent un modèle pour $P(Y|X)$. Les paramètres (ou les coefficients dans le cas des méthodes non-paramétriques) de ce modèle sont estimés par minimisation du coût de classification, éventuellement régularisé par une pénalité sur les coefficients. Ces approches, très efficaces en pratique, ne seront abordées explicitement que dans le chapitre 8, mais la vision que nous avons des méthodes génératives (chapitres 2 à 7) est très liée à l'objectif « discrimininatif » de classification ou de régression. Nous introduisons donc ici ces approches avec plus de détail.

– la Minimisation du Risque Empirique (ERM) est généralement un problème combinatoire très difficile à

résoudre car la fonction à minimiser n'est pas dérivable et le problème n'est pas convexe,

- la minimisation du risque L_1 est un problème convexe non dérivable, et peut être résolu dans les cas les plus simples par programmation linéaire ou quadratique sous contrainte linéaire. C'est la méthode utilisée dans les classifieurs SVM, et c'est notamment celle qui sera utilisée dans le chapitre 8,
- en classification, la minimisation du coût logistique, équivalent au maximum de vraisemblance conditionnelle, est un problème convexe et infiniment dérivable. La solution est obtenue par descente du gradient. nous utiliserons cette fonction de coût pour estimer les modèles génératifs de manière discriminative (voir les chapitres 2 et 5),
- en régression, la minimisation du risque quadratique L_2 est lui aussi équivalent au maximum de vraisemblance conditionnelle, sous l'hypothèse de normalité de la variable de sortie $Y|X$. C'est le problème des moindres carrés dont la solution est explicite dans le cas linéaire.

Pour les trois premières fonctions de coût, les résultats en classification et en régression sont en général très proches [71, 158].

La régression logistique Le modèle probabiliste généralement adopté pour modéliser $P(Y|X)$ est une loi multinomiale dont les paramètres dépendent de X à travers une fonction seuil dérivable appelée *softmax*. Dans le cas binaire c'est une loi de Bernoulli dont le paramètre de proportion correspond à la fonction logistique $\left(1+\exp(-\beta^T X-\beta_0)\right)^{-1}$ où β est le vecteur de paramètres.

Pour obtenir une règle de décision non linéaire, les variables d'entrées X sont transformées en d'autres variables $\tilde{X} = \varphi(X)$ à partir desquelles une méthode de classification linéaire est appliquée. La fonction $\varphi(X)$ définie plus haut pour le paragraphe sur la régression logistique peut prendre des formes très variées et dépend fortement du domaine d'application. Pour les *réseaux de neurones*, par exemple, φ est un ensemble de fonctions logistiques appliquées aux entrées, dont les poids sont obtenus eux aussi par minimisation d'une des fonctions de coût précédemment décrites. En général, les problèmes de minimisation sont très complexes et le choix de φ est assez difficile. Cette fonction peut elle-même dépendre de paramètres qui ne sont pas choisis en minimisant le risque empirique (par exemple le rayon des fonctions radiales de base). Les méthodes à noyau permettent de ne pas définir explicitement cette fonction φ et garantissent l'unicité de la solution.

Les méthodes à noyau Les *Machines à Vecteur Support* (SVM) sont un ensemble de méthodes non paramétriques de classification discriminatives, de régression ou d'estimation de densité, basées sur des estimateurs de fonctions à noyau, dont la forme générale est

$$f(x) = \sum_i \alpha_i K(x, x_i)$$

où $\{x_i\}_{i=1,\ldots,n}$ est l'ensemble des données d'apprentissage, α_i des coefficient réels et $K(x, x')$ une fonction noyau. En général, ce type de fonction a des propriétés d'intégrabilité, de positivité et de symétrie [145].

Comme nous l'avons précisé plus haut, l'avantage principal des SVM est que la fonction φ n'a pas besoin d'être spécifiée explicitement. À la place, on détermine la *fonction noyau* $K(x, x')$ entre deux entrées x et x'. Le choix du noyau s'avère cependant un problème ardu qui nécessite de véritables expertises, en particulier lorsque des données extérieures doivent être ajoutées au problème.

Dans les faits, de nombreux chercheurs construisent directement des données X en très grande dimension, puis obtiennent les meilleurs résultats de classification avec un classifieur SVM linéaire (voir par exemple [31, 37, 4] en vision par ordinateur et [119, 60] en classification textuelle). Il n'y a donc pas d'utilisation de fonction noyau. Dans ce cas, il est important de souligner qu'une simple régression logistique régularisée donnerait des résultats très similaires en classification par rapport au classifieur SVM linéaire puisqu'ils sont basés sur des fonctions de coût asymptotiquement équivalentes [158].

Un autre avantage des SVM est qu'elles permettent d'obtenir une règle de classification non-paramétrique « parcimonieuse », dans le sens où elle ne dépend que d'un nombre réduit de vecteurs supports [145, 153]. Cette caractéristique, intéressante lorsque le nombre de données d'apprentissage est important et que la mémoire disponible est limitée, n'est en réalité pas spécifique aux méthodes à noyau, puisqu'il suffit à toute méthode basée sur la minimisation d'une fonction de coût, d'ajouter une pénalité ou une contrainte sur les coefficients non nuls pour obtenir un estimateur parcimonieux.

Autres méthodes discriminatives Il existe bien d'autres méthodes de classification, en particulier *la méthode des k plus proches voisins*, *les arbres de décision*. Ces derniers peuvent être stabilisés par *boosting* pour offrir une règle de décision particulièrement simple et efficace. Ces méthodes non-linéaires ne sont pas détaillées ici car la loi de $P(Y|X)$ n'a pas une forme simple, mais elles sont très appréciées en pratique, autant pour leur simplicité

que pour leurs performances en classification.

Intérêts et défauts de l'approche discriminative Les partisans d'une approche purement discriminative ont comme argument que la minimisation d'une fonction de coût empirique répond précisément à l'objectif recherché sur un échantillon de test. En effet, les paramètres du modèle sont tous utilisés pour estimer la frontière de classification et pas autre chose (ceci n'est pas vrai pour les méthodes génératives).

Les approches discriminatives sont surtout efficaces lorsque les données X ont une distribution inconnue et difficilement modélisable. Dans ce contexte, les résultats théoriques sur les bornes du risque empirique peuvent être intéressants puisqu'ils cherchent à minimiser le risque pour la distribution la plus pénalisante pour le modèle [24] (approches de type *minimax*).

Cependant, ces méthodes incorporent difficilement des invariances ou des structures propres aux données, comme des indépendances conditionnelles ou des variables latentes, et même si elles sont optimales asymptotiquement, elles peuvent être dépassées par des approches génératives pour des tailles finies d'échantillon d'apprentissage.

Dans les études comparant des méthodes génératives et discriminatives de classification, ce sont ces dernières qui donnent les meilleurs résultats (voir par exemple [89, 166, 62] pour la classification de textes, [31] pour la catégorisation d'objets). Cependant, ces comparaisons sont souvent biaisées, car les méthodes génératives accèdent à tout leur potentiel discriminatif si une véritable *modélisation* probabiliste des données est effectuée.

1.3.3 La modélisation générative

La modélisation générative cherche en premier lieu à trouver une structure pour la distribution jointe des entrées X et sorties Y. Elle correspond à une vaste catégorie de classifieurs, dont les plus simples (et souvent très performants) sont l'*Analyse Discriminante Linéaire* (LDA) et les *classifieurs de Bayes Naïfs* (NB).

On parle de *modélisation générative* car une loi de probabilité jointe est définie pour *toutes* les variables possibles, c'est-à-dire pour les données à prédire Y, les données d'entrée X *et* des variables annexes Z non observées (cachées ou latentes). En ce sens, un travail de modélisation supplémentaire est nécessaire par rapport aux approches discriminatives. Cependant, même si plus de variables doivent être modélisées, il est souvent bien plus

facile de définir une loi de probabilité jointe des données. En effet, le développement récent des modèles graphiques et des modèles à classes latentes sont des outils particulièrement adaptés à la construction d'un modèle paramétrique cohérent.

Les méthodes génératives seront présentées plus en détail dans le chapitre suivant, et font l'objet de la majeure partie de la thèse (six chapitres sur sept). En particulier, nous verrons qu'il y a plusieurs moyens d'estimer les paramètres, suivant la fonction de coût (générative ou discriminative) que l'on minimise. Les différents points abordés dans les chapitres suivants sont la régression générative dans le cas des mélanges de distribution, le choix d'un modèle génératif en classification, l'introduction d'un intermédiaire entre l'estimation générative et discriminative, et enfin deux applications en vision par ordinateur et en fiabilité industrielle.

1.3.4 Modèles paramétriques ou non paramétriques ?

L'apprentissage statistique nécessite la définition de modèles de densités (jointes ou conditionnelles) pour les variables considérées. Dans de nombreuses applications, on ne dispose pas directement d'un modèle adapté à des données multidimensionnelles. Le choix du modèle de densité est bien entendu très important pour obtenir un taux de classification convenable. On peut modéliser ces densités par un *modèle paramétrique* ou un modèle *non paramétrique*.

Le point de vue d'un modèle paramétrique est de contraindre les distributions à épouser un modèle rigide comportant un nombre fini de paramètres, c'est-à-dire que la famille de probabilités qu'il définit peut être paramétrée par un vecteur θ dans un espace Θ de dimension finie. Lorsqu'on a une idée précise de la structure des données, il est généralement aisé de paramétrer la distribution. Les modèles génératifs proposés dans cette thèse seront paramétriques.

En revanche, les méthodes non paramétriques considèrent un espace paramétrique dont la dimension dépend de l'échantillon d'apprentissage, et permettent asymptotiquement d'obtenir une classe de probabilités dense dans L_2. Ainsi, toute probabilité intégrable peut être approchée par une fonction de la classe considérée. Cette propriété de consistance se paye par le besoin de *régularisation* pour éviter le *sur-apprentissage* : afin de ne pas reproduire dans le modèle toute l'information issue des données d'apprentissage, il est nécessaire de contraindre les solutions à être suffisamment « lisses ». Ces estimateurs non paramétriques sont efficaces dans les méthodes discriminatives

car ils se basent sur un minimum d'hypothèses sur la frontière de classification ou la fonction à prédire. Notons que ces méthodes considèrent souvent une base de fonctions (noyaux, splines, Fourier), ce qui permet de percevoir un problème de classification non linéaire dans l'espace d'origine de manière linéaire dans l'espace fonctionnel (communément appelé *feature space*). Nous étudierons un estimateur non paramétrique à noyau dans la deuxième partie de la thèse consacrée à l'approche discriminative.

1.3.5 Les modèles graphiques

Les applications de l'apprentissage statistique nécessitent généralement la prise en compte de plusieurs centaines de variables. Face à cette complexité croissante, des modèles modulaires pouvant être représentés sous la forme de graphe sont des outils devenus indispensables à la modélisation et à l'interprétation.

Les modèles graphiques encore appelés *réseaux bayésiens*, ont l'avantage de fournir un formalisme qui allie :

– un graphe de dépendance entre les variables,

– un sens probabiliste des arcs et des noeuds clairement défini.

La construction d'un modèle de grande taille se fait partie par partie, tout en étant garanti que le modèle final correspond à une loi de probabilité.

La majorité des modèles statistiques existants peuvent se mettre sous la forme d'un modèle graphique, en incluant accessoirement des variable inobservées. Nous nous efforcerons de donner la structure des modèles génératifs considérés dans cette thèse sous la forme d'un modèle graphique.

1.3.6 La statistique bayésienne

La statistique bayésienne consiste à considérer que les paramètres d'un modèle sont aléatoires. Il est donc nécessaire de définir une distribution *a priori* de ces paramètres de dépendant pas des observations. Autrefois très constestée pour le caractère subjectif de cette loi *a priori*, l'approche bayésienne est aujourd'hui un outil reconnu de l'apprentissage statistique. Elle permet d'incorporer de manière probabiliste des informations extérieures aux données, souvent cruciales pour la reconnaissance. Lorsque la quantité de données disponibles ne suffit pas à estimer le modèle de manière fiable, l'ajout d'informations permet d'améliorer considérablement les performances d'une méthode d'apprentissage statistique. On parle généralement dans ce cas de *régularisation bayésienne*.

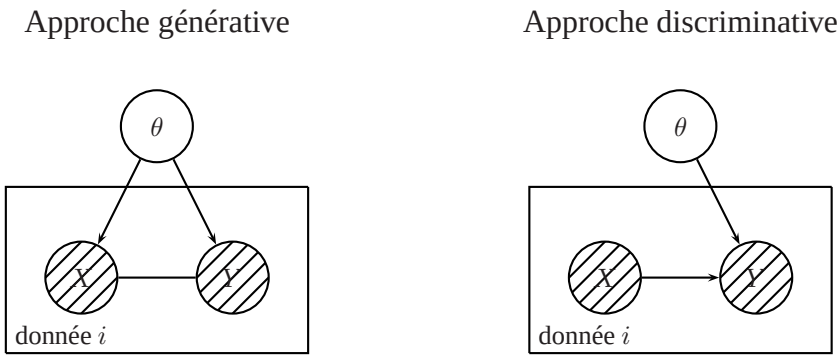


FIG. 1.2 – Modèles graphiques illustrant les approches bayésiennes pour la classification supervisée. X représente les entrées (ou covariables), Y les sorties à prédire et θ le vecteur des paramètres. Les variables observées sont indiquées par des hachures. Les cadres (*plates* en anglais) indique une répétition indépendante des variables.

Les approches génératives et discriminatives dans un cadre bayésien sont représentées sous la forme modèles graphiques sur la figure 1.2. L'approche générative modélise la densité jointe des entrées et des sorties alors que l'approche discriminative définit une distribution des sorties conditionnellement aux entrées et à aux paramètres. Dans la majorité des cas, la distribution *a priori* des paramètres ne dépend pas des entrées, ce qui explique l'absence de lien entre X et θ.

Le paradigme bayésien est utilisé dans le chapitre 7, consacré à l'étude d'un composant physique soumis à des dégradations, car on dispose d'une forte information *a priori* sur les paramètres liée à des études antérieures. Cette information est déterminante dans l'estimation des risques liés à la défaillance des composants, car très peu de données sont disponibles dans le problème considéré.

1.4 Contributions et organisation de la thèse

Cette thèse propose plusieurs outils novateurs pour l'apprentissage statistique supervisé. La plupart des chapitres sont consacrés à la modélisation générative. Seul le chapitre 8 présente une méthode discriminative.

– Le *chapitre 2* définit l'approche générative dans un cadre de classification. Les deux principales manières d'estimer les paramètres sont introduites : l'estimation générative maximisant la vraisemblance jointe et l'estimation discriminative, maximisant la vraisemblance conditionnelle des sorties sachant les entrées. Les

aspects théoriques de ces deux types d'estimation sont analysés afin de mieux comprendre leurs intérêts respectifs. Nous rappelons que le modèle d'Analyse Discriminante Linéaire (LDA) estimé de manière discriminative correspond à la régression logistique linéaire (LLR). En particulier, nous montrons que ce résultat est vrai en utilisant la paramétrisation générative de LDA, composée des moyennes des classes et de la matrice de covariance commune. Cette paramétrisation permet d'associer à la régression logistique un modèle gaussien pour chaque classe. Ce résultat sera utilisé dans le chapitre 5 pour proposer un nouveau type d'estimation.

– Le *chapitre 3* présente les mélanges de régressions, aussi appelés mélanges d'experts (ME). Lorsque nous modélisons la densité des covariables par un mélange de distributions gaussiennes, nous montrons que l'estimation générative des paramètres se fait aussi par l'algorithme EM. Cette modification a plusieurs avantages par rapport à l'algorithme discriminatif (*i.e.* conditionnel). Il est plus facile d'introduire des contraintes sur les paramètres du modèle, en fonction des volontés du modélisateur (robustesse, hétéroscétasticité, multimodalité, etc.). La complexité algorithmique est aussi plus faible, car l'estimation d'un modèle linéaire généralisé est remplacée par la résolution d'un problème de moindres carrés linéaires pondérés. Enfin, l'estimateur génératif a une variance plus faible que l'estimateur conditionnel. Nous montrons dans une application de prévision du taux d'ozone atmosphérique que cette approche est en effet très stable, ce qui se traduit par de meilleures prévisions lorsqu'il y a peu de données d'apprentissage.
– Dans le *chapitre 4*, le choix d'un modèle probabiliste pour l'analyse discriminante est étudié. Les critères classiques de sélection de modèle privilégient l'adéquation du modèle à la distribution jointe des variables explicatives et de la variable de groupe plutôt que la minimisation du taux d'erreur du classifieur associé. Nous proposons un nouveau critère, le *Bayesian Entropy Criterion* (BEC), qui permet de sélectionner un classifieur prenant en compte l'objectif décisonnel par la minimisation de l'entropie intégrée de classification. Il représente une alternative intéressante à la validation croisée qui est très coûteuse. Les propriétés asymptotiques du critère BEC sont présentées, et des expériences numériques sur des données simulées et des données réelles montrent que ce critère a un comportement meilleur que BIC pour choisir le modèle minimisant l'erreur de classification, et analogue à celui de la validation croisée.
– Dans le *chapitre 5*, un nouveau type d'estimation des modèles génératifs est proposé : un intermédiaire

entre l'estimation générative et discriminative (*Generative-Discriminative Tradeoff, GDT*). L'estimateur est obtenu en maximisant la somme pondérée des vraisemblances jointe et conditionnelle. Plutôt que de choisir entre l'une ou l'autre des méthodes d'estimation, le coefficient de pondération entre les deux log-vraisemblances doit être sélectionné par validation croisée. Le taux de classication sur des données de test est amélioré par cette méthode car le biais de classification propre à l'estimateur génératif est réduit sans pour autant avoir une variance aussi élevée que l'estimateur discriminatif. Des résultats théoriques donnent les hypothèses sous lesquelles l'estimation GDT est meilleure que l'estimation générative *et* discriminative. Deux expériences numériques, une avec des covariables discrètes, l'autre avec des covariables gaussiennes, illustrent ce phénomène. Les résultats sur données réelles en utilisant le classifieur de Bayes naïf montrent que ce type d'estimation est une alternative prometteuse aux estimateurs génératifs et discriminatifs.

– Le *chapitre 6* est un exemple d'application des méthodes génératives à la *catégorisation d'objets* en vision par ordinateur. Nous proposons un modèle génératif qui décrit la loi de probabilité des points d'intérêt (*i.e*. descripteurs locaux d'images numériques). Un modèle hiérarchique des parties permet de prendre en compte les corrélations spaciales des points d'intérêts. L'algorithme d'apprentissage gère efficacement plusieurs centaines de points d'intérêt, ce qui rend la méthode particulièrement robuste aux occlusions et aux images bruitées. La méthode est totalement invariante par transformation d'échelle, et les performances en classification sur des données multiclasses sont très prometteuses.
– Le *chapitre 7* est un exemple d'application de l'apprentissage supervisé à la fiabilité industrielle. Des éléments métalliques sont soumis à des contraintes de fabrication et de fonctionnement et sont susceptibles de fissurer. Un modèle graphique modélise l'apparition de ces fissures, ainsi que des les informations extérieures (température, contrainte de fabrication, type de matériau) qui peuvent influencer la date de leur apparition. Les paramètres du modèle graphique sont estimés de manière bayésienne, car l'information *a priori* est très présente dans ce modèle. L'échantillonnage de Gibbs est utilisé, avec une étape de simulation particulièrement délicate qui fait intervenir un calcul FORM de fiabilité des structures. La particularité de l'approche est que les observations de fissures sont en très forte contradiction avec la loi *a priori*, et les paramètres sont réactualisés de manière significative dès qu'une fissure est observée. Après la constatation ou non des défaillances lors des contrôles de maintenance sur site, les risques de défaillance sont réactualisés

par le modèle et la stratégie de maintenance peut être modifiée.

– Dans *chapitre 8*, nous proposons un estimateur à noyau permettant de résoudre le problème de l'estimation de frontière. Ce type de problème correspond à l'estimation du supremum conditionnel à une ou plusieurs covariables, c'est donc un cas typique d'estimation discriminative. L'estimateur correspond à la minimisation de l'aire délimitée par la courbe estimée, sous la contrainte d'être au dessus de tous les les points de l'échantillon d'apprentissage. Cette minimisation sous contrainte est identifiée avec un problème de programmation linéaire. La solution est parcimonieuse dans le sens ou elle ne dépend que d'un nombre limité de points supports. La preuve de la convergence de l'estimateur sous l'hypothèse d'une distribution uniforme des points sous la frontière est donnée. Les performances de l'estimateur sont justifiées de manière théorique et sur données simulées.

Le *chapitre 9* dresse la conclusion de tous les travaux réalisés, et indique des développements futurs des méthodes et les nouveaux problèmes apparus dans cette thèse.

Chapitre 2

Approche générative pour l'apprentissage statistique supervisé

Ce chapitre bibliographique présente l'approche générative pour l'apprentissage statistique supervisé. Comme cela a été présenté dans l'introduction, notre objectif est d'estimer la distribution d'une sortie Y en fonction de variables mesurées regroupées dans un vecteur d'entrée X.

Au lieu de modéliser directement la distribution $\boldsymbol{p}(Y|X)$, le parti pris « génératif » consiste dans un premier temps à modéliser toutes les données, à savoir la distribution jointe $\boldsymbol{p}(X, Y)$. Ensuite, l'application de la loi de Bayes permet d'en déduire la solution recherchée $\boldsymbol{p}(Y|X)$. Le modèle de densité jointe nécessite donc naturellement plus de paramètres.

La modélisation générative, parfois qualifiée d'approche *informative*, tient son nom de l'aspect explicatif qui la lie aux données. Elle répond à la question « Comment pourrait-on générer les données que l'on observe ? ». Evidemment, dans les applications, les données ne sont pas toujours le fruit d'un processus purement aléatoire, et leurs valeurs sont parfois déterministes, mais considérer la loi « génératrice » est un moyen simple de représenter l'information complexe contenue dans les données. D'ailleurs, un moyen de définir un modèle génératif consiste à proposer un processus pour générer aléatoirement les différentes variables du modèle, en incluant éventuellement des variables annexes inobservées. Notons que la simulation effective des données n'est pas toujours réalisable,

comme pour les champs de Markov cachés.

Le choix d'une approche générative a des implications profondes sur les hypothèses inhérentes au modèle. La classe des distributions $\boldsymbol{p}(X, Y)$ exactes pour le modèle est plus petite que dans l'approche conditionnelle, puisque dans ce dernier cas, les hypothèses d'exactitude du modèle portent uniquement sur $\boldsymbol{p}(Y|X)$ et non sur la distribution des entrées $\boldsymbol{p}(X)$. Un modèle génératif est donc par nature plus « contraint » qu'un modèle conditionnel.

Un exemple : la classification de séries temporelles Ce sont justement les hypothèses portant sur la distribution $\boldsymbol{p}(X)$ qui, si elles sont vérifiées, permettent de gagner en qualité d'estimation. Un bon exemple est la classification de séries temporelles : on cherche à trouver la classe Y d'une série (par exemple un signal sonore à reconnaître) en fonction d'une suite de mesures (par exemple des fréquences instantanées) ordonnées dans le temps $X_1, \cdots, X_T$.

Une application « naïve » de l'approche discriminative (*i.e.* conditionnelle) consiste à appliquer une méthode de classification quelconque sur les données d'entrée. A aucun moment on ne prend en compte l'aspect temporel des covariables X_t puisque cela suppose qu'on fait une hypothèse sur leur distribution.

En revanche, l'approche générative consisterait à trouver un modèle adapté à la série $\{X_t\}_t$ pour chaque classe Y possible, par exemple par des chaînes de Markov cachées ou des modèles autorégressifs. Après apprentissage des paramètres de toutes les séries temporelles, on en déduit le classifieur génératif, qui dans les applications donne des résultats bien meilleurs que l'approche discriminative « naïve ».

Naturellement, pour remédier à ce problème, de nombreuses approches discriminatives comportant un aspect temporel ont été proposées, mais d'une manière ou d'une autre, le modèle sous-jacent revient à faire des suppositions sur la distribution des entrées. Par exemple, dans le cas des réseaux de neurones, les connexions entre neurones éloignés temporellement sont interdites. Comme ces hypothèses ne sont en général pas explicites, ces modèles sont souvent liés à un domaine très particulier (reconnaissance de la parole, prédiction boursière) et restent difficiles à manipuler. Ils sont rarement efficaces dans des contextes plus généraux.

Contenu du chapitre Dans ce chapitre, nous formalisons le problème de l'apprentissage dans l'approche générative. Les différentes manières d'estimer les paramètres, à savoir l'estimation générative et l'estimation discriminative, sont analysées. Nous rappellons les principaux résultats théoriques qui s'appliquent à ces deux types d'estimation, et soulevons leurs avantages et inconvénients.

Exemples de classifieurs génératifs Plusieurs exemples de méthodes de classification génératives sont décrites. Nous insistons particulièrement sur l'Analyse Discriminante Linéaire (LDA), dont la contrepartie discriminative est la régression logistique. Nous montrons à ce sujet qu'il est possible de conserver la paramétrisation générative de LDA pour effectuer la régression logistique. Il suffit pour cela d'estimer les moyennes et variances de manière discriminative. Même si la solution n'est pas unique, toutes les solutions qui maximisent la vraisemblance conditionnelle donnent la même règle de classification.

Un autre exemple de classifieur génératif est décrit : l'Analyse Discriminante par Mélange (MDA). Ce classifieur consiste à modéliser les densités des groupes $f_k(x) = \boldsymbol{p}(X = x|Y = k)$ par un mélange de gaussiennes. Les problèmes de sélection de modèle liés à cette méthode motivent le chapitre suivant.

2.1 Modèles de densité et règles de classification

On cherche à prédire au mieux la valeur d'une variable $Y \in \mathcal{Y}$ en fonction de variables associées $X \in \mathcal{X}$ (aussi appelées covariables), où $\mathcal{X}$ et $\mathcal{Y}$ sont des espaces mesurables quelconques. D'un point de vue probabiliste, notre objectif est d'estimer la distribution conditionnelle $\boldsymbol{p}(Y|X)$. Ne connaissant pas les valeurs des variables X qui seront utilisées dans le futur, nous supposons qu'elles suivent une distribution $\boldsymbol{p}(X)$. Cela revient à considérer que la distribution jointe des données $\boldsymbol{p}(X,Y) = \boldsymbol{p}(X)\boldsymbol{p}(Y|X)$ existe.

2.1.1 Le modèle

Comme nous l'avons précisé dans l'introduction, le parti pris de la classification générative est de modéliser la densité jointe $\boldsymbol{p}(X,Y)$ par un modèle $\boldsymbol{p}(X,Y;\theta)$ paramétré par $\theta \in \Theta$, puis de prédire les données de test en appliquant la règle de Bayes pour trouver $\boldsymbol{p}(Y|X)$. L'approche générative se définit comme suit :

Définition 1 *Soit $\boldsymbol{p}(X,Y;\theta)$, $\theta \in \Theta$ un modèle pour la densité jointe de X et de Y définie sur $\mathcal{X} \otimes \mathcal{Y}$. Une* méthode d'apprentissage génératif *est définie par :*

1. *une phase d'*apprentissage *: recherche d'un estimateur $\hat{\theta}$ de θ à partir des données d'apprentissage $(\mathbf{x},\mathbf{y}) = \{(x_i,y_i), i = 1,\ldots,n\}$,*
2. *une phase de* test *: à partir d'une donnée X, la valeur y' est prédite avec une probabilité :*

$$\boldsymbol{p}(Y = y|X = x;\hat{\theta}) = \frac{\boldsymbol{p}(X = x, Y = y;\hat{\theta})}{\int_{\mathcal{Y}} \boldsymbol{p}(X = x, Y = y';\hat{\theta})dy'}, \tag{2.1}$$

Dans le cadre de la *classification supervisée,* la variable à prédire Y est discrète et définit l'index de la classe. La formule (2.1) s'écrit dans ce cas :

$$\boldsymbol{p}(Y = k|X = x;\hat{\theta}) = \frac{\pi_k f_k(x;\hat{\theta})}{\sum_{l=1}^{K} \pi_l f_l(x_i;\hat{\theta})}, \tag{2.2}$$

pour $k = 1,\cdots,K$, où $\pi_k = P(Y = k)$ et f_k est la densité de la k-ième classe.

Les classifieurs génératifs portent différents noms dans la littérature. Ripley (1996) parle de *plug-in rule classifier* [137]. Plusieurs auteurs utilisent le terme *Bayesian Classifiers,* ce qui peut être trompeur car le paradigme bayésien — incluant des loi *a priori* et *a posteriori* — n'est généralement pas utilisé [40, 36, 106].

2.1.2 La règle de décision

Dans un objectif décisionnel, on souhaite prédire une valeur unique pour Y au lieu d'une distribution, on peut associer à chaque donnée son mode *a posteriori* :

$$\hat{y} = \underset{\mathcal{Y}}{\operatorname{argmax}}\, \boldsymbol{p}(X = x, Y = y).$$

Cette règle est généralement utilisée en classification car elle minimise le taux d'erreur de classement. Lorsque les valeurs de Y sont ordonnées — par exemple dans un cadre de régression — on peut préférer choisir de minimiser l'erreur quadratique de prédiction, ce qui revient à choisir l'espérance conditionnelle :

$$\hat{y} = \int_{\mathcal{Y}} y \boldsymbol{p}(Y = y | X = x; \theta) d\boldsymbol{p}(y).$$

2.1.3 L'apprentissage des paramètres

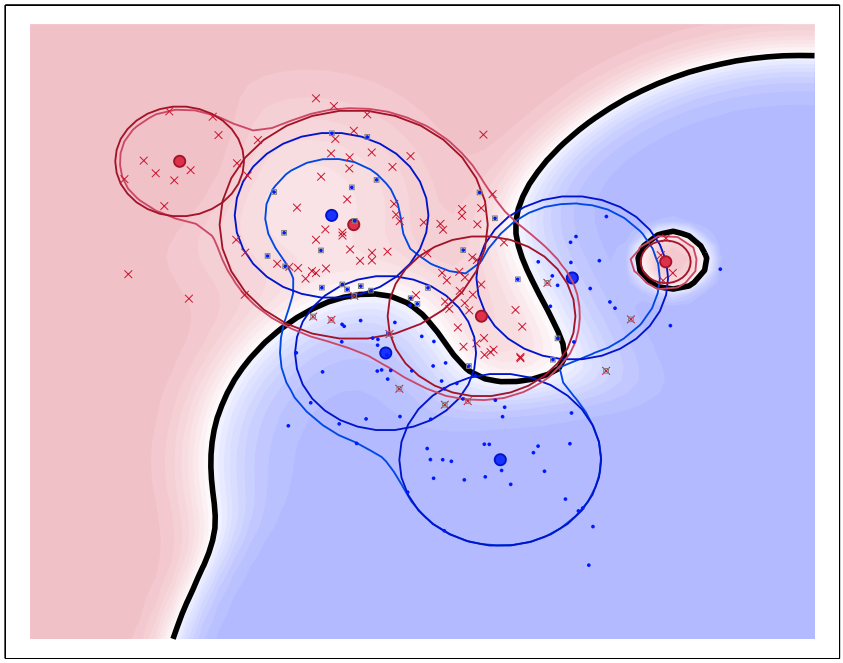


FIG. 2.1 – Frontière de décision du classifieur génératif basé sur un mélange de distributions gaussiennes sphériques. Les volumes des distributions gaussiennes ne sont pas contraints à être égaux, de manière à modéliser différents degrés de régularité avec un nombre limité de composants.

Pour simplifier, nous nous placerons dans un cadre paramétrique en supposant que Θ est de dimension finie.

Une fois que le modèle paramétrique est défini, il faut choisir un estimateur de θ pour déterminer complètement la méthode. La notion d'apprentissage se traduit ainsi par l'estimation du paramètres θ de manière à satisfaire au mieux les objectifs du modélisateur, à savoir approcher par le modèle $\boldsymbol{p}(Y|X;\theta)$ la distribution conditionnelle des données $\boldsymbol{p}(Y|X)$. Classiquement, les estimateurs sont solutions d'un problème d'optimisation, généralement sous la forme d'une minimisation d'une fonction de coût dépendant des données d'apprentissage labellisées $(\mathbf{x}, \mathbf{y})$ et du vecteur de paramètres θ. Cette fonction peut être perçue comme une mesure de distance entre la distribution empirique des données et la famille paramétrique. Toute mesure de distance entre distributions peut potentiellement être utilisée (voir par exemple [138, 34]). Dans cette thèse, la divergence de *Kullback-Leibler* sera souvent utilisée, car l'estimateur résultant correspond au maximum de vraisemblance. Il est possible de maximiser deux types de vraisemblance :

– La vraisemblance jointe des données :

$$\mathcal{L}_J(\theta) = \sum_{i=1}^{n} \log \boldsymbol{p}(x_i, y_i; \theta), \tag{2.3}$$

est souvent utilisée car c'est la vraisemblance « naturelle » du modèle paramétrique. Elle donne lieu à l'estimateur *génératif* (parfois appelé informatif [142]) :

$$\hat{\theta}_J = \operatorname*{argmax}_{\theta \in \Theta} \mathcal{L}_J(\theta), \tag{2.4}$$

– La vraisemblance conditionnelle des données :

$$\mathcal{L}_C(\theta) = \sum_{i=1}^{n} \log \boldsymbol{p}(y_i|x_i; \theta). \tag{2.5}$$

Les figures 2.1 et 2.2 illustrent ces deux types d'estimation en utilisant un modèle de mélange pour $\boldsymbol{p}(X|Y)$. La quantité $-\mathcal{L}_C(\theta)$ est aussi appelée *entropie de classification* car elle mesure la faculté d'un modèle à séparer les données. Sa maximisation est moins classique, mais plusieurs travaux récents considèrent cet estimateur comme plus efficace puisqu'il minimise un coût de classification correspondant aux objectifs du modélisateur [20, 100, 50, 61, 96, 64]. Elle donne lieu à l'estimateur *discriminatif* :

$$\hat{\theta}_C = \operatorname*{argmax}_{\theta \in \Theta} \mathcal{L}_C(\theta). \tag{2.6}$$

Le type d'apprentissage le plus répandu est évidemment l'apprentissage génératif, qui correspond à l'estimation « naturelle » des paramètres du modèle. L'apprentissage discriminatif a été introduit dans la thèse de Léon Bottou

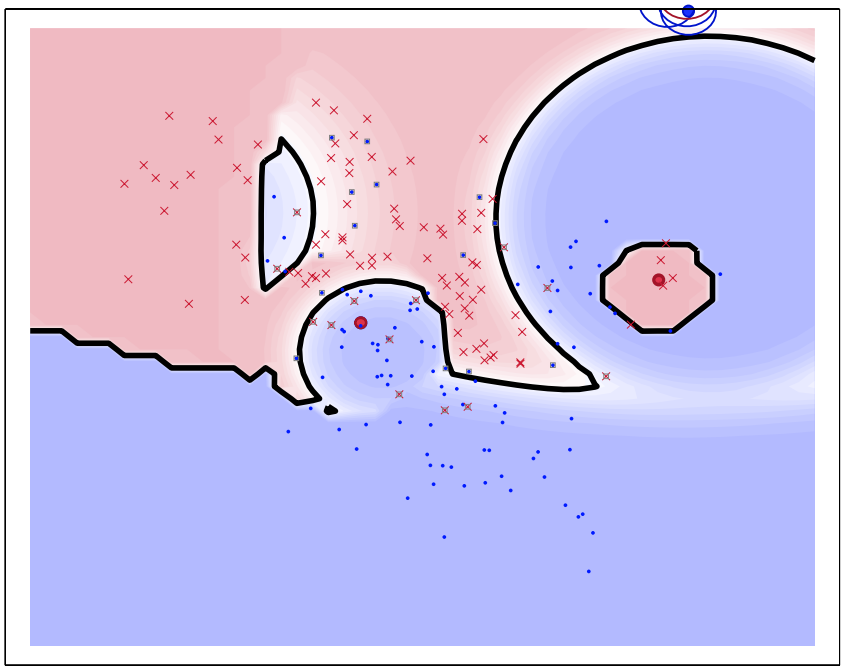


FIG. 2.2 – Estimation discriminative de la frontière de classification sur les données de la Figure 2.1.

(1991,[20]) pour l'estimation de modèles de Markov Cachés en reconnaissance de la parole. Dans ce contexte, des améliorations notables de performances de classification ont été observées. Des considérations théoriques montrent que ce type d'estimateur converge vers la règle de classification ayant une entropie de classification minimale sur les données de test [100]. Des tentatives d'estimation discriminative des paramètres pour des réseaux bayésiens quelconques montrent des gains de performances [50, 64] pour certains jeux de données à catégoriser. La supériorité de l'estimation discriminative n'est donc pas systématique et dépend à la fois du type de donnée et du modèle. Par exemple, une autre étude portant sur l'estimation discriminative des modèles de mélange de lois gaussiennes conclut sur l'inefficacité de l'estimation discriminative [61]. En effet, l'estimation discriminative soulève certains problèmes :

– Si les données sont séparables (et c'est souvent le cas en grande dimension, surtout avec des frontières de décisions non-linéaires), alors le maximum de vraisemblance n'est pas défini. Dans ce cas, des procédures de *régularisation* sont nécessaires. Par exemple, on maximise une vraisemblance *pénalisée* de manière à obtenir de meilleures propriétés pour le maximum. La pénalité a souvent une interprétation bayésienne mais les *a priori* ne sont pas toujours évidents à définir. Bien sûr, l'estimation générative nécessite elle aussi d'être

régularisée lorsque qu'il y a peu de données d'apprentissage, mais l'effet de régularisation doit généralement être plus modéré.

– Les paramètres du modèle sous-jacent perdent leur sens informatif, un exemple étant le modèle gaussien multivarié : les moyennes des classes sont des paramètres qui correspondent à des positions dans l'espace $\mathcal{X}$ des covariables et permettent de minimiser l'erreur de classification, mais il arrive avec l'estimation discriminative que des moyennes ne se situent plus au centre des données, mais hors de leur enveloppe convexe. Un paramètre de moyenne situé « à côté » peut apparaître déroutant pour le modélisateur ! On comprend à quel point les paramètres génératifs issus de l'estimation discriminative ne sont pas interprétables… Ceci est illustré dans le cas des mélanges de gaussiennes sur la figure 2.2 : les moyennes des classes ne sont pas centrées sur des données. Avec l'estimation générative, il est facile de montrer que cela ne peut pas arriver, par exemple en montrant que les moyennes des clusters se situent obligatoirement dans l'enveloppe convexe des points (l'étape M de l'agorithme EM correspond à une moyenne pondérée, *i.e.* une combinaison convexe des points).

– L'estimation discriminative se fait généralement pas descente de gradient et pose des difficultés d'estimation lorsque le nombre de paramètres est grand. La version de l'algorithme EM présentée dans [85] est trop complexe pour se généraliser à des modèles graphiques complexes.

2.1.4 Lien entre l'estimation générative et discriminative

Remarquons que la fonction à maximiser lors de l'estimation discriminative se décompose de manière intéressante :

$$L_{disc}(\theta; \boldsymbol{x}, \boldsymbol{y}) = \sum_{k=1}^{K} \sum_{\{i; y_i = k\}} \log \frac{\pi_k f_k(x_i; \theta_k)}{\sum_{l=1}^{K} \pi_l f_l(x_i; \theta_l)} \tag{2.7}$$

$$= \underbrace{\sum_{k=1}^{K} \sum_{i; y_i = k} \log \pi_k f_k(x_i; \theta_k)}_{L_J(\theta, \boldsymbol{x}, \boldsymbol{y})} \underbrace{- \sum_{i=1}^{n} \log \sum_{k=1}^{K} \pi_k f_k(x_i; \theta_k)}_{L_x(\theta, \boldsymbol{x})} \tag{2.8}$$

On voit que cette fonction est la différence entre :

$L_J(\theta; \boldsymbol{x}, \boldsymbol{y})$: la log-vraisemblance de la distribution jointe des (x_i, y_i), c'est-à-dire la fonction objectif de l'approche générative,

$L_x(\theta, \boldsymbol{x})$: la log-vraisemblance marginale des x_i, i.e. des données non classées, c'est donc la log-vraisemblance d'un *mélange de K distributions de densité f_k, $k = 1, \cdots, K$*,

En fait, cela revient à prendre le logarithme de l'expression $p(\boldsymbol{y}|\boldsymbol{x}) = \frac{p(\boldsymbol{x},\boldsymbol{y})}{p(\boldsymbol{x})}$. La minimisation du coût est équivalent à maximiser la vraisemblance L *pénalisée* par L_x. Ainsi, la différence entre l'approche générative et l'approche discriminative est précisément la présence ou l'absence de ce terme de pénalité L_x appliqué à la fonction de coût.

On peut cependant difficilement parler de *vraisemblance pénalisée* car le terme de pénalisation L_x est dépendant d'une partie des données, à savoir les x_i. A titre informatif, l'interprétation bayésienne (à $\boldsymbol{x}$ fixé) de ce terme est la pénalisation des modèles qui donnent des probabilités élevées aux x_i. Cela peut paraître à première vue contre-intuitif, mais nous verrons que la solution au problème de maximisation de la vraisemblance conditionnelle permet de mieux appréhender la différence entre les deux approches.

En dérivant l'expression (2.8) par rapport à θ, on obtient pour $k = 1, \cdots, K$:

$$\begin{cases} \frac{\partial L_C}{\partial \theta_k} = \sum_{\{i;y_i=k\}} \frac{\partial \log f_k(x_i;\theta_k)}{\partial \theta_k} - \sum_{i=1}^n \frac{\pi_k f_k(x_i;\theta_k)}{\sum_{l=1}^K \pi_l f_l(x_i;\theta_l)} \frac{\partial \log f_k(x_i;\theta_k)}{\partial \theta_k}, \\ \frac{\partial L_C}{\partial \pi_k} = \frac{n_k}{\pi_k} - \sum_{i=1}^n \frac{1}{\pi_k} \frac{\pi_k f_k(x_i;\theta_k)}{\sum_{l=1}^K \pi_l f_l(x_i;\theta_l)}. \end{cases} \tag{2.9}$$

ce qui se simplifie :

$$\begin{cases} \frac{\partial L_C}{\partial \theta_k} = \sum_{i=1}^n (\mathbf{1}_{\{y_i=k\}} - \tau_{ki}) \frac{\partial \log f_k(x_i;\theta_k)}{\partial \theta_k} \\ \frac{\partial L_C}{\partial \pi_k} = \frac{1}{\pi_k} \left(n_k - \sum_{i=1}^n \tau_{ki}\right) \end{cases} \tag{2.10}$$

avec

$$\tau_{ki} = \frac{\pi_k f_k(x_i;\theta_k)}{\sum_{l=1}^K \pi_l f_l(x_i;\theta_l)}. \tag{2.11}$$

En supposant (dans un premier temps) que L_C n'a qu'un extremum, nous cherchons la valeur de θ qui annule les équations (2.10). On remarque que les équations

$$\sum_{i=1}^n (\tau_{ki} - \mathbf{1}_{\{y_i=k\}}) \frac{\partial \log f_k(x_i;\theta_k)}{\partial \theta_k} = 0 \tag{2.12}$$

sont des équations score classiques et peuvent être résolues par un algorithme de descente de gradient, tel que Gauss-Newton ou Newton-Raphson, en prenant garde de rester dans l'ensemble Θ des paramètres [4]. Les données

[4] Il y a souvent des paramètres strictements positifs (variance par exemple) ou des paramètres qui somment à 1 (comme les proportions). Dans ces cas, une simple reparamétrisation permet de s'affranchir de ces contraintes.

$\boldsymbol{x}$ influent sur l'estimation de θ_k proportionnellement aux poids $w_{ki} = \mathbf{1}_{\{y_i=k\}} - \tau_{ki}$. En examinant les poids w_{ik} on remarque que, pour un groupe k donné, les données appartenant à ce groupe ont des poids positifs (d'autant plus que sa probabilité τ_{ik} d'être bien classée est faible), celles qui n'appartiennent pas au groupe ont des poids négatifs (d'autant plus négatifs que ces données ont une forte chance d'être mal classées dans le groupe k). On voit donc que les données bien classées ont un poids plus faible que les données difficiles à classer.

2.2 Aspects théoriques des estimateurs génératifs et discriminatifs

L'étude du comportement asymptotique des estimateurs permet de comprendre les caractéristiques des deux méthodes d'estimation.

2.2.1 Hypothèse du « vrai modèle » : optimalité de l'estimateur génératif

Un argument important en faveur de l'estimateur génératif est qu'il est asymptotiquement de variance minimum. Nous faisons ici l'hypothèse que la distribution des données appartient à la famille de probabilités $\boldsymbol{p}(X, Y; \theta)$ paramétrée par $\theta \in \Theta$, On notera θ_0 le paramètre du « vrai modèle » tel que

$$\boldsymbol{p}(X, Y) = \boldsymbol{p}(X, Y; \theta_0) \quad \forall X, Y \in \mathcal{X} \otimes \mathcal{Y}.$$

Sous cette hypothèse, l'estimateur du maximum de vraisemblance de la distribution jointe est asymptotiquement sans biais, *i.e.* $\hat{\theta}_J \to \theta_0$ lorsque $n \to \infty$. Ainsi, en supposant que la fonction de vraisemblance est suffisament régulière dans un voisinage de θ_0, la variance de l'estimateur génératif $\hat{\theta}_J$ atteint asymptotiquement la borne de Cramer-Rao :

$$nVar\left[\hat{\theta}_J\right] \underset{n\to\infty}{\to} J_J^{-1} \tag{2.13}$$

où $-J_J$ est la matrice hessienne[5] de $E\left[\log \boldsymbol{p}(X, Y)\right]$ au point θ_0. Ainsi, il n'est pas possible de construire un estimateur sans biais qui ait une variance plus petite que l'estimateur génératif. Pour comparer cet estimateur à l'estimateur discriminatif, il est possible de comparer leur variance asymptotique. La même démarche nous permet

[5]Un résultat classique de la théorie asymptotique est que la matrice $-J_J$ est égale à la matrice d'information de Fisher $E\left[\frac{\partial}{\partial\theta}\log \boldsymbol{p}(X, Y; \theta_0)\frac{\partial}{\partial\theta}\log \boldsymbol{p}(X, Y; \theta_0)^T\right]$ lorsque le modèle est exact sur les données.

d'établir, lorsque l'estimateur discriminatif est unique, la distribution asymptotique de ce dernier :

$$nVar\left[\hat{\theta}_C\right] \underset{n\to\infty}{\to} J_C^{-1} \tag{2.14}$$

avec $-J_C$ la matrice hessienne de $E\left[\log \boldsymbol{p}(Y|X;\theta)\right]$ au point θ_0. La relation $J_J \geq J_C$ peu aisément être établie. En dérivant deux fois l'équation

$$\log \boldsymbol{p}(X,Y;\theta) = \log \boldsymbol{p}(X;\theta) + \log \boldsymbol{p}(Y|X;\theta),$$

nous obtenons $J_J = J_C + J_M$. On en déduit que :

$$J_J \geq J_C. \tag{2.15}$$

Ainsi, à biais égal, l'estimateur génératif a une variance asymptotique inférieure ou égale à l'estimateur discriminatif, il est donc asymptotiquement plus efficace pour estimer θ_0. Cependant, ce résultat ne suffit pas à conclure sur la supériorité de l'estimateur génératif car l'objectif n'est pas d'estimer θ_0, mais d'approcher au mieux la densité conditionnelle $\boldsymbol{p}(Y|X;\theta_0)$. Pour mesurer la qualité d'approximation, la notion de *risque* doit être prise en compte en fonction de l'objectif décisionnel.

A chaque valeur de du paramètre θ, on peut associer un risque, caractérisant l'objectif du modélisateur. Nous considérerons généralement l'entropie de classification espérée :

$$L_C(\theta) = -E\left[\log \boldsymbol{p}(X,Y;\theta)\right].$$

En classification, on peut s'intéresser au taux d'erreur espéré :

$$L_{01}(\theta) = E\left[\mathbb{I}\{Y = \underset{k\in\{1,\cdots,K\}}{\operatorname{argmax}} \boldsymbol{p}(X,Y=k;\theta)\}\right].$$

Pour ces deux types de risque, l'hypothèse que la distribution des données appartient à la classe paramétrique fait que l'estimateur génératif est sans biais pour l'estimation du paramètre θ. Ainsi, à biais égal, la méthode de prédilection est intuitivement celle qui a une variance minimale. La proposition suivante permet, sous l'hypothèse que la famille paramétrique est exacte sur les données, de déduire la supériorité de l'estimation générative, pour plusieurs coûts de classification L possibles :

Proposition 1 *Si les trois conditions suivantes sont vérifiées,*

– *$L(\theta)$ et $\boldsymbol{p}(X, Y; \theta)$ sont des fonctions dérivables au second ordre en θ,*

– *la distribution des données $\boldsymbol{p}(X, Y)$ appartient à la famille paramétrique $\boldsymbol{p}(X, Y; \theta)$, qui admet un estimateur génératif consistant,* i.e. *il existe un unique θ_0 tel que*

$$\theta_0 = \underset{\theta \in \Theta}{\operatorname{argmax}} \log \boldsymbol{p}(\mathbf{x}, \mathbf{y}; \theta),$$

– *les matrices $J_M = -\frac{\partial^2}{\partial\theta\partial\theta^T}\boldsymbol{p}(X; \theta_0)$ et $\frac{\partial^2}{\partial\theta\partial\theta^T}L(\theta_0)$ sont définies positives,*

alors le classifieur basé sur l'estimation générative donne asymptotiquement un risque plus faible que le classifieur utilisant l'estimation discriminative, i.e.

$$E[L(\hat{\theta}_J)] - E[L(\hat{\theta}_C)] \quad \rightarrow \quad \alpha < 0$$

lorsque la taille d'échantillon n tend vers l'infini.

Cette proposition s'applique aux risques L_{01} et L_C lorsque l'hypothèse de « définie-positivité » de $\frac{\partial^2}{\partial\theta\partial\theta^T}L(\theta_0)$ est vérifiée, ce qui est les cas lorsque la solution discriminative est unique.

PREUVE. Le théorème central-limite nous donne la distribution du maximum de vraisemblance de la loi jointe et de la loi conditionnelle :

$$\sqrt{n}(\hat{\theta}_J - \theta_0) \quad \xrightarrow{\mathcal{D}} \quad \mathcal{N}\left(0, J_J^{-1}\right)$$
$$\sqrt{n}(\hat{\theta}_C - \theta_0) \quad \xrightarrow{\mathcal{D}} \quad \mathcal{N}\left(0, J_C^{-1}\right).$$

Les distributions asymptotiques de $L(\hat{\theta}_J)$ et $L(\hat{\theta}_C)$ sont obtenues grâce à un résultat classique de la théorie asymptotique du maximum de vraisemblance (voir par exemple [107, 76]) :

Lemme 1 *Soit un estimateur $\hat{\tau}$ de θ_0 tel que*

$$\sqrt{n}(\hat{\tau} - \theta_0) \quad \xrightarrow{\mathcal{D}} \quad \mathcal{N}\left(0, \Sigma\right)$$

et une fonction deux fois dérivable L telle que θ_0 minimise $L(\theta)$ sur Θ, alors

$$n\left(L(\hat{\tau}) - L(\theta_0)\right) \quad \xrightarrow{\mathcal{D}} \quad z^T B z$$

où $z \sim \mathcal{N}(0, \Sigma)$ et $B = \frac{1}{2}\frac{\partial^2}{\partial\theta\partial\theta^T}L(\theta_0)$

Nous appliquons ce résultat à l'estimateur génératif ($\hat{\tau} = \hat{\theta}$) et discriminatif ($\hat{\tau} = \hat{\theta}_C$). En définissant les variables $z_J \sim \mathcal{N}(0, J_J^{-1})$ et $z_C \sim \mathcal{N}(0, J_C^{-1})$, la loi des grands nombres nous permet d'écrire

$$\begin{aligned} n\left(E[L(\hat{\theta}_J)] - E[L(\hat{\theta}_C)]\right) \quad &\rightarrow \quad E\left[z_J^T B z_J\right] - E\left[z_C^T B z_C\right] \\ &= \quad \mathrm{tr}\left(BE\left[z_J^T z_J\right]\right) - \mathrm{tr}\left(BE\left[z_C^T z_C\right]\right) \\ &= \quad \mathrm{tr}\left(BJ_J^{-1}\right) - \mathrm{tr}\left(BJ_C^{-1}\right). \end{aligned}$$

Comme J_M est définie positive et que $J_J - J_C = J_M$, la matrice $J_J^{-1} - J_C^{-1}$ est définie négative. De plus, $B = \frac{1}{2}\frac{\partial^2}{\partial\theta\partial\theta^T}L(\theta_0)$ est défini positif par hypothèse, ce qui permet d'avoir une inégalité stricte dans l'équation $\mathrm{tr}\left(B(J_J - J_C)\right) < 0$. Ainsi

$$n\left(E[L(\hat{\theta}_J)] - E[L(\hat{\theta}_C)]\right) \quad \rightarrow \quad \mathrm{tr}\left(B(J_J^{-1} - J_C^{-1}\right) < 0,$$

ce qu'il fallait démontrer. □

Les hypothèses sur L de la proposition 1 sont en général vérifiées pour le taux d'erreur $L = L_{01}$ et l'entropie de classification $L = L_C$, ce qui permet d'appliquer les résultats de manière équivalente à ces deux types de risque.

2.2.2 Modèles biaisés : optimalité de l'estimateur discriminatif

Le défaut principal de l'estimateur génératif vient du fait qu'un modèle, par essence, ne correspond pas exactement à la réalité. Ainsi, à partir d'une certaine taille d'échantillon d'apprentissage, le biais intrinsèque au modèle devient supérieur à la variance des estimateurs, et il est préférable de choisir un estimateur dont le biais est minimal, c'est-à-dire l'estimateur discriminatif.

La solution optimale au sens du risque L est :

$$\theta_0 = \underset{\theta\in\Theta}{\mathrm{argmin}}\, E\left[L(\theta)\right]$$

Formellement, un modèle biaisé se traduit par le fait que le vecteur score de la vraisemblance générative $\log \boldsymbol{p}(\mathbf{x}, \mathbf{y}; \theta_0)$ n'a pas une espérance égale à zéro. En utilisant l'estimateur du maximum de vraisemblance conditionnelle pour la solution discriminative, nous ne pouvons établir un résultat général que pour le risque L_C (*i.e.* l'entropie de classification).

Proposition 2 *Si les trois conditions suivantes sont vérifiées :*

– $E\left[\frac{\partial}{\partial\theta}\log \boldsymbol{p}(X,Y;\theta_0)\right] \neq 0$,

– $\theta_0 = \operatorname{argmin}_{\theta\in\Theta} E\left[L(\theta)\right]$ *a une solution unique,*

– $\boldsymbol{p}(X,Y;\theta)$ *est bornée et dérivable dans un voisinage de* θ_0,

alors asymptotiquement la solution discriminative donne un risque L_C *plus faible que la solution générative sur des données de test :*

$$E\left[L_C(\hat{\theta}_C)\right] < E\left[L_C(\hat{\theta}_J)\right]$$

PREUVE. A taille d'échantillon n fixé,

$$\hat{\theta}_C = \operatorname*{argmax}_{\theta} \log \boldsymbol{p}(\mathbf{y}|\mathbf{x};\theta) = \operatorname*{argmin}_{\theta} \sum_{i=1}^{n} L_C(\theta) \quad \underset{n\to\infty}{\overset{a.s}{\to}} \quad \theta_0,$$

et donc $E\left[L_C(\hat{\theta}_C)\right] = L_C(\theta_0)$. Soit $\theta_1 = \lim_{n\to\infty}\hat{\theta}_J$. L'estimateur génératif vérifie $\frac{\partial}{\partial\theta}\log \boldsymbol{p}(\mathbf{x},\mathbf{y};\hat{\theta}_J) = 0$, pour toute taille d'échantillon n. Par convergence dominée on en déduit que $\frac{\partial}{\partial\theta}\log \boldsymbol{p}(X,Y;\theta_1) = 0$, Or θ_1 ne peut être égal à θ_0 sans contredire la première hypothèse (modèle biasé). Puisque θ_0 est l'unique maximum de $L_C(\theta)$, $E\left[L_C(\hat{\theta}_J)\right] = L_C(\theta_1) < L_C(\theta_0) = E\left[L_C(\hat{\theta}_C)\right]$, ce qu'il fallait démontrer. □

2.2.3 Comportement non asymptotique

L'hypothèse de l'exactitude du modèle sur les données est évidemment irréaliste, mais celui-ci peut parfois être un très bon approximateur. A taille d'échantillon fixée, savoir quel estimateur est le plus adéquat est un problème difficile. Ng et Jordan [127] ont prouvé que dans un cas d'un modèle simple (le classifieur de Bayes naïf basé sur des gaussiennes de même variance), l'estimateur génératif est plus adéquat pour de petits échantillons d'apprentissage, alors que l'estimateur discriminatif a un biais de classification plus faible et donne donc de meilleurs résultats sur de grands échantillons. Ce phénomène se vérifie empiriquement sur les autres modèles génératifs. Intuitivement, tous les paramètres contribuent à classer les données, et le phénomène de sur-apprentissage apparaît pour un échantillon de taille plus faible.

Choisir entre l'une ou l'autre des méthodes d'estimation reste cependant une question ouverte, car l'adéquation des données au modèle est toujours difficile à évaluer. En pratique, la validation croisée semble le moyen le plus simple pour choisir entre les deux estimateurs.

2.2.4 Aspects algorithmiques

La possibilité de pouvoir estimer les paramètres de manière fiable et efficace est souvent déterminante dans le choix d'une méthode de classification. L'estimation générative des modèles génératifs est étroitement liée aux méthodes non-supervisées d'estimation de densité, puisqu'on cherche à estimer la distribution $\boldsymbol{p}(X, Y; \theta)$ la plus « proche » des données d'apprentissage sans tenir compte de l'objectif du modélisateur. Ainsi, les algorithmes classiques de maximisation de la vraisemblance peuvent être utilisés tels quels pour obtenir une méthode de classification supervisée. Le maximum de vraisemblance peut parfois être calculé de manière explicite. C'est le cas lorsque les densités des classes sont gaussiennes[6], ou discrètes et qu'il n'y a pas de donnée manquante. En présence de données manquantes ou inobservées, l'algorithme EM est un outil puissant pour maximiser la vraisemblance,

Un avantage important de l'estimation générative est que les paramètres des classes peuvent parfois être estimés de manière indépendante, ce qui peut notablement simplifier le problème de maximisation. Ceci est particulièrement avantageux lorsque le nombre de classes est relativement grand. En effet, les densités f_k des classes ont généralement des paramètres différents. On peut donc exprimer les paramètres θ du modèle sous la forme d'un ensemble de paramètres fonctionnellement indépendants $\theta = (\theta_1, \cdots, \theta_K)$. La log-vraisemblance jointe $\mathcal{L}_J$ se décompose alors en somme de vraisemblances partielles :

$$
\begin{aligned}
L_J(\theta) &= \sum_{i=1}^{n} \log p(Y = y_i, X = x_i) && (2.16)\\
&= \sum_{i=1}^{n} \log \pi_{y_i} f_{y_i}(x_i; \theta_k) && (2.17)\\
&= \sum_{k=1}^{K} \sum_{\{i; y_i = k\}} \log f_k(x_i; \theta_k) + \sum_{k=1}^{K} n_k \log \pi_k && (2.18)\\
&= \mathcal{L}_{J1}(\theta_1) + \cdots + \mathcal{L}_{JK}(\theta_K) + \sum_{k=1}^{K} n_k \log \pi_k && (2.19)
\end{aligned}
$$

où n_k est le nombre d'observations appartenant à la classe k. La maximisation de $\mathcal{L}_J$ se réduit donc à l'estimation du maximum de vraisemblance pour chacune des classes indépendamment.

[6] La solution n'est pas toujours explicite si on fixe des contraintes particulières sur les paramètres. Par exemple, lorsque la densité des classes est une gaussienne multivariée, une contrainte d'égalité sur la « forme » des gaussiennes avec des volumes libres necessite un algorithme itératif pour maximiser la vraisemblance [15].

La régression logistique classique choisit θ qui maximise $L_c(\theta; \boldsymbol{x}, \boldsymbol{y})$. On interprète souvent cela comme la minimisation d'un coût $C(\theta; \boldsymbol{x}, \boldsymbol{y})$ égal à $-L_c(\theta; \boldsymbol{x}, \boldsymbol{y})$ (coût logistique). On prend parfois le coût 0-1 qui minimise le taux d'erreur empirique, mais le coût logistique est une bonne approximation de cette erreur d'apprentissage et a l'avantage d'être plus facile à minimiser car il est dérivable.

On peut noter l'existence d'une version de l'algorithme EM appelée *Conditional EM* (CEM) qui maximise la vraisemblance conditionnelle [85, 87]. Bien que l'idée soit judicieuse et la mise en œuvre originale — inversion de l'inégalité de Jensen pour borner la densité $\boldsymbol{p}(x)$ dans l'expression $\log \boldsymbol{p}(y|x) = \log \boldsymbol{p}(x, y) - \log \boldsymbol{p}(x)$ — cette méthode reste difficile à utiliser et se limite dans les études existantes à des mélanges de gaussiennes et de modèles de Markov Cachés de faible dimension. Les résultats proposés ne suffisent pas pour conclure sur l'intérêt de l'algorithme. Notre point de vue est plutôt d'utiliser un algorithme de descente de gradient pour effectuer la maximisation.

Lorsqu'un modèle probabiliste est bien adapté aux données, la différence entre l'estimation générative et discriminative des paramètres est relativement faible. Ainsi, l'estimation générative est souvent préférée car l'algorithme EM [35] peut être utilisé pour maximiser la vraisemblance, ce qui simplifie considérablement l'estimation des paramètres.

2.3 Exemples de classifieurs génératifs

Puisqu'il y a autant de classifieurs génératifs qu'il y a de modèles de probabilités possibles, nous avons choisi de détailler dans cette section deux exemples de classifieurs génératifs. Le premier est un classifieur linéaire : l'analyse discriminante de Fisher, le second est basé sur des mélanges de gaussiennes mutivariées. Ces classifieurs permettent d'obtenir des résultats de classification satisfaisants pour une grande variété de distributions rencontrées dans des problèmes réels.

2.3.1 L'analyse discriminante linéaire et la régression logistique linéaire

Introduite en 1936 par Fisher, l'Analyse Linéaire Discriminante (LDA) est probablement la méthodes générative la plus ancienne et la plus populaire [47]. De nombreuses extensions non linéaires de LDA ont été proposées.

On peut citer l'Analyse Discriminante Quadratique [23], l'analyse discriminante flexible (FDA) et des méthodes à noyau basées sur LDA (Kernel Discriminant Analysis [117]).

La régression logistique linéaire (LLR) est aussi une méthode standard en classification supervisée, suite au développement des *modèles linéaire généralisés* [113, 63, 7]. De même, de nombreuses extensions non-linéaires ont été proposées, incluant la régression logistique quadratique, les réseaux de neurones (see e.g. [137]), les modèles additifs généralisés (GAM) [69] et enfin la régression logistique généralisée[7] [167].

Dans de nombreuses applications, LDA et LLR donnent des taux d'erreur satisfaisants. Elle sont particulièrement compétitives pour des problèmes où les données sont en grande dimension. Comme ces deux méthodes fournissent une règle de classification linéaire les taux de classification sont souvent très proches, mais elles sont de nature totalement différente :

– La régression logistique linéaire est la méthode de type discriminatif de référence. Elle modélise directement la distribution conditionnelle de la variable Y sachant le prédicteur $X \in \mathbb{R}^d$ par une loi binomiale dont le paramètre dépend de manière logistique des covariables :

$$\boldsymbol{p}(Y = 1|X = x; \beta) = \frac{1}{1 + e^{-\beta'_{1:d}x - \beta_0}}, \tag{2.20}$$

où $\beta = (\beta_0, \theta_{1:d})$ est le vecteur de paramètres à estimer.

– LDA modélise la densité de chaque groupe par une distribution gaussienne multivariée, avec une matrice de covariance commune. On en déduit une règle de classification par la formule de Bayes (2.2). C'est donc une méthode générative pour laquelle la densité des classes s'écrit :

$$f_k(x; \theta) = \frac{1}{\sqrt{2\pi}^d |\Sigma|^{\frac{1}{2}}} \exp\left\{\frac{1}{2}(x - m_k)^T \Sigma^{-1}(x - m_k)\right\},$$

où $\theta = (m_1, \cdots, m_K, \text{vec}(\Sigma^{-1}))$ est le vecteur de paramètres à estimer.

La frontière de classification $\boldsymbol{p}(Y|X = x; \theta)$ résultante a la forme logistique (2.20) en définissant une fonction $g : \mathbb{R}^{d+1} \mapsto \Theta$ égale à :

$$\begin{cases} \beta_{1:d} &= \Sigma^{-1}(m_1 - m_2) \\ \beta_0 &= \frac{m_1 + m_2}{2}\Sigma^{-1}(m_1 - m_2) \end{cases} \tag{2.21}$$

[7] On oublie souvent de rappeler que cette version de la régression logistique à été à l'origine des travaux de Vapnick sur les Support Vector Machine. Il précise [158] que ces deux méthodes sont très prochent par nature et donnent des performances très similaires [158], hormis l'aspect « parcimonieux » de la solution de discrimination $\delta(x) = \sum_{i=1}^n \alpha_i y_i K(x_i, x)$ qui possède de nombreux coefficients α_i nuls.

dans le cas de $K = 2$ classes de proportions égales[8]. On construit la fonction g de manière à avoir :

$$\beta = g(\theta) \quad \Rightarrow \quad \forall (X, Y) \in \mathcal{X} \otimes \mathcal{Y} \quad \log \boldsymbol{p}(Y|X;\theta) = \log \boldsymbol{p}(Y|X;\beta). \tag{2.22}$$

Soit $\hat{\theta}_C$ un estimateur discriminatif (équation (2.6)) de θ, c'est-à-dire vérifiant :

$$\mathcal{L}_C(\hat{\theta}_C) = \max_{\Theta} \mathcal{L}_C(\theta), \tag{2.23}$$

où nous avons défini $\mathcal{L}_C(\theta)$ en (2.5). Soit $\tilde{\beta}$ une solution de la régression logistique linéaire :

$$\mathcal{L}_L(\tilde{\beta}) = \max_{\mathbb{R}^{d+1}} \mathcal{L}_L(\beta), \tag{2.24}$$

où $\mathcal{L}_L(\beta) := \boldsymbol{p}(\mathbf{y}|\mathbf{x};\beta)$. Le théorème suivant montre l'équivalence entre l'estimation discriminative de LDA et la régression logistique linéaire.

Theorème 1 *L'estimation discriminative du modèle gaussien homoscédastique et la régression logistique linéraire donnent la même distribution conditionnelle de Y sachant X,* i.e. *:*

$$\forall (X, Y) \in \mathcal{X} \otimes \mathcal{Y} \quad \boldsymbol{p}(Y|X;\tilde{\beta}) = \boldsymbol{p}(Y|X;\hat{\theta}).$$

De plus, si la solution $\tilde{\beta}$ de la régression logistique est unique, $g(\hat{\theta}) = \tilde{\beta}$.

PREUVE. Pour obtenir l'égalité, nous allons démontrer les deux inégalités suivantes :

$$\forall \tilde{\beta} \qquad \mathcal{L}_L(\tilde{\beta}) \geq \mathcal{L}_C(\hat{\theta}_C) \quad \text{et}$$

$$\mathcal{L}_L(\tilde{\beta}) \leq \mathcal{L}_C(\hat{\theta}_C).$$

Posons $\hat{\beta} = g(\hat{\theta})$. D'après (2.22), on a $\mathcal{L}_L(\hat{\beta}) = \mathcal{L}_C(\hat{\theta}_C)$. Or la solution de (2.24) implique $\mathcal{L}_L(\tilde{\beta}) \geq \mathcal{L}_L(\hat{\beta})$. On en déduit que $\mathcal{L}_L(\tilde{\beta}) \geq \mathcal{L}_C(\hat{\theta}_C)$.

Inversement, pour tout $\tilde{\beta} \in \mathbb{R}^{d+1}$, l'ensemble $\mathcal{S} = \{\theta; g(\theta) = \tilde{\beta}\}$ n'est pas vide. Par exemple, on peut toujours trouver un élément $\tilde{\theta} = (\tilde{m}_1, \tilde{m}_2, \text{vec}(\tilde{\Sigma}^{-1}))$ appartenant à $\mathcal{S}$ de la manière suivante :

$$\begin{cases} \tilde{m}_1 &= \frac{\tilde{\beta}_0}{\|\tilde{\beta}_{1:d}\|}\tilde{\beta}_{1:d} + \frac{1}{2}\tilde{\Sigma}\tilde{\beta}_{1:d} \\ \tilde{m}_2 &= \frac{\tilde{\beta}_0}{\|\tilde{\beta}_{1:d}\|}\tilde{\beta}_{1:d} - \frac{1}{2}\tilde{\Sigma}\tilde{\beta}_{1:d} \\ \tilde{\Sigma} &= I_d \end{cases} . \tag{2.25}$$

[8] Pour ne pas alourdir les notations, nous considérons uniquement le cas de deux classes. Le cas $K > 2$ ne posant pas de difficulté théorique majeure.

Si $\|\tilde{\beta}_{1:d}\| = 0$, on peut prendre $\tilde{m}_1 = \tilde{m}_2 = (0, \cdots, 0)^T$. Pout tout $\tilde{\beta}$, nous avons donc prouvé l'existence de $\tilde{\theta}$ vérifiant $\mathcal{L}_C(\tilde{\theta}) = \mathcal{L}_L(\tilde{\beta})$. De plus, d'après (2.23), $\mathcal{L}_C(\hat{\theta}_C) \geq \mathcal{L}_C(\tilde{\theta})$, on a donc $\mathcal{L}_L(\tilde{\beta}) \leq \mathcal{L}_C(\hat{\theta}_C)$.

Ansi, on a nécessairement $\mathcal{L}_L(\tilde{\beta}) = \mathcal{L}_C(\hat{\theta}_C) = \mathcal{L}_L(\hat{\beta})$ et si la solution de la régression logistique est supposée unique, on a nécessairement $\tilde{\beta} = \hat{\beta} = g(\hat{\theta})$. □

Notons que le résultat précédent est immédiat lorsqu'on utilise la paramétrisation logistique. Nous avons ici justifié le fait de conserver les paramètres génératifs pour effectuer la régression logistique linéaire. Le fait de n'avoir pas de maximum unique peut parfois poser des problèmes de maximisation, mais nous verrons dans le chapitre 5 un moyen de définir de manière unique l'estimateur discriminatif de LDA.

Un classifieur génératif très efficace lorsque la dimension des données est élévée est le classifieur de Bayes naïf (NB) qui suppose l'indépendance entre toutes les entrées [40] conditionnellement à la classe. Ce type de classifieur est obtenu naturellement en contraignant la matrice de variance commune à être diagonale. Ainsi, conditionnellement à la classe $Y = k$, toutes les covariables sont indépendantes et suivent des lois gaussiennes univariées. Tous les résultats précédents restent applicables, on peut ainsi conserver la paramétrisation de NB pour effectuer la régression logistique. En réalité, même si les hypothèses du modèle sont irréalistes, ce type de classifieur est efficace dans de nombreuses applications [36].

2.3.2 Mélanges de distributions gaussiennes pour la discrimination

LDA est une méthode de référence en classification supervisée. Dans les cas où ses performances sont médiocres, il est nécessaire de disposer d'une méthode complémentaire permettant des frontières de décision non linéaires. Les atouts principaux d'une telle méthode doivent être la souplesse, la simplicité et la parcimonie. Les méthodes purement discriminatives du type SVM semblent répondre à ces besoins, mais nous souhaitons montrer qu'une méthode générative peut aussi avoir ces qualités. A travers le choix du nombre de composants dans les mélanges, la complexité de la frontière de discrimination peut varier considérablement. Elle peut être linéaire ou quadratique, lorsqu'un seul composant par groupe est sélectionné. Dans les cas plus complexes, les mélanges de distributions permettent de séparer des groupes non connexes et aux contours irréguliers.

Nous proposons ici de modéliser les distributions des classes par des mélanges de distributions gaussiennes. Cela permet de définir de manière simple une méthode de classification adaptée à des distributions très différentes.

Ce type de modèle génératif sera ensuite utilisé à plusieurs reprises dans les différents chapitres de cette thèse.

Les modèles de mélange sont des modèles très appréciés pour modéliser des distributions de forme *a priori* inconnue [115, 34]. Il a déjà été remarqué qu'utiliser pour chaque classe un mélange de distributions gaussiennes avec des matrices de covariances égales est une extension directe de l'analyse discriminante [70]. Un modèle équivalent mais permettant l'affectation partielle des composants aux classes a aussi été proposé, pour des résultats équivalents [155]. C'est d'ailleurs le modèle de prédilection en classification non supervisée utilisé en fouille de données[25]. Des comparaisons de performance en classification pour différents types de paramètrisation de la matrice de covariance (pleine, diagonale, PPCA[9], spherical) montrent que le choix de modèle n'est pas facile et dépend vraiment de l'approche considérée [121].

Nous insistons sur le caractère universel des distributions de mélanges, puisque toute distribution intégrable peut être approximée par un mélange fini de gaussiennes [34]. Mais cette qualité d'« approximateur universel » des mélanges n'est pas seulement asymptotique, et de nombreuses distributions peuvent êtres représentées par un nombre relativement limité de composants. Il est par exemple très difficile de différentier une distribution Gamma et le mélange de trois distributions gaussiennes convenablement choisies [115]. Dans un cadre discriminatif, nous pouvons aller plus loin dans la parcimonie puisque le but recherché n'est pas l'adéquation aux données mais plutôt à la frontière de discrimination. En effet, le problème de classification ne s'intéresse qu'au taux d'erreur et des groupes à la structure complexe mais très séparés peuvent, par application du principe de parcimonie [10], être modélisés par un seul composant.

Classification par boules gaussiennes

Nous considérons un problème de classification à K classes dont les données sont $\boldsymbol{x} = \{\mathbf{x}, \mathbf{y}\}$ où $\mathbf{x} = (x_1, \cdots, x_n)$ est un ensemble de vecteurs dans $\mathbb{R}^d$ et $\mathbf{y} = (y_1, \cdots, y_n)$ correspond aux labels des classes.

Lorsque le nombre de composants dans une classe donnée k vaut R_k, $k = 1, \cdots, K$, le modèle de densité de

[9] Le modèle Probabilistic Principal Component Analysis (PPCA) consiste à paramétrer la matrice de variance sous la forme $\Sigma = \partial I_d + \sum_{r=1}^{R} \lambda_r v_r v_r^T$ où R est le nombre de composantes principales (ayant vocation à être petit). Ce modèle est pariculièrement adapté aux données de grande dimension ayant de fortes corrélations [154].

[10] Le philosophe *Ockham* est souvent cité dans ce cas : « *Pluralitas non est ponenda sine neccesitate* », ce qui peut se traduire par « Les choses ne devraient pas se multiplier si ce n'est pas nécessaire ».

cette la $k^{\text{ème}}$ classe s'écrit :

$$f_k(\boldsymbol{x};\theta_k) = \sum_{r=1}^{R_k} \pi_r \phi(\boldsymbol{x};\boldsymbol{\mu}_r,\sigma_r^2 I_d) \tag{2.26}$$

où π_r, μ_r and σ_r sont respectivement le poids, la moyenne et l'écart-type du $r^{\text{ème}}$ composant et $\phi(\boldsymbol{x};\boldsymbol{\mu},\Sigma)$ désigne la densité d'une distribution gaussienne multivariée de moyenne μ et de matrice de covariance Σ. On note θ_k l'ensemble des paramètres de la classe k. Contraindre la matrice de variance à être proportionnelle à la matrice identité permet d'avoir un modèle à la fois stable (la matrice de variance n'est dégénérée que lorsque $\sigma_r \to 0$) et parcimonieux, *i.e.* avec un nombre limité de de paramètres par composant. Ainsi, un composant du mélange aura $\nu_r = d + 1$ paramètres, à comparer aux $2d$ paramètres dans le cas de covariances diagonales et $d + d(d + 1)/2$ paramètres pour des covariances libres. Grâce à cette relative simplicité des composants, leur nombre peut varier significativement entre les classes. Les modèles de mélange gardent une certaine souplesse d'ajustement aux données car les paramètres de variance σ_r^2 ne sont pas contraints à être égaux au sein d'une même classe.

L'estimateur du maximum de vraisemblance génératif des paramètres peut être obtenu en maximisant les vraisemblances partielles des classes séparément. L'algorithme EM est utilisé, puisque l'affectation des données aux composants au sein d'un classe est inconnue. Les affectations initiales sont obtenues par l'algorithme des k-means.

La Figure 2.1 donne une illustration de la frontière obtenue avec cette méthode de classification générative. Ces données simulées sont issues de Hastie et al (2001)[71]. Elles consistent en 200 points en dimension 2 séparés en deux classes équiprobables[11]. Sur la figure, les classes sont identifiées grâce à des symboles différents. La distribution estimée par l'algorithme EM est représentée par des cercles correspondant à l'isocontour contenant 80% de la masse des composants.

Sélection du nombre de composants des mélanges

Cette méthode d'Analyse Discriminante par Mélange (MDA) basée sur des matrices de variance sphériques peut donner de bons résultats en classification car elle est à la fois souple et parcimonieuse. Cependant, le choix du nombre de composants $\{R_k\}_{k=1,\cdots,K}$ des mélanges est un problème difficile. En effet, si nous considérons que nous voulons tester tous les modèles avec au plus M composants par classe, le nombre de modèles à tester s'élève à M^K, ce qui est exponentiel en fonction du nombre de classes. Le fait que les paramètres sont estimés

[11] Le jeu de données est disponible à l'adresse `http ://www-stat.stanford.edu/ElemStatLearn.`

indépendemment dans chaque classe permet de réduire le temps d'apprentissage, puisque MK estimations de mélanges par l'algorithme EM seront nécessaires. Cette simplification n'est pas toujours possible en discrimination. Par exemple, dans le cas de MDA telle qu'elle a été définie par Hastie et Tibshirani [70], le fait que les matrices de variance de tous les clusters sont égales ne permet pas de d'estimer les densités des classes de manière indépendante. Un autre problème apparaît lorsqu'il faut déterminer lequel des ces M^K modèles est le plus adapté à la discrimination. La validation croisée necessite νMK estimations de paramètres, et νM^K calculs de taux d'erreur, où ν est le nombre de divisions de l'échantillon d'apprentissage. Des critères tels que BIC [147] semblent plus adaptés, mais sont sous-optimaux dans un cadre de classification supervisée (voir chapitre 4). La table 2.1 donne une illustration du choix obtenu par validation croisée pour $R_1 \leq 7$ et $R_2 \leq 6$. On remarque dans ce cas que le modèle MDA avec des distributions sphériques est meilleur que LDA pour le modèle contenant 4 composants dans chaque classe et est capable de trouver une frontière de classification très proche de la frontière optimale de Bayes. (voir [71], p. 22).

		R_1						
		1	2	3	4	5	6	7
R_2	1	0.297	0.284	0.255	0.247	0.244	0.249	0.256
	2	0.273	0.262	0.235	0.226	0.226	0.233	0.241
	3	0.268	0.254	0.230	0.223	0.224	0.228	0.234
	4	0.256	0.244	0.225	**0.219**	0.220	0.223	0.229
	5	0.252	0.243	0.228	0.219	0.219	0.221	0.224
	6	0.250	0.243	0.229	0.221	0.221	0.221	0.223

TAB. 2.1 – Taux moyen d'erreur en test sur les données simulées estimé par half-sampling sur 500 jeux d'apprentissage/tests aléatoires. Le taux d'erreur de test pour LDA est de 0.283. Le taux d'erreur optimal pour MDA avec des distributions gaussiennes sphériques apparaît en gras.

D'un point de vue quantité de calculs, le critère BIC est attractif. En effet, on peut constater que BIC$=\sum_k \text{BIC}_k$, BIC_k étant le critère BIC calculé pour la classe k. Ainsi, comme pour l'étape d'estimation, il est possible de calculer le vecteur $R = (R_k, k = 1, \ldots, K)$ optimal en MK opérations aux lieux de M^K évaluations par

validation croisée.

Cependant, supposer que les densités des classes peuvent être réduites à un nombre fini de « boules » gaussiennes peut être perçu comme un modèle de densité très approximatif, et BIC mesure l'adéquation du mélange aux données plutôt que les performances en classification. Cela veut dire que dans de nombreuses situations, et particulièrement en grande dimension, le nombre de composants obtenus par BIC n'est pas optimal. Son comportement est en réalité assez difficile à appréhender (il a une légère tendance à sous-estimer le nombre de composants sur des simulations, et à sur-évaluer leur nombre sur données réelles), et la validation croisée reste le critère de référence pour sélectionner le vecteur R du nombre de composants.

Différents travaux sur les méthodes de type MDA utilisant des distributions gaussiennes sphériques ont été effectuées. Nous pouvons dresser les conclusions suivantes :

- Lorsque LDA donne de bonnes performances de classification, les taux de classifications de MDA sphérique sont comparables,
- Lorsque LDA est inefficace, typiquement pour des problèmes de classification non-linéaires, la méthode MDA sphérique permet d'améliorer considérablement les performances,
- En très grande dimension, MDA sphérique a des performances limitées car les données se situent généralement dans un sous-espace de dimension réduite, et l'utilisation d'une densité *remplissant* l'espace biaise l'estimation ("curse of dimensionality").

Le défi principal en vue d'améliorations futures de cette méthode est de proposer une procédure plus simple pour déterminer le nombre de composants. Il faudrait par exemple définir un critère de selection du nombre de composants qui soit à la fois fiable vis-à-vis de l'objectif de discrimination, tout en allégeant la charge de calcul liée à la validation croisée. Le chapitre 4 de cette thèse traite de ce problème.

2.4 Méthodes concurrentes

Plusieurs méthodes permettant de classer des données structurées ont été proposées. Nous en donnons un bref aperçu.

2.4.1 Réseaux bayésien à marge maximale

L'entropie de classification est parfois remise en cause. Pour apprendre les paramètres de manière discriminative, certains auteurs ne maximisent pas explicitement la vraisemblance conditionnelle, mais utilisent des classifieurs à marge maximale (Max-Margin Markov Networks [103, 152] ou Maximal Entropy Discrimination [151]) ou construisent une fonction de coût adaptée à la discrimination [26, 27]. Voir [9] pour une synthèse de ces approches.

2.4.2 Noyaux probabilistes

Les machines à noyaux offrent un outil discriminatif très performant. Plusieurs études se sont intéressées à la construction d'un noyau à partir d'un modèle probabiliste.

Le *Noyau de Fisher* [81] et ses extensions [156, 80] se basent sur deux modèles : un génératif et un discriminatif. La classification des données se fait elle aussi en deux étapes.

- un modèle génératif m_1 avec des paramètres θ dont la distribution $\boldsymbol{p}(X;\theta)$ est définie sur l'espace d'origine des données $\mathcal{X}$
- un modèle discriminatif m_2 de paramètres α basé sur une fonction de coût $C(\tilde{X}|Y;\alpha)$ est défini sur $\mathbb{R}^d$ où d est le nombre de paramètres du modèle génératif $m1$.

Les paramètres θ et α étant fixés, une donnée X est classée dans la classe $Y \in \{1, \cdots, K\}$ en deux étapes :

1. calcul du vecteur tangent au modèle : $\tilde{X} = \nabla_\theta \boldsymbol{p}(X, \theta)$,
2. classification de ce vecteur avec le modèle discriminatif :

$$Y = \underset{y \in \{1,\cdots,K\}}{\operatorname{argmin}} C(\tilde{X}|Y = y;\alpha)$$

En général, le modèle génératif est appris par maximisation de la vraisemblance. Pour le modèle discriminatif, un classifieur SVM linéaire est souvent utilisé. Cependant, le fait de baser l'apprentissage sur deux étapes séparées plutôt que de minimiser une fonction de coût globale est par nature sous-optimal : rien n'empêche la première partie de l'apprentissage (estimation générative) de choisir des directions orthogonales à la frontière de classification optimale.

Enfin, nous pouvons noter que d'autres méthodes de construction de noyaux basées sur des modèles probabilistes ont été proposées. Dans ces approches, les noyaux sont liés à l'estimation de la distance séparant deux

modèles probabilistes. Les outils de géométrie différentielle permettent de définir formellement cette distance en accord avec le modèle statistique. Le *heat kernel* [65, 104] et les noyaux basés sur la divergence de Kullback-Leibler [123] ont des justifications théoriques convaincantes mais restent difficiles à calculer. Une nouvelle approche nommée *probability product kernels* [86] est plus facile à estimer car elle se base sur l'estimation du modèle génératif avec un seul échantillon d'apprentissage. Elle inclut de plus de nombreux noyaux existants comme cas particuliers.

2.4.3 Les champs de Markov conditionnels

Plusieurs modèles ne cherchent pas à retrouver une seule variable Y, mais une série de valeurs $Y_1, \ldots, Y_R$ ayant des relations entre elles. Tout en restant conditionnel à X, il est possible de spécifier la structure d'indépendance condititionnelle des Y_i. On modélise ainsi $P(Y|X)$ par un modèle graphique. Cette approche, est souvent appelée *Conditional Random Field* (CRF) et a été étendue à différents types de modèle. Par exemple, [10] définit les CRF pour les familles de distributions exponentielles.

Ce type d'approche a été appliqué avec succès en tant qu'extension discriminative des modèles de Markov cachés [105]. Les hypothèse sous-jacentes aux CRF sont plus générales que les modèles de Markov cachés (génératifs). La méthode généralement employée pour l'apprentissage est un algorithme de descente de gradient [61] Récemment, de nouvelles approches simplifiant le problème de descente de gradient ont été proposées [105, 149].

2.5 Discussion

Nous avons mis en avant plusieurs aspect inhérents à la classification générative :

- La classification générative a des avantages incontestables pour définir des modèles cohérents sur des données à structure complexe,
- La classification basée sur des mélanges de distributions pour modéliser les classes est une alternative intéressante à l'Analyse Discriminante Linéaire,
- La classification générative est très simple dans son principe de base (modéliser la jointe pour obtenir la conditionnelle) mais elle peut avoir des performances sous-optimale lorsque le nombre de données d'ap-

prentissage est important,

– L'apprentissage discriminatif permet de réduire le biais du modèle, mais l'estimation des paramètres est parfois difficile et ne donne pas toujours les meilleures performances sur des données de test,

– La sélection de modèle est primordiale pour obtenir des taux de classification satisfaisants.

Nous tenterons dans la suite de résoudre les problèmes de performances des méthodes génératives. Dans le chapitre 4, le problème de sélection de modèle est étudié, et dans le chapitre 5, nous proposons une méthode d'estimation des paramètres qui optimisent les performances de prédiction en classification ou en régression.

Chapitre 3

Modèles à classes latentes en régression

Lorsque la variable à prédire Y est continue, la majorité des méthodes de régression estiment la fonction $f(X) = E[Y|X]$. C'est le cas de la régression linéaire, et des méthodes non linéaires telles que les réseaux de neurones et la régression de type SVM. Dans certaines applications, la distribution conditionnelle est multimodale et l'estimation de la fonction f par un modèle homogène perd son sens. Les mélanges de régressions sont adaptés à ce type de problème en modélisant la distribution conditionnelle $P(Y|X)$ sous la forme d'un mélange. La *switching regression* [132] fut probablement le premier modèle de ce type, mais il suppose que les mélanges ne dépendent pas de X, ce qui peut être irréaliste. D'une manière générale, ces modèles sont appelés *mélange d'experts* (ME) [83], car ils combinent plusieurs modèles de régression simples[12] (les experts) pour résoudre un problème non linéaire. La thèse de Steven Waterhouse [161] donne un aperçu complet de ces modèles, et montre de bons résultats sur une large gamme de problèmes de classification et de régression.

Les mélanges d'experts sont définis de manière discriminative, *i.e*. sans suppostion sur la distribution des covariables X. Dans ce chapitre, les mélanges d'experts sont étudiés sous leur forme générative, c'est à dire en modélisant explicitement la distribution jointe des entrées X et des variables de sorties Y sous la forme d'un modèle de mélange gaussien. Cette modification du modèle est appelé *mélange d'experts locaux* [120], car les composants des régressions correspondent à des valeurs localisées dans l'espace des covariables. Nous montrons

[12] Les modèles combinées sont généralement des régression linéaires en régression et des modèles logistiques linéaires en classification supervisée.

que la solution classique des ME est obtenue en maximisant la vraisemblance conditionnelle de Y sachant X (estimation discriminative). Les modèles graphiques correspondant aux approches discriminatives et génératives sont donnés sur la figure 3.1, en introduisant la variable cachée discrète Z correspondant à l'index du composant du mélange. Les deux modèles sont équivalents, puisqu'ils ne définissent aucune indépendance conditionnelle, mais le sens des flèches entre X et Z diffère et montre que les paramétrisations sont différentes.

Nous étudierons le cas où les experts sont des régressions linéaires. Nous montrerons que l'utilisation de l'estimateur génératif a plusieurs avantages :

- d'après la proposition 1 du chapitre précédent, dans le cas où la distribution des covariables est effectivement gaussienne, l'estimateur est optimal,
- l'apprentissage des paramètres du modèle est beaucoup plus rapide, car il ne nécessite pas l'estimation d'une régression logistique,
- l'estimation est plus stable, dans le sens ou la variance des estimateurs est réduite par rapport à l'estimation discriminative et est plus rarement bloquée dans des maxima locaux,
- des modèles additionnels peuvent être définis grâce à l'utilisation explicite des moyennes et des variances des covariables.

Une application originale des mélanges de régressions est la reconstruction de la pose tridimensionnelle d'un corps humain (*i.e*. la position des membres dans l'espace) à partir d'une seule image (Argawal and Triggs, 2004 [5]). C'est un problème de régression multiple : les entrées X sont dans l'espace des images 2D et les sorties Y sont dans l'espace des poses tridimensionnelles. Dans ce problème, l'absence d'information spatiale dans les images 2D

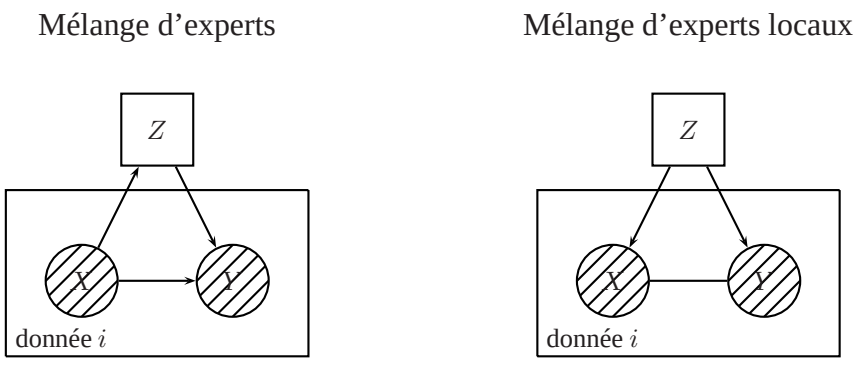


FIG. 3.1 – Modèles graphiques du mélange d'experts standard et du mélange d'experts locaux.

génére parfois des ambiguités, puisque des configurations 3D très différentes pouvant correspondre à des silhouette semblables. En n'utilisant qu'une dimension pour les entrées et les sorties, la figure 3.2 illustre ce phénomène. Pour la silhouette en bas à gauche, on ne sait pas si le bras saillant est celui de gauche ou de droite. De même on ne peut pas savoir si la jambe avant est celle de gauche ou de droite. La présence de cette ambiguité se caractérise par une distribution conditionnelle $P(Y|X)$ multimodale : les deux modes de la distribution conditionnelle $P(Y|X)$ correspondent aux configurations « jambe gauche à l'avant, bras droite à l'avant » et « jambe droite à l'avant, bras gauche à l'avant ».

Dans ce chapitre, nous évaluons les performances des mélanges de régression dans une application de prédiction du taux d'ozone maximale d'une journée. Le mélange de régression est adapté à ce cas car il existe des phénomènes météorologiques latents qui influent sur les corrélations entre les variables. En particulier, la présence de pluie la veille de la prédiction diminue la dépendance entre les taux d'ozone de deux journées consécutives.

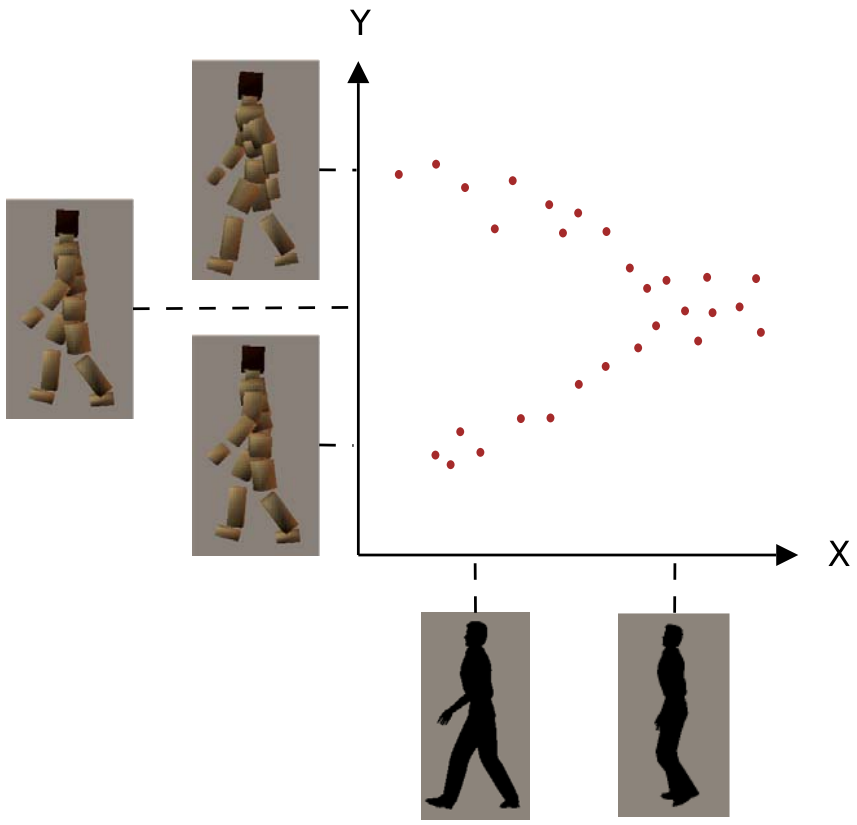


FIG. 3.2 – Un exemple d'application pour laquelle la distribution de Y conditionelle à X est multimodale. En abscisses sont représentées les silhouettes à partir desquelles on cherche à retrouver la pose tridimensionnelle du modèle humanoïde. Celle-ci est représentée sur l'axe des ordonnées, dont les valeurs peuvent être interprétées comme la position du bras droit et de la jambe gauche relative au centre de gravité du modèle. Les points représentés sont fictifs, mais les figures sont tirées de [5].

3.1 Les mélanges de régressions

Considérons un modèle de régression, où la variable dépendante Y peut être expliquée par un ensemble de variables données $X_1, \cdots, X_{d+1}$. Nous supposons que X_{d+1} est une variable discrète non observée. Ce régresseur est appelé *variable latente*. Une manière naturelle d'effectuer la régression est d'expliquer Y en fonction des d régresseurs restants. Le problème est que si la variable latente est très informative pour le problème à traiter, il peut être crucial d'essayer de retrouver (de manière statistique) sa valeur afin d'améliorer la régression. Dans certains cas, le modèle de régression peut être totalement différent pour chaque valeur de la variable latente X_{d+1} ; nous sommes alors dans un cas typique de « switching regression ». L'information manquante peut être estimée de manière efficace grâce à un mélange de régressions[133]. Les différents types de mélange de régressions ont été analysés par [72].

La switching regression est bien connue en économétrie. C'est un cas particulier des mélanges de régression pour lesquels les proportions des mélanges ne dépendent pas des régresseurs. Ce modèle a été introduit par Quandt en 1972 [132]. Kiefer [93], a donné des résultats de consistence de l'estimateur du maximum de vraisemblance. Une analyse bayésienne du modèle a aussi été proposée [77].

Dans un cadre plus général, les mélanges de régression sont souvent appelés *Mélanges d'experts* (Mixtures of Experts, ME) car ils ont tout d'abord été introduits dans la communauté du Machine Learning [83]. Les ME considèrent qu'une fonction particulère (appelée *gating network*) donne la distribution de la variable latente conditionnellement aux régresseurs. Ce type de modèle fait partie de la classe des *modèles de mélanges conditionnels*. Des résultat théoriques concernant le taux de convergence de l'algorithme EM ont été établis [165] de même que les conditions d'identifiabilité du modèle [88]. Une extension importante des ME est le ME hiérarchique pour lequel la distribution de la classe latente conditionnellement aux régresseurs a une structure d'arbre [90].

La motivation principale du travail présenté dans ce chapitre est l'étude des mélanges de régressions lorsque l'on modélise la distribution des régresseurs pour chaque classe latente par une distribution gaussienne multivariée. Ce modèle est appelé *mélanges d'experts localisés* (ou *localized mixture of experts* [120]) et a initialement été introduit par Xu en 1995 [164] comme alternative algorithmiquement satisfaisante des ME. Ce modèle est parfois appelée *normalized Gaussian networks* [144]. Notre approche a été d'analyser l'influence de la modélisation générative sur ce type de modèle, c'est-à-dire d'analyser les avantages de modéliser la distribution jointe des données.

Cela nous a permis de relier les *mélanges d'experts localisés* aux modèles de mélanges gaussiens classiques, et profiter de ce fait de leurs outils théoriques existants [115]. De cette manière, nous avons fourni une version de l'algorithme EM qui permet de réduire de manière significative le temps d'apprentissage des paramètres. Inversement, cela nous a permis de définir des contraintes spécifiques sur les paramètres des mélanges de gaussiennes classiques qui peuvent naturellement être introduites dans l'algorithme EM.

3.2 Le modèle

Considérons les relations entre trois variables X, Y et H :

- X dans $\mathbb{R}^d$ est un vecteur de d régresseurs,
- Y dans $\mathbb{R}$ est la variable à expliquer,
- H dans $\{1, \cdots, K\}$ est la variable latente (inobservée).

Soit $(x, y) = \{(x_i, y_i)_{i=1,\cdots,n}\}$ un échantillon iid d'observations du couple (X, Y). Comme H n'est pas observée, la densité de (X, Y) est obtenue par marginalisation :

$$p(X, Y) = \sum_{k=1}^{K} p(X, Y, H = k). \tag{3.1}$$

La règle de Bayes appliquée à $p(X, Y, H)$ permet de trouver deux expressions utiles de la densité jointe :

$$p(X, Y) = \sum_{k=1}^{K} p(X)p(H = k|X)p(Y|X, H = k) \tag{3.2}$$

$$p(X, Y) = \sum_{k=1}^{K} p(H = k)p(X|H = k)p(Y|X, H = k). \tag{3.3}$$

Pour ces deux paramétrisations, la distribution de Y conditionnellement à $H = k$ et $X = x$ est, comme pour la régression linéaire une variable gaussienne univariée de moyenne $\beta_k' x + \alpha_k$ et de variance τ_k^2 :

$$Y|X = x, H = k \sim \mathcal{N}(\beta_k' x + \alpha_k, \tau_k^2). \tag{3.4}$$

Nous présentons maintenant les moyens d'estimer β_k,α_k et τ_k.

3.2.1 Mélanges d'experts classiques

L'expression (3.2) correspond au modèle de mélange conditionnel, puisque maximiser sa log-vraisemblance ne nécessite pas de connaître la distribution des régresseurs X. Il est donc équivalent de travailler avec la distribution conditionnelle de Y sachant X :

$$p(Y|X) = \sum_{k=1}^{K} \underbrace{p(H=k|X)}_{gating\ network} \underbrace{p(Y|X,H=k)}_{expert} \tag{3.5}$$

Dans ce cas, le « gating network » est une sorte de classifieur retournant $p(H|X)$. Le modèle *logit multinomial*, aussi appelé fonction *softmax*, est souvent utilisé. C'est un *modèle linéaire généralisé* dont la forme de la distribution conditionnelle est :

$$p(H=k|X) = \frac{p_k e^{v_k x}}{\sum_{l=1}^{K} p_l e^{v_l x}}, \quad k = 1, \cdots, K, \tag{3.6}$$

où les vecteurs v_k et les proportions p_k sont des paramètres tels que $v_K = 0$, $0 < p_k < 1$ et $\sum_{k=1}^{K} p_k = 1$.

3.2.2 Mélanges d'experts localisés

Modélisant de manière générative les paramètres, nous optons pour une paramétrisation qui correspond aux modèles de mélanges classiques. Chaque composant a une densité qui se décompose sous la forme $p(X|H = k)p(Y|X, H = k)$. La variable H est discrète et suit une distribution multinomiale :

$$H \sim \mathcal{M}(1, p), \tag{3.7}$$

où $p = (p_1, \cdots, p_K)'$ est un vecteur de proportions ($\sum_{k=1}^{K} p_k = 1$). Sachant le composant de mélange H, les régresseurs X sont supposés gaussiens :

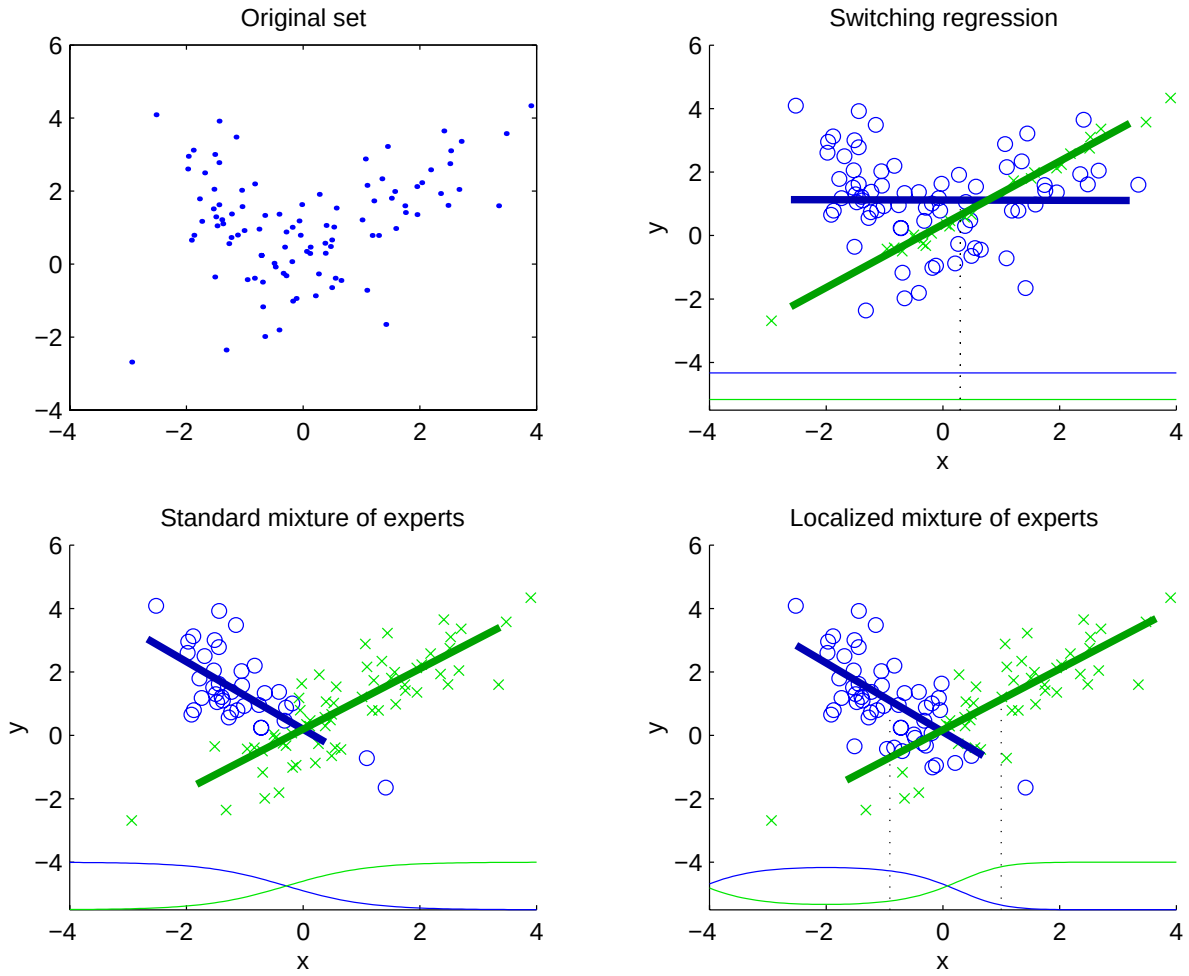
$$X|H = k \sim \mathcal{N}(\mu_k, \Sigma_k). \tag{3.8}$$

Avec la paramétrisation gaussienne, les composants peuvent être interprétés de manière plus naturelle que pour les ME classiques [165], puisque les moyennes μ_k donnent une idée de la position des régresseurs. La fonction gating network est obtenue par une application directe de la règle de Bayes :

$$p(H=k|X=x) = \frac{p(H=k)p(X=x|H=k)}{p(X=x)} \tag{3.9}$$

$$= \frac{p_k|\Sigma_k|^{-\frac{1}{2}} e^{-\frac{1}{2}(x-m_k)'\Sigma_k^{-1}(x-m_k)}}{\sum_{l=1}^{K} p_l|\Sigma_l|^{-\frac{1}{2}} e^{-\frac{1}{2}(x-m_l)'\Sigma_l^{-1}(x-m_l)}}. \tag{3.10}$$

FIG. 3.3 – Illustration d'un mélange de régressions : à partir d'un jeu de données où une simple régression linéaire n'est pas adaptée (en haut à gauche). Pour les modèles estimés, les proportions sont représentées en bas de chaque graphe. Le modèle de switching regression (en haut à droite) considère que les deux droites de régression ne dépendent pas des covariables (proportions constantes). Le modèle classique des mélanges d'experts (en bas à gauche) suppose que les proportions dépendent des régresseurs à travers un lien logistique. Les ME localisés supposent que la distribution des régresseurs dans chaque groupe est normale. Le paramètre de proportion est égal au rapport des densités. On voit que les deux modèles du bas donnes des résultats très similaires.



Cela correspond exactement à la forme logistique du gating network proposé par [165]. Ici, cette paramétrisation diffère de la fonction *softmax* classique $1/(1+\exp(\beta x))$ par la forme quadratique de la fonction de lien. Une

étude empirique [120], compare les deux types de gating network et conclut sur une légère supériorité du gating network de type linéaire, ce qui correspond à contraindre les variances de gaussiennes associées aux régresseurs à être égales. Les mélanges d'experts localisés ont été appliqués avec succès dans des applications de reconnaissance du langage [51] and [52]. Il est montré que de tels modèles permettent de réduire considérablement le temps d'apprentissage grâce à une procédure d'initialisation non-supervisée (sur les régresseurs X seulement) des paramètres μ_k and Σ_k. Cependant, cette étude empirique n'étudie pas les avantages théoriques des ME localisés.

La distribution jointe des observations (X', Y), $X \in \mathbb{R}^p$, $Y \in \mathbb{R}$ est un mélange de gaussiennes $d+1$ dimensionnelles. Les proportions sont les p_k, $k = 1, \cdots, K$ définies plus haut, la moyenne et la matrice de covariance du k-ième composant sont

$$m_k = \begin{pmatrix} \mu_k \\ \mu_k'\beta_k + \alpha_k \end{pmatrix}, \quad \Gamma_k = \begin{bmatrix} \Sigma_k & \Sigma_k\beta_k \\ \beta_k'\Sigma_k & \tau_k^2 + \beta_k'\Sigma_k\beta_k \end{bmatrix}. \tag{3.11}$$

Ainsi, le ME localisés est un simple mélange de gaussiennes avec une paramétrisation spécifique.

3.2.3 Contraintes sur les paramètres

Le nombre de paramètres dans le modèle mélange de régression que nous venons de présenter est une fonction quadratique de la dimension d des données d'entrée :

$$\nu = (\frac{d^2}{2} + \frac{5}{2}d + 3)K - 1,$$

ce qui peut être élevé lorsque les données sont en grande dimension. Pour obtenir un modèle plus parcimonieux et ainsi éviter le surapprentissage, il est possible de contraindre certains paramètres à être nuls ou égaux entre eux. Une contrainte classique est de supposer que les matrices Σ_k sont diagonales ($[\Sigma_{(k)}]_{ij} = 0$ pour $i \neq j$), c'est-à-dire que les covariables sont supposées indépendantes au sein de chaque cluster. Un tel modèle contient $K(2d+3)-1$ paramètres, ce qui est linéaire en d. Cette contrainte ne portant que sur le modèle de densité des covariables, la forme de la densité conditionnelle $\boldsymbol{p}(y|x,\theta)$ reste inchangée. Ainsi l'estimation des paramètres devient plus stable, sans pour autant introduire un biais dans la régression. Nous obtenons de cette manière des matrices de covariance

particulières Γ_k pour la distribution jointe des données du composant k :

$$\Gamma_k = \begin{bmatrix} \sigma_{k1}^2 & 0 & \dots & \sigma_{k1}^2\beta_{k1} \\ \vdots & \ddots & 0 & \vdots \\ 0 & \dots & \sigma_{kd}^2 & \sigma_{kd}^2\beta_{kd} \\ \sigma_{k1}^2\beta_{k1} & \dots & \sigma_{kd}^2\beta_{kd} & \tau_k^2 + \prod_{i=1}^{d} \beta_{ki}^2\sigma_{ki}^2 \end{bmatrix}. \tag{3.12}$$

Notons que cette matrice de covariance peut avoir son intérêt dans des problèmes différents de la régression, par exemple dans les modèles de mélange classiques avec des dépendances particulières entre les variables.

D'autres modèles peuvent être obtenus en forçant certains paramètres à être égaux entre les groupes :

1. $p_k = p$: les proportions des composants sont égales. Cette contrainte permet d'obtenir des groupes de tailles approximativement égales en terme de nombre de données. En pratique, cette contrainte améliore l'estimation du maximum de vraisemblance car elle permet de réduire le nombre de maxima locaux de la fonction de vraisemblance.

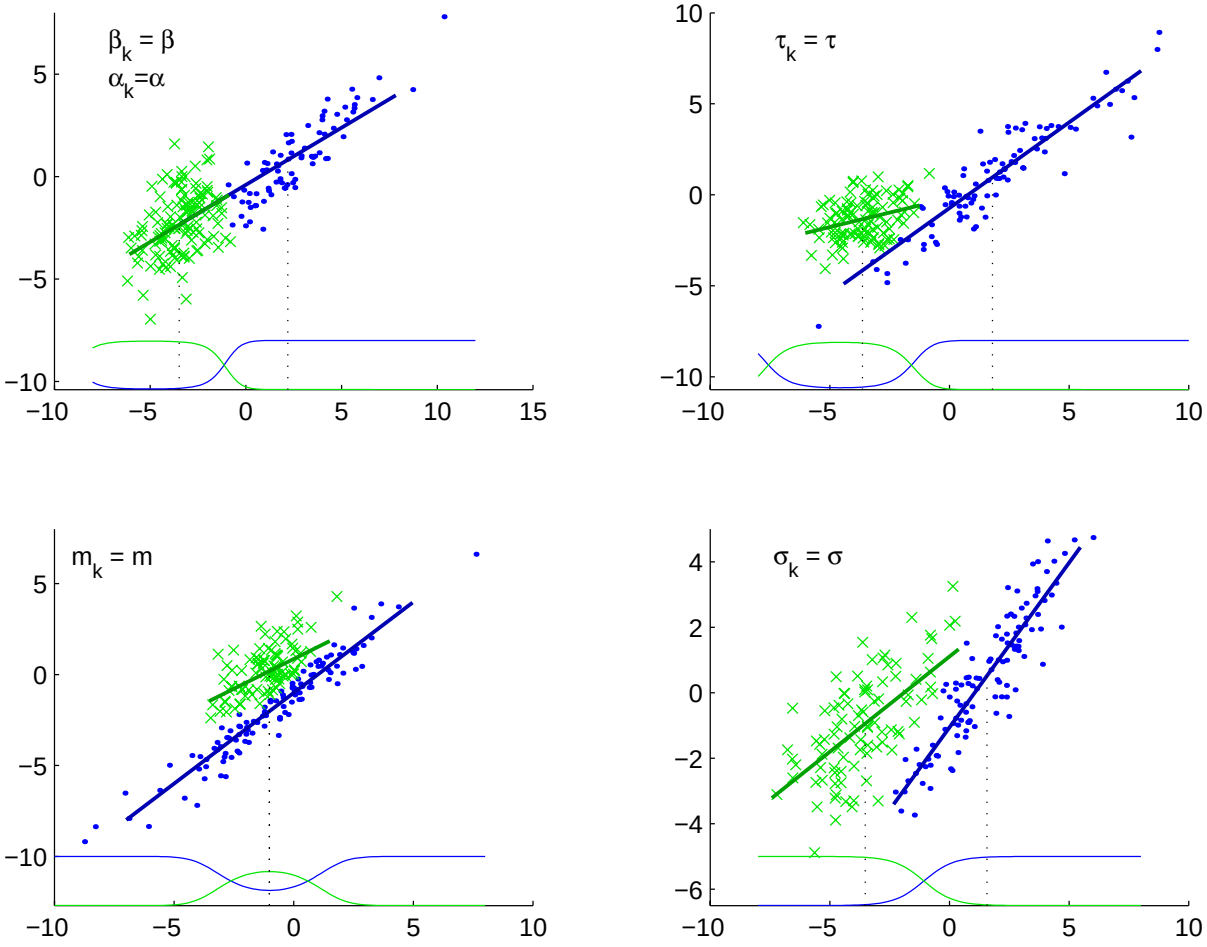
2. $\beta_k = \beta$: une pente commune entre les composants. Le modèle devient une régression linéaire *hétéroscédatique*, c'est-à-dire que la distribution de l'erreur autour de sa moyenne dépend de la valeur des régresseurs (Figure 3.4 en haut à gauche).

3. $\tau_k = \tau$: variance de l'erreur de régression commune entre les clusters. (en haut à groite sur la Figure 3.4).

4. $\Sigma_k = \Sigma$: matrice de variance des regresseurs commune. Cette contrainte est utile lorsque l'on souhaite avoir des séparations linéaires entre les groupes au lieu de frontières quadratiques. Ceci est illustré sur la Figure 3.4 en bas à droite : Avec la contrainte $\sigma_k = \sigma$, les probabilités des composants sont séparées entre droite et gauche, contrairement aux autres modèles.

Il est aussi possible de définir d'autres contraintes telles que $\alpha_k = \alpha$ ou $\mu_k = \mu$ (Figure 3.4 à gauche), *i.e.* supposer que les composants ont des ordonnées à l'origine ou des moyennes égales, ce qui donne des modèles très particuliers. Notons que les contraintes qui viennent d'être définies peuvent être combinées afin d'obtenir une famille de modèles très variée.

FIG. 3.4 – Quelques illustrations de mélanges de régression sur des modèles contraints. Les courbes en bas de chaque figure représentent les probabilité de chaque cluster.



3.3 Estimateur du maximum de vraisemblance

Avant de définir l'estimateur du maximum de vraisemblance, nous devons nous assurer que le modèle est identifiable. Ce problème a été abordé par [73], qui a donné des conditions nécéssaires pour l'existence d'un estimateur consistant des mélanges de régressions avec des régresseurs aléatoires : la distribution des régresseurs ne doit pas donner de probabilité positive à un hyperplan de dimension $(d-1)$. Cette condition est en général justifiée, et dans le contexte de l'apprentissage génératif, il suffit de vérifier que les matrices Σ_k ne sont pas singulières.

Nous décrivons l'algorithme EM [35] pour trouver l'estimateur génératif, *i.e*. celui qui maximise la vraisemblance jointe des données. Nous notons $\tilde{\beta}_k = (\alpha_k, \beta_k')'$ pour $k = 1, \cdots, K$. Soit θ le vecteur des paramètres contenant p_k, $\tilde{\beta}$, τ_k, μ_k et Σ_k pour $k = 1, \cdots, K$.

Etape E. Cette étape nécessite le calcul de l'espérance de la vraisemblance des données complétées :

$$Q(\theta|\theta^{(t)}) = E\{L_c(\theta; x, y)|x, y, \theta^{(t)}\} \tag{3.13}$$

où $\theta^{(t)}$ est le vecteur des paramètres à l'étape t et $L_c(\theta; x, y)$ est la vraisemblance complète. Le terme h désignant la densité de $X|H$ et g la densité de $Y|X, H$, nous avons

$$L_c(\theta; x, y) = \sum_{i=1}^{n}\sum_{k=1}^{K} c_{ik} \log\left(p_k h(x_i; \mu_k, \Sigma_k) g(y_i; x_i, \tilde{\beta}_k, \tau_k)\right). \tag{3.14}$$

Ici, c_{ik} égale 1 si la donnée i vient du composant k, et 0 dans le cas contraire. Son espérance sous le modèle paramétré par $\theta^{(t)}$ est :

$$w_{ik}^{(t)} = \frac{p_k^{(t)} h(x_i; \mu_k^{(t)}, \Sigma_k^{(t)}) g(y_i; x_i, \tilde{\beta}_k^{(t)}, \tau_k^{(t)})}{\sum_{l=1}^{K} p_l^{(t)} h(x_i; \mu_l^{(t)}, \Sigma_l^{(t)}) g(y_i; x_i, \tilde{\beta}_l^{(t)}, \tau_l^{(t)})}. \tag{3.15}$$

M step. L'étape de maximisation consiste à trouver les paramètres maximisant l'espérance de la vraisemblance complète :

$$\theta^{(t+1)} = \underset{\theta}{\operatorname{argmax}}\, Q(\theta|\theta^{(t)}). \tag{3.16}$$

A partir des équations (3.13) et (3.14) nous obtenons :

$$Q(\theta|\theta^{(t)}) = \sum_{i=1}^{n}\sum_{k=1}^{K} w_{ik}^{(t)} \log\left[p_k^{(t)} h(x_i; \mu_k^{(t)}, \Sigma_k^{(t)}) g(y_i; x_i, \tilde{\beta}_k^{(t)}, \tau_k^{(t)})\right] \tag{3.17}$$

Soit $X = [x_1, \cdots, x_n]'$ la matrice des régresseurs et $\tilde{X} = [\mathbf{1}_n\ X]$ où $\mathbf{1}_n$ est un vecteur unité de taille $n \times 1$. Y est le vecteur des y_i et $W_k^{(t)}$ sont des matrices $n \times n$ diagonales d'éléments diagonaux $w_{ik}^{(t)}$. L'expression (3.17) est maximisée en annulant ses dérivées partielles relativement à p_k, $\tilde{\beta}$, τ_k, μ_k and Σ_k. On obtient une solution

explicite :

$$p_k^{(t+1)} = \frac{1}{n} trW_k^{(t)}, \tag{3.18}$$

$$\tilde{\beta}_k^{(t+1)} = (\tilde{X}' W_k^{(t)} \tilde{X})^{-1} \tilde{X} W_k^{(t)} Y, \tag{3.19}$$

$$\tau_k^{2(t+1)} = \frac{1}{trW_k^{(t)}} (Y - \tilde{X}\tilde{\beta}_k^{(t+1)})' W_k^{(t)} (Y - \tilde{X}\tilde{\beta}_k^{(t+1)}), \tag{3.20}$$

$$\mu_k^{(t+1)} = \frac{1}{trW_k^{(t)}} X' W_k^{(t)} \mathbf{1}', \tag{3.21}$$

$$\Sigma_k^{(t+1)} = \frac{1}{trW_k^{(t)}} (X - \mathbf{1}\mu_k^{'(t)})' W_k^{(t)} (X - \mathbf{1}\mu_k^{'(t)}). \tag{3.22}$$

Notons que les équations (3.19) et (3.20) correspondent à des moindres carrés pondérés et que les équations (3.21) et (3.22) sont des estimations de moyenne et variance pondérées.

3.3.1 Estimation des modèles contraints

Afin d'adapter l'algorithme précédent aux modèles contraints définis plus haut, la procédure est la même : les dérivées de l'expression (3.17) sont calculées relativement aux paramètres, et l'annulation des ces dérivées permet d'obtenir une solution explicite, hormis pour la contrainte $\tilde{\beta}_k^{(t+1)}$. Dans ce cas, le système non linéaire suivant doit être résolu :

$$\begin{cases} \tau_k^{2(t+1)} = \frac{1}{trW_k^{(t)}} (Y - \tilde{X}\tilde{\beta}_k^{(t+1)})' W_k^{(t)} (Y - \tilde{X}\tilde{\beta}_k^{(t+1)}), \\ \tilde{\beta}^{(t+1)} = \left(\tilde{X}' (\sum_{k=1}^K \frac{1}{\tau_k^{2(t+1)}} W_k^{(t)}) \tilde{X} \right)^{-1} \tilde{X} (\sum_{k=1}^K \frac{1}{\tau_k^{2(t+1)}} W_k^{(t)}) Y. \end{cases} \tag{3.23}$$

Comme aucune solution explicite ne peut être obtenue, nous remplaçons simplement le terme $\tau_k^{2(t+1)}$ par sa valeur à l'étape précédente $\tau_k^{2(t)}$ dans l'expression de $\tilde{\beta}_k^{(t+1)}$. On peut prouver que dans ce cas, l'étape M augmente la vraisemblance. (La preuve nécessite de prouver que $\tilde{\beta}_k^{(t+1)} - \tilde{\beta}_k^{(t)}$ a une produit scalaire positif avec $\nabla L(\tilde{\beta}_k^{(t)})$, L désignant la vraisemblance. Ainsi, nous avons défini un algorithme EM *généralisé* qui a les mêmes propriétés de convergence que EM dans sa version standard [35].

3.3.2 Diminution de la complexité algorithmique

Lorsque nous considérons le modèle sans contrainte sur les paramètres, l'estimateur du maximum de vraisemblance peut être obtenu en estimant un modèle de mélange gaussien sur les données jointes (X, Y), puisque

nous avons prouvé l'équivalence entre les deux modèles. Cette approche se distingue de [51] car il n'effectue un apprentissage non supervisé que sur les données d'entrée X. La manière la plus simple d'estimer μ_k et Γ_k est l'algorithme EM appliqué à des mélanges de gaussiennes multidimensionnelles [115]. A partir des estimateurs $\hat{\mu}_k$ and $\hat{\Gamma}_k$, nous avons $\hat{\mu}_k = \begin{bmatrix} e_k \\ f_k \end{bmatrix}$ et $\hat{\Gamma}_k = \begin{bmatrix} A_k & b_k \\ b_k' & c_k \end{bmatrix}$, A_k est une matrice $d \times d$, b_k et e_k des vecteurs dans $\mathbb{R}^d$ et f_k et e_k des valeurs réelles. Les équations (3.11) sont ensuite résolues, pour finalement obtenir $\hat{\mu}_k = e_k$, $\hat{\Sigma}_k = A_k$, $\hat{\beta}_k = \hat{\Sigma}_k^{-1} b_k$, $\hat{\tau}_k = c_k - \hat{\beta}_k' \hat{\Sigma}_k \hat{\beta}_k$, et $\alpha_k = f_k - \hat{\mu}_k' \hat{\beta}_k$. Cette estimation est plus simple et plus rapide que l'algorithme EM précédent si on compare les étapes M[13]. En effet, dans le premier algorithme, la résolution d'un problème de moindre carrés pondérés (3.19) est nécessaire à chaque étape et pour chaque composant. En revanche, en passant par un apprentissage de mélanges gaussiens de dimension $d + 1$, la résolution de système linéaire n'intervient qu'après la dernière étape de l'algorithme, ce qui réduit notablement la quantité de calculs globale.

3.3.3 Données réelles : prédiction du taux d'ozone atmosphérique

Le modèle de mélange de régressions a été appliqué à des données météorologiques pour lesquelles le taux de concentration en ozone de l'air[14] de Rennes (France) doit être estimé en fonction de quatre régresseurs influants mesurés le jour précédent : la température à J-12 heures (en °C), la concentration maximale en ozone le jour précédent (en ppbv), la vitesse du vent d'est à J-12 heures (en m.s^{-1}) et la valeur maximale de la couverture nuageuse (échelle variant de 0 $\equiv$ pour un ciel totalement dégagé à 10 $\equiv$ pour un brouillard opaque). Le jeu de donnée comprend 1562 exemples. Nous avons observé que l'apprentissage conditionnel (*i.e.* discriminatif), correspondant au mélange d'experts standard, donne de meilleurs résultats que l'estimation générative, en testant les performances de prédiction par half-sampling. Ce phénomène n'est pas surprenant car nous avons vu (chapitre 2) que l'estimation discriminative est supérieure à l'estimation générative lorsque le modèle est biaisé et que le nombre de données d'apprentissage est suffisant, ce qui est probablement cette situation. Afin de tester la qualité de la régression lorsque le nombre de données d'apprentissage est réduit, nous avons sélectionné aléatoirement

[13] Les deux algorithmes ont une étape E de complexité similaire : ils necessitent la décomposition de Choleski d'une matrice de taille d ou $d + 1$ pour calculer la densité des gaussiennes.

[14] La concentration en ozone est mesurée en ppbv : parties par milliard en volume d'air.

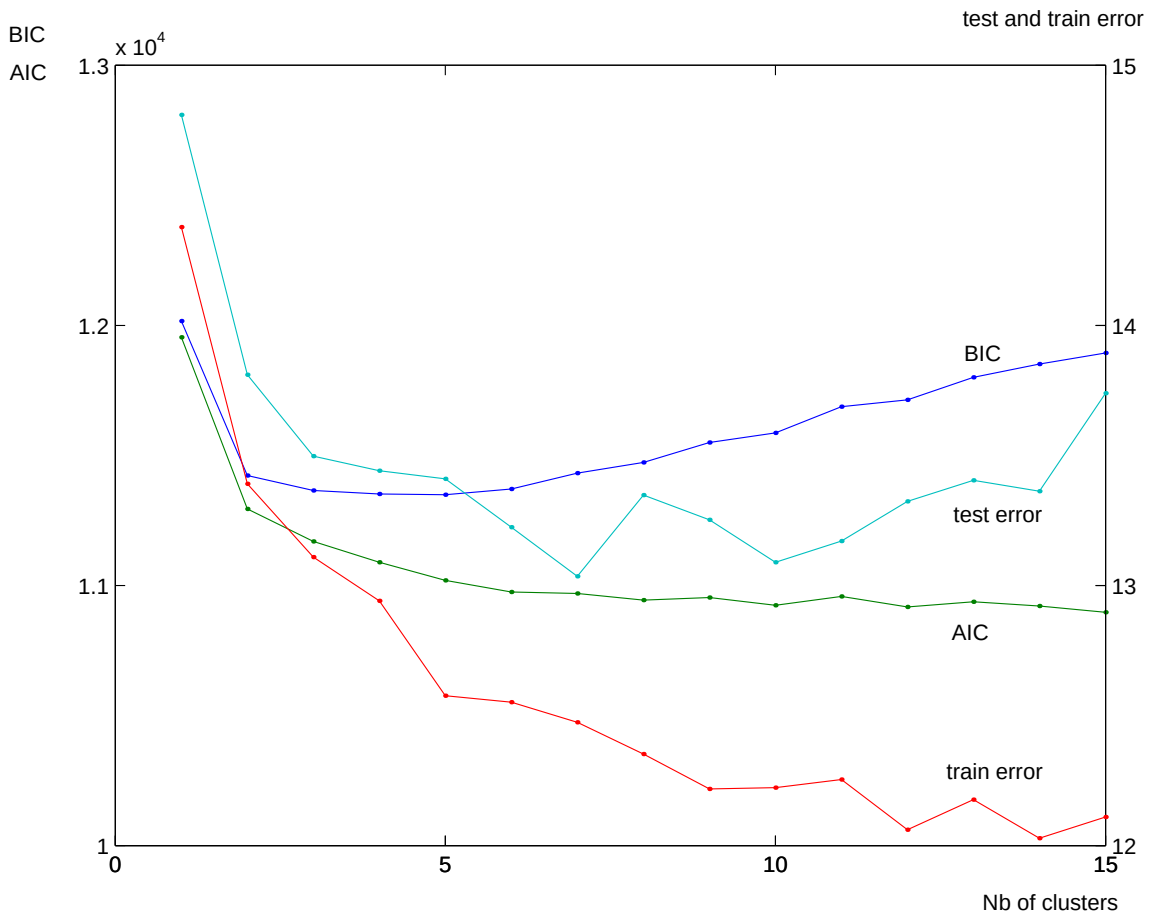


FIG. 3.5 – Critère de sélection du nombre de composants pour les données de prédiction d'ozone.

10% des données pour effectuer l'apprentissage, et comparé les estimations gnératives et discriminatives sur les 90% de données restantes.

Afin de choisir le nombre de composants, les critères AIC et BIC sont calculés. Leurs valeurs sont données sur la Figure 3.5 et sont comparées au taux d'erreur quadratique sur l'échantillon de test. On constate que BIC sélectionne quatre composants, ce qui est satisfaisant du point de vue de l'erreur de test. En revanche, le critère AIC ne semble pas pénaliser suffisamment la complexité du modèle et ne donne pas un nombre de composants satisfaisant. Dans cette application, la régression linéaire simple (1 seul composants) donne des résultats nettement inférieurs à ceux obtenus par mélanges de régression.

Pour $K = 2$ composants, le modèle estimé est représenté sur la figure 3.6, où le type de point représente le

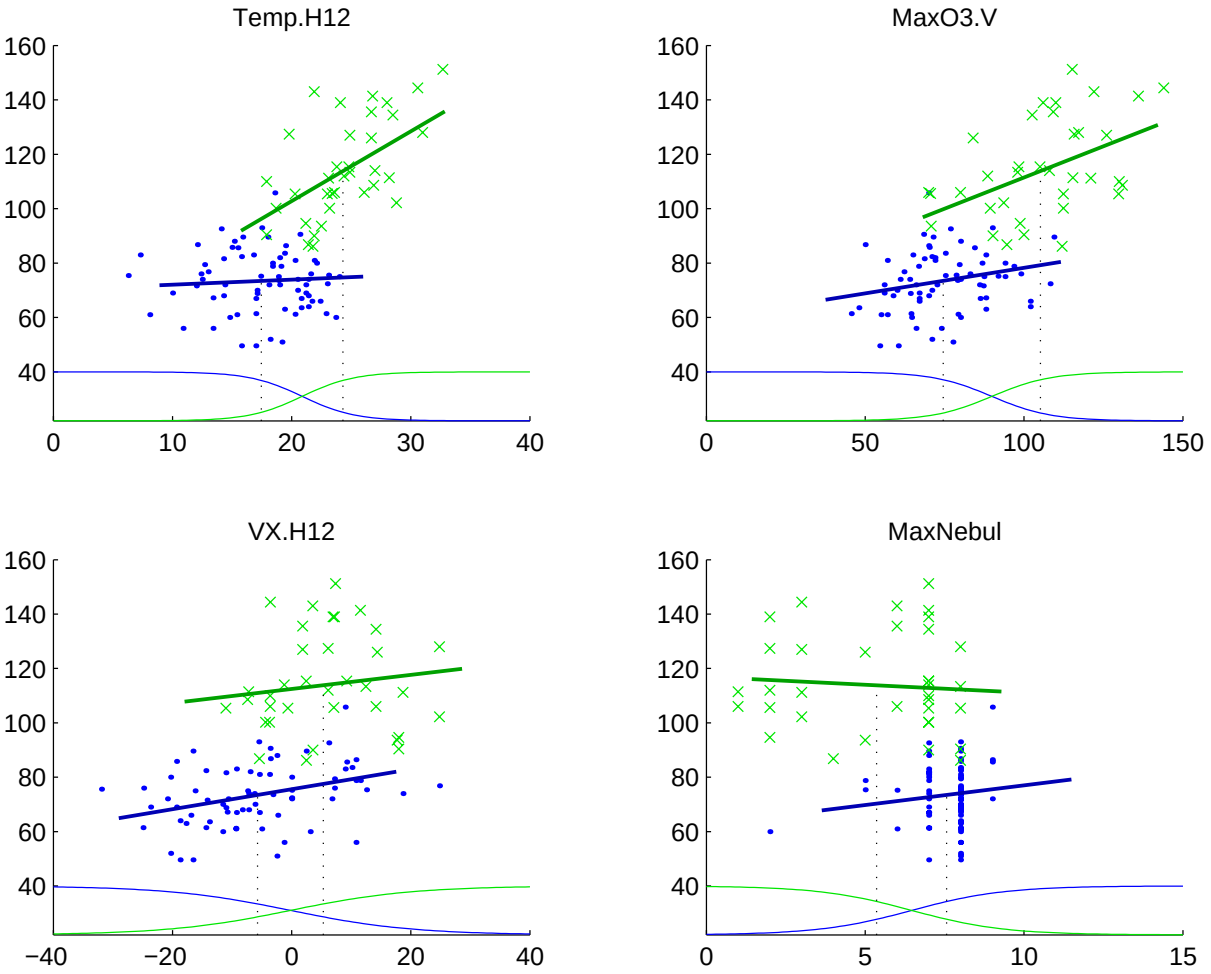


FIG. 3.6 – Représentation d'un mélange de deux composants pour la prédiction du taux d'ozone. Chaque graphe représente la concentration d'ozone à prédire en fonction d'un régresseur : températeur à J-12 heure, taux maximum d'ozone du jour précédent, vitesse maximale du vent d'ouest et mesure de la nébulosité.

composant le plus probable. Les jours de forte nébulosité (cluster de points) montrent une faible pente en fonction de la température (premier graphe), mais l'autre composant (croix) correspond à des jours où la concentration et la température sont fortement corrélées. On peut donner une interprétation au premier composant : il devrait correspondre aux jours de pluie, pour lesquels l'ozone de l'air est éliminée par la pluie, quelle que soit la température.

Pour comparer les performances en prédiction, nous avons utilisé $K = 4$ composants. Les résultats sont

	Max. Vrais. Conditionnelle		Max. Vrais. Jointe	
MSE	moyenne	ecart-type	moyenne	ecart-type
Apprentissage	12.0	0.187	12.4	0.0096
Test	15.1	0.324	14.6	0.0132

TAB. 3.1 – Résultats sur le jeu de données de prédiction de l'ozone atmosphérique.

donnés sur la Table 3.1, où MSE correspond à Mean Squared Error (MSE), *i.e.* l'erreur quadratique moyenne. Les valeurs ont été moyennées sur 100 séparations aléatoires en ensembles d'apprentissage et de test. Cette fois, c'est l'estimation générative qui donne des performances nettement meilleures que l'estimation conditionnelle. Cela montre que l'hypothèse de normalité des composants, bien que fausse sur les données réelles, peut être utile pour réduire la variance lorsque le nombre de données est petit. Dans ce cas précis, la variance de l'estimateur conditionnel domine le biais de l'estimateur génératif.

3.4 Discussion

Nous avons étudié les mélanges de gaussiennes dans un objectif de régression, et montré que cela correspond aux modèle de mélanges d'experts avec une estimation générative. La paramétrisation générative donne une interprétation naturelle aux clusters et permet d'introduire des contraintes sur les paramètres. Tous les sous-modèles ainsi définis peuvent servir à des objectifs variés, incluant la régression linéaire hétéroscédatique ou robuste. Le fait de pouvoir modéliser les erreurs par un mélange de gaussiennes enrichit considérablement le champ d'applications des mélanges de régressions linéaires. Des hypothèses particulières sur la distribution des covariables permettent d'obtenir des modèles avec un nombre limité de paramètres, et donc particulièrement performants en généralisation. Enfin, nous avons proposé un modèle efficace d'estimation lorsqu'il n'y a pas de contrainte sur les paramètres.

Chapitre 4

Sélection de modèle pour la classification générative

En classification générative, une travail de modélisation probabiliste est nécessaire. Cette étape nécessite de faire des choix sur la forme des distributions qui peuvent avoir un impact très important sur le taux d'erreur final du classificateur. En effet, une grande variété de modèles de densité ont été proposés, et même si le modélisateur a une idée précise sur la forme des données à classer, il est souvent difficile de savoir quel modèle sera le plus adapté pour la classification. Ce choix est d'autant plus délicat que la plupart des modèles de densités ont un paramètre de complexité généralement difficile à régler. Dans l'exemple du modèle génératif basé sur les mélanges de distributions gaussiennes (chapitre 2), le paramètre « critique » est le nombre de composants des mélanges.

Dans ce chapitre, nous considérons le problème de classification supervisée basée sur l'estimateur génératif, c'est-à-dire maximisant la vraisemblance jointe des variables explicatives et des variables expliquées. Le modèle recherché est cependant celui qui maximise la vraisemblance conditionnelle. Le fait que la fonction objectif soit différente pour l'estimation des paramètres et la sélection de modèle rend sous-optimaux, voire inefficaces les critère classiques de sélection de modèle tels que AIC ou BIC.

En approchant la *vraisemblance conditionnelle intégrée*, nous obtenons un critère de sélection de modèle que nous nommons *Bayesian Entropy Criterion* (BEC). Il nécessite l'estimation du maximum de vraisemblance margi-

nale des paramètres, effectuée par l'algorithme EM. Le critère proposé ne fait pas intervenir directement le nombre de paramètres libres des modèles candidats.

Nous ne considérons que le cas où l'on dispose d'un nombre fini de modèles candidats. Lorsqu'un paramètre continu doit être choisi (par exemple un paramètre de lissage), il suffit de considérer une grille de valeurs possibles. Quelques résultats théoriques sont établis : lorsqu'il n'y a pas de modèle emboité, le critère sélectionne asymptotiquement le modèle ayant la plus faible entropie de classification sur des données de test. En présence de modèles emboités, BEC a un comportement « en plateau », c'est-à-dire qu'il se stabilise à une valeur constante lorsqu'il a atteint le modèle de complexité minimale.

Des simulations sur différentes familles de modèle montrent que le critère a un comportement satisfaisant. Sur une expérience de catégorisation d'objet, BEC permet de choisir une complexité donnant un taux d'erreur similaire à la validation croisée. Cette application montre qu'une méthode générative simple peut concurrencer des approches purement discriminatives de type SVM, considérées comme optimales pour ce type de problèmes.

4.1 Introduction

En apprentissage statistique, les modèles de classification supervisée font des suppositions sur les densités par groupe. Beaucoup de méthodes paramétriques ou non paramétriques ont été conçus ou peuvent être présentées dans ce cadre (cf. [114]). Dans beaucoup de problèmes pratiques, il peut être utile de mettre en compétition plusieurs modèles génératifs pour construire une règle de décision minimisant le taux d'erreur futur. Des exemples où une famille de méthodes est considérée dans le but d'en extraire le meilleur représentant sont [70] et [15]. Dans cette perspective, une tâche importante est de sélectionner un modèle pertinent dans la collection de modèles génératifs en compétition. Une manière naturelle de traiter ce problème de sélection de modèles est d'évaluer le taux d'erreur estimé par validation croisée des modèles. Cependant, ce type d'évaluation est chère et l'emploi d'autres critères fait sens. Mais, les critères classiques de sélection de modèles ne prennent pas en compte l'objectif de classification et peuvent s'avérer décevants. Dans ce chapitre, nous proposons un nouveau critère de sélection de modèle qui prend en compte l'objectif de classification. Avant de le présenter, nous rappelons les points de vue à partir desquels les critères classiques de sélection de modèles ont été conçus. En statistique inférentielle, sélectionner un modèle parcimonieux dans une collection de modèles est une tâche importante mais difficile. Ce problème général a fait l'objet de nombreuses recherches depuis les articles pionniers de [8] et de [147]. Il s'agit essentiellement de résoudre le dilemne biais/variance : un modèle trop simple produira une erreur d'approximation importante (sous-apprentissage) tandis qu'un modèle trop complexe produira une erreur d'estimation importante (sur-apprentissage).

Une approche classique consiste à pénaliser une mesure d'ajustement d'un modèle par une mesure de complexité. Une mesure d'ajustement répandue d'un modèle $m \in \mathcal{M}$ est sa *déviance*

$$d(\boldsymbol{x}) = 2[\log \boldsymbol{p}(\boldsymbol{x}) - \log \boldsymbol{p}(\boldsymbol{x}|\hat{\theta}_m)]$$

où $\boldsymbol{p}(\boldsymbol{x}) = \prod_{1=1}^{n} \boldsymbol{p}(x_i)$ désigne la vraie distribution des données $\boldsymbol{x} = (x_1, \ldots, x_n)$ (pour faire simple, les $x_i s$ sont supposés indépendants et identiquement distribués (iid) ; $\boldsymbol{p}(\boldsymbol{x}|\theta_m) = \prod_{i=1}^{n} \boldsymbol{p}(x_i|\theta_m)$ est la distribution sous le modèle m, de paramètre θ_m ; et $\hat{\theta}_m$ est l'estimateur du maximum de vraisemblance θ_m. Sous l'approche du maximum de vraisemblance et dans une perspective prédictive, une manière naturelle de pénaliser vient de l'idée que la déviance sera plus petite pour un ensemble d'apprentissage que pour un ensemble test de taille comparable,

puisque les paramètres sont choisis de sorte à minimiser la déviance sur l'ensemble d'apprentissage. Ainsi, la pénalisation doit refléter la différence de déviance entre ces deux ensembles d'apprentissage et de test. Autrement dit, il doit approximer $nD(X) - E(d(\boldsymbol{x}))$ où :

$$D(X) = 2E[\log \boldsymbol{p}(X) - \log \boldsymbol{p}(X|\hat{\theta}_m)]$$

est la déviance attendue pour une observation test X. Sous l'hypothèse que les données sont issues d'une distribution de la famille de modèles considérés, Akaike a proposé d'estimer cette différence par $2\nu_m$ où ν_m est le nombre de paramètres indépendants du modèle m [8, 137]. Cela conduit au critère AIC :

$$\text{AIC}(m) = 2 \log \boldsymbol{p}(\boldsymbol{x}|\hat{\theta}_m) - 2\nu_m. \tag{4.1}$$

La suppression de cette hypothèse irréaliste conduit à définir d'autres critères comme le critère NEC *Network Information Criterion* [124]. (Des détails sont dans [137], pp.32-34 and 61.)

Une autre approche consiste à fonder la sélection d'un modèle sur la vraisemblance marginale d'un modèle en se plaçant dans une perspective bayésienne ([91]. Cette vraisemblance marginale s'écrit :

$$\boldsymbol{p}(\boldsymbol{x}|m) = \int \boldsymbol{p}(\boldsymbol{x}|\theta_m)\pi(\theta_m)d\theta_m, \tag{4.2}$$

$\pi(\theta_m)$ étant une distribution *a priori* pour le paramètre θ_m. Le problème technique à résoudre est d'approximer le logarithme de cette vraisemblance intégrée. Une approximation asymptotique classique conduit au critère BIC [147]

$$\text{BIC}(m) = \log \boldsymbol{p}(\boldsymbol{x}|\hat{\theta}_m) - \frac{\nu_m}{2} \log(n). \tag{4.3}$$

Cette approximation est valide sous des conditions de régularité sur les vraisemblances de la collection de modèles $\mathcal{M}$ et n'est précise que lorsque la loi a priori $\pi(\theta_m)$ est centrée sur l'estimateur du maximum de vraisemblance $\hat{\theta}_m$ [134]. On doit noter à ce sujet, que cette hypothèse n'est réaliste que si un et un seul modèle est le bon modèle (cf. [16], chapitre 6).

Face à la difficulté inhérente au problème de sélection de modèles, un nombre croissant d'auteurs suggèrent qu'il est peu réaliste de déconnecter la sélection de modèles du but de la modélisation. Ainsi, choisir le nombre de composants d'un modèle de mélange peut grandement dépendre du but de l'utilisateur. Si le modèle de mélange est considéré comme un outil d'estimation semi paramétrique de densités, alors le critère BIC s'avère satisfaisant

en pratique [141, 48]. Mais ce modèle est utilisé dans un but de classification non supervisée. D'autres critères, comme le critère ICL, prenant ce but en compte peuvent s'avérer plus pertinents (cf. [17] ou [115], chapitre 6) Dans ce chapitre, nous sommes confrontés au problème de choix de modèles probabilistes pour la classification supervisée. Des critères comme AIC et BIC ne prennent pas en compte cet objectif de classification et ont des pénalités fixes. En fait, dans ce contexte, il existe un critère de référence qui est le taux d'erreur évalué par validation croisée. Mais ce critère est cher. Nous proposons un nouveau critère de vraisemblance pénalisée qui prend en compte l'objectif de classification. Ce critère est de même nature que le critère BIC, mais c'est une approximation asymptotique de la vraisemblance conditionnelle intégrée du modèle génératif de classification et non, comme BIC, de la vraisemblance jointe intégrée. Ce critère offre une alternative intéressante à la validation croisée.

Le plan du chapitre est le suivant. Le paragraphe 4.2 présente le problème de sélection de modèle pour des modèles génératifs en classification supervisée. Notre critère, dénommé BEC, est présenté au paragraphe 4.3. Son comportement asymptotique est analysé au paragraphe 4.4, et des expériences numériques sur des données simulées et réelles sont présentées au paragraphe 4.5 pour illustrer son comportement pratique. Le chapitre se termine par une courte discussion.

4.2 Classifieurs génératifs et sélection de modèles

En classification supervisée, il s'agit de deviner la classe $Y \in \{1, \ldots, K\}$ d'un vecteur observé $\boldsymbol{X}$ à valeurs dans $\mathbb{R}^d$. Dans ce but, une fonction de décision, appellée classifieur, $\delta(\boldsymbol{X}) : \mathbb{R}^d \to \{1, \ldots, K\}$ est construite à l'aide d'un échantillon d'apprentissage $(\boldsymbol{x}_i, y_i), i = 1, \ldots, n$. Pour des raisons de simplicité, les $\boldsymbol{x}_i's$ sont supposés iid. Une approche classique est de modéliser les densités conditionnelles par des densités paramétriques $\boldsymbol{p}(\boldsymbol{X}|Y = k, \theta_m)$ pour $k = 1, \ldots, K$, où m désigne les paramètres du modèle $\theta_m \in \Theta_m$. Nous supposons que l'espace des paramètres est de dimension finie. Les observations $\boldsymbol{X}$ sont affectées à la classe k qui maximise la probabilité conditionnelle, $\boldsymbol{p}(Y = k|\boldsymbol{X}, \theta_m)$. Par la règle de Bayes, cela conduit au classifieur :

$$\delta(\boldsymbol{X}) = \underset{k \in \{1,\ldots,K\}}{\operatorname{argmax}} \boldsymbol{p}(\boldsymbol{X}, Y = k|\hat{\theta}_m), \tag{4.4}$$

où $\hat{\theta}_m$ est un estimateur de θ_m fondé sur l'échantillon d'apprentissage. Cette approche est connue sous le nom d'approche générative [85, 148]. L'estimation du maximum de vraisemblance (ML) fondée sur les distributions

conditionnelles aux classes est un estimateur populaire pour lequel la vraisemblance jointe des observations $\boldsymbol{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ et des labels $\boldsymbol{y} = (y_1, \ldots, y_n)$ est maximisée.

$$\hat{\theta} = \arg\max_{\theta} \boldsymbol{p}(\boldsymbol{x}, \boldsymbol{y}|\theta_m). \tag{4.5}$$

En classification supervisée, il est souvent utile de construire la fonction de classification en utilisant plusieurs modèles et en choisissant le modèle qui fournit le plus petit taux d'erreur sur un ensemble test. Dans le contexte génératif, plusieurs méthodes requièrent une telle étape de sélection de modèles. Des exemples récents et qui seront considérés au paragraphe 4.5 sont des distributions gaussiennes multivariées avec différentes hypothèses sur la décompostion spectrale des matrices de variances [15], et l'analyse discriminante par mélange (MDA) [70]. Par exemple, dans l'approche MDA où chaque densité par classe est supposée être un mélange de lois gaussiennes, les nombres de composants du mélange par classes sont des paramètres de réglage importants. Ils peuvent soit être fournis par l'utilisateur [70], ce qui est évidemment sous optimal, ou choisis de sorte à minimiser l'erreur de classement évalué par une procédure v-*fold* de validation croisée, comme dans [49] ou dans [15] pour d'autres paramètres de réglage. Cela peut être considéré comme une solution satisfaisante bien que le choix de v puisse être délicat et que cela soit coûteux. Donc, le problème de bien choisir de tels paramètres de réglage avec un critère de vraisemblance pénalisée dans l'esprit de BIC est souhaitable dans maintes circonstances. Dans le contexte de la classification, BIC prend la forme :

$$\text{BIC}(m) = \log \boldsymbol{p}(\boldsymbol{x}, \boldsymbol{y}|\hat{\theta}_m) - \frac{\nu_m}{2} \log(n), \tag{4.6}$$

où ν_m est la dimension de θ_m. Mais, BIC mesure l'ajustement de m aux données $(\boldsymbol{x}, \boldsymbol{y})$ plutôt que sa capacité à fournir un bon classifieur. Aussi, dans beaucoup de situations, BIC fournit des résultats décevants pour choisir le modèle à taux d'erreur minimum. Dans le but de répondre à cette limitation, nous proposons un critère de vraisemblance pénalisée qui prend en compte l'objectif de classification pour évaluer les performances d'un modèle.

4.3 Le critère d'entropie bayésienne

Comme indiqué ci-dessus, un classifieur déduit d'un modèle m affecte une observation $\boldsymbol{X}$ à la classe k qui maximise $\boldsymbol{p}(y = k|\boldsymbol{X}, \hat{\theta}_m)$. Ainsi, du point de vue de la classification, la vraisemblance conditionnelle $\boldsymbol{p}(\boldsymbol{y}|\boldsymbol{x}, \theta_m)$

joue un rôle capital. Pour sélectionner un modèle pertinent m, nous proposons d'utiliser la vraisemblance conditionnelle intégrée *integrated conditional likelihood* :

$$\boldsymbol{p}(\boldsymbol{y}|\boldsymbol{x},m)=\int \boldsymbol{p}(\boldsymbol{y}|\boldsymbol{x},\theta_m)\pi(\theta_m|\boldsymbol{x})d\theta_m, \tag{4.7}$$

où

$$\pi(\theta_m|\boldsymbol{x})=\frac{\pi(\theta_m)\boldsymbol{p}(\boldsymbol{x}|\theta_m)}{\boldsymbol{p}(\mathbf{x}|m)}$$

est la distribution a posteriori de θ_m sachant $\boldsymbol{x}$. Comme pour la vraisemblance intégrée, cette intégrale est généralement difficile à calculer et doit être approximée. L'approximation de $\log \boldsymbol{p}(\boldsymbol{y}|\boldsymbol{x},m)$, que nous présentons maintenant, conduit au critère BEC *Bayesian Entropy Criterion*. Nous avons :

$$\boldsymbol{p}(\boldsymbol{y}|\boldsymbol{x},m)=\frac{\boldsymbol{p}(\boldsymbol{x},\boldsymbol{y}|m)}{\boldsymbol{p}(\boldsymbol{x}|m)} \tag{4.8}$$

$$\boldsymbol{p}(\boldsymbol{x},\boldsymbol{y}|m)=\int \boldsymbol{p}(\boldsymbol{x},\boldsymbol{y}|\theta_m)\pi(\theta_m)d\theta_m \tag{4.9}$$

$$\boldsymbol{p}(\boldsymbol{x}|m)=\int \boldsymbol{p}(\boldsymbol{x}|\theta_m)\pi(\theta_m)d\theta_m. \tag{4.10}$$

Nous pouvons appliquer l'approximation de Laplace à ces deux intégrales (4.9) et (4.10) puis approximer leurs logarithmes comme il est fait pour obtenir BIC [134]. Soit $\tilde{\theta}_m=\arg\max_\theta \boldsymbol{p}(\boldsymbol{x}|\theta_m)$ et faisant l'hypothèse que la distribution a priori $\pi(\theta_m)$ de θ_m peut être approchée par une loi normale de moyenne θ_m^0 et de variance V_m^0, nous pouvons écrire [137] :

$$\log \boldsymbol{p}(\boldsymbol{x},\boldsymbol{y}|m)\approx \log \boldsymbol{p}(\boldsymbol{x},\boldsymbol{y}|\hat{\theta}_m)-\frac{\nu_m}{2}\log n-\frac{1}{2}\log|\hat{J}_J|-\frac{1}{2}(\hat{\theta}_m-\theta_m^0)^t V_0^{-1}(\hat{\theta}_m-\theta_m^0) \tag{4.11}$$

$$\log \boldsymbol{p}(\boldsymbol{x}|m)\approx \log \boldsymbol{p}(\boldsymbol{x}|\tilde{\theta}_m)-\frac{\nu_m}{2}\log n-\frac{1}{2}\log|\tilde{J}_M|-\frac{1}{2}(\tilde{\theta}_m-\theta_m^0)^t V_0^{-1}(\tilde{\theta}_m-\theta_m^0), \tag{4.12}$$

où ν_m est la dimension de θ_m, et $\hat{J}_J=J_J(\hat{\theta}_m)$ et $\tilde{J}_M=J_M(\tilde{\theta}_m)$ sont les matrices hessiennes normalisées de l'opposée des logvraisemblances jointe et marginale en $\hat{\theta}_m$ et en $\tilde{\theta}_m$ respectivement :

$$J_J(\theta_m)=-\tfrac{1}{n}\tfrac{\partial^2}{\partial\theta_m\partial\theta_m{}^T}\log \boldsymbol{p}(\boldsymbol{x},\boldsymbol{y}|\theta_m),\quad J_M(\theta_m)=-\tfrac{1}{n}\tfrac{\partial^2}{\partial\theta_m\partial\theta_m{}^T}\log \boldsymbol{p}(\boldsymbol{x}|\theta_m).$$

La différence des deux expressions (4.11) et (4.12) donne

$$\begin{aligned}\log \boldsymbol{p}(\boldsymbol{y}|\boldsymbol{x}, m) \approx & \quad \log \boldsymbol{p}(\boldsymbol{x}, \boldsymbol{y}|\hat{\theta}_m) - \log \boldsymbol{p}(\boldsymbol{x}|\tilde{\theta}_m) \\ & - \quad \frac{1}{2}\log|\hat{J}_J \tilde{J}_M^{-1}| - \frac{1}{2}(\hat{\theta}_m - \tilde{\theta}_m)^t V_0^{-1}(\hat{\theta} + \tilde{\theta} - 2\theta_m^0) \\ \approx & \quad \log \boldsymbol{p}(\boldsymbol{x}, \boldsymbol{y}|\hat{\theta}_m) - \log \boldsymbol{p}(\boldsymbol{x}|\tilde{\theta}_m) \\ & - \quad \frac{1}{2}\log|I_d + \hat{J}_C \tilde{J}_M^{-1}| - \frac{1}{2}(\hat{\theta}_m - \tilde{\theta}_m)^t V_0^{-1}(\hat{\theta} + \tilde{\theta} - 2\theta_m^0).\end{aligned}$$

où $\hat{J}_C = J_C(\hat{\theta}_m)$ avec

$$J_C(\theta_m) = -\frac{1}{n}\frac{\partial^2}{\partial\theta_m \partial{\theta_m}^T}\log \boldsymbol{p}(\boldsymbol{y}|\boldsymbol{x}, \theta_m).$$

Supprimant les termes d ordre O(1) donne :

$$\log \boldsymbol{p}(\boldsymbol{y}|\boldsymbol{x}, m) \approx \log \boldsymbol{p}(\boldsymbol{x}, \boldsymbol{y}|\hat{\theta}_m) - \log \boldsymbol{p}(\boldsymbol{x}|\tilde{\theta}_m). \tag{4.13}$$

Ainsi l'approximation de $\log \boldsymbol{p}(\boldsymbol{y}|\boldsymbol{x}, m)$ que nous proposons est

$$\text{BEC} = \log \boldsymbol{p}(\boldsymbol{x}, \boldsymbol{y}|\hat{\theta}_m) - \log \boldsymbol{p}(\boldsymbol{x}|\tilde{\theta}_m). \tag{4.14}$$

Quelques commentaires méritent d'être faits.

- La vraisemblance conditionnelle intégrée peut s'interpréter comme l'entropie bayésienne [53, 122] de la classification déduite du modèle m, d'où son nom *Bayesian Entropy Criterion* (BEC).
- L'approximation sur lequel BEC est fondée, *i.e.* l'équation (4.13), est valide à une constante près. Cela signifie qu'en général, l'erreur ne disparaît pas lorsque n tend vers l'infini. Aussi BEC est une approximation relativement grossière de $\log \boldsymbol{p}(\boldsymbol{y}|\boldsymbol{x}, m)$. Cependant, les termes dépendant de n domineront pour peu que suffisamment de données soient disponibles. Le critère BEC sera plus précis en pratique lorsque $\hat{\theta} \approx \tilde{\theta}$. Typiquement cela se produit quand la distribution jointe des données $(\boldsymbol{x}, \boldsymbol{y})$ est celle de l'un des modèles considérés. Mais, c'est rarement, le cas en pratique.
- Une approximation à la BIC de $\log \boldsymbol{p}(\boldsymbol{y}|\boldsymbol{x})$:

$$\log \boldsymbol{p}(\boldsymbol{y}|\boldsymbol{x}) \approx \log \boldsymbol{p}(\boldsymbol{y}|\boldsymbol{x}, \theta_m^*) - \frac{\nu_m}{2}\log n, \tag{4.15}$$

  où

$$\theta_m^* = \arg\max_{\theta_m} \boldsymbol{p}(\boldsymbol{y}|\boldsymbol{x}, \theta_m),$$

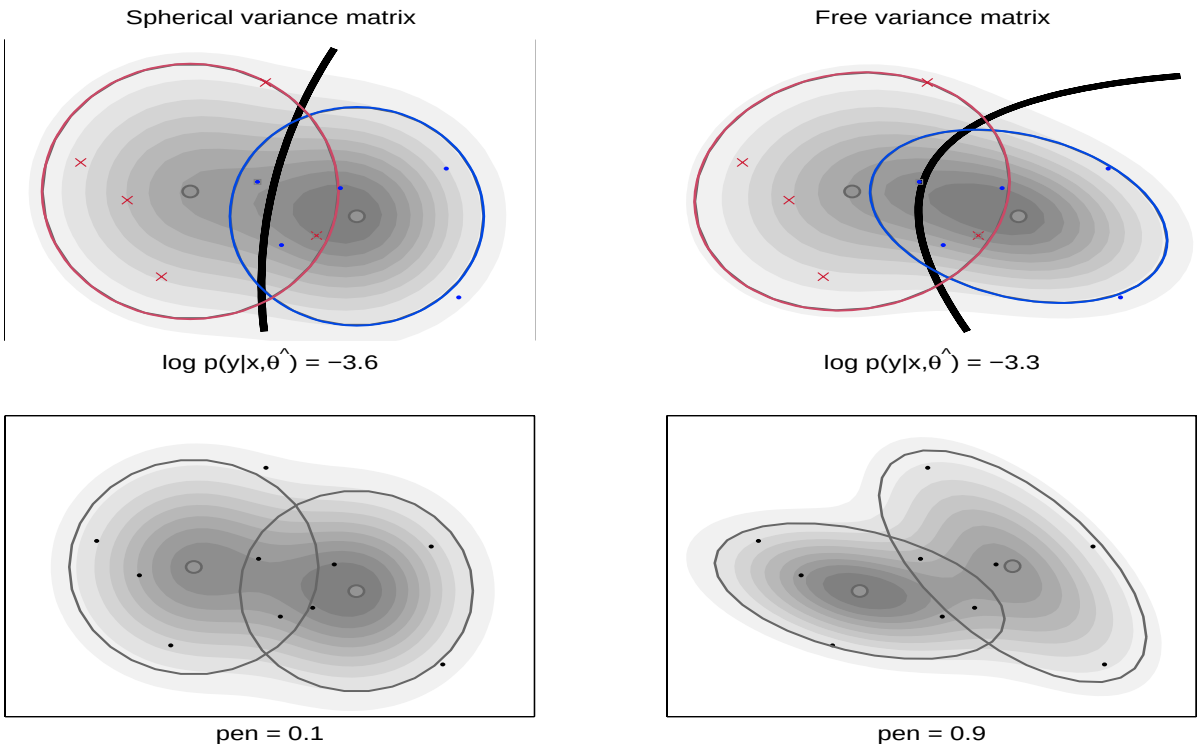


FIG. 4.1 – Une illustration du comportement de BEC. Un modèle gaussien sphérique (colonne de gauche) est comparé à un modèle gaussien avec matrices de variances libres (colonne de droite) pour un problème à deux classes. Dans la ligne du dessus, les estimateurs ML de la distribution jointe sont données, ainsi que la frontière de classification du classifieur correspondant. Les observations d'une classe sont notées par une croix, celles de l'autre par un point. Dans la ligne du bas, estimateurs ML des distributions marginales obtenues par EM sont données. Le niveau de gris est proportionnel à la densité. 'pen' est la valeur de la pénalité apparaissant dans (4.17). Le modèle fournissant la plus grande valeur de BEC, *i.e.* le modèle 1, a été retenu.

n'est pas valide car, pour tout modèle génératif, la distribution a posteriori $\pi(\theta_m|\boldsymbol{x})$ dans (4.7) dépend de n et ne peut pas être négligée. Cependant dans une approche discriminative de la classification supervisée où $\boldsymbol{x}$ ne dépendrait pas de θ, cette approximation serait valide.

– Le critère BEC nécessite le calcul de $\tilde{\theta} = \arg\max_\theta \boldsymbol{p}(\boldsymbol{x}|\theta_m)$. Comme, pour $i = 1, \ldots, n$,

$$\boldsymbol{p}(\boldsymbol{x}_i|\theta_m) = \sum_{k=1}^{K} \boldsymbol{p}(y_i = k)\boldsymbol{p}(\boldsymbol{x}_i|y_i = k, , \theta_m), \tag{4.16}$$

$\tilde{\theta}_m$ est l'estimateur ML du mélange fini. Il peut être obtenu par l'algorithme EM [115]. Heureusement, dans ce contexte, les défauts bien connus de EM comme la forte dépendance à sa position initiale et les situations de convergence lentes sont évitées. En effet, EM peut être initialisé de manière naturelle par $\hat{\theta}_m$. De la sorte, le calcul de $\tilde{\theta}_m$ est simple. Nous devons aussi noter que les proportions $\boldsymbol{p}(y_i = k), k = 1, \ldots, K$ ne dépendent pas de θ_m. Quand l'échantillon d'apprentissage a été obtenu comme un échantillon rétrospectif, il correspond au regroupement de K sous-échantillons dont la taille n'est pas aléatoire. Alors, les proportions du mélange (4.16) sont fixes : $\boldsymbol{p}(y_i = k) = n_k/n$ où $n_k = \text{card}\{i \text{ such that } y_i = k\}$ pour $k = 1, \ldots, K$. Quand l'échantillon d'apprentissage a été obtenu sous le schéma de mélange, les proportions du mélange doivent être estimées par EM. Mais, de nouveau, les proportions initiales peuvent être initialisées dans EM de manière naturelle par $\boldsymbol{p}(y_i = k) = n_k/n$ for $k = 1, \ldots, K$.

– Dans le but d'interpréter BEC comme un critère de vraisemblance pénalisé, on peut écrire :

$$\begin{aligned} \text{BEC} &= \log \boldsymbol{p}(\boldsymbol{x}, \boldsymbol{y}|\hat{\theta}_m) - \log \boldsymbol{p}(\boldsymbol{x}|\hat{\theta}_m) + \log \boldsymbol{p}(\boldsymbol{x}|\hat{\theta}_m) - \log \boldsymbol{p}(\boldsymbol{x}|\tilde{\theta}_m) \\ &= \log \boldsymbol{p}(\boldsymbol{y}|\boldsymbol{x}, \hat{\theta}_m) - \left(\log \boldsymbol{p}(\boldsymbol{x}|\tilde{\theta}_m) - \log \boldsymbol{p}(\boldsymbol{x}|\hat{\theta}_m)\right). \end{aligned} \tag{4.17}$$

La quantité $\log \boldsymbol{p}(\boldsymbol{x}|\tilde{\theta}_m) - \log \boldsymbol{p}(\boldsymbol{x}|\hat{\theta}_m)$ est positive car $\tilde{\theta}$ maximise la vraisemblance marginale $\boldsymbol{p}(\boldsymbol{x}|\theta_m)$. Elle peut s'interpréter comme une pénalité de complexité appliquée à la logvraisemblance conditionnelle. Cette pénalité est toujours positive et est minimum quand $\hat{\theta} = \tilde{\theta}$. Sa dépendance implicite avec la complexité du modèle est illustrée dans l'exemple jouet de la figure 4.1. Il s'agit d'un problème à deux classes. Dans l'ensemble d'apprentissage, cinq points proviennent de chaque classe, représentés par une croix et un point dans l'image du dessus de la figure 4.1. Deux modèle gaussiens sont considérés : un modèle "simple"avec des matrices variance sphériques et un "complexe" avec des matrices variances libres. BEC choisit le modèle le plus simple car la pénalité pour le modèle complexe surpasse l'augmentation obtenue pour la logvraisemblance conditionnelle $\log(\boldsymbol{p}(\boldsymbol{y}|\boldsymbol{x}, \hat{\theta})$.

– Enfin, de (4.17), on tire que BEC est toujours négatif puisque $\log \boldsymbol{p}(\boldsymbol{y}|\boldsymbol{x}, m, \hat{\theta}) \leq 0$.

4.4 Comportement asymptotique du critère BEC

Nous étudions ici quelques propriétés asymptotiques de BEC lorsque la taille de l'échantillon d'apprentissage tend vers l'infini et quand la distribution d'échantillonnage appartient au moins à l'un des modèles en compétition. Comme mentionnée ci-dessus, cette dernière hypothèse est irréaliste dans la plupart des situations, mais il s'agit ici de voir si BEC se comporte de manière cohérente dans une situation idéale. Le but de BEC est de trouver le modèle minimisant l'erreur de classification dans une collection de modèles. S'il existe un et un seul modèle m^* auquel la distribution d'échantillonnage appartient, on s'attend à ce que BEC le sélectionne puisque c'est l'unique modèle qui atteint asymptotiquement le taux d'erreur optimal de Bayes. Notons $\theta_m^0 = \operatorname{argmax}_{\theta\in\Theta} E\left[\boldsymbol{p}(\boldsymbol{X}, Y|\theta_m)\right]$ la limite asymptotique de l'estimateur ML. La proposition suivante prouve que BEC choisit asymptotiquement l'unique vrai modèle s'il existe.

Proposition 3 *Si la distribution d'échantillonnage jointe appartient à exactement un modèle m^* dans une famille finie de modèles candidats $\{m_1, \ldots, m_M\}$, et sous des conditions standard de régularité sur la famille de modèles en compétition, alors le critère BEC sélectionne m^* ou un modèle m' de même taux d'erreur attendu que m^*,* i.e. *:*

$$E\left[\log \boldsymbol{p}(Y|\boldsymbol{X}, \theta_{m'}^0)\right] = E\left[\log \boldsymbol{p}(Y|\boldsymbol{X}, \theta_{m^*}^0)\right]$$

presque sûrement quand n tend vers l'infini.

PREUVE. Si la distribution d'échantillonnage $\boldsymbol{p}$ est issue du modèle m^*, il existe $\theta_{m^*}^0$ vérifiant $\boldsymbol{p}(\boldsymbol{X}, Y) = \boldsymbol{p}(\boldsymbol{X}, Y|\theta_{m^*}^0)$. Le critère normalisé $\frac{1}{n}\text{BEC}(m^*)$ est la différence de $\frac{1}{n}\log \boldsymbol{p}(\boldsymbol{x}, \boldsymbol{y}|\hat{\theta}_{m^*})$ et de $\frac{1}{n}\log \boldsymbol{p}(\boldsymbol{x}|\tilde{\theta}_{m^*})$. Par la loi des grands nombres, $\hat{\theta}_{m^*} \to \theta_{m^*}^0$ et $\tilde{\theta}_{m^*} \to \theta_{m^*}^0$ quand $n \to \infty$. Alors, sous des conditions de régularité classiques, $\frac{1}{n}\log \boldsymbol{p}(\boldsymbol{x}, \boldsymbol{y}|\hat{\theta}_{m^*})$ and $\frac{1}{n}\log \boldsymbol{p}(\boldsymbol{x}|\tilde{\theta}_{m^*})$ tend ps vers $E\left[\log \boldsymbol{p}(\boldsymbol{X}, Y|\theta_{m^*}^0\right]$ et $E\left[\log \boldsymbol{p}(\boldsymbol{X}|\theta_{m^*}^0)\right]$, respectivement. Donc,

$$\frac{1}{n}\text{BEC}(m^*) \to E\left[\log \boldsymbol{p}(Y|\boldsymbol{X}, \theta_{m^*}^0)\right] \quad \text{as} \quad n \to \infty. \tag{4.18}$$

D'autre part, pour tout autre modèle $m \neq m^*$ ne contenant pas la distribution d'échantillonnage, $\hat{\theta}_m \to \theta_m^1$ et $\tilde{\theta}_m \to \theta_m^2$, de telle sorte que pour tout modèle $m \neq m^*$

$$\begin{aligned} \frac{1}{n}\text{BEC}(m) &\to E\left[\log \boldsymbol{p}(\boldsymbol{X}, Y|\theta_m^1)\right] - E\left[\log \boldsymbol{p}(\boldsymbol{X}|\theta_m^2)\right] && (4.19) \\ &= \underbrace{E\left[\log \boldsymbol{p}(Y|\boldsymbol{X}; \theta_m^1)\right]}_{\leq E\left[\log \boldsymbol{p}(Y|\boldsymbol{X}; \theta_{m^*}^0)\right]} - \underbrace{E\left[\log \boldsymbol{p}(\boldsymbol{X}|\theta_m^2) - \log \boldsymbol{p}(\boldsymbol{X}|\theta_m^1)\right]}_{\geq 0}. && (4.20) \end{aligned}$$

La première inégalité vient du fait que l'espérance de la logprobabilité est maximisée pour la vraie valeur $\theta^{\,0}_{m^*}$ du paramètre (ou de manière équivalente, la divergence de Kullback-Leibler est minimum en $\theta^{\,0}_{m^*}$). La seconde inégalité vient du fait que θ_m^2 maximise l'espérance de la vraisemblance marginale dans l'espace des paramètres du modèle m. L'égalité n'a lieu que si $E\left[\log \boldsymbol{p}(\boldsymbol{X}, Y|\theta_{m^*}^0\right]$ et $E\left[\log \boldsymbol{p}(\boldsymbol{X}|\theta_m^1)\right]$, *i.e.* l'espérance des taux d'erreur des modèles m et m^* sont égaux. □

La proposition 3 ne s'applique pas pour des modèles emboîtés. Dans ce cas, la vraie distribution peut appartenir à plusieurs modèles candidats.

Proposition 4 *Supposons que la vraie distribution $\boldsymbol{p}(\boldsymbol{X}, Y)$ appartient à deux modèles emboîtés m et m', avec ν et ν' paramètres, quel que soit $\varepsilon > 0$ pour n suffisamment grand, nous avons*

$$E(\textit{BEC}(m)) - E(\textit{BEC}(m')) < \varepsilon.$$

PREUVE. Supposons que $\nu' > \nu$. La statistique du rapport de vraisemblance des deux modèles emboîtés est asymptotiquement une distribution du χ^2 à $\delta_\nu = \nu' - \nu$ degrés de liberté. Lorsque nous calculons la différence du critère BEC pour les deux modèles m et m', une statistique de rapport de vraisemblance apparaît

$$\begin{aligned} \text{BEC}(m) - \text{BEC}(m') &= \log \boldsymbol{p}(\boldsymbol{x}, \boldsymbol{y}|\hat{\theta}_m) - \log \boldsymbol{p}(\boldsymbol{x}, \boldsymbol{y}|\hat{\theta}_{m'}) - \left(\log \boldsymbol{p}(\boldsymbol{x}; \tilde{\theta}_m) - \log \boldsymbol{p}(\boldsymbol{x}; \tilde{\theta}_{m'})\right) \\ &= \log\{\text{LR of } m \text{ vs. } m' \text{ for } \boldsymbol{p}(X, Y)\} - \log\{\text{LR of } m \text{ vs. } m' \text{ for } \boldsymbol{p}(X)\} \\ &\xrightarrow{\mathcal{D}} \frac{1}{2}\chi_{\delta_\nu}^{'2} - \frac{1}{2}\chi_{\delta_\nu}^2 \qquad (4.21) \end{aligned}$$

où $\chi_{\delta_\nu}^{'2}$ et $\chi_{\delta_\nu}^2$ sont deux variables dépendantes suivant une loi du χ^2 à δ_ν degrés de liberté. La dernière approximation est valide pour n suffisamment grand et est $O_p(1)$. la prouve que la variable aléatoire $\text{BEC}(m) - \text{BEC}(m')$ est asymptotiquement de moyenne nulle.

□

Cela signifie que le critère BEC pondère les deux modèles équitablement. Ainsi, même asymptotiquement, on peut trouver BEC$(m') < BEC(m)$ et en conséquence choisir le modèle le plus complexe. En pratique, cela arrivera rarement car cela exige que deux conditions, un échantillon de grande taille et une collection de modèles approximant très bien la distribution d'échantillonnage. Mais, quand des modèles emboîtés sont en compétition, il est utile de tracer les variations de BEC en fonction du nombre de paramètres des modèles. Si un plateau apparaît sur ce graphe, cela signifie que la collection de modèles s'ajuste bien aux données. Dans un tel cas, nous recommandons de choisir le modèle le plus simple sur ce plateau. Ce type de comportement de BEC est illustré au paragraphe 4.5.1, figure 4.2.

Cela dit, ces considérations théoriques sont asymptotiques et ne renseignent pas sur le comportement à taille d'échantillon finie. Le paragraphe suivant étudie cette question à partir d'expérimentations numériques sur des données simulées et des données réelles.

4.5 Expériences numériques

Nous reportons maintenant les résultats d'études de cas analysant la capacité pratique de BEC à sélectionner un modèle de classification approprié. BEC est comparé à d'autres critères tels que la validation croisée, AIC et BIC. Tout d'abord, des expériences de Monte-Carlo sont présentés dans des situations simples pour mettre en lumière les caractéristiques notables de BIC. Puis, nous considérons le problème de sélectionner un modèle pertinent dans le contexte de *Eigenvalue Decomposition Discriminant Analysis* (EDDA) [15] et de *Mixture Discriminant Analysis* (MDA) [70]. par des expériences de Monte-Carlo sur des ensembles de données de référence. Enfin, une étude concernant un problème de reconnaissance des formes en analyse d'images est présenté.

4.5.1 Simulations de Monte-Carlo

Tout d'abord, nous comparons deux modèles. Cinq cent échantillons de $n = 120$ observations dans $\boldsymbol{R}^2$ provenant de deux classes de même proportion ont été simulés avec les densités par classe suivantes :

$$X|Y = 1 \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 2 & 0.5 \\ 0.5 & 1 \end{bmatrix}\right)$$

séparation	modèle	$\overline{err}$	$-$BIC	$-$BEC	BIC choisi(%)	BEC choisi(%)
$\Delta = 1$	DIAG	**0.250**	502.331	**64.108**	24	98
$\Delta = 1$	SPHE	0.268	**500.422**	69.665	76	2
$\Delta = 3.5$	DIAG	**0.070**	502.331	**22.067**	24	94
$\Delta = 3.5$	SPHE	0.076	**500.422**	26.120	76	6
$\Delta = 5$	DIAG	**0.019**	502.331	**6.081**	24	84
$\Delta = 5$	SPHE	0.023	**500.422**	8.310	76	16
$\Delta = 7$	DIAG	**0.002**	502.331	**0.458**	24	80
$\Delta = 7$	SPHE	0.004	**500.422**	1.046	76	20
$\Delta = 10$	DIAG	**0.000**	502.331	**0.001**	24	60
$\Delta = 10$	SPHE	0.000	**500.422**	0.002	76	40

TAB. 4.1 – Comparaison des critères BEC et BIC pour choisir entre les modèles DIAG et SPHE. La colonne $\overline{err}$ donne le taux d'erreur moyen évalué sur un échantillon test independant de taille 50 000. Les moyennes sont calculés à partir de 500 échantillons.

et

$$X|Y = 2 \sim \mathcal{N}\left(\begin{bmatrix} \Delta \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 0.5 \\ 0.5 & 2 \end{bmatrix}\right).$$

Les deux modèles comparés sont des distributions gaussiennes avec des matrices variances diagonales pour chaque classe (DIAG) et des matrices variances proportionnelles à la matrice identité pour chaque classe (SPHE). Les performances des critères BEC et BIC sont comparées dans le tableau 4.1. Dans ce tableau, la colonne $\overline{err}$ donne le taux d'erreur obtenu avec un échantillon test indépendant de taille 50 000. BEC choisit le modèle de taux d'erreur minimum avec une plus grande fréquence que BIC. Ce dernier critère sélectionne souvent le modèle sphérique car il est meilleur en termes d'estimation de la densité. Quand la séparation des classes augmente, BEC a lui tendance à choisir le modèle le plus parcimonieux plus souvent comme espéré.

La deuxième simulation de Monte-Carlo illustrate la possibilité que BEC, considéré comme fonction de la complexité, produise un plateau pour des modèles emboîtés. On considère pour cela un problème à deux classes

FIG. 4.2 – Une illustration du comportement typique de BEC pour des modèles emboîtés. Pour chaque vignette, la courbe en trait plein donne les variations de -BEC (échelle de gauche) et la courbe en tirets donne les variations du taux d'erreur (échelle de droite) pour un ensemble test de taille 50 000.

dans $\boldsymbol{R}^2$. Deux ensembles de données de taille $n = 300$ sont considérés. POur le premier, chaque densité par classe est gaussienne avec la matrice de variance $\begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}$ et des moyennes par classe $(0,0)^t$ et $(1.5,0)^t$. Pour le deuxième, chaque densité par classe est un mélange de trois lois gaussiennes avec des matrices variances égales à l'identité et des moyennes par composant $(0,0)^t, (3,0)^t$ et $(0,3)^t$ pour la classe 1, et $(1.5,0)^t, (-1.5,0)^t$ and $(1.5,-3)^t$ pour la classe 2. Les modèles comparés sont, pour chaque classe, un mélange de lois gaussiennes sphériques de volume égal dont le nombre de composants varie entre un et huit. La figure 4.2 donne les variations de BEC en fonction du nombre de composants gaussiens pour chaque ensemble de données. Dans cette figure les variations de BEC sont en trait plein. La figure 4.2 montre aussi les variations du taux d'erreur (tirets), calculées sur un échantillon test de taille 50 000. Comme attendu, le dessin de droite fait apparaître un plateau à partir de la vraie valeur du nombre de composants du mélange. La règle préconisée au paragraphe 4.4 conduit à choisir le bon nombre de composants, à savoir trois, qui produit le plus petit taux d'erreur. POur le premier jeu de données, un tel plateau n'apparaît pas et BEC choisit un mélange avec un nombre moindre de composants. Ce n'est pas celui qui produit le plus petit taux d'erreur, mais dans ce cas il y a peu de différence sur les taux d'erreur en fonction du nombre de composants du mélange comme l'indique la courbe en tirets.

4.5.2 Choix de la paramétrisation d'une matrice variance

L'analyse discriminante linéaire (LDA) et l'analyse discriminante quadratique (QDA) sont des modèles génératifs de classification répandus. Ces deux méthodes supposent des densités par classe gaussiennes avec une matrice variance commune pour LDA et des matrices variances libres pour QDA. La considération de la décomposition spectrale des matrices variances par classe conduit à des nombreux modèles alternatifs [15]. Soit $\Sigma_k = L_k D_k A_k D_k$ la décomposition de la matrice variance de la classe k, où $L_k = |\Sigma_k|^{1/d}$ définit le volume de la classe, D_k (la matrice des vecteurs propres de Σ_k) définit son orientation, et A_k (la matrice diagonale des

valeurs propres normalisées Σ_k), définit sa forme. Autoriser ou non ces quantités à varier selon les classes conduit à plusieurs modèles intéressants pour la classification. De plus, supposer que Σ_k est diagonale ou proportionnelle à la matrice identité conduit à d'autres modèles parcimonieux.

Dans [15], 14 modèles différents fondés sur cette décomposition spectrale sont proposés et le modèle minimisant le taux d'erreur évalué par validation croisée est sélectionné. Cette méthode est appellée EDDA– *Eigenvalue Decomposition Discriminant Analysis*. Ici, nous analysons la possibilité de sélectionner l'un des modèles à l'aide des critères AIC, BEC ou BIC au lieu d'utiliser le taux d'erreur évalué par validation croisée. Pour des raisons de simplicité, les deux cas avec une même orientation et des formes différentes ne sont pas inclus dans l'étude car ces cas nécessitent l'emploi d'un algorithme assez lourd pour estimer les paramètres associés. Il s'agit en effet ici d'évaluer la capacité des critères à sélectionner un bon modèle et non d'évaluer les performances de EDDA.

Pour cette étude, des données répertoriées dans le *UCI Machine Learning Database Repository* (disponible à `http://www.ics.uci.edu/~mlearn/`) ont été utilisées. Comme la réduction de dimension améliore souvent les performances des classifieurs, les expériences ont été faites dans l'espace de dimension quatre généré par les $K-1$ axes de l'analyse discriminante factorielle, complété si nécessaire par les axes de l'ACP opérée sur le sous-espace orthogonal à l'espace discriminant. (Pour chaque jeu de données, la dimension originale d est donnée dans le tableau 4.3.)

Tout d'abord, le comportement de BEC est illustré en detail sur le jeu *Australian Credit Approval*. Cet ensemble de données contient des variables discrètes et des variables continues. Les variables ordinales ont été considérées continues et les variables binaires ont été supprimées pour éviter les problèmes numériques.

La procédure suivante a été répétée 100 fois : pour chaque expérimentation, 200 observations parmi les) ont été sélectionnées aléatoirement pour constituer l'ensemble d'apprentissage pour estimer les 12 modèles et calculer les valeurs des critères BIC, AIC, BEC, et le taux d'erreur évalué par validation croisée procédure *3-fold* (CV3). Le modèle optimisant chaque critère a été sélectionné et ses performances sont évaluées sur l'ensemble test. Les proportions de choix des différents modèles sont données dans le tableau 4.2. Il en ressort que BEC choisit un modèle satisfaisant. De plus, BEC et CV3 ont un comportement similaire. En revanche, comme il arrive souvent avec d'autres jeux de données, BIC choisit des modèles notablement sous-optimaux en termes de taux d'erreur. BIC sélectionne un modèle ayant un taux d'erreur de 27.5% la plupart du temps, loin du minimum de 22.9%, souvent

modèle	ν	BIC	AIC	BEC	CV3	test error
λI	10	0	0	0	0	0.293
$\lambda_k I$	13	0	0	0	0	0.289
λB	13	0	9	32	28	0.23
$\lambda_k B$	14	0	0	1	1	0.264
λB_k	16	0	0	0	0	0.287
$\lambda_k B_k$	17	93	0	0	0	0.276
$\lambda D^t A D$	19	0	23	38	36	**0.229**
$\lambda_k D^t A D$	20	0	0	0	1	0.261
$\lambda D_k^t A D_k$	25	0	68	25	34	0.23
$\lambda_k D_k^t A D_k$	26	0	0	3	0	0.258
$\lambda D_k^t A_k D_k$	28	0	0	0	0	0.291
$\lambda_k D_k^t A_k D_k$	29	7	0	1	0	0.274

TAB. 4.2 – Comparaison des différents critères de sélection de modèles sur les données *Australian credit*, avec deux classes, un ensemble d'apprentissage de taille 200 et un ensemble test de taille 490 ; Dix variables continues ramenées à quatre dimensions. Chaque nombre représente la proportion de modèles choisis parmi les 12 modèles proposés. Dans la première colonne, I indique la matrice identité, B une matrice diagonale et la présence d'un indice k indique que l'élement associé peut varier selon les classes.

obtenu avec CV3 et BEC. BEC et CV3 hésitent entre les trois modèles de plus petit taux d'erreur. (AIC a un comportement analogue, malgré une légère tendance à préférer le plus complexe des trois modèles.) Ces résultats suggèrent que BEC est aussi intéressant dans une optique de combinaison de méthodes où chaque fonction de décision est pondérée selon les probabilités a posteriori des modèles en compétition (voir par exemple [75].)

Les autres données utilisées pour évaluer les performances des critères AIC, BEC, BIC et CV3 sont *Abalone* (classification entre males, femelles et enfants), *Bupa* (maladies du foie), *Haberman* (données de survie), *Page-blocks* (classification de documents), *teaching* (évaluation pédagogique), *Diabetes* (détection de diabète chez les indiens Pima), *German* (risque dedéfaut de crédit), *Heart* (risque d'attaque cardiaque). Une description complète

Dataset	K	N	d	BIC	AIC	BEC	CV3	oracle
Abalone	3	4177	7	47.3	47.4	46.1	**45.9**	45.4
Bupa	2	345	6	37.5	38.3	**33.5**	34.6	31.6
Haberman	2	306	3	25.0	25.0	25.1	**24.9**	23.7
Pageblocks	5	5473	10	4.4	4.4	**2.8**	**2.8**	2.5
Teaching	3	151	5	63.8	63.3	63.8	**61.1**	56.9
Australian	2	690	14	26.3	26.4	**22.6**	22.8	21.9
Diabetes	2	768	8	26.0	25.6	**23.9**	24.2	23.0
German	2	1000	20	25.3	25.4	25.1	**24.9**	24.0
Heart	2	270	10	17.5	18.3	17.6	**17.3**	15.6

TAB. 4.3 – Les taux d'erreur test des classifieurs choisis par les quatre critères. Ces taux sont moyennés sur 100 découpages aléatoires appentissage/test. K est le nombre de classes, N le nombre total d'échantillons et d est la dimension de l'espace des descripteurs.

de ces jeux de données se trouve dans le répertoire UCI. Pour nos expériences, les données binaires ont été retirées de *Abalone*, *Australian* et *Heart*.

Les expériences menées sont analogues à celles du jeu *Australian credit*. Mais, pour chaque découpage apprentissage/test effectué, le taux d'erreur de chaque modèle choisi par chaque critère a été sauvé. Le taux d'erreur moyen est donné dans le tableau 4.3 pour chaque jeu et chaque critère. Nous donnons aussi les performances de l'*oracle*, à savoir le taux d'erreur obtenu si nous avions choisi pour chaque cas le modèle fournissant l'erreur test minimum parmi tous ceux obtenus avec les quatre critères considérés.

Ces expériences montrent que BEC fournit des taux d'erreur plus faibles que AIC et BIC. BEC et BIC approximent des vraisemblances intégrées des modèles considérés, mais comme BIC ne tient pas compte des performances de classification, il s'avère qu'il peut choisir un modèle sous-boptimal du point de vue de la prédiction. C'est visible ici pour les jeux *Bupa, pageblocks, Australian and diabetes* en termes de taux d'erreur où la différence entre BIC, d'une part et BEC et CV3 d'autre part dépasse 2%. Sinon, sauf pour *Teaching dataset* pour lequel EDDA s'avère mauvais, les performances de BEC et de CV3 sont très proches. Cela invite à penser que BEC est une

alternative intéressante à la validation croisée pour évaluer l'erreur de classement de différents modèles de classification. Cependant, ces expériences sur des données de référence demeurent superficielles. Dans la suite, nous presentons les performances de BEC dans un contexte plus réaliste.

4.5.3 Choix du nombre de composants des mélanges dans MDA

Nous considérons maintenant le problème de sélectio de modèle qui se pose pour les classifieurs MDA [70]. Nous nous restreignons à des mélanges gaussiens sphériques qui représentent une famille de modèles attractive par sa simplicité et sa flexibilité [21]. Les densités par classe s'écrivent $p_k(\boldsymbol{x}|\theta_k) = \sum_{r=1}^{R_k} \pi_r \phi(\boldsymbol{x}|\boldsymbol{\mu}_r, \sigma_r^2 I_d)$ où R_k, $k = 1, \cdots, K$ désigne le nombre de composants du mélange pour chaque classe, et π_r, μ_r et σ_r sont respectivement la proportion, la moyenne et l'écart-type du r^{me} composant avec $\phi(\boldsymbol{x}; \boldsymbol{\mu}, \Sigma)$ qui représente la densité d'une distribution gaussienne de moyenne μ et de matrice variance Σ. L'ensemble des paramètres de la classe k est

$$\theta_k = (\pi_1, \ldots, \pi_{R_k - 1}, \mu_1, \ldots, \mu_{R_k}, \sigma_1, \ldots, \sigma_{R_k}).$$

Bien sûr, la sélection du nombre de composants du mélange $\{R_k\}_{k=1,\cdots,K}$ est importante pour obtenir de bons résultats avec cette méthode. L'évaluation par validation croisée du taux d'erreur est particulièrement chère dans ce contexte du fait que les nombres de modèles à tester est considérable. De ce point de vue, BIC peut être utile pour contourner la difficulté. Mais supposer que les densités par classes sont des mélanges de gaussiennes sphériques est une hypothèse assez grossière. Aussi, on peut craindre que BIC soit décevant puisqu'il s'attache avant tout à mesurer la qualité du modèle de mélange sphérique à l'échantillon d'apprentissage plutôt que sa capacité à produire un classifieur efficace. En pratique, BIC s'avère effectivement décevant pour sélectionner un nombre pertinent de composants des mélanges par classes [21]. Pour illustrer ce fait, nous présentons tout d'abord une petite étude de Monte-Carlo avant de détailler une application de MDA à un problème de reconnaissance des formes en analyse d'images.

Une étude de Monte-Carlo

Nous avons considéré la même distribution d'échantillonnage utilisée pour comparer une matrice variance diagonale contre un matrice variance proportionnelle à l'identité au paragraphe 4.5.1. Le modèle considéré ici est le mélange de gaussiennes sphériques présenté ci-dessus, et le problème est de sélectionner le nombre de

séparation	modèle	$\overline{err}$	-BIC	-BEC	BIC choisi(%)	BEC choisi(%)
$\Delta = 1$	1 components	0.266	1.6e+03	114	29	0
$\Delta = 1$	2 components	**0.25**	**1.6e+03**	98.5	70	12
$\Delta = 1$	3 components	0.251	1.62e+03	**94.6**	1	88
$\Delta = 3.5$	1 components	0.0748	1.6e+03	43.1	29	0
$\Delta = 3.5$	2 components	0.0625	**1.6e+03**	31.2	70	18
$\Delta = 3.5$	3 components	**0.0617**	1.62e+03	**28.8**	1	82
$\Delta = 5$	1 components	0.0227	1.6e+03	14.1	29	0
$\Delta = 5$	2 components	0.0151	**1.6e+03**	8.48	70	18.5
$\Delta = 5$	3 components	**0.0149**	1.62e+03	**7.16**	1	81.5
$\Delta = 7$	1 components	0.00357	1.6e+03	2.1	29	3.5
$\Delta = 7$	2 components	0.00174	**1.6e+03**	0.9	70	28
$\Delta = 7$	3 components	**0.0015**	1.62e+03	**0.702**	1	68.5
$\Delta = 10$	1 components	8.55e-05	1.6e+03	0.121	29	88
$\Delta = 10$	2 components	**1.8e-05**	**1.6e+03**	**0.0187**	70	5.5
$\Delta = 10$	3 components	**1.8e-05**	1.62e+03	0.033	1	6.5

TAB. 4.4 – Une comparaison des critères BEC et BIC dans le choix du nombre de composants d'un modèle de mélange sphérique. La colonne $\overline{err}$ donne le taux d'erreur évalué sur un ensemble test de taille 50 000. Les valeurs reproduites ont été calculés sur 500 replications.

composants $R_k, k = 1, 2$. Pour des raisons de simplicité, nous supposons que $R_1 = R_2$. Le comportement des critères BEC et BIC sont comparés dans le tableau4.4. On peut voir que BEC choisit une complexité bien adapté au problème de classification. Par exemple, pour des classes très séparés ($\Delta = 10$), les taux d'erreur des différents modèles sont identiques et choisit le plus souvent le plus simple. De son côté, BIC choisit toujours le même modèle sans tenir compte de la séparation entre les classes.

Un exemple de sélection de modèle en analyse d'images

La catégorisation d'objets vise à classer des objets ayant des attributs en commun. Nous nous intéressons ici au problème de trouver des images contenant une moto. C'est un exemple typique de catégorisation d'objets car différentes sortes de motos existent. Le problème est d'apprendre à tirer des caractéristiques générales d'un type d'objet particulier. Les motos et les données d'arrière-plan considérées [15] ici ont été originellement étudiées par [45], et plusieurs auteurs ont comparé des méthodes de catégorisation d'objets se fondant sur la détection de points d'intérêt sur ces données [129, 31, 37]. Ces données contiennent respectivement 826 et 900 images. La moitié de chaque ensemble de données a été sélectionné au hasard pour construire le classifieur, l'autre moitié étant utilisée comme ensemble test.

Pour classifier ces données, une méthode simple et rapide de type « bag of features » [31] a été utilisée. Cette méthode est basée sur la *quantization* de descripteurs locaux d'images. [16] Pour chaque image, des descripteurs k-dimensionnels invariants par échelle sont calculés (k étant le nombre de vecteurs quantizés) et sont utilisés en entrée d'une méthode de classification. Nous détaillons brièvement de quelle manière ces vecteurs sont générés et nous nous focalisons sur le classifieur génératif et le problème de sélection de modèle.

– Les images sont mises à la même échelle (au maximum 320×160 pixels), en préservant le rapport hauteur/largeur initial.

– Un détecteur de points d'intérêts de Harris-Laplace invariant par échelle extrait m points de l'image $\mathcal{I}$. En fonction de la complexité de l'image, entre 100 et 300 points sont détectés. Pour chacun de ces points, un vecteur de dimension 128 est calculé à partir des pixels du voisinage. Ce *descripteur d'image* est appelé SIFT (*Scale Invariant Feature Tranform*). Il code l'apparence visuell autour du point d'intérêt.

– Pour chaque image, l'ensemble des vecteurs d'apparence est quantizé en un vecteur de 1000 dimensions en classant les veteurs d'apparence en 1000 clusters différents : une affectation floue des vecteurs au centre le plus proche est calculée en utilisant une distribution gaussienne de variance 0.36. Ensuite, chaque cluster est associé à un vecteur de la manière suivante : La valeur donnée à chacune des 1000 coordonnées est la

[15] disponibles sur `http://www.vision.caltech.edu/html-files/archive.html`

[16] La quantization est une discrétisation de l'espace qui consiste à approcher un vecteur par la valeur du vecteur "référence" le plus proche. La détermination des vecteurs de référence est souvent effectué par clustering sur un grand nombre d'exemples. Cette méthode provient de la compression d'image (les couleurs sont remplacées par des couleurs de référence) et a été popularisée par Kohonen [95].

probabilité maximale d'une affectation au cluster correspondant. Les centres utilisés pour la quantization sont estimés par k-means sur l'ensemble des apparences des images d'apprentissage. de manière à ce que les clusters couvrent à peu près toutes les apparences.

IL faut noter que la dimension est plus grande que la taille de l'ensemble d'apprentissage. Certaines étude ont montré que des classifieurs discriminatifs régularisés sont bien adaptés à cette situation $[6,\ 129,\ 31]$, et ici, la question est d'analyser si un classifieur génératif modélisant la distribution jointe des données $(\boldsymbol{x}, \boldsymbol{y})$ peut obtenir des performances similaires. Dans le cas présent, nous avons utilisé un classifieur génératif modélisant les densités pas classe par un mélange de lois gaussiennes ayant une matrice variance diagonale. Ce type de modèle peut sávérer bien adapté pour les données considérées car plusieurs groupes de motos peuvent être présents, et les images d'arrière-plan sont censées être associées à beaucoup de catégories différentes conduisant à une distribution multimodale.

Le problème important à résoudre ici est de trouver le nombre adéquat de composants du mélange pour décrire chaque classe à discriminer. Aucune information a priori n'est disponible pour aider à répondre à cette question. Ainsi, le problème à résoudre est un problème de sélection de modèle. Les mélanges estimés avaient de une à cinq composants pour la classe des images de motos et de un à sept composants pour les images d'arrière-plan. Les critères -BEC, -BIC et CV10 (*10-fold* validation croisée) ont été calculés sur l'échantillon d'apprentissage et le taux d'erreur a été évalué sur l'ensemble test. Le tableau 4.5 donne les valeurs des differents critères pour tous les modèles possibles.

Comparé à d'autres études sur ces données, le taux d'erreur de 3.84% apparaît compétitif. Quelques exemples d'images mal classées sont donnés dans la figure 4.3. BIC tend clairement à sélectionner un modèle trop simple avec $R_1 = 3$ et $R_2 = 3$ composants. BEC est minimisé pour le même modèle que l'erreur sur l'ensemble test avec $R_1 = 3$ et $R_2 = 6$ composants. De plus, pour les autres complexités, les valeurs de BEC reproduisant un comportement très semblable à celui de l'erreur sur l'ensemble test. Cela illustre le fait que BEC pénalise la logvraisemblance conditionnelle de manière adéquate pour obtenir des performances quasi optimales. Maintenant, pour 20 découpages différents entre ensembles d'apprentissage et ensemble test, nous avons calculé l'amélioration realtive de BEC par rapport à BIC par

$$\alpha = \frac{e\bar{r}r_{\text{BIC}} - e\bar{r}r_{\text{BEC}}}{\min_{\{R_1,R_2\}} e\bar{r}r_{\text{test}}},$$

$-$-BIC ($\times 10^5$)

	R_1				
R_2	1	2	3	4	5
1	-9.111	-9.227	-9.255	-9.263	-9.264
2	-9.260	-9.257	-9.126	-9.243	-9.271
3	-9.279	**-9.281**	-9.275	-9.273	-9.126
4	-9.242	-9.270	-9.278	-9.279	-9.275
5	-9.272	-9.122	-9.239	-9.267	-9.275
6	-9.276	-9.271	-9.269	-9.115	-9.231
7	-9.259	-9.267	-9.268	-9.264	-9.261

$-$BEC ($\times 10^3$)

	R_1				
R_2	1	2	3	4	5
1	3.06	1.18	0.91	0.75	0.63
2	1.35	1.27	6.24	1.09	0.75
3	0.51	0.46	0.46	0.39	6.93
4	1.99	0.80	0.52	0.48	0.37
5	0.32	7.95	2.35	0.80	0.53
6	0.45	0.34	**0.29**	8.57	2.44
7	0.91	0.58	0.51	0.38	0.32

CV10 error rate ($\times 100$)

	R_1				
R_2	1	2	3	4	5
1	7.19	9.04	6.61	4.98	6.95
2	6.95	7.42	9.04	7.18	6.61
3	6.26	5.91	4.98	4.75	9.62
4	7.42	6.61	5.79	5.45	4.87
5	4.75	9.85	6.84	5.79	5.68
6	5.33	4.29	**4.09**	11.47	6.61
7	5.91	6.14	5.79	4.72	4.72

Test error rate ($\times 100$)

	R_1				
R_2	1	2	3	4	5
1	6.26	8.34	5.56	6.49	4.85
2	5.56	5.10	7.76	6.72	5.91
3	5.91	5.33	5.56	4.87	8.69
4	6.95	5.68	5.56	5.21	5.21
5	4.98	9.50	6.84	5.45	5.91
6	4.87	4.52	**3.84**	10.08	6.84
7	5.33	6.03	4.75	4.85	4.59

TAB. 4.5 – Les valeurs des différents critères pour les modèles de mélange avec R_1 composants pour les images de moto et R_2 composants pour les images d'arrière-plan. Le dernier tableau donne le taux d'erreur calculé sur un ensemble test indépendant issue de la base des 863 images. Les taux d'erreur calculés supposent que les images des deux classes ont les mêmes probabilités a priori.

où $e\bar{r}r_c$ représente l'erreur test pour un critère donné c. La valeur moyenne de α est de 27.7 et l'intervalle de confiance à 95% est $[16.5, 38.9]$. Cela signifie que choisir un modèle avec BEC plutôt qu'avec BIC améliore en

moyenne les performances du classifieur de 27.7%. Une comparaison analogue entre CV10 et BEC donne un intervalle de confiance de $[-10.7, 8.8]$, ce qui signifie que les deux critères ont des performances analogues.

Enfin, sur les mêmes découpages entre ensembles d'apprentissage et de test, nous avons comparé le classifieur sélectionné avec BEC avec une approche discriminative pour voir si les modèles retenus sont compétitifs dans ce contexte de grande dimension. Un classifieur SVM avec noyau gaussien doone un taux d'erreur de 3.46% sur les ensembles tests, la largeur du noyau et le "slack-variable coefficient" ayant éte choisi par validation croisée (10-fold CV). Le classifieur génératif choisi avec BEC donne lieu un taux d'erreur de 3.97% sur les ensembles tests. Du fait que les approches purement discriminatives sont les méthodes de réference dans de tels contextes [31], le petit déficit de performance est acceptable car il ouvre la porte à des modèles génératifs plus sophistiqués, ajoutant par exemple des informations telles que la localisation et la taille des descripteurs. De telles extensions s'avèrent beaucoup plus difficiles à introduire pour les approches discriminatives (see for instance [22]).

FIG. 4.3 – Exemples d'images mal classées avec les détecteurs de Harris invariants par transformation d'échelle.

4.6 Discussion

Nous avons proposé un critère de sélection de modèle, BEC, qui prend en compte l'objectif de classification pour sélectionner un modèle génératif en classification supervisée. Il procure une alternative efficace à la validation croisée lorsque la collection de modèles en compétition est grande. Le critère BEC peut être vu comme un critère à la BIC pour approximer l'entrpie de classification d'un modèle génératif. Il sélectionne fréquemment des modèles produisant un taux d'erreur significativement plus petit que ceux obtenus avec BIC.

Chapitre 5

Entre l'estimation générative et discriminative

Nous avons vu dans le chapitre précédent qu'un critère de sélection de modèle basé sur l'entropie de classification $\log p(\mathbf{y}|\mathbf{x})$ s'avère efficace pour sélectionner un classifieur génératif performant. Nous avons aussi remarqué que les modèles probabilistes sont en général inexacts, ce qui ce traduit par un biais pénalisant des performances de classification. En principe, l'estimation discriminative, c'est-à-dire la recherche des paramètres maximisant l'entropie de classification permet de réduire ce biais, mais au dépend d'une augmentation de la variance des estimateurs.

Depuis une trentaine d'années, les chercheurs hésitent entre les estimateurs génératifs et discriminatifs. B. Efron, en 1975, [42], compare les deux méthodes linéaires de référence en classification supervisée, à savoir l'analyse discriminante et la régression logistique linéraire. De manière équivalente, cette étude peut être utilisée pour comparer les estimation jointes et conditionnelles du modèle d'analyse discriminante linéaire (voir chapitre 2). Le résultat majeur de Efron est que la régression logistique nécessite 30% de données supplémentaires lorsque la distribution des classe est gaussienne, ce qui est un argument non négligeable en faveur des classifieurs génératifs.

Ainsi, les arguments en faveur de l'estimation générative se basent sur l'exactitude du modèle de densité jointe des entrées et des sorties, alors que les défenseurs de l'estimation discriminative se basent sur la politique du « pire

cas » en considérant que la distribution des données est en contradiction avec les hypothèses du modèle. D'un point de vue expérimental, les études comparatives ne donnent pas clairement une supériorité à l'une ou l'autre des méthodes d'estimation. Ng et Jordan (2002) montrent que le choix entre le classifieur de Bayes naïf et la régression logistique linéaire dépend de la taille de l'échantillon d'apprentissage [127]. Greiner et al. (2002) [64] proposent une étude expérimentale complète basée sur l'estimation générative et discriminative des réseaux bayésiens. Ils insistent sur l'amélioration des performances liées à l'estimation générative, alors que plus du quart des expériences montrent que l'estimation générative donne un taux d'erreur plus faible. Ainsi privilégier systématiquement l'un ou l'autre type d'estimation ne semble pas être un bon point de vue et il semble nécessaire de choisir entre les deux par validation croisée.

Plutôt que de se poser la question entre la maximisation de la vraisemblance jointe ou la vraisemblance conditionnelle, nous proposons un juste milieu entre ces deux extrêmes en pondérant ces deux vraisemblances. Ainsi, le modèle appris est « partagé » entre une bonne estimation de la densité et de bonnes performances en classification. Ce type d'estimation, dénoté GDT (Generative-Discriminative Tradeoff), est original et ne correspond pas aux estimateurs classiques basés sur une vraisemblance pénalisée ou un *a posteriori* bayésien, car les deux termes de la fonction objectif dépendent des données d'apprentissage.

La justification théorique de la validité de la méthode GDT s'inspire des travaux de T. O'Neil (1980) [128], dans lesquels il étend de manière significative l'idée de Efron en dérivant la distribution asymptotique du taux d'erreur d'un classifieur génératif ([128], théorème 1). L'originalité de ce travail repose dans l'utilisation d'une paramétrisation commune pour les différents estimateurs, en définissant une transformation entre les paramètres de la densité jointe $\boldsymbol{p}(x, y)$ et ceux de la distribution conditionelle $\boldsymbol{p}(y|x)$. Les différentes formes de régression logistiques ne sont justifiées dans cet article qu'après une modélisation explicite de la densité des classes. Nous adoptons le même point de vue en utilisant un modèle génératif, puis en considérant les distributions asymptotiques des estimateurs génératifs, discriminatifs et GDT. A la différence de T. O'Neil, nous utiliserons la paramétrisation de la densité jointe au lieu de la conditionnelle. Dans le cas particulier des distributions gaussiennes, cette paramétrisation n'est pas minimale dans le sens où l'estimation discriminative a une infinité de solutions. Cela n'est pas vraiment un problème car toutes ces solutions donne la même règle de classification.

Prenons l'exemple des mélanges de gaussiennes sphériques étudiés dans le chapitre 2. La simplicité du modèle

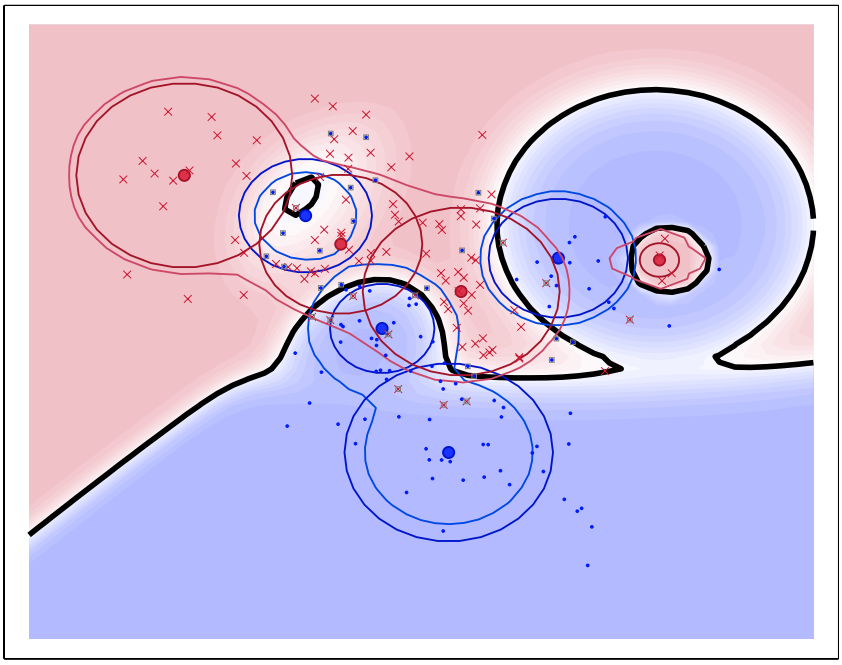


FIG. 5.1 – Estimation GDT pour l'exemple du mélange de gaussiennes sphérique sur l'exemple de discrimination étudié dans l'introduction.

se paie précisément par un biais important dans l'estimation générative. A l'opposé, l'estimation discriminative des modèles de mélanges est très peu biaisée : un agencement intelligent des paramètres de moyenne et de variance permet d'approcher une grande variété de frontières de classification. Cependant, les paramètres peuvent perdre totalement leurs sens (e.g. les moyennes ne sont plus centrées sur les données, comme le montre la figure 2.2 page 33) et certains chercheurs ont remarqué que l'estimation discriminative des modèles de mélange a des performances très médiocres [61]. Dans l'estimation GDT, on se satisfait d'une correction partielle de ce biais, tout en conservant le sens du modèle paramétrique. La figure 5.1 illutre bien ce phénomène : les paramètres de moyenne et de variance sont très proches de l'estimateur génératif (figure 2.1, page 31) mais la frontière de classification sépare mieux les données difficiles à classer, comme le suggère la différence entre les figures 5.1 et 2.1 dans la zone de chevauchement des deux classes. On peut d'ailleur remarquer que la frontière de classification est relativement proche de celle obtenue par l'estimation discriminative (figure 2.2, page 33).

Peu d'approches intermédiaires entre les estimations génératives et discriminatives ont été proposées. On peut recenser un modèle hybride génératif-discriminatif basé sur le classifeur de Bayes naïf [135] et des modèles spéci-

fiques pour la classification de la parole par modèles de Markov cachés [108, 94]. Ces approches ne minimisent pas clairement une fonction de coût globale, mais proposent des heuristiques pour minimiser le biais de classification. Ce travail se différencie des approches précédentes par l'introduction d'un nouveau type d'estimateur, justifié par des argument théoriques et expérimentaux. Il améliore les performances de classification des modèles génératifs de manière *simple* et *cohérente*.

L'estimation est en effet relativement simple car il n'y a qu'une seule fonction à maximiser, contrairement à l'utilisation du noyau de Fisher, qui nécessite d'abord l'estimation des paramètres par maximum de vraisemblance, puis la recherche de la frontière optimale de classification par une méthode à noyau (e.g. [81]). Un autre avantage est que l'estimation peut se faire par maximisation d'une fonction sans contrainte, ce qui n'est pas le cas de la Discrimination par Entropie Maximale (MED), qui introduit autant de contraintes que de données d'apprentissage [151], conduisant à des problèmes d'optimisation extrèmement difficiles à résoudre. La cohérence vient de l'aspect intermédiaire entre génératif et discriminatif, ce qui n'apparaît pas dans les méthodes existantes. En effet, MED ne fournit qu'un estimateur purement discriminatif, parfois régularisé pour éviter le phénomène de sur-apprentissage, mais ne permet pas, dans le cas d'une très forte régularisation d'obtenir les paramètres du maximum de vraisemblance de la densité jointe.

5.1 Joint or conditional learning in generative classification ? A difficult choice

In supervised classification, the inputs x and their labels y arise from an unknown joint probability $p(x, y)$, and the overall goal is to find the classification rule with the smallest error rate. This depends only on the conditional density $p(y|x)$. *Discriminative* methods directly model the conditional distribution, without assuming anything about the input distribution $p(x)$.

Conversely, if we can approximate $p(x, y)$ using a parametric family of models $\mathcal{G} = \{p_\theta(x, y), \theta \in \Theta\}$, then a natural classifier is obtained by first estimating the class-conditional densities, then classifying each new data point to the class with highest posterior probability. This approach is called *generative* classification. Unsupervised density estimation methods such as mixture distributions and graphical models are powerful tools to model structures and correlations in the data. Generative classifiers are able to exploit these structures.

Classifiers based on models of the form $\boldsymbol{p}_\theta(x, y)$, must estimate θ from the training data. Two estimators have been proposed in the literature :

– the *generative estimator* is the parameter maximizing the *joint* likelihood of the inputs x and the outputs y [50, 137, 100],

– the *discriminative estimator* is the parameter maximizing the likelihood of y *contitionally* on x [64, 84, 61].

Generative estimators are “natural” because they maximize the likelihood for which the model was built. Linear Discriminant Analysis (LDA) and Naive Bayes (NB) are popular classification methods based on generative estimator. The estimator converges to the best parameter (ML solution) for the joint distribution $p(x, y)$ but the resulting conditional density is usually a biased classifier unless $p_\theta(x)$ is an accurate model for $p(x)$.

The discriminative estimator is equivalent to the minimization of a specific classification loss, sometimes called classification entropy. It can be viewed as a smooth approximation of the error rate. Note that for discriminative classifiers for which no assumption on the input distribution holds, such as the *logistic regression*, the conditional likelihood is usually the quantity that is maximized. This explains why Greiner & Zhou [64] called “Extended Logistic Regression” the discriminative learning of Bayesian Networks.

In real world problems the assumed generative model is rarely exact, and asymptotically, discriminative es-

timation should typically be preferred [158, 127]. The key argument is that the discriminative estimator of this generative model converges to the conditional density $\boldsymbol{p}_\theta(y|x)$ that minimizes the negative log-likelihood classification loss against the true density $p(x, y)$ (see [100],pp.270–276 for a formal proof). For finite sample sizes, there is a bias-variance tradeoff and it is less obvious how to choose between generative and discriminative classifiers.

For some specific distributions discriminative learning of the parameters is equivalent to fitting a logistic model. For example, modelling the class-conditional distributions as multivariate Gaussians (Linear or Quadratic Discriminant Analysis) leads to linear or quadratic logistic regression, while assuming that the inputs are independent (NB classifier) corresponds to fitting a Generalized Additive Model (GAM). Many authors have already studied these generative-discriminative pairs e.g. [127, 142]. Under the unrealistic assumption that the underlying distributions are Gaussian with equal covariances, it is known that LDA requires less data than its discriminative counterpart, linear logistic regression [42].

In the following, we will first consider the parameter estimation problem for generatively parametrized models, focusing on the theoretical distinction between generative and discriminative estimators. Then we propose a new technique for combining the two approaches : the Generative-Discriminative Trade-off (GDT) estimator. It is based on a continuous class of cost functions that interpolate in a convex way between generative and discriminative estimation. The goal is to find the parameters that maximize the classification performance on the underlying population, but we do this by defining a cost function that is intermediate between the joint and the conditional log-likelihoods and optimizing this on training and validation sets. Given that the generative model based on maximum likelihood (ML) hang minimum variance but is possibly biased in terms of classification loss, while the discriminative is more variable but avoids bias, there are good reasons for thinking that an intermediate method such as the GDT estimate should sometimes be preferred. A theoretical justification is given, and some experiments on simulations and real datasets illustrate the benefits of the estimator.

5.2 Preliminaries

Using independent training samples $\{x_i, y_i\}_{i=1,\ldots,n}$, $x_i \in \mathbb{R}^d$,and $y_i \in \{1, \ldots, K\}$ sampled from the unknown distribution $p(x, y)$, we aim to find the classification rule that gives the lowest error rate on new data. This is closely

related to estimating the conditional probability $p(y|x)$.

For each of the K classes, the class-conditional probability $p(x|y = k)$ is modeled by a parametric model f_k with parameters θ_k. The y follows a multinomial distribution with parameters $p_1, \ldots, p_K$. The full parameterization of the joint density is $\theta = (p_1, \ldots, p_K, \theta_1, \ldots, \theta_K)$. Given θ, new data points x are classified to the group k giving the highest conditional probability

$$\boldsymbol{p}(Y = k|X = x; \theta) = \frac{p_k f_k(x; \theta_k)}{\sum_{l=1}^{K} p_l f_l(x_i; \theta_l)}. \tag{5.1}$$

Given data $\{x_i, y_i\}_{i=1,\ldots,n}$, there are two standard ways to estimate θ :

– the *generative estimate* is the maximum of the joint log-likelihood $\mathcal{L}_J(\theta) = \log \boldsymbol{p}(\mathbf{x}, \mathbf{y}; \theta)$. This maximum is assumed to be unique :

$$\hat{\theta}_J = \underset{\theta \in \Theta}{\operatorname{argmax}} \mathcal{L}_J(\theta), \qquad \mathcal{L}_J(\theta) = \textstyle\sum_{i=1}^{n} \log p_{y_i} f_{y_i}(x_i; \theta). \tag{5.2}$$

– the *discriminative estimate* is the maximum of the conditional loglikelihood[17] $\mathcal{L}_C(\theta) = \log \boldsymbol{p}(\mathbf{y}|\mathbf{x}; \theta)$

$$\hat{\theta}_C = \underset{\theta \in \Theta}{\operatorname{argmax}} \mathcal{L}_C(\theta), \qquad \mathcal{L}_C(\theta) = \textstyle\sum_{i=1}^{n} \log \frac{p_{y_i} f_{y_i}(x_i;\theta)}{\sum_k p_k f_k(x_i;\theta)}. \tag{5.3}$$

Let $\mathcal{D} = \{p_\theta(y|x) = p_\theta(x, y)/\sum_z p_\theta(x, z), \theta \in \Theta\}$ be the set of conditional densities derived from the generative model. The discriminative estimator finds the conditional density model in $\mathcal{D}$ that minimizes a particular classification loss function on the training set : the negative conditional log-likelihood $-\mathcal{L}_C$, which can be viewed as a differentiable approximation to the training set error rate [71]. In this sense, it is a form of *empirical risk minimization* — an estimator that finds the classification rule with the smallest training error rate.

The discriminative estimate is often non unique, and simplification of equation (5.3) may allow parameters that influence only $p(x)$, not $p(y|x)$ to be eliminated. For example, the conditional likelihood does not depend on the parameters of the shared covariance matrix in LDA, this leads to the linear logistic regression over a lower dimensional parameter space. However, we will not use this reduction, as we need to maintain a common parameterization for the discriminative and generative cases. The ambiguity does not affect the classification performance, because for any solution of (5.3) gives the same classification rule.

[17] We assume here that the discriminative estimate is unique. If it is not unique, one can choose for $\hat{\theta}_C$ any maximizer of $\mathcal{L}_C(\theta)$. For any solution $\hat{\theta}_C$, the conditional probability $\boldsymbol{p}(Y = k|X = x; \hat{\theta}_C)$ is the same.

See chapter 2 for a precise definition of the discriminative solution.

The quantity $\mathcal{L}_C$ can be expanded as follows :

$$\mathcal{L}_C(\theta) = \underbrace{\sum_{i=1}^{n} \log p_{y_i} f_{y_i}(x_i;\theta)}_{\mathcal{L}_J(\theta)} - \underbrace{\sum_{i=1}^{n} \log \sum_{k=1}^{K} p_k f_k(x_i;\theta)}_{\mathcal{L}_M(\theta)} \tag{5.4}$$

The difference between the two objective functions $\mathcal{L}_J$ and $\mathcal{L}_C$ is thus the *marginal likelihood*

$$\mathcal{L}_M(\theta) = \sum_{i=1}^{n} \sum_{k} \log p_\theta(x_i, k),$$

i.e. the log-likelihood of the input space probability model $p_\theta(x)$. Equation (5.4) shows that compared to the discriminative approach, the generative strategy tends to favor parameters that give high likelihood on the training data.

5.3 Between Generative and Discriminative classifiers

To get a natural trade-off between the two approaches, we can introduce a new objective function $\mathcal{L}_\lambda$ based on a parameter $\lambda \in [0, 1]$ that interpolates linearly between the discriminative and generative objective functions :

$$\mathcal{L}_\lambda(\theta) = \mathcal{L}_J(\theta) - (1-\lambda)\mathcal{L}_M(\theta) \tag{5.5}$$

$$= \lambda\mathcal{L}_J(\theta) + (1-\lambda)\mathcal{L}_C(\theta). \tag{5.6}$$

Definition 1 *For $\lambda \in [0, 1]$, the GDT estimate is*

$$\hat{\theta}_\lambda = \begin{cases} \operatorname{argmax}_{\theta\in\Theta} \mathcal{L}_\lambda(\theta) & \textit{if} \quad \lambda > 0 \\ \lim_{\lambda \to 0^+} \hat{\theta}_\lambda & \textit{if} \quad \lambda = 0. \end{cases} \tag{5.7}$$

Taking $\lambda = 0$ leads to the discriminative estimate $\hat{\theta}_C$ (if unique), while $\lambda = 1$ leads to the generative one $\hat{\theta}_J$. We expect that the GDT estimates $\hat{\theta}_\lambda$ ($0 < \lambda < 1$) will sometimes have better generalization performances than these two extremes. Even if the discriminative estimate (5.3) is not unique, the maximum of (5.7) is unique for all $\lambda \in [0, 1)$ if the ML estimate $\hat{\theta}_J$ is unique.

5.3.1 Justification of the estimator as a constraint optimization problem

For a fixed λ, the GDT estimate can be thought of providing the model that gives the best *classification* performance on the training data, among all models that fit the *joint distribution* of the training data well. We now

develop this remark.

Let c be a real number. The set $\mathcal{S} = \{\theta \in \Theta \; ; \; \mathcal{L}_J(\theta) \geq c\}$ contains all "good candidates", i.e. parameter values that give a likelihood on the training data greater than c. Within $\mathcal{S}$, we would like to choose the parameter $\tilde{\theta}$ that minimizes the classification loss $-\mathcal{L}_C(\theta)$ on the training set. This corresponds to a constrained optimization problem :

$$\max_{\theta \in \Theta} \mathcal{L}_C(\theta) \quad \text{such that} \quad \mathcal{L}_J(\theta) \geq c. \tag{5.8}$$

Assuming that the solution is unique, the Lagrange multiplier theorem says that $\exists \nu \geq 0$ such that $\nabla \mathcal{L}_C(\theta) + \nu \nabla \mathcal{L}_J(\theta) = 0$. Given the Lagrange multiplier ν, the optimal θ is

$$\tilde{\theta} = \underset{\theta \in \Theta}{\operatorname{argmax}} \left(\mathcal{L}_C(\theta) + \nu \mathcal{L}_J(\theta) \right). \tag{5.9}$$

By choosing $\lambda = \frac{\nu}{1+\nu}$ to get a value between 0 and 1, $\tilde{\theta}$ is equal to our estimate GDT estimate $\hat{\theta}_\lambda$ defined in (5.7). Here, ν is a non-decreasing function of c since the constraint $\mathcal{L}_J^{(n)}(\theta^{(n)}) \geq c$ becomes more and more restrictive as c increases. Hence, choosing λ is equivalent to finding the best classifier among models having a joint likelihood greater than c. However, instead of choosing the threshold c of the likelihood, we directly set the constant λ, which leads to the family of estimates $\{\hat{\theta}_\lambda^{(n)}, \lambda \in [0,1]\}$ defined in (5.7). Similarly, GDT is equivalent to the constrained problem $\max \mathcal{L}_J$ such that $\mathcal{L}_C > c'$.

5.3.2 Parameter estimation

The estimator $\hat{\theta}_\lambda$ has no closed-form expression and has to be computed numerically. Since we use a differentiable classification loss, the maximization problem (5.7) can be solved by any gradient ascent method. The Newton algorithm converges rapidly but requires the computation of the Hessian matrix, The Conjugate Gradient (CG) algorithm may be more suitable for large scale problems : it needs only the first derivative and it is possible to avoid the storage of the quasi-Hessian matrix which can be huge when the number of parameters is large.

To illustrate the gradient computation, we assume here that the parameters θ_k of the different class densities are independent. Taking the derivative of (5.5) with respect to θ_k and p_k, we get

$$\begin{cases} \frac{\partial}{\partial \theta_k} \mathcal{L}_\lambda(\theta_k) = \sum_{i=1}^n (\mathbf{I}_{\{y_i = k\}} - (1-\lambda)\tau_{ki}) \frac{\partial \log f_k(x_i;\theta_k)}{\partial \theta_k} \\ \frac{\partial}{\partial p_k} \mathcal{L}_\lambda(\theta_k) = \frac{1}{p_k} \left(n_k - (1-\lambda) \sum_{i=1}^n \tau_{ki} \right) \end{cases} \tag{5.10}$$

with $n_k = \sum_{i=1}^{n} \mathbf{I}_{\{y_i=k\}}$ and $\tau_{ki} = \frac{p_k f_k(x_i;\theta_k)}{\sum_{l=1}^{K} p_l f_l(x_i;\theta_l)}$. The optimal parameters are zeros of the equations (5.10) for $k = 1, \ldots, K$.

For a given class k, these equations are analogous to the ML equations on weighted data, although unlike ML, the weights can be negative here. Each point has a weight $\mathbf{I}_{\{y_i=k\}} - (1-\lambda)\tau_{ki}$. The observations that have most influence on the θ_k-gradient are those that belong to the class k with a low probability (τ_{ki} is small), and conversely those that do not belong to the class k but that are assigned to it with a high probability. The influence of the assignment probabilities is controlled by the parameter λ. This remark may ultimately help us to link our approach to boosting, and similar algorithms that iteratively re-weight misclassified data. It also shows that the generative estimator (λ=1) is not affected by the classification rate of the data points.

Class-conditional distribution in the exponential family We now study GDT estimation when the distributions are modelled with a distribution in exponential family. For any distribution belonging to the exponential family, the pdf can be writen under the form

$$\log \boldsymbol{p}(x, \theta) = \phi(x)^T \theta - \alpha(\theta) + \psi(x) \tag{5.11}$$

where $\psi(x)$ is the reference density, θ is the natural parameter, $\phi(x)$ is the sufficient statistic, $\alpha(\theta)$ is a normalizing constant. The parameter space – i.e., the set of values of θ for which this function is integrable – is necessarily convex [136]. Some examples of standard distributions belonging to the exponential family are given in Table 5.1.

distribution		$\alpha(\theta)$	$\psi(x)$
Gaussian (known covariance)	$x \sim \mathcal{N}(\theta, \Sigma)$	$\frac{1}{2}\|\theta\|^2$	$-\frac{1}{2}(\|x\|^2 - d\log(2p))$
Poisson	$x \sim \mathcal{P}(\exp(\theta))$	$\exp(\theta)$	$-\log(x!)$
Exponential	$x \sim \mathcal{E}(-\theta)$	$-\log(-\theta)$	0
Binomial	$x \sim \mathcal{B}(1/(1+\exp(-\theta)))$	$\log(1+\exp(-\theta))$	0

TAB. 5.1 – Log-linear parametrization of standard distributions in the exponential family.

When discriminating two classes whose distribution belongs to the exponential family, the class-conditional

distributions take the form

$$\log \boldsymbol{p}(x|y=1,\theta_1) = \phi_1(x)^T\theta_1 - \alpha_1(\theta_1) + \psi_1(x)$$

$$\log \boldsymbol{p}(x|y=0,\theta_0) = \phi_0(x)^T\theta_0 - \alpha_0(\theta_0) + \psi_0(x)$$

where α is a convex function and ψ a normalization factor. We assume that the transformations of the input data is the same for the two classes, *i.e.* $\phi_1 = \phi_0 = \phi$. Thus, applying the ϕ function on the input variables, we can set ϕ to the identity without loss of generality. For these models, there are simple matrix formulae for the first and second derivatives of the joint and conditional likelihood. The calculations are given in Appendix A. Since the Hessian is positive definite in standard exponential family distributions, the Newton-Raphson algorithm can be applied.

LinearGDT : a smooth transition between LDA and the linear logistic regression When the class-conditional distributions are modelled as with multivariate gaussian dentities with a shared covariance matrix, the conditional distribution $\boldsymbol{p}(y|x;\theta)$ has a linear discriminant form. Assuming that the class proportions $\pi_1, \cdots, \pi_K$ are known, we define the *LinearGDT* estimor as follows :

Definition 2 LinearGDT *is the GDT estimator based on the LDA model :*

$$\hat{\theta}_\lambda = \text{argmax}_{(m_1,m_2,\Sigma)}\, \mathcal{L}^{lin}_\lambda(\theta)$$

$$\mathcal{L}^{lin}_\lambda(\theta) = -\frac{\lambda n}{2}\log|\Sigma| + \frac{\lambda}{2}\sum_{i=1}^n (x_i - m_{y_i})^T\Sigma^{-1}(x_i - m_{y_i}) + (1-\lambda)\sum_{i=1}^n \log\frac{\pi_{y_i} e^{(x_i-m_{y_i})\Sigma^{-1}(x_i-m_{y_i})}}{\sum_{k=1}^K \pi_k e^{(x_i-m_k)\Sigma^{-1}(x_i-m_k)}},$$

for $0 < \lambda \leq 1$ *and* $\hat{\theta}_0 = \lim_{\lambda\to 0}\hat{\theta}_\lambda$.

In chapter 2, we saw that the generative and discriminative estimates of the mean and variance parameters correspond to standard classification methods : LDA and linear logistic regression, respectively.

In many situations, LDA and linear logistic regression give similar classification results due to the simple form of the classification boundary (see e.g. [71]) However, in high dimensional settings, the difference can be significant and it is interesting to see how our intermediate model behaves compared to these standard linear classifiers.

In the next section, we give some sufficient conditions for GDT to have asymptotically better classification performance than LDA and linear logistic regression, and in the experimental section, it will be shown that LinearGDT also works well on real datasets.

5.4 Theoretical properties of the GDT etimators

5.4.1 Asymptotics

In this section, we give some necessary conditions for GDT to be better than the generative and the discriminative approaches. To compare estimators based on n training samples, we use a *loss function* η_n which corresponds (up to a constant) to the expected log-likelihood classification loss on test data :

$$\eta_n(\lambda) = E\left[-\log \frac{\boldsymbol{p}(Y|X;\hat{\theta}_\lambda^{(n)})}{\boldsymbol{p}(Y|X)}\right]. \tag{5.12}$$

where $\boldsymbol{p}(x,y)$ is the true conditional distribution of the test and training data. Note that the expectation is taken on two identical distributions : the sample distribution (through the estimator $\hat{\theta}_\lambda^{(n)}$) and the test distribution (defined on (X,Y)). Here, the estimator $\hat{\theta}_\lambda^{(n)}$ is the GDT estimator based on a training sample of size n. The *optimal* parameter λ_n^* is the one minimizing η_n :

$$\eta_n(\lambda_n^*) = \min_{\lambda\in[0,1]} \eta_n(\lambda) \tag{5.13}$$

The role of λ_n^* is important because it determines the best estimator. One should choose :

- the discriminative estimate if $\lambda_n^* = 0$,
- the generative estimate if $\lambda_n^* = 1$,
- the GDT estimate if $0 < \lambda_n^* < 1$.

We consider three disjoint assumptions to the parametric family :

1. $\exists\theta_0 \in \Theta, \boldsymbol{p}(X,Y;\theta_0) = \boldsymbol{p}(X,Y)$, *i.e.* the joint distribution of the data belongs to the parametric family, which implies that the conditional distribution is also true at θ_0 : $\boldsymbol{p}(Y|X;\theta_0) = \boldsymbol{p}(Y|X)$,
2. $\exists\theta_0 \in \Theta, \boldsymbol{p}(Y|X;\theta_0) = \boldsymbol{p}(Y|X)$ and $\forall\theta \in \Theta, \boldsymbol{p}(X,Y;\theta) \neq \boldsymbol{p}(X,Y)$ *i.e.* the conditional distribution belongs to the parametric family, but the joint distribution is false,
3. $\forall\theta \in \Theta, \boldsymbol{p}(Y|X;\theta) \neq \boldsymbol{p}(Y|X)$ *i.e.* both joint and conditional distributions are false.

The least restrictive and most general assumption[18] is the third one, but for simplicity we will consider the first two cases. We intend to prove that

- with assumptions 1, the generative solution is asymptotically optimal ($\lambda_n^* = 1$), and

[18] In general (*i.e.* in real applications), the model assumptions are ot true (for both the joint and the conditional distributions).

– with assumptions 2, the GDT solution is asymptotically optimal for a particular $0 < \lambda_n^* < 1$ which decreases to 0 as the training sample size n increases.

The first statement is very similar to a result given Chapter 2. We refer the reader to the Proposition 1 page 37, in which the discriminative estimator can be replaced by any GDT estimator with $\lambda < 1$. This result is intuitive because the ML estimator $\hat{\theta}_1$ estimates the true model parameters θ_0 which also minimizes the classification loss since the Bayes rule is optimal at θ_0. The second case is more interesting, since it justifies the use of the GDT estimation when the conditional distribution is true, but the joint distribution is false. We give the main result of this chapter :

Theorem 1 *If the following conditions are satisfied :*

(i) For all $\varepsilon > 0$, the solution of $\max_{\theta \in \Theta} \log \boldsymbol{p}(Y|X;\theta) + \varepsilon \log \boldsymbol{p}(X,Y;\theta)$ is unique,

(ii) the conditional model belongs to the parametric family, i.e. *there exist $\theta_0 \in \Theta$ such that $\log \boldsymbol{p}(Y|X;\theta_0) = \log \boldsymbol{p}(Y|X)$,*

(iii) the asymptotic generative solution $\theta_1^* = \lim_{n\to\infty} \theta_1^{(n)}$ has a higher classification loss than the asymptotic discriminative solution $\theta_0^* = \lim_{n\to\infty} \theta_0^{(n)}$, i.e. *$E\left[\log \boldsymbol{p}(Y|X;\theta_1^*)\right] < E\left[\log \boldsymbol{p}(Y|X;\theta_0^*)\right]$.*

(iv) the model pdf $\boldsymbol{p}(X,Y;\theta)$ is twice differentiable with respect to the parameter θ around θ_0^* and the second derivative is bounded,

then the value $\lambda_n^* = \min_{\lambda\in[0,1]} \eta_n(\lambda)$ tends to 0 as n tends to ∞. Moreover, for n sufficiently large, there exists a constant $C > 0$ such that

$$\frac{C}{n} \leq \lambda_n^* < 1. \tag{5.14}$$

The proof of this theorem is given in appendix B. The main point is that the optimal value of λ is neither 0 nor 1, there are intermediary estimates ($\lambda \in]0,1[$) that have better generalisation performance than the standard generative and discriminative estimators. In addition, the theorem gives a lower bound on the convergence speed of λ_n^* to 0.

We apply this theorem to the LinearGDT estimator, *i.e.* when we use multivariate Gaussian distributions with a shared covariance matrix to model the class-conditional pdfs. This characterizes a class of distributions for which the GDT estimate has better generalization performance than LDA and linear logistic regression.

Corollary 1 *If the sample distribution satisfies these conditions (i) and (ii) :*

(i) There exist $\beta \in \mathbb{R}^d$ *such that* $\boldsymbol{p}(Y = 1|X) = \exp(\beta' X)\boldsymbol{p}(Y = 2|X)$,

(ii) $\boldsymbol{p}(Y = 1|X)$ *is not a Gaussian distribution,*

then, for a sufficiently large training size n*, the expected loss of the LinearGDT estimator is strictly smaller than the expected loss of LDA and the linear logistic regression.*

In other words, Corollary 1 states that the optimal tuning parameter $\lambda_n^* = \min_{\lambda\in[0,1]} \eta_n(\lambda)$ satisfies :

$$\eta_n(\lambda_n^*) < \eta_n(0) \qquad \text{(better than the linear logistic regression) and}$$

$$\eta_n(\lambda_n^*) < \eta_n(1) \qquad \text{(better than the linear discriminant solution).}$$

Hence, for any non-Gaussian distribution for which the linear logistic model is true, the GDT estimation with $\lambda = \lambda_n^*$ have better prediction performances than LDA and the logistic regression.

5.4.2 Choice of λ.

In practice, the value λ_n^* is not directly available as it requires knowledge of the sample distribution $\boldsymbol{p}(X, Y)$.

The tuning parameter λ functions like the smoothing parameter in regularization methods. λ cannot be set on the basis of minimum classification loss on the training set, since by definition, $\lambda = 0$ gives the optimal θ for training set classification. Instead, λ is set to the value $\hat{\lambda}$ that minimizes the cross-validated classification loss

$$\hat{\lambda} = \underset{\lambda\in\Lambda}{\operatorname{argmin}} \sum_{i=1}^{\nu} -\log \boldsymbol{p}(\mathbf{y}^{(-i)}|\mathbf{x}^{(-i)}; \hat{\theta}_\lambda^{(i)}) \tag{5.15}$$

where Λ is a set containing possible values of λ (between 0 and 1), $\hat{\theta}_\lambda^{(i)}$ is the GDT parameter estimate based on the i^{th} training set and $(\mathbf{x}^{(-i)}, \mathbf{y}^{(-i)})$ is the i^{th} validation set. It might seen more natural to minimize 0-1 classification loss at this point, but experimentally, we find that it leads to less stable estimates of λ.

If the optimal $\hat{\lambda}$ is close to one, the generative classifier is preferred. This suggests that the bias in $p_\theta(x, y)$ (if any) does not affect the discrimination of the model too much. Similarly, if $\hat{\lambda}$ is close to zero, it suggests that the model $p_\theta(x, y)$ does not fit the data well, and the bias of the generative classifier is too high to provide good classification results. In this case, a more complex model — i.e. with more parameters, or less constrained ones —

may be needed to reduce the bias. For intermediate $\hat{\lambda}$, there is an equilibrium between the bias and the variance, meaning that the model complexity is well adapted to the amount of training data.

It would be usefull to find computationally efficient alternatives to cross-validation for estimating the value of λ that give good classification performance on the test data.

5.4.3 Bias-variance decomposition

We now give some theoretical insights about the optimal value for λ. The classification error can be decomposed into bias and variance terms. The relative importance of these two terms is controlled by the GDT parameter λ. This explains why the generalization performance of the method is related to the choice of lambda.

$$\theta^*_\lambda = \lim_{n \to \infty} \hat{\theta}^{(n)}_\lambda$$

By the law of large numbers, the parameters θ^*_λ are the expected values of their respective estimator $\hat{\theta}^{(n)}_\lambda$. We denote θ^*_λ the values of the parameters minimizing the expected GDT loss function :

$$\theta^*_\lambda = \underset{\theta \in \Theta}{\operatorname{argmin}} \, E\left[\lambda \log \boldsymbol{p}(X, Y; \theta) + (1 - \lambda) \log \boldsymbol{p}(Y|X; \theta)\right]. \tag{5.16}$$

We assumed that the parameter estimates are consistent, *i.e.* that $\hat{\theta}_\lambda \to \theta^*_\lambda$ almost surely as n tends to infinity.[19].

The "best" parameter in terms of classification loss is $\theta^*_C = \theta^*_0$, for which the discriminative estimator $\hat{\theta}_0$ is unbiased. However, this unbiasedness is associated with a high estimation variance. The GDT estimator, allows some bias in return for lower variance. Without loss of generality, we look for the estimator $\hat{\theta}_\lambda$ minimizing the quantity $\eta(\lambda)$ defined in (5.12). Let $L_C(\theta) = E\left[\log \boldsymbol{p}(Y|X; \theta)\right]$. The overall classification loss $\eta(\lambda)$ can be split into bias and variance terms :

$$\eta(\lambda) = \underbrace{E\left[L_C(\hat{\theta}_\lambda) - L_C(\theta^*_\lambda)\right]}_{\text{variance}(\lambda)} + \underbrace{L_C(\theta^*_\lambda) - L_C(\theta^*_C)}_{\text{bias}(\lambda)} + \text{bias}^2_0 \tag{5.17}$$

where bias^2_0 is the irreducible minimal model bias $L_C(\theta^*_C) - E\left[\log \boldsymbol{p}(Y|X)\right]$. Due to the fact that the generative estimation is the minimum variance estimator of θ, variance(λ) is minimal for $\lambda = 1$ (the proof is found in the

[19] Sometimes the discriminative solution is not unique, but the GDT estimators for $\lambda > 0$ are consistent. In this case we define $\theta^*_0 = \lim_{\lambda \to 0} \theta^*_\lambda$. This appears for example in the NB classifier and the LDA model.

appendix). Conversely, bias$(0) = 0$ *i.e.* the bias term is minimum for the discriminative estimator. As for many other learning methods, the estimator should be chosen in order to balance these two quantities.

5.5 Simulations

5.5.1 A toy example with binary regressor

To illustrate the fact that the GDT estimator provides a way to balance the bias and the variance and can improve the generalization performance of the generative classifier, we study a toy example involving only one parameter : predicting a binary variable Y given one binary variable X.

The advantage of discrete variables is that we can define the true distribution (i.e. the distribution of the sample) with a saturated parametric model. In this case, we consider that $\boldsymbol{p}(Y = 1) = \boldsymbol{p}(Y = 0) = \frac{1}{2}$ so that the saturated model is defined by the proportions p_1^* and p_0^* of the conditional distributions $P(X|Y = 1)$ and $P(X|Y = 0)$.

$$X|Y = 1 \sim \mathcal{B}(p_1^*)$$

$$X|Y = 0 \sim \mathcal{B}(p_0^*)$$

The fitted model is parameterized by only one parameter :

$$X|Y = 1 \sim \mathcal{B}(\theta)$$

$$X|Y = 0 \sim \mathcal{B}(2\theta),$$

so that the proportion parameter of the class $Y = 0$ is constrained to be twice the proportion parameter of the class $Y = 1$. We set $p_1^* = .45$ and $p_0^* = 0.70$, so that p_1^* is different from $2p_0^*$, and the model assumptions are not satisfied. For each experiment, thirty data points were generated (15 for each class), For each value of λ, the GDT estimator was estimated by a simple maximization of the GDT objective function defined in (5.5) on the $[0, 1]$ range[20].

Table 5.2 gives the key quantities to show how the GDT estimation balances between bias and variance. The bias term is non-random and equals $L_C(\theta_\lambda^*) - L_C(\theta_C^*)$. It grows with λ, illustrating the well known fact that the generative estimation does not minimize the objective function L_C. There exists also a bias for the discriminative

[20]The `fminbnd` function of MATLAB was used

λ	d	bias^2 $L_C(\theta^*_\lambda) - L_C(\theta^*_C)$	variance $E\left[L_C(\hat{\theta}_\lambda) - L_C(\theta^*_\lambda)\right]$	bias^2+variance $E\left[L_C(\hat{\theta}_\lambda) - L_C(\theta^*_C)\right]$
Saturated model	2	0	0.2382	0.2382
0 (Discriminative)	1	0.1271	0.1354	0.2625
0.05	1	0.1324	0.0640	0.1964
0.10	1	0.1409	0.0519	0.1928
0.25	1	0.1572	0.0369	0.1942
0.50	1	0.1923	0.0182	0.2105
1 (Generative)	1	0.2249	0.0184	0.2433

TAB. 5.2 – Evaluation of the bias and variance of the GDT estimator in the binary variable experiment. The values of the expectation were approximated by averaging over 1000 estimations based on independent samples. The saturated model estimation involves two parameters instead of one for the GDT estimator. The column d gives the dimension of the parameter space (number of free parameters), not including the tuning parameter λ.

classifier, since the conditional model cannot model the conditional distribution of the simulated data [21]. The saturated model has a bias term of zero as it converges asymptotically to the joint distribution of the simulated data. So, the resulting conditional probability is optimal.

We can now interpret the variances. Although the term $E\left[L_C(\hat{\theta}_\lambda) - L_C(\theta^*_\lambda)\right]$ does not correspond exactly to a variance, it depends only on the variability of the sample. Contrary to the bias, it tends to zero as the amount of data grows. We named it variance because it can be interpreted as such using a quadratic approximation of the function L_C This term was computed by taking the mean over the 1000 experiments. Notice that the saturated model gives much larger variance that the GDT estimators. This is simply due to the fact that we need to estimate two parameters as opposed of one for the GDT estimator.

The objective quantity is $\eta(\lambda) = E\left[L_C(\hat{\theta}_\lambda) - L_C(\theta^*_C)\right]$. We look for the estimator that has the smallest conditional loss on average. The function $\eta(\lambda)$ is the sum of the bias and the variance terms previously mentioned. Its values are given in the last column of table 5.2. The best performance are reached by the GDT estimator.

It is interesting to see that the estimator of the saturated model has better performance (score 0.2382) than both

[21] The asumptions of the theorem 1 hold only if the bias of the discriminative model is equal to 0, *i.e.* bias$(0) = 0$.

the discriminative model (0.2625) and the generative one (0.2433). However, The GDT estimate gives even better performance for all of the intermediate models computed ($\lambda = 0.05$,0.10,0.25 and 0.50). Hence, when choosing between the generative estimator, the discriminative estimator and the saturated model, we conclude that the one parameter model is useless and the saturated model has to be chosen. But, when the GDT estimate is included, the one-parameter model has still some benefit.

Figure 5.2 gives a graphical illustration of the GDT estimate in this binary variable example. Each training sample is a point in the saturated parameter space whose coordinates are the empirical proportions

$$\hat{p}_1 = \frac{card\{x_i = 1, y_i = 1\}}{card\{y_i = 1\}} \quad \text{and} \quad \hat{p}_2 = \frac{card\{x_i = 1, y_i = 0\}}{card\{y_i = 0\}}.$$

The model space is a one-dimensional manifold (a line) in the saturated parameter space, so the estimation can be viewed as a projection on the parameter space according to the natural metric of the model. The plots illustrate well the decrease of the estimation variance (the projections are more concentrated in the last plot). Additionally, we see the increase in bias : the projections are centered around the optimal parameter value (the square) in the first plot but not in the last one.

The last experiment on this toy example is important to fully understand GDT estimation. For various training sample sizes and various values of α, we computed the optimal λ by minimizing the error on a test set of size 50 000 over λ. We also compared the error of the resulting classifier with a more complicated model, namely the saturated model. Figure 5.3 shows the results. The grey area represents the cases where the saturated model had the best classification rate so that the one parameter model assumptions reduce the performance.

Note that the "best" estimate was never the discriminative one — the saturated model always outperformed it. Outside of the grey area, the optimal value of λ ranges from about 0.05 to 1. For large training samples more discriminative estimates are preferred. The reduced model is exact for $\alpha^* = p_0^*/p_1^* \approx 1.55$ located at the bottom x-axis of the graph. At this value of α, the reduced generative model is chosen.

5.5.2 Class-conditional Gaussian distributions

To illustrate the behavior of the GDT method in the case of continuous distributions, we study its performance on two synthetic test problems. We define the true distributions of the data as follows :

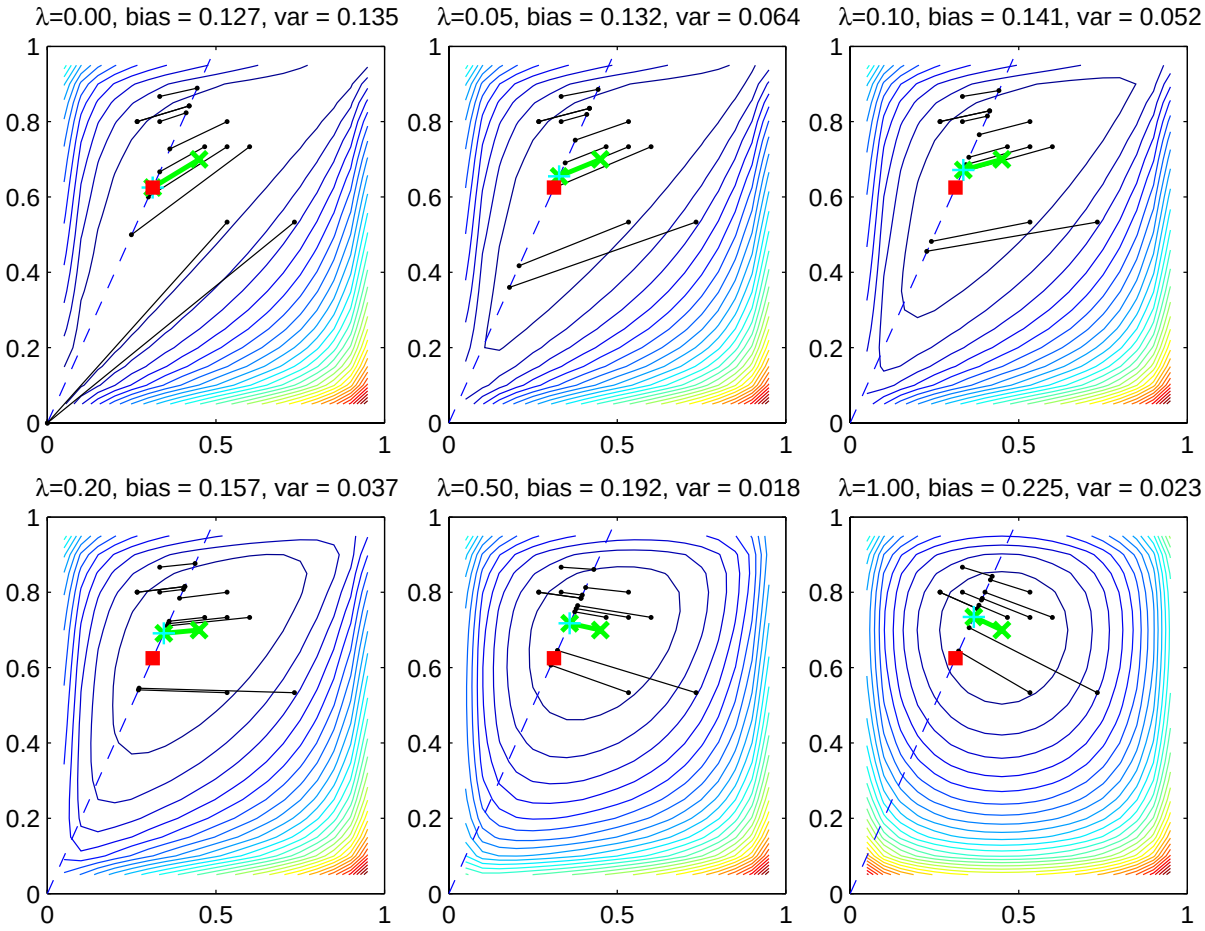


FIG. 5.2 – An illustration of GDT estimation with one binary regressor. The parameter space of the saturated model is the square $[0,1]^2$, and the model parameter space is the one-dimensional manifold represented by the dashed line. In each figure, the black points (away from the dashed line) are independent estimates of the saturated model. They are linked to the GDT estimate based on the same sample in order to show that GDT estimation can be interpreted as oblique projection on the model space. The sign $\square$ represent the parameter minimizing the conditional loss (the "best" parameter if we use the model), the sign $+$ is the mean of the GDT estimator and the sign $\times$ (aways from the dashed line) is the parameter of the saturated model minimizing the conditional loss.

– In the first experiment, the class conditional probabilities are Gaussian with identity covariance matrix and means $m_1 = (1.25, 0, 0, 0)$ and $m_0 = (-1.25, 0, 0, 0)$.

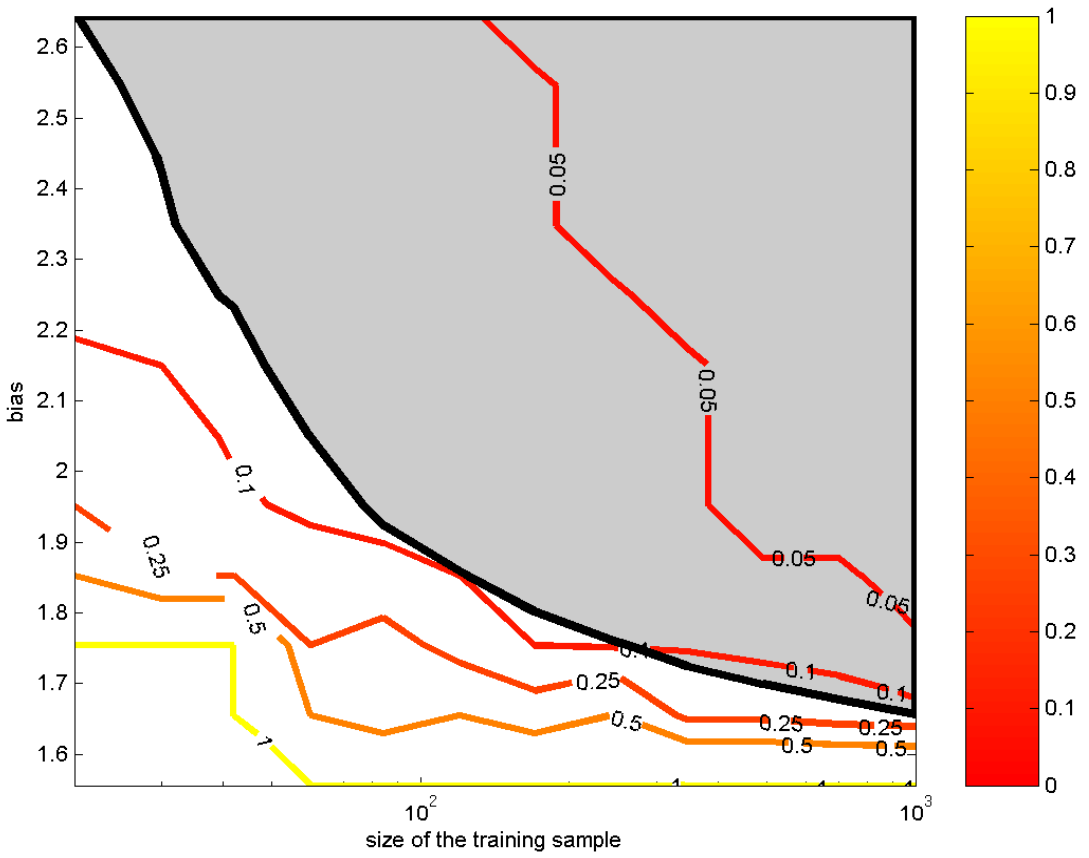


FIG. 5.3 – Optimal λ (selected by computation of the true classification loss) for different values of the factor α and the training size (the more α is different from 1.55, the more the model is biased). The grey region is the region for which the saturated model gives better classification performances. The left scale map the color of the isocontours to the optimal λ value.

– In the second case, we simulate x according to a uniform density with correlated covariates : $x^{(1)} \sim \mathcal{U}[0;1]$ and $x^{(d)} \sim \mathcal{U}[x^{(d-1)}; 1+x^{(d-1)}]$ with $d \in \{2,3,4\}$ and $x^{(i)}$ denotes the i^{th} covariate. Then $y|x$ is simulated according to a Bernoulli distribution with parameter $1/\exp(-2.5x^{(1)})$.

The LDA model may suite the first distribution, as the class-condiional densities are Gaussians with shared covariance matrix. In the second case the LDA model is not true, but the conditional distribution $\boldsymbol{p}(y|x)$. Hence, we tested on these distribution the diagonal LDA model (Gaussian distribution for each class with shared diagonal covariance matrices) and prior class probabilities equal to $\frac{1}{K}$. Note that the linear logistic model is true in the two experiments. Hence,

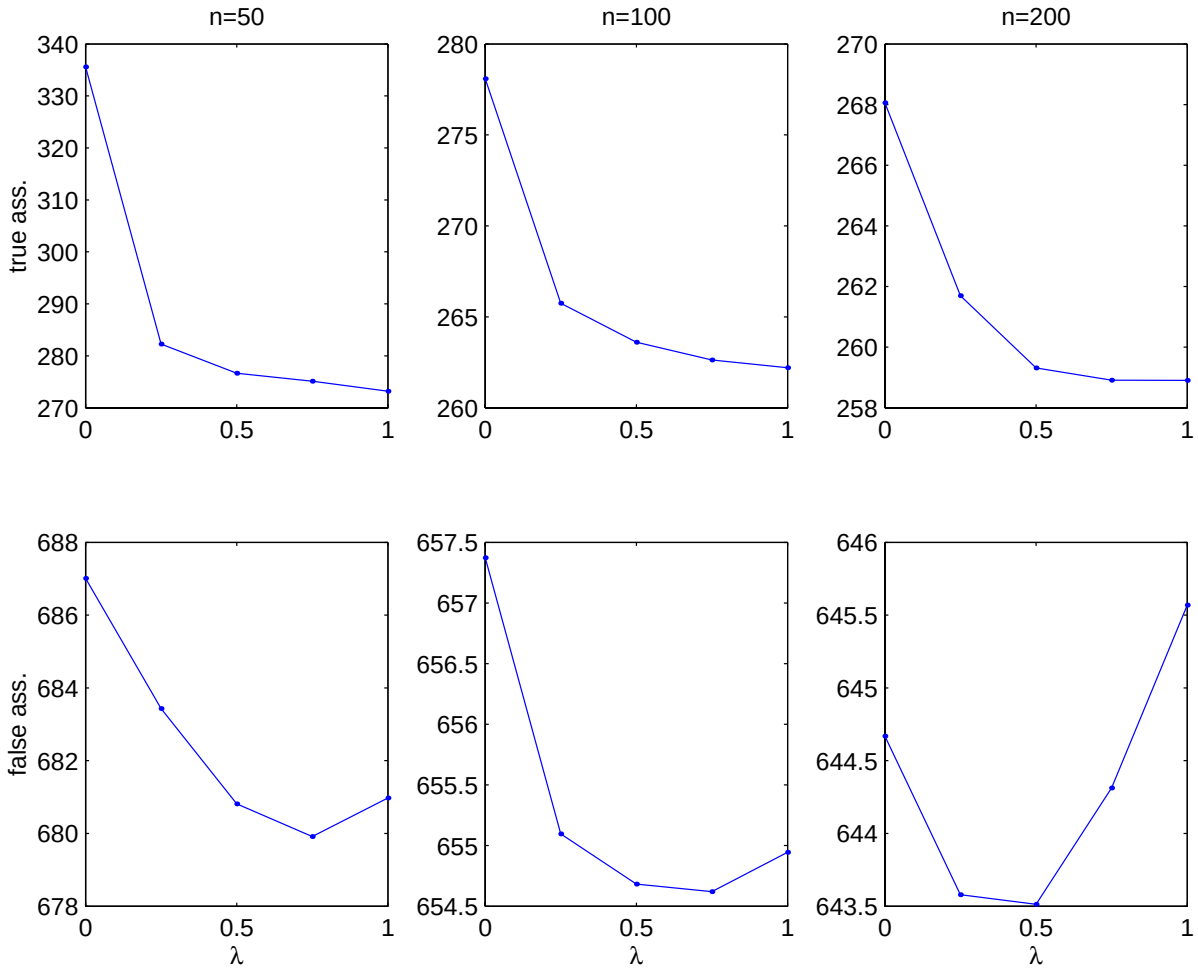


FIG. 5.4 – The full lines plot the conditional loss computed on test sets against the tuning parameter λ. Each plotted value is the median of 200 experiments. The rows correspond to the first and second simultations. The columns correspond to different training sample sizes.

– in the first experiment the joint distribution and the conditional distribution are true,

– in the second experiment, the joint distribution is not false but the conditional distribution is true – the assumptions of theorem 1 are valid.

Even if the model does not correspond exactly to the true density in the second experiment, the model is reasonable since Gaussian distributions provide good approximations to unimodal distributions.

In each case, we estimated the true classification loss of the classifiers learned on training samples of size 50, 100 and 200 by estimating it on test samples of size 10^5. The results are plotted in figure (5.4). We used standard

plug-in estimates for $\lambda = 1$ and closed form logistic regression[22] for $\lambda = 0$. For intermediate estimates, the conjugate gradient method was used. The first row (experiment 1) illustrates the fact that the generative classifier performs better than the other estimates, but this difference tends to decrease when the sample size increases. In the second row (experiment 2), the best performance is from the GDT estimator for all training set sizes, and the optimal value of λ (the one that minimizes the expected loss) decreases with n, confirming that the discriminative approach becomes optimal when n tends to infinity.

5.6 Experiments on real datasets

We tried our classification method on 14 publicly available benchmark datasets from the UCI machine learning repository[23]. We treated the discrete variables as continuous.

We used the LinearGDT classifier interpolating between the Gaussian NB classifier and linear logistic regression. We compared the changes in test error rate as the training sample sizes. A similar experiment was already presented in the literature [127] to show the better convergence rate of the generative estimator and the smaller bias of the discriminative one.

On the diabetes and german datasets, we learned the GDT estimators for $\lambda \in \{0, 0.01, 0.5, 0.1, 1\}$ on randomly selected training samples, and computed the test error rate on the remaining data. This experiment was repeated 200 times. The results are plotted on the figure 5.5. The phenomenon pointed out by [127] is still present, i.e. the generative model is better for the small training sizes, while the discriminative one has the smallest error rate asymptotically. But, the GDT estimator with $\lambda = 0.1$ outperforms both generative and discriminative ones for a large range of training sizes, namely 20 to 120 in the diabetes dataset and all sizes for the german dataset. The behavior of the other GDT estimates strongly confirms the theoretical results that we gave in section 5.9.1, since for any sample size n, there exists a value $0 < \hat{\lambda} < 1$ that minimizes the test error rate and this optimal value tends to decrease to 0 as n increases. These are promising results that also appear on other datasets, but generally for different values of $\hat{\lambda}$. However, they do not directly show the performance of GDT classification because we fixed

[22] We saw in chapter 2 that the discriminative training can be done using the full parametrization (means and variances) or equivalently by simple linear logistic regression.

[23] Available at `http://www.ics.uci.edu/~mlearn/MLRepository.html`.

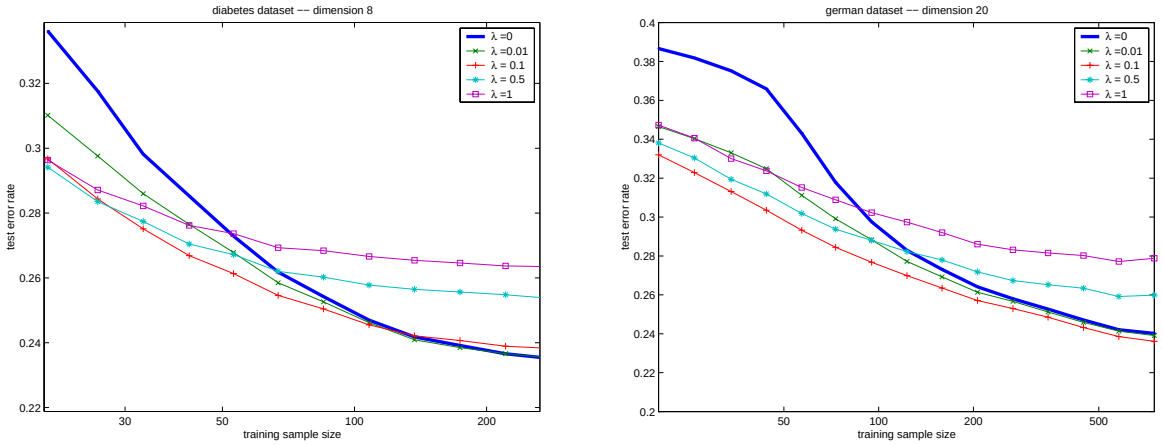


FIG. 5.5 – Comparison the GDT estimators for various training sample sizes. The left graph corresponds to the Pima indian diabete dataset, the right one corresponds to the german credit scoring dataset.

λ rather than selecting it by cross-validation on the training set.

The next experiment is similar, but we compare the results of only three estimators : the generative, the discriminative and the GDT estimators where the λ parameter was selected by cross-validation. We used the 10-fold cross-validated conditional likelihood given in equation (5.15). The test error rates were computed on 50 random test/train splits. The plots of the error rate against the training sample size are given on Figure 5.6 for the 14 datasets.

Except for the *contraceptive* and the *australian* datasets, the GDT classification has better classification performances than the other methods for at least one training sample size n. In *contraceptive* dataset, the generative model is outperfomed by the discriminative one even for small sample sizes. In this case, using a generative model only has a negative effect on the classification. Conversely, for the *australian* dataset, the generative estimator performs well even for relatively large sample sizes, indicating that the generative model suits the data well.

We already know that small training samples favor the generative classifier and large training samples favor the discriminative classifier. To show the possible improvements that GDT estimation can give, we select a training size between these two extremes. The value of n for which $\min(\text{err}_{GEN}, \text{err}_{DISC}) - \text{err}_{GDT}$ is maximal was used to plot the results in Table 5.3. They show that substantial improvements in the classification rate can be obtained for these intermediate values of λ. One can see that the training sample size is generally of the same order as the number of parameters, which equals 3 times the dimension d. We also note that the optimal λ can very close to

1, especially when the data are well separated (have a small error rate). This is mainly due to the fact that the conditional and joint likelihoods do not have the same scales or "strengths", but cross-validation over a wide range of λ values enables us to find a good balance between the two objective functions L_J and L_C.

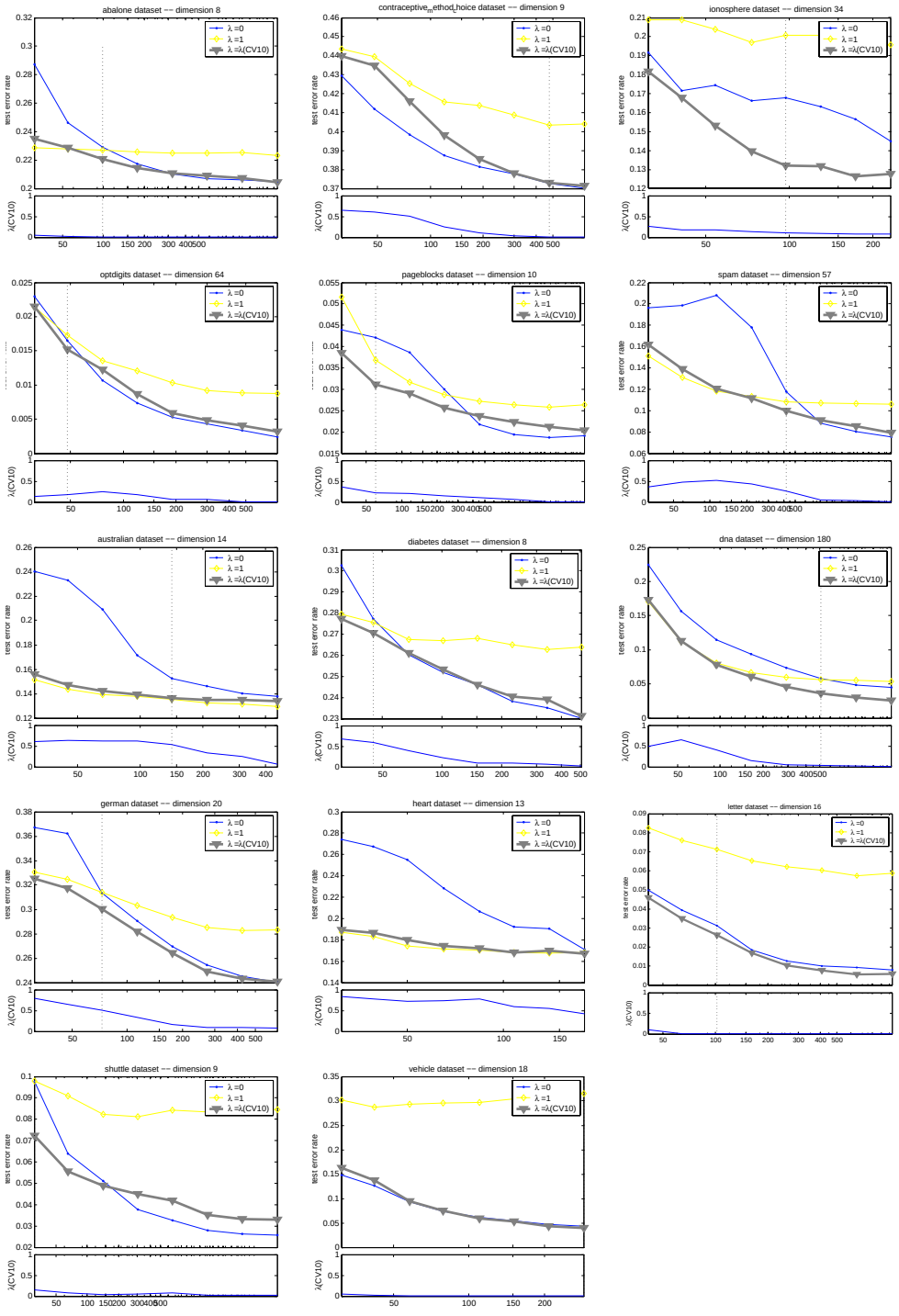


FIG. 5.6 – Average test error rates of the linear logistic regression (points), BN classifier (diamonds), and LinearGDT classifier (triangles) plotted against the training sample size. The GDT parameter λ was selected by a 10-fold cross-validation. The vertical lines indicate the training size for which the GDT estimatior leads to the highest improvement compared to the generative and discriminative estimators.

dataset	d	n	err_{DISC}	$\text{err}_{\hat{\lambda}}$	err_{GEN}	$\hat{\lambda}$
abalone	8	99	22.9 ± 1.5	22.1 ± 1.1	22.7 ± 0.34	0.012
contraceptive method choice	9	475	37.3 ± 1.5	37.3 ± 1.5	40.3 ± 2.2	0.01
ionosphere	34	97	16.8 ± 3.2	13.2 ± 1.5	20.1 ± 5.6	0.103
optdigits	64	48	1.65 ± 1.2	1.52 ± 1.1	1.73 ± 1.2	0.174
pageblocks	10	60	4.21 ± 2.1	3.11 ± 0.73	3.69 ± 1.3	0.215
spam	57	420	11.8 ± 1.3	10 ± 0.96	10.8 ± 0.78	0.262
australian	14	142	15.3 ± 1.6	13.6 ± 0.97	13.5 ± 0.9	0.54
diabetes	8	45	27.7 ± 3.1	27 ± 2.4	27.6 ± 2.4	0.588
dna	180	514	5.82 ± 0.83	3.59 ± 0.44	5.64 ± 0.52	0.0302
german	20	73	31.4 ± 2.7	30 ± 2.7	31.4 ± 2.5	0.5
heart	13	179	17.1 ± 3.3	16.7 ± 2.7	16.8 ± 2.9	0.426
letter	16	102	3.12 ± 1.6	2.63 ± 0.84	7.12 ± 1.5	0.00541
shuttle	9	31	9.76 ± 3.9	7.23 ± 2.9	9.79 ± 3.2	0.143
vehicle	18	288	4.37 ± 1.5	3.9 ± 1.5	31.5 ± 4.1	0.001

TAB. 5.3 – Average and standard deviation of the test error rate (in %) on real datasets over 50 random test/train splits. The data dimension and the training sample size are given in the first two columns. The columns err_{GEN}, $\text{err}_{\hat{\lambda}}$ and err_{DISC} correspond to the generative estimator ($\lambda = 1$), the GDT estimator with parameter $\hat{\lambda}$ selected by CV and the discriminative estimator ($\lambda = 1$), respectively. The last column is the average of $\hat{\lambda}$.

5.7 Bayesian Formulation of the GDT estimator

The generative-discriminative tradeoff was derived from in a *frequentist* point of view, that is to say, assuming that the parameters θ have fixed values. A Bayesian interpretation — viewing the parameter θ as random with distribution $\boldsymbol{p}(\theta)$ — is also usefull for two reasons. First, since we presented GDT as a discriminative estimate penalized by a generative term, one might wonder wether the penalty has a link with Bayesian prior, just as standard penalized criterion does. Secondly, adding the Bayesian paradigm to our method naturally extends the range of applications of GDT estimation.

Once the sample $(\mathbf{x}, \mathbf{y})$ is observed, the *a posteriori* distribution of θ is

$$\boldsymbol{p}(\theta|\mathbf{x}, \mathbf{y}) \propto \boldsymbol{p}(\mathbf{x}, \mathbf{y}|\theta)\boldsymbol{p}(\theta). \tag{5.18}$$

Similarly, modelling only the uncertainty in y, the discriminative version of the posterior distribution is

$$\boldsymbol{p}(\theta|\mathbf{x}, \mathbf{y}) \propto \boldsymbol{p}(\mathbf{y}|\mathbf{x}, \theta)\boldsymbol{p}(\theta). \tag{5.19}$$

the discriminative Bayesian estimator is obtained by maximizing this quantity.

The tradeoff beetween the generative and the discriminative Bayesian estimators is simply the maximization of a weighted sum of these generative and discriminative terms :

$$\hat{\theta}_{\lambda}^{Bayes} = \underset{\theta\in\Theta}{\operatorname{argmax}}\, \lambda \log \boldsymbol{p}(\mathbf{y}|\mathbf{x}\theta) + (1-\lambda)\log \boldsymbol{p}(\mathbf{x}, \mathbf{y}|\theta) + \boldsymbol{p}(\theta) \tag{5.20}$$

which naturally extents (5.7) by adding a prior term. This prior could be interpreted as a regularization term. The GDT estimator gives a pointwise estimator instead of a full distribution over the parameter space. A pure Bayesian approach would consider the *a posteriori* distribution given in (5.19). The *GDT posterior distribution* can be written as follows :

$$\boldsymbol{p}_{\lambda}(\theta|x, y) = \frac{\boldsymbol{p}(\theta)\boldsymbol{p}(\mathbf{x}|\theta)^{\lambda}\boldsymbol{p}(\mathbf{y}|\mathbf{x}, \theta)}{\int_{\Theta} \boldsymbol{p}(\theta)\boldsymbol{p}(\mathbf{x}|\theta)^{\lambda}\boldsymbol{p}(\mathbf{y}|\mathbf{x}, \theta)d\theta}. \tag{5.21}$$

With this definition of the Bayesian Generative-Discriminative Tradeoff, we have a smooth transition between the usual (generative) posterior and the discriminative posterior, which amount to assuming that the x-distribution is uniform. Moreover, we recover the MAP solution 5.20 when computing the maximum of 5.21.

Regularized LinearGDT We show here that the Bayesian estimates is usefull to regularize the estimation of the LinearGDT parameters. The *a priori* distribution of the LDA parameters is defined up to a normalization constant :

$$\boldsymbol{p}(\theta) \propto \exp\left(\nu \sum_{k>k'} (m_k - m_{k'})^T \Sigma^{-2} (m_k - m_{k'})\right) \tag{5.22}$$

where ν is a hyperparameter that defines the strength of the *a priori*. The proposed *a priori* distibution puts more weight on means that are close to the others and on high variances. The classes are therefore more likely to ovelap, so that this *a priori* has the same role as a reglarizer.

Definition 3 *Based on $\mathcal{L}_\lambda^{lin}$ of definition 2, the* Regularized LinearGDT *estimator is :*

$$\hat{\theta}_\lambda = \text{argmax}_{(m_1,m_2,\Sigma)} \mathcal{L}_\lambda^{reg}(\theta)$$

$$\mathcal{L}_\lambda^{reg}(\theta) = \mathcal{L}_\lambda^{lin}(\theta) - \nu \sum_{k>k'} (m_k - m_{k'})^T \Sigma^{-2} (m_k - m_{k'})$$

for $0 < \lambda \leq 1$ *and* $\hat{\theta}_0 = \lim_{\lambda \to 0} \hat{\theta}_\lambda$.

One interesting consequence of the *a priori* (5.22) is that the discriminative estimator is equivalent to a logistic regression regularized with the euclidian norm : in regularized logistic regression with two classes, the estimator is

$$\hat{\beta} = \text{argmin} \sum_{i=1}^{n} \log(1 + e^{\beta_0 + \beta_{1:d}^T x_i}) + \nu \|\beta_{1:d}\|^2.$$

And, according to the chapter 2, a continuous mapping between the LDA parameters $(m_{\,1}, m_2, \Sigma)$ and the linear logistic regression parameters β can be defined such that $\beta_{1:d} = \Sigma^{-1}(m_2 - m_1)$. Now, we see that the *a priori* defined in (5.22) can be rewritten $C^{te} \exp(\nu \beta_{1:d}^T \beta_{1:d})$, *i.e.* the penalization $\nu \|\beta_{1:d}\|^2$ of the regularized logistic regression.

To illustrate this regularization, we performed a small experiment on the benchmark datasets. To suit high dimensional distributions, we constrained the LinearGDT classifier to have a diagonal covariance matrices (NB model). For simplicity, we used only binary classifiers : in each dataset, the two classes with the largest sample size were selected. The number of training data was set to (at most) 4 times the feature dimensions. The error rate was computed on the remaining dat. Each experiment was repeated 20 times. In Figure 5.7 we give classification performance of this regularized classifier on the previous datasets. Clearly, regularization has a tendency to reduce the test error rate. The best performances were obtained either by more regularized classifiers or by more "generative" ($\lambda \to 1$) classifiers. Moreover, the models for which the generative asumptions (diagonal Gaussian distribution) are reasonable (*australian, diabetes* and *heart* are roughly Gaussians) tend to perform well with the GDT classification. In practice, the hyperparameter ν should be chosen by cross-validation.

This extension to the regularized logistic regression may have interesting applications. For example, this naturally helps us to include unlabelled data into logistic regression.

Dataset	australian	diabetes	dna	german	heart	letter	satimage
$n_{train}/d/n_{test}$	56/ 14/ 634	32/ 8/ 736	720/180/ 812	80/ 20/ 920	52/ 13/ 218	64/ 16/ 1491	144/ 36/ 2092
λ=0.01,ν=0.01	0.202	0.298	0.046	0.307	0.236	0.021	**0.014**
λ=0.01,ν=0.1	0.183	0.289	0.044	0.302	0.221	**0.021**	0.014
λ=0.01,ν=0.3	0.172	0.284	0.042	0.296	0.209	0.021	0.015
λ=0.01,ν=1	0.160	0.280	0.040	0.287	0.196	0.024	0.018
λ=0.01,ν=5	0.156	0.294	0.037	**0.276**	0.183	0.033	0.028
λ=0.01,ν=0.0	0.173	0.287	0.038	0.295	0.211	0.022	0.019
λ=0.02,ν=0.0	0.166	0.284	0.037	0.290	0.202	0.023	0.021
λ=0.05,ν=0.0	0.159	0.280	**0.037**	0.284	0.191	0.028	0.025
λ=0.1,ν=0.0	0.155	0.278	0.038	0.282	0.184	0.033	0.030
λ=0.15,ν=0.0	0.154	**0.276**	0.039	0.281	0.180	0.039	0.033
λ=.25,ν=0.0	**0.153**	0.277	0.042	0.283	0.176	0.046	0.037
λ=.5,ν=0.0	0.153	0.280	0.045	0.292	**0.175**	0.055	0.042
λ=1,ν=0.0	0.154	0.286	0.046	0.308	0.175	0.060	0.046

TAB. 5.4 – Test error rate (averaged over 20 experiments) on statlog datasets for regularized linearGDT classifier, and closest mean classification (generative). In this experiment, the number of data in equal to 4 times their dimension.

5.8 A robust GDT variant

We proposed a *linear* combination of the joint and the conditional likelihoods. Yet it is possible to define different mixes of these objective functions.

In this section, we propose a "robust" combination that puts more weight on features that do not fit well the generative model. Let $\varepsilon \geq 0$ and consider the optimization problem :

$$\max_{\theta \in \Theta} \log \boldsymbol{p}(\mathbf{y}|\mathbf{x};\theta) + \log\left(\varepsilon + \boldsymbol{p}(\mathbf{x};\theta)\right)$$

Thus, we obtain the generative solution for $\varepsilon = 0$, and the discriminative one when $\varepsilon \to \infty$. Data which give a small probability for $p(x)$ have less influence in the generative *regularization* term/ Hence the solution should be less sensitive to the *outliers*. In practice we link λ and ε by the relation $\epsilon = -\log(\lambda)$.

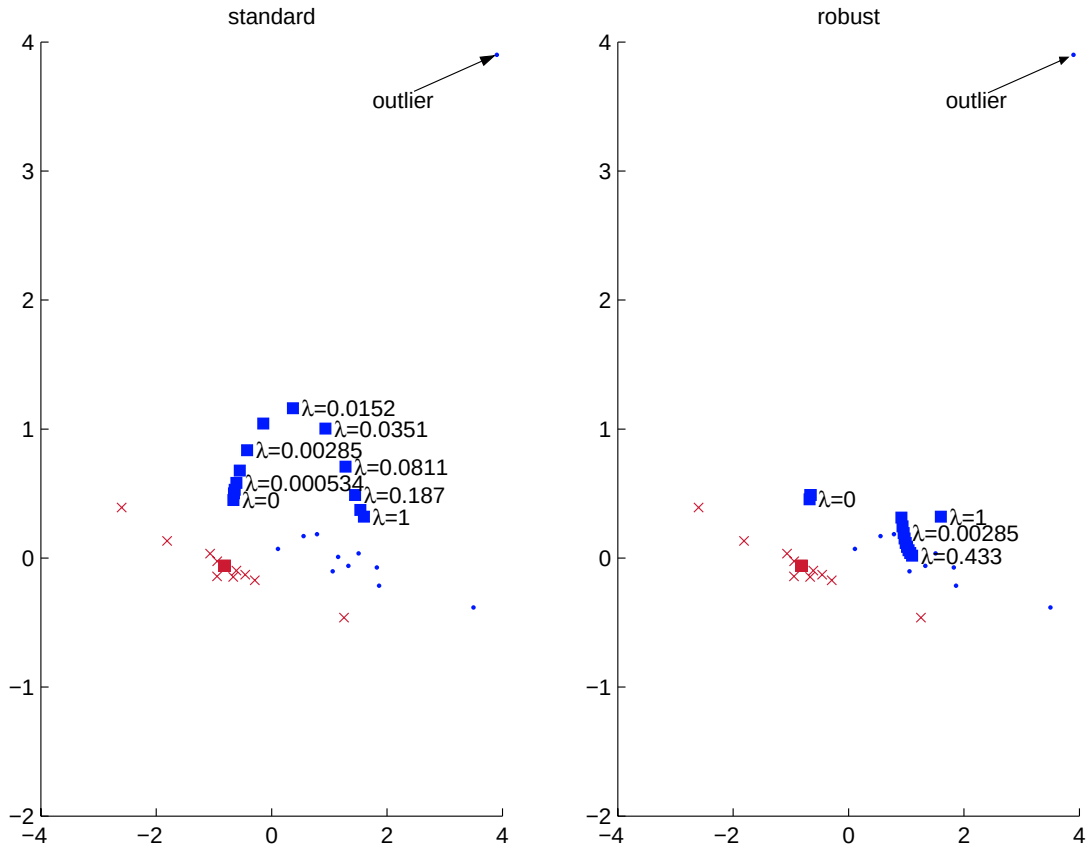


FIG. 5.7 – Comparaison between standard GDT and its robust variant. The squares represent the estimated means of the model for different values of the tuning paramter (λ for the standard GDT, $\varepsilon = -\log(\lambda)$ for the robust GDT variant).

A simple experiment involving two classes modelled with 1 Gaussian each was performed. We assumed that the mean and the variance of the first class is known, so that only the parameters of the second class are estimated. An outlier was added to the training sample of the second class to see its impact on the estimation. Multiple values for ε were tested. This is represented on figure 5.7. One can see that the mean of the robust GDT estimate does not move far away from the initial empirical mean (generative solution). Practically, it was difficult to observe a continuous transition of the robust GDT when $\varepsilon \to \infty$, essentially due to numerical instabilities[24]. Deeper theoretical anderstanding of this robust version of GDT are necessary for further studies.

[24]The logarithm representation of the the probabilities was still insufficient to garantee numerical stability for the algorithm.

5.9 Conclusion

In this study, the relationship between generative and discriminative estimates of generatively parametrized classifiers has been clarified : they correspond to two different of cost functions defined over the parameter space. By interpolating linearly between the two objective functions, we introduced the GDT estimator. The cost function that we defined keeps the probabilistic interpretation of the model outputs (as recommended in [143]) but still answers the users need for good classification performance. This can be seen either as a less biased variant of the generative solution, or as an improvement of the discriminative classifier. The regularization is “natural” in the sense that the parameters are encouraged to fit the inputs. Testing on real data showed that the intermediate model often gives better classification performances than both the discriminative and generative classifiers.

Currently, the main difficulty with the GDT method is the choice of the tuning parameter, as this requires an expensive cross-validation computation. We believe that more computationally efficient criteria can be developed by analyzing the solutions on the training set, in the spirit of AIC [8] or BIC [147].

Recent work focusing on discrimative parameter learning [87, 163, 64] may solve many of the difficulties resulting from the underlying optimization problem. Extentions of these methods to GDT estimation should be possible, and are promising directions for further work.

Appendix A : GDT computation for log-linear models

If the class proportions p and $1-p$ are known, the total parameter vector is $\theta = (\theta_1, \theta_0)$. To allow common parameters between classes, we introduce a more convenient notation by defining A_0 and A_1 so that $\theta_1 = A_1\theta$ and $\theta_0 = A_0\theta$. When the parameters are not functionally linked, the matrices A_1 and A_0 are

$$A_1 = \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 & 0 & \cdots & 0 \\ & & \ddots & & & \vdots & \\ 0 & 0 & \cdots & 1 & 0 & \cdots & 0 \end{bmatrix} \quad A_0 = \begin{bmatrix} 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 0 & 1 & \cdots & 0 \\ & \vdots & & & & \ddots & \\ 0 & \cdots & 0 & 0 & 0 & \cdots & 1 \end{bmatrix}$$

This notation is useful in the case of dependent parameters : in the case of LDA, the class-conditional densities are Gaussian with a common covariance matrix. Hence, the whole set of parameters is

$$\theta = (m_{11}, \cdots, m_{1d}, m_{01}, \cdots, m_{0d}, \tau_{11}, \tau_{12}, \cdots, \tau_{dd})$$

where $m_{i1}, \cdots, m_{id}$ are the mean parameters of the class i and $\tau_{11}, \tau_{12}, \cdots, \tau_{dd}$ are the parameters of the inverse (shared) covariance matrix. In this case, the matrices A_i have the form $A_1 = [I_d, 0, I_d]$ and $A_0 = [0, I_d, I_d]$.

Now, both class-conditional densities depend on a common parameter vector θ :

$$\log \boldsymbol{p}(x|y=k,\theta) = \langle x^T, A_k\theta\rangle - \alpha_k(A_k\theta) + \psi_k(x) \quad \text{for } k \in \{0,1\} \tag{5.23}$$

Derivatives of the joint likelihood

The generative likelihood of a sample $(\mathbf{x}, \mathbf{y})$ is

$$\begin{aligned} \mathcal{L}_J &= \sum_{i=1}^{n} \langle x_i, A_{y_i}\theta\rangle - \alpha_{y_i}(A_{y_i}\theta) + \psi_{y_i}(x_i) + y_i \log p + (1-y_i)\log(1-p) \\ &= n\left(\langle \overline{\mathbf{xy}}, A_1\theta\rangle - \bar{\mathbf{y}}\alpha_1(A_1\theta) + \langle \overline{\mathbf{x}(1-\mathbf{y})}, A_0\theta\rangle - (1-\bar{\mathbf{y}})\alpha_0(A_0\theta)\right) + \mathrm{C}^{\mathrm{te}}(\mathbf{x},\mathbf{y}) \\ &= n\left(\langle A_1^T\overline{\mathbf{xy}} - A_0^T\overline{\mathbf{x}(1-\mathbf{y})}, \theta\rangle - \bar{\mathbf{y}}\alpha_1(A_1\theta) - (1-\bar{\mathbf{y}})\alpha_0(A_0\theta)\right) + \mathrm{C}^{\mathrm{te}}(\mathbf{x},\mathbf{y}) \end{aligned} \tag{5.24}$$

where $\overline{\boldsymbol{z}} = \frac{1}{n}\sum_i \boldsymbol{z}_i$. Equation (5.24) shows that the joint distribution is still an exponential family with parameter θ. The ϕ, α and ψ functions for this family for a single sample are :

$$\begin{aligned}
\phi(x,y) &= A_1^T\overline{\mathbf{xy}} - A_0^T\overline{\mathbf{x}(1-\mathbf{y})} \\
\alpha(\theta) &= \bar{\mathbf{y}}\alpha_1(A_1\theta) + (1-\bar{\mathbf{y}})\alpha_0(A_0\theta) \\
\psi(x,y) &= \sum_{i=1}^{n}(y_i\psi_1(x_i) + (1-y_i)\psi_0(x_i)) + \bar{\mathbf{y}}\log p + (1-\bar{\mathbf{y}})\log(1-p)
\end{aligned}$$

First derivative Differenting (5.24) by θ gives

$$\frac{\partial \mathcal{L}_J}{\partial\theta}(\theta) = n\left(A_1^T\overline{\mathbf{xy}} - A_0^T\overline{\mathbf{x}(1-\mathbf{y})} - \bar{\mathbf{y}}A_1^T\alpha_1'(A_1\theta) - (1-\bar{\mathbf{y}})A_0^T\alpha_0'(A_0\theta)\right) = nv \tag{5.25}$$

where $\alpha_k'(\tau) := \frac{\partial}{\partial\tau}\alpha_k(\tau)$. In generative ML, one sets this quantity to zero and solves it for θ. Since α_k are strictly convex function, the solution is unique.

Hessian of the joint likelihood Another differentiation for θ gives

$$\begin{aligned}
\frac{\partial^2 \mathcal{L}_J}{\partial\theta\partial\theta^T}(\theta) &= -n\left(\bar{\mathbf{y}}A_1^T\alpha_1''(A_1\theta)A_1 + (1-\bar{\mathbf{y}})A_0^T\alpha_0''(A_0\theta)A_0\right) \\
&= -n\left(\bar{\mathbf{y}}H_1 + (1-\bar{\mathbf{y}})H_0\right) \qquad (5.26)
\end{aligned}$$

with $\alpha_k''(\tau) := \frac{\partial^2}{\partial\tau\partial\tau^T}\alpha_k(\tau)$ and $H_k := A_k^T\alpha_k''(A_k\theta)A_k$ for $k \in \{0,1\}$. Since we assumed α_k to be convex, the Hessian matrix is definite negative at the solution, and the generative likelihood has a unique maximum.

Conditional likelihood derivatives

Noting that

$$\log\frac{\boldsymbol{p}(x,y=1;\theta)}{\boldsymbol{p}(x,y=0;\theta)} = x^T(A_1-A_0)\theta - \alpha_1(A_1\theta) + \alpha_0(A_0\theta) + \psi_1(x) - \psi_0(x) + \log\frac{p}{\mathbf{1}-p},$$

the probability of assigning a point x to the class 1 has the logistic form

$$\boldsymbol{p}(y=1|x;\beta,\beta_0) = \frac{1}{1+e^{-x^T\beta-\beta_0-(\psi_1(x)-\psi_0(x))}} \tag{5.27}$$

where the parameters β and β_0 depend on θ :

$$\beta = (A_1-A_0)\theta \tag{5.28}$$

$$\beta_0 = -(\alpha_1(A_1\theta) - \alpha_0(A_0\theta)) + \log\frac{p}{\mathbf{1}-p} \tag{5.29}$$

Hence, the conditional likelihood depends only on β and β_0 :

$$\mathcal{L}_C(\theta) \quad = \quad \sum_{i=1}^{n} \log \boldsymbol{p}(x_i|y_i, \beta, \beta_0) = \sum_{i=1}^{n} \log p_i \tag{5.30}$$

where $p_i = \boldsymbol{p}(x_i|y_i, \beta, \beta_0)$ for $i = 1, \cdots, n$.

Equation (5.27) is a linear logistic regression when $\psi_1 = \psi_0$. In this case the discriminative estimate of (β, β_0) can be found rising standard logistic estimation techniques, such as the *Iteratively Reweighted Least Squares* algorithm. An important point here is that the parameterization (β, β_0) is sufficient to classify new data, without finding the full parameter set θ. So, once the GDT estimator $\hat{\theta}_\lambda$ is found on the training data, it suffice (but it is not necessary) to save the minimal parametrization $(\hat{\beta}_\lambda, \hat{\beta}_{0\lambda})$.

First derivative. Using the chain to differentiate (5.27) :

$$\begin{aligned} \frac{\partial \mathcal{L}_C}{\partial \theta}(\theta) &= \sum_{i=1}^{n} \left(\frac{\partial \beta}{\partial \theta} x_i - \frac{\partial \beta_0}{\partial \theta} \right) (y_i - \boldsymbol{p}(x_i|y_i = 1; \beta, \beta_0)) \\ &= \sum_{i=1}^{n} \left(A_1^T (x_i - \alpha_1'(A_1\theta)) - A_0^T (x_i - \alpha_0'(A_0\theta)) \right) (y_i - p_i) \\ &= \sum_{i=1}^{n} \tilde{x}_i \, (y_i - p_i) \end{aligned} \tag{5.31}$$

were we defined $\tilde{x}_i := A_1^T (x_i - \alpha_1'(A_1\theta)) - A_0^T (x_i - \alpha_0'(A_0\theta))$.

Hessian of the conditional likelihood

$$\begin{aligned} \frac{\partial^2 \mathcal{L}_C}{\partial \theta \partial \theta^T}(\theta) = & - \sum_{i=1}^{n} \left(A_1^T \alpha_1''(A_1\theta) A_1 - A_0^T \alpha_0''(A_0\theta) A_0 \right) (y_i - p_i) \\ & - \sum_{i=1}^{n} \tilde{x}_i \tilde{x}_i^T p_i (1 - p_i) \\ & = -(H_1 - H_0) \sum_{i=1}^{n} (y_i - p_i) - \sum_{i=1}^{n} \tilde{x}_i \tilde{x}_i^T p_i (1 - p_i). \end{aligned} \tag{5.32}$$

where H_k are the matrices defined in (5.26). Here, we cannot guarantee on the definitness of the Hessian. It has to be checked for each individual model.

Newtow-Raphson algorithm

To maximize the GDT loss function, we use the Newton-Raphson algorithm. Starting from a initial point θ^0, we update the parameter with

$$\theta^{t+1} = \theta^t - \left(\frac{\partial^2 \mathcal{L}_\lambda(\theta^t)}{\partial\theta\partial\theta^T} \right)^{-1} \frac{\partial \mathcal{L}_\lambda(\theta^t)}{\partial\theta}$$

In matrix notation,

$$\frac{\partial \mathcal{L}_\lambda(\theta)}{\partial\theta} = \lambda n v + (1-\lambda)\tilde{X}^T(\boldsymbol{y} - \boldsymbol{p}) \tag{5.33}$$

$$-\frac{\partial^2 \mathcal{L}_\lambda(\theta)}{\partial\theta\partial\theta^T} = \lambda n \left(\bar{\mathbf{y}} H_1 + (1-\bar{\mathbf{y}})H_0\right) + (1-\lambda)(\bar{\mathbf{y}} - \bar{\boldsymbol{p}})(H_1 - H_0) + \tilde{X}^T W \tilde{X} \tag{5.34}$$

where v is defined in (5.25) and $\tilde{X}$ is the matrix containing the vectors $\tilde{x}_i$ in its rows.

Appendix B : Proofs

5.9.1 Asymptotic optimality

We already now that the generative estimator has a smaller asymptotic classification bias than the discriminative one. Thus, the variance is a continuous function such that variance$(1) >$ variance(0) which tends to zero as n grows. Conversely, the bias term does not depend on the estimate and is at a minimum for the discriminative solution : bias$(0) = b_0$, where b_0 is the minimum classification loss that can be reached with the parametric family.

The GDT sums two objective functions. It is not obvious that the sum performs better than either of the initial functions. In order to show that the GDT estimator is m !aningful, we give in this section the theroretical justification that the bias-variance tradeoff exists, a least asymptotically. The proofs are given for a one-dimensional parameter space. The extension to the multidimensional case involves no major difficulties and is discussed at the end of the proofs (in Appendix B).

We look at the behavior of the bias term :

Lemma 1 *If $E(\log(\boldsymbol{p}(Y|X;\theta))$ and $E(\log(\boldsymbol{p}(X;\theta))$ have bounded second derivatives around θ^*_C,*

$$\frac{\partial}{\partial\lambda} \textit{bias}(\lambda)\Big|_{\lambda=0} = 0, \tag{5.35}$$

where bias$(\lambda) = L_C(\theta^*_\lambda) - L_C(\theta^*_C)$.

This leads to a proposition in favor of discriminative estimation :

Proposition 5 *If the score of the marginal likelihood is not zero at the asymptotic generative solution* $\hat{\theta}_1$, i.e.

$$E\left[-\frac{\partial}{\partial\theta}\log \boldsymbol{p}(X;\theta^*_C)\right] \neq 0,$$

then $\lambda^* \to 0$ *as* $n \to \infty$.

PROOF. The variance tends to zero as n tends to infinity and the score different from zero implies that the bias is strictly positive. Hence, for sufficiently large training size n', the variance is small compared to the bias. Since the bias is non negative, the lemma 1 states that for all $\lambda > 0$, there exist $0 < \lambda' < \lambda$ such that bias$(\lambda') <$ bias(λ) for a training size n'' large enough. Hence, choosing $n = \max(n', n'')$, a solution with a smaller λ has to be preferred. This means that the optimal λ tends to zero as n grows. □ Note that this proposition enables us to associate the bias to score of the marginal model. It is important since a the joint model can both be wrong and have zero bias. In this case, the generative estimation is better than any discriminative or GDT estimator, since it is an unbiased estimator that reaches the Cramer-Rao bound.

At this point, one could state that the discriminative estimator should be chosen. In fact, there is an advantage in using a GDT estimator, since lemma 1 also implies that the bias grows less that linearly with the parameter λ, With the additional remark that the variance decrease linearly from $\lambda = 0$, the overall loss $\eta(\lambda)$ will also decrease linearly from 0. The second lemma gives some necessary conditions to obtain a negative derivative for the variance term :

Lemma 2 *If*

1. *the discriminative model is true, i.e. there exists a parameter* $\theta^*_C \in \Theta$ *such that* $\boldsymbol{p}(y|x;\theta^*_C)$ *equals the sample distribution* $\boldsymbol{p}^*(y|x)$,
2. *the regularity conditions required for the maximum likelihood estimation of* θ^*_C *are satisfied and*
3. *the marginal likelihood is locally convex around the discriminative solution, i.e.*

$$E\left[-\frac{\partial^2}{\partial\theta\partial\theta^T}\log \boldsymbol{p}(X;\theta^*_C)\right] > 0$$

then

$$\frac{\partial}{\partial\lambda}\textit{variance}(\lambda)\Big|_{\lambda=0} = -\frac{C^{te}}{n} < 0 \tag{5.36}$$

holds, where C^{te} *does not depend on* λ *nor* n *and variance(*λ*)=*$E\left[L_C(\hat{\theta}_\lambda) - L_C(\theta^*_\lambda)\right]$.

The crucial point in this lemma appears is the third assumption : the term $E\left[-\frac{\partial^2}{\partial\theta\partial\theta^T}\log \boldsymbol{p}(X;\theta^*_C)\right]$ is the Fisher information matrix of the marginal model at the discriminative solution. Assuming that this term is positive states that that the distribution of X contains some information about θ. This is precisely the reason why the GDT estimation should work : the estimator uses this information to improve the estimation. The performance improvement is illustrated in the lemma 2 : the variance of the test classification loss decreases when λ moves away from 0.

The first assumption in lemma 2 can be thought of restrictive since in general, the conditional distribution does not belong to a particular parametric family. However, we do not impose [25] a specific form for the distribution of the regressors x. It is therefore far less restrictive than assumming that the joint density belongs to $\mathcal{F}$.

To be more precise, the assumption that the conditional distribution $p(y|x;\theta)$ belongs to the parametric family can be relax. See the proof of the lemma 2 for details. These more general asumption involves a bound on the third derivatives according to the parameters, and the more the bias is high, the more the asumption is restrictive.

The results from the two lemmas can be added to state that $\frac{\partial}{\partial\lambda}\eta(0) < 0$. This shows that there exists λ^*_1 such that

$$E\left[L(\hat{\theta}_{\lambda^*})\right] < E\left[L(\hat{\theta}_C)\right], \tag{5.37}$$

that it to say we can do better than the discriminative estimator.

If we assume that the generative solution is biased, then for n sufficiently large, the discriminative estimator has a smaller classification loss on average, prooving that the GDT estimator $\hat{\theta}_{\lambda^*}$ should be chosen among the three estimators, i.e.

$$E\left[L(\hat{\theta}_{\lambda^*})\right] < E\left[L(\hat{\theta}_C)\right] < E\left[L(\hat{\theta}_J)\right], \tag{5.38}$$

That is to say, for n sufficiently large, we can increase the classification performances of the discriminative *and* the generative estimator. This is summarized in the theorem 1 page 107.

[25] The distribution of the inputs X was used to build the model, as usual in generative classification, but it is not required here that this model is true on the data.

This result gives an additional information which is a bound to the speed of decrease (up to a scale factor) of the optimal λ to 0. It is proportional to the inverse of the training sample size, and they are strong reasons for thinking that this bound is actually reached (see the proof in the appendix for details). Yet, the theorem does not directly give the true solution λ^* since the scale factor is not known, and the bound (5.14) is very loose. It can be noticed that the constant C include term up to the fourth derivative according to the model parameters. Hence, apart toy examples, a direct approximation of this coefficient is intractable. Some further studies could be done on the approximation of this constant.

Notations used in the proofs

The GDT loss function for one sample (X, Y) is $L_\lambda(\theta) = -E\left[\log \boldsymbol{p}(Y|X;\theta) - \lambda \log \boldsymbol{p}(X;\theta)\right]$. The parameter minimizing this quantity is θ^*_λ. When λ is replaced by C, it corresponds to the discriminative (or Conditional) solution $\lambda = 0$. For example, $\theta^*_C = \theta^*_0$. The marginal loss function is $L_M(\theta) = -E\left[\log \boldsymbol{p}(x;\theta)\right]$. The first derivative of this quantity is

$$S_M(\theta) = \frac{\partial}{\partial\theta} L_M(\theta).$$

The second derivatives of the loss functions are

$$J_\lambda(\theta) = \frac{\partial^2}{\partial\theta^2} L_\lambda(\theta), \quad J_C(\theta) = \tfrac{\partial^2}{\partial\theta^2} L_0(\theta), \quad \text{and} \quad J_M(\theta) = \frac{\partial^2}{\partial\theta^2} L_M(\theta).$$

Finally, we define the GDT Fisher information :

$$K_\lambda(\theta) \quad = \quad E\left[(\frac{\partial}{\partial\theta}\log \boldsymbol{p}(y|x;\theta) + \lambda\frac{\partial}{\partial\theta}\log \boldsymbol{p}(x;\theta))^2\right],$$

which is a second degree polynom in λ :

$$K_\lambda(\theta) = K_C + 2\lambda\tilde{K} + \lambda^2 K_M$$

where K_C=K_0 and

$$\tilde{K} = E\left[\frac{\partial}{\partial\theta}\log \boldsymbol{p}(y|x;\theta)\frac{\partial}{\partial\theta}\log \boldsymbol{p}(x;\theta)\right] \quad \text{and} \quad K_M = E\left[(\frac{\partial}{\partial\theta}\log \boldsymbol{p}(x;\theta))^2\right].$$

Proof of lemma 1

We prove here that the first derivative of the bias^2 term vanishes at the discrimative solution. Using the chain rule,

$$\left.\frac{\partial \text{bias}(\lambda)}{\partial \lambda}\right|_{\lambda=0} = \left.\frac{\partial L_C(\theta^*_\lambda)}{\partial \lambda}\right|_{\lambda=0} = \frac{\partial}{\partial \theta} L_C(\theta^*_C) \left.\frac{\partial \theta^*_\lambda}{\partial \lambda}\right|_{\lambda=0}. \tag{5.39}$$

We know that the expected values of the GDT estimator is the solution (assumed to be unique) of the following minimization problem :

$$\theta^*_\lambda = \operatorname{argmin} L_C(\theta) + \lambda L_M(\theta) = \operatorname{argmin} L_\lambda(\theta). \tag{5.40}$$

This implies $\frac{\partial}{\partial \theta} L_\lambda(\theta^*_\lambda) = 0$. In particular, for $\lambda = 0$, we have $\frac{\partial}{\partial \theta} L_C(\theta^*_C)$=0. Hence, the result holds if the second part is bounded. Differentiating 5.40 according to λ gives

$$J_\lambda(\theta^*_\lambda)\frac{\partial \theta^*_\lambda}{\partial \lambda} + S_M(\theta^*_\lambda) = 0.$$

Since the second derivatives are bounded by hypothese, the quantity

$$\frac{\partial \theta^*_\lambda}{\partial \lambda} = -\frac{S_M(\theta^*_\lambda)}{J_\lambda(\theta^*_\lambda)} \tag{5.41}$$

is bounded, so that the expression (5.39) equals 0, prooving the lemma. □

Proof of lemma 2

To prove that the quantity $E\left[L(\hat{\theta}_\lambda) - L(\theta^*_\lambda)\right]$ that we named *variance* in (5.17) decreases when λ moves away from zero, we follow the steps :

1. the asymptotic distribution of the estimator $\hat{\theta}_\lambda$ is found :

$$\sqrt{n}(\hat{\theta}_\lambda - \theta^*_\lambda) \xrightarrow{\mathcal{D}} \mathcal{N}\left(0, \frac{K_\lambda(\theta^*_\lambda)}{J^2_\lambda(\theta^*_\lambda)}\right), \tag{5.42}$$

2. the variance is expressed as a function of λ :

$$\text{variance}(\lambda) = \frac{1}{2n}\frac{J_C(\theta^*_\lambda)K_\lambda(\theta^*_\lambda)}{J^2_\lambda(\theta^*_\lambda)},$$

3. then the derivative of this is considered in 0 :

$$n\frac{\partial}{\partial\lambda}\text{variance}(0) = -\frac{J_M}{J_C^2} + \frac{S_M(K_C\frac{\partial}{\partial\theta}J_C - \frac{1}{2}J_C\frac{\partial}{\partial\theta}K_C)}{J_C^3} \tag{5.43}$$

   where (to lighten the notations) all the terms are computed at point θ^*_λ.

4. Finally, we prove that the true discriminative model assumption implies that $2K_C\frac{\partial}{\partial\theta}J_C - J_C\frac{\partial}{\partial\theta}K_C = 0$ so that

$$\frac{\partial}{\partial\lambda}\text{variance}(0) = -\frac{J_M}{nJ_C^2}$$

   which is negative as soon as the third hypothese $J_M > 0$ is verified.

We now follow the steps in order. The large sample theory gives us the asymptotic distribution of any minimizer of the form $\hat{\theta} = \text{argmin}_{\theta\in\Theta}\sum_{i=1}^n C(z_i;\theta)$, where z_i are i.i.d. random variables. By the the law of large number, $\hat{\theta}$ will tend to the minimum of the function $E\left[C(Z;\theta)\right]$, Z having the same distribution as the data. Let θ^* be the parameter value minimizing this function. Under weak regularity conditions on the data distribution and on the cost function, $\hat{\theta}\to\theta^*$ almost surely (See for example for Huber, 1967 [76], Ripley, 1996 [137]). Now we consider a useful result about the asymptotic distribution of such estimators : Under regularity conditions (A), $\sqrt{n}(\hat{\theta} - \theta^*) \longrightarrow^{\mathcal{L}} \mathcal{N}(0, J^{-1}KJ^{-1})$ where $J = E\left[\frac{\partial^2 C(z;\theta^*)}{\partial\theta\partial\theta^T}\right]$ and $K = Var\left[\frac{\partial C(z;\theta^*)}{\partial\theta}\right]$. See Ripley [137] for a proof on the special case of maximum likelihood estimators where $C(z;\theta) = -\log\boldsymbol{p}(z;\theta)$. The extension to a general cost functions is straightforward. Here, this result is applied with $z_i = (x_i, y_i)$ and $C = -\log\boldsymbol{p}(y|x;\theta) - \lambda\log\boldsymbol{p}(x;\theta)$, leading to the equation (5.42). Here we showed only the result for the one-dimensioanl case.

The second step is the application of the function L_C to the variable $\hat{\theta}_\lambda$. Since it is twice differentiable around θ^*_λ, the first step implies that

$$n(L_C(\hat{\theta}_\lambda) - L_C(\theta^*_\lambda)) \xrightarrow{\mathcal{D}} \frac{1}{2}J_C(\theta^*_\lambda)w^2$$

where $w\sim\mathcal{N}\left(0, \frac{K_\lambda(\theta^*_\lambda)}{J^2_\lambda(\theta^*_\lambda)}\right)$. Taking the expectation gives

$$\text{variance}(\lambda) = \frac{1}{2}J_C(\theta^*_\lambda)E\left[z^2\right] = \frac{1}{2n}\frac{J_C(\theta^*_\lambda)K_\lambda(\theta^*_\lambda)}{J^2_\lambda(\theta^*_\lambda)}.$$

The third step, standard derivation rules are applied :

$$2n\frac{\partial}{\partial\lambda}\text{variance}(\lambda) = J_C\frac{J_\lambda\mathbf{D}_\lambda K_\lambda - 2K_\lambda\mathbf{D}_\lambda J_\lambda}{J^3_\lambda} \tag{5.44}$$

where $\mathbf{D}_\alpha$ is the total differentiation operator according to the parameter α. We have simplified the notations since all the functions are evaluated at point θ^*_λ : $J_\lambda = J_\lambda(\theta^*_\lambda)$, $J_M = J_M(\theta^*_\lambda)$, $K_\lambda = K_\lambda(\theta^*_\lambda)$ and $S_M = S_M(\theta^*_\lambda)$. Remarking that $\frac{\partial \theta^*_\lambda}{\partial \lambda}$ was given in formula (5.41), we can find the two remaining differentiations.

$$\mathbf{D}_\lambda J_\lambda = J_M - \frac{S_M}{J_\lambda}\frac{\partial J_\lambda}{\partial \theta} \tag{5.45}$$

$$\mathbf{D}_\lambda K_\lambda = 2(\tilde{K} + \lambda K_M) - \frac{S_M}{J_\lambda}\frac{\partial K_\lambda}{\partial \theta} \tag{5.46}$$

At the discriminative solution θ^*_C, $\tilde{K}$ equal zeros :

$$\tilde{K} = E\left[E\left[\frac{\partial}{\partial \theta}\log \boldsymbol{p}(y|x;\theta^*_C)\middle| x\right]\frac{\partial}{\partial \theta}\log \boldsymbol{p}(x;\theta^*_C)\right] \tag{5.47}$$

$$= E\left[\frac{\partial}{\partial \theta}E\left[\log \boldsymbol{p}(y|x;\theta^*_C)|\, x\right]\frac{\partial}{\partial \theta}\log \boldsymbol{p}(x;\theta^*_C)\right] = 0 \tag{5.48}$$

since θ^*_C maximizes $E\left[\log \boldsymbol{p}(x;\theta^*_C)\right]$. The two terms (5.45) and (5.46) are now injected into equation (5.44) to get the expression (5.43). At this point we get a sufficient condition on the parametric family to give a negative derivative for the variance at $\lambda = 0$:

$$\frac{J_M}{S_M} > K_C\left(\frac{\partial}{\partial \theta}\log J_C - -\frac{1}{2}\frac{\partial}{\partial \theta}\log K_C\right). \tag{5.49}$$

In the last step we show that this condition is satisfied if the sample distribution $\boldsymbol{p}(y|x)$ belongs to the parameteric family. First, it is well known that this assumption implies $K_C = J_C$ (see [137], proposition 2.2 pp.32, applied to the conditional distribtuion of Y given X) so that the condition (5.49) is verified. It is easy to check that for any three times diffentiable function u with derivatives u', u'' and u''',

$$(\log(u))''' + (\log(u))'(\log(u))'' = (2(u')^3 + u^2u''')/u^3.$$

We apply this formula to $u = \log \boldsymbol{p}(y|x;\theta)$: $\frac{\partial}{\partial \theta}K_C = \frac{1}{2}\frac{\partial}{\partial \theta}J_C$.

$$\frac{\partial}{\partial \theta}K_C - \frac{1}{2}\frac{\partial}{\partial \theta}J_C = E\left[\frac{\partial}{\partial \theta}\log \boldsymbol{p}(y|x;\theta^*_C)\frac{\partial^2}{\partial \theta^2}\log \boldsymbol{p}(y|x;\theta^*_C) + \frac{\partial^3}{\partial \theta^3}\log \boldsymbol{p}(y|x;\theta^*_C)\right]$$

$$= 2E\left[\left(\frac{\partial}{\partial \theta}\log \boldsymbol{p}(y|x;\theta^*_C)\right)^3\right] + E\left[\int \frac{\partial^3}{\partial \theta^3}\boldsymbol{p}(y|x;\theta^*_C)dy\right] = 0$$

since the asymptotic law of the score $\frac{\partial}{\partial \theta}\log \boldsymbol{p}(y|x;\theta^*_C)$ is symetric and the bounded third derivatives imply that $\int \frac{\partial^3}{\partial \theta^3}\boldsymbol{p}(y|x;\theta^*_C)dy = \frac{\partial^3}{\partial \theta^3}\int \boldsymbol{p}(y|x;\theta^*_C)dy = \frac{\partial^3}{\partial \theta^3}1 = 0$. Thus, the inequality (5.49) is true, which concludes the fourth step and the lemma. □

Proof of theorem 1

Assuming that the model admit bounded fourth derivatives, a Tailor expansion of the expected loss function η can be obtained

$$\eta(\lambda) = E\left[L(\hat{\theta}_\lambda) - L(\theta_C^*)\right] = a - \frac{b}{n}\lambda + c\frac{\lambda^2}{2} + o(\lambda^3)$$

The Taylor expansion is valid since λ is arbitrarly close to zero for n large enough. From lemmas 1 and 2, we know that $b > 0$ and the constants a and b do not depend on n. Note that the coefficient c equals

$$c = \frac{\partial^2}{\partial\lambda^2}\left(\text{bias}(\lambda) + \text{variance}(\lambda)\right) = \frac{\partial^2}{\partial\lambda^2}\text{bias}(\lambda) + o(\frac{1}{n})$$

since the variance decreases a speed $\frac{1}{n}$. A full differentiation at $\lambda = 0$ gives

$$c \approx \frac{\partial^2}{\partial\lambda^2}\text{bias}(0) = -\frac{S_M(S_M - \frac{\partial}{\partial\theta}J_C)}{J_C^3} \tag{5.50}$$

where each term is evaluated at the discriminative solution θ_C^*. They are three possible situations concerning the minimum of η. depending on the sign of the constant c.

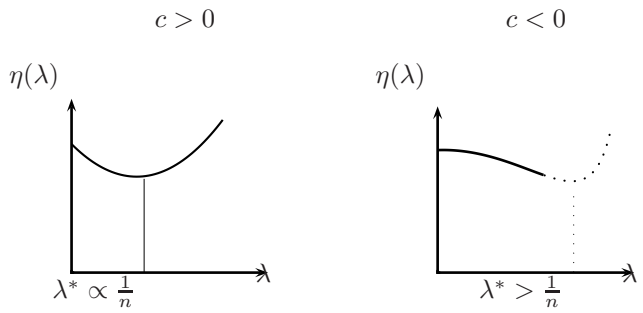


FIG. 5.8 – Different behaviors of the expected loss η against λ. The constant c is the value for the second derivative at $\lambda = 0$. For positive value of this constant, the optimal GDT parameter decrease at a speed $\frac{1}{n}$, whereas a negative c implies a minimum superior to this bound.

- If $c < 0$ the function η is locally concave. The minimum cannot be reached superior in the interval (say $[0, \frac{1}{n}]$) for which the Taylor expansion is a valid approximation. Hence λ^* decrease to zero at a speed strictly superior to $\frac{1}{n}$.
- If $c > 0$ the function η is convex so the value $\lambda^* = \frac{b}{nc}$ for which the derivative equals zero is a local minimum. Due to the fact that $\lambda^* \to 0$ (by hypothese), it becomes the global minimum for n sufficiently

large. The proposition 5 implies that this global minimum is unique for n sufficiently large.

– If $c = 0$, the tailor expansion must be extended to higher orders to decide if the function is locally concave or convex. If q is the minimum order that does not give a zero derivative, then, the same reasoning gives the bound $\lambda^* \geq \frac{b}{c} n^{-\frac{1}{q-1}} > \frac{b}{cn}$. Hence the speed is slower than for $c > 0$.

The cases $c < 0$ and $c > 0$ are illustrated on Figure 5.8. Hence, for all value of c, $\lambda^* \geq \frac{\text{C}^{\text{te}}}{n}$. We assumed that the marginal model is biased, which implies $\lambda^* < 1$ for n sufficiently large (due to proposition 5), thus the theorem is established. □

If the quantity $\frac{\partial J_C}{\partial \theta}$ equals zero (as for linear regression with linear constrains), or if the bias S_M is large compared to it, the expression (5.50) can be replaced by $\frac{S_M J_M}{J_C^3}$ and since $b = \frac{J_M}{J_C^2}$, the optimal parameter has a simple expression :

$$\lambda^* = \frac{J_C J_M}{n S_M^2}. \tag{5.51}$$

This expression is interesting, since it mixes three important quantities in the GDT estimation : J_C represents the fit to the conditional model, n is the training sample size and $\frac{S_M^2}{J_M}$ is a measure of the relative bias in the marginal model. A correct choice of λ should balance these three terms.

Multidimensional parameter space The proofs were given for a scalar parameter only. To extend it to vector parameters, the second derivatives are transformed into Hessian matrices. The semi-definiteness is guaranted by the model assumptions, so that fractions are replaced by inverse matrices. The details are not given here due to the cumbersome notations of the third and fourth derivatives (the matrix notation is not possible for these quantities, so the equations need to be decomposed into sums). The details of this extended proof will be given in future studies.

Chapitre 6

Un modèle hiérarchique des parties pour la catégorisation d'objets

Dans de nombreuses applications, la prise en compte de la structure des données est déterminante pour obtenir une règle de classification efficace. Ce chapitre donne un exemple de modélisation générative pour résoudre un problème de *catégorisation d'objets*. Nous cherchons à reconnaître la classe d'un objet dans une image numérique. Afin de construire la règle de classification, nous disposons de plusieurs images d'apprentissage dont la catégorie est connue. Un objet est généralement défini par une forme générique, plus ou moins variable. Par exemple, les images de chaises, d'animaux, de corps humain ont une forme généralement plus variable qu'une image de voiture ou une tête humaine. Les objets que nous considérons peuvent être localisés dans une image et se différencient de manière plus ou moins nette du fond. Dans ces exemples, la structure spatiale des primitives graphiques est déterminante. Nous proposons un modèle génératif qui tient compte de la géométrie et de l'apparence des catégories d'objets. Des descripteurs locaux d'images invariants par changement d'échelle, appelés *points d'intérêt*, sont utilisés comme primitives graphiques. Considérant que les objets sont composés de plusieurs *parties* distinctes, celles-ci sont associées à un ensemble de points d'intérêts. Les affectations entre parties et points d'intérêts se font de manière probabiliste. Notre méthode modélise la distribution de ces points d'intérêts. Nous sommes donc dans un cadre typique de modèle génératif : l'entrée X représente les points d'intérêt et la sortie Y la catégorie de

l'objet.

L'ensemble des positions des descripteurs et des parties est modélisé de manière hiérarchique, afin de coder des déformations géométriques de parties d'objets contenant plusieurs détections. L'affectation des sous-parties aux parties n'est pas déterminée *a priori*, mais apprise sur des images d'apprentissage. La méthode permet de gérer efficacement plusieurs centaines de descripteurs locaux, ce qui la rend particulièrement adaptée aux méthodes actuelles basées sur les points d'intérêt [111, 118]. De plus, le fait de prendre en compte un grand nombre de primitives rend la méthode plus robuste à des erreurs de détection.

Nous effectuons un apprentissage génératif du modèle car l'algorithme EM s'obtient facilement. De plus, cela permet d'estimer les paramètres des classes indépendamment les uns des autres. Les objets n'ont pas besoin d'être localisés *a priori* sur l'image car la position des parties du modèle est traitée comme une variable cachée. L'initialisation de la position des parties se fait de manière hiérarchique par une procédure de vote des sous-parties pour les parties du niveau supérieur, jusqu'au niveau terminal représentant le centre de gavité de l'objet.

Afin d'obtenir un classifieur multi-classes performant, les probabilités d'appartenance aux catégories sont normalisées de manière discriminative (en utilisant le classifieur softmax, extension de la régression logistique à plusieurs classes). Cela correspond donc à une approche associant apprentissage génératif et discriminatif.

Dans l'ensemble, l'apprentissage est très rapide et des expériences sur des bases d'images réelles montrent la capacité de la méthode à capturer la structure de classes d'objets complexes. Les taux de classification obtenus par cette méthode sont nettement meilleurs que les approches antérieures de type « constellation » [45, 44].

6.1 Modélisation hiérarchique des objets visuels

En catégorisation d'images digitales, les modèles géométriques existants sont en général très spécifiques d'une classe d'objet (par exemple les modèles 3D humains). Il y a un réel besoin d'objets génériques qui puissent être adaptés à des catégories plus larges. Les modèles basés sur des « parties » ou des « fragments » qui combinent les caractéristiques locales d'une image au travers de liens géométriques basiques fournissent une solution cohérente à ce problème [110, 157, 45, 44, 109]. Les modèles de constellation [45, 44] prennent en compte l'apparence et localisation de descripteurs locaux au sein d'un modèle probabiliste. Une de leurs limitations principales est le fait qu'ils nécessitent une énumération exhaustive des mises en correspondances possibles, ce qui est très coûteux en terme de quantité de calcul, et limitent le nombre de parties à 6 ou 7. Cela veut dire qu'une quantité non négligeable d'information disponible dans l'image doit être ignorée, en particulier pour les objets comportant de nombreuses parties, soit naturellement, soit parce qu'un grand nombre de caractérisques locales sont nécessaires pour les décrire (e.g les objets texturés). En fait, ces approches structurelles ont souvent des difficultés à obtenir les mêmes résultats que des approches de type « bag of features » (basées sur l'apparence seulement) car ces dernières utilisent mieux l'information disponible [110, 118, 32]. Il est ainsi utile d'explorer les modèles structurels qui supportent efficacement plusieurs centaines de primitives graphiques.

Ensuite, de nombreuses catégories naturelles (humains, animaux, photos d'objets divers prises en condition réelles) ont une forme relativement rigide, mais une variabilité d'échelle assez importante, et des primitives graphiques proches sur l'image ont une forte corrélation alors que les points plus éloignés sont beaucoup moins corrélés. Cependant, ces corrélations ne sont pas toujours locales et peuvent être extrêmement complexes, comme pour les expressions de visage et les objets 3D avec de petites variations de pose, pour lesquels un modèle par parties peut approcher les déplacement à différentes profondeurs. Les dépendances entre parties et sous-parties se retrouvent à plusieurs échelles dans les images, donnant lieu à une structure hiérarchique de parties. Au final, le modèle global devient un modèle à structure arborescente [92].

Dans ce chapitre, nous proposons un modèle hiérarchique capable de prendre en compte des centaines de primitives graphiques de manière efficace. Il convient à des descripteurs d'images très basiques. La position de l'objet dans l'image ainsi que la structure du modèle sont traitées comme des variables cachées et sont estimées par EM après une initialisation adéquate. La méthode est entièrement invariante par échelle, et les paramètres du

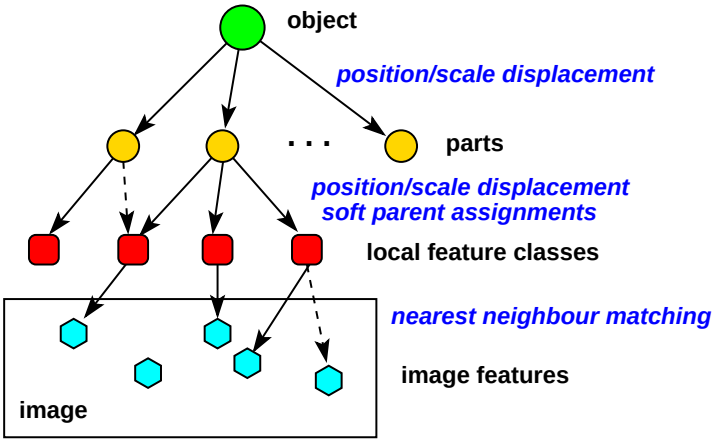


FIG. 6.1 – La structure globale du modèle hiérarchique.

modèle sont appris par maximum de vraisemblance. Les seuls paramètres à modifier sont le nombre de parties à chaque niveau de la hiérarchie. Des expériences de validation croisé montrent que l'utilisation de ce modèle est avantageuse par rapport aux méthodes existantes.

Dans la suite, nous présentons le modèle probabiliste, puis la méthode d'apprentissage, incluant l'initialisation et les étapes EM. Enfin, des expériences sur des images réelles montrent que ce modèle est efficace pour classer des objets dans des catégories.

6.2 Structure du modèle

Notre modèle (voir Figure 6.1) est une hiérarchie de parties et de sous-parties avec l'objet au plus haut niveau et les classes d'apparences-position au plus bas niveau. A chaque niveau de la hiérarchie, les parties sont affectées de manière probabiliste aux parents du niveau supérieur. Ces affectations « floues » sont utilisées principalement pour permettre à la structure du modèle de s'adapter à la classe de l'objet durant l'apprentissage : une fois que les modèles sont appris, la plupart des parties ont une affectation pratiquement déterministe, c'est-à-dire qu'une forte probabilité est donnée à un seul parent. La Table 6.1 définit les paramètres et les variables utilisés.

paramètres	
$\mu_{pp'}^{(\ell)}$	position moyenne de la partie p par rapport à la partie p' au niveau $\ell+1$ de la hiérarchie
$\tau_{pp'}^{(\ell)}$	probabilité pour la partie p-ième d'être affectée à la partie p'
π_c	probabilité d'observer la c-ième primitive graphique
$\Sigma_p^{(\ell)}$	variance des parties au niveau ℓ
α_c	apparence moyenne de la primitive graphique c
Σ_c^{α}	variance de l'apparence

Variables aléatoires dépendant de l'image i	
$\bar{x}_{ip}^{(\ell)}$	position/échelle de la partie p au niveau ℓ
$\bar{a}_{ic}$	apparence de la primitive graphique
η_{ic}	index de la détection qui correspond à la primitive graphique c
$B_{ip}^{(\ell)}$	index de la partie associée à la partie p au niveau supérieur dans la hiérarchie.
O_{ic}	valeur binaire indiquant si la primitive c est observée

TAB. 6.1 – Résumé des paramètres et variables du modèle.

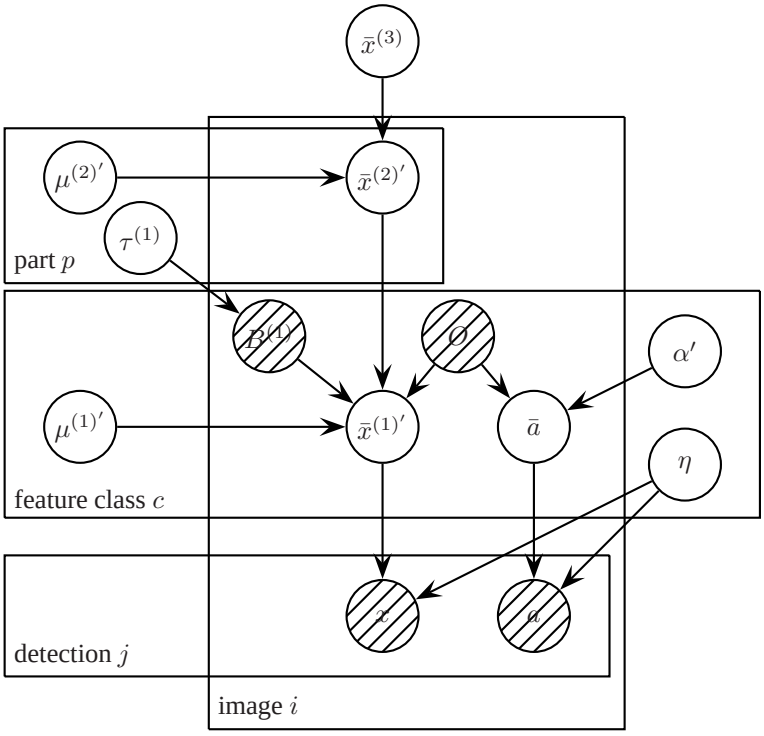


FIG. 6.2 – Modèle graphique du modèle à trois niveaux. Au niveau ℓ, $\mu^{(\ell)'} = (\mu^{(\ell)}, \Sigma^{(\ell)})$ est le vecteur des paramètres codant pour la géométrie, $\bar{x}^{(\ell)'} = (\bar{x}^{(\ell)}, s^{(\ell)})$ est la position/échelle de l'objet dans l'image, $\alpha' = (\alpha, \Sigma^{\alpha})$ est le vecteur de paramètres pour l'apparence (moyenne et variance). .

Structure spatiale

Un objet est constitué de plusieurs parties à différents niveaux de la hiérarchie $\ell = 1, \cdots, L$. Chaque partie et son sous-arbre sont attachés à leur parents par une transformation spatiale non déterministe. Dans les expériences présentées plus loin, seuls les translations et les changements d'échelle entre les parties et leur parent ont été considérés[26]. Nous supposons que la transformation réelle $\mathbf{T}_{pp'}$ entre les deux parties suit une loi normale sumr l'espace des paramètres de transformation et une loi log-normale sur l'espace des échelles. La variance de cette distribution peut être considérée comme une matrice 3×3 en utilisant la paramétrisation $(u, v, \log s)$. Celle-ci est supposée diagonale dans la suite. Nous écrivons la structure de manière récursive : au niveau ℓ de la hiérarchie, les positions des parties $\bar{x}_{ip}^{(\ell)}$ sachant les positions au niveau supérieur $\bar{x}_{ip'}^{(\ell)}$, $p' = 1, \cdots, P^{(\ell)}$ suivent des mélanges de Gaussiennes dont les proportions sont $\tau_{pp'}^{(\ell)}$:

$$\begin{pmatrix} \bar{x}_{ip}^{(\ell)} \\ \log s_{ip}^{(\ell)} \end{pmatrix} \sim \sum_{p'=1}^{P} \tau_{pp'}^{(\ell+1)} \mathcal{N}\left(\bar{x}_{ip'}^{(\ell+1)} + s_i^{(\ell+1)} \mu_{pp'}^{(\ell)}, (s_{ip'}^{(\ell+1)})^2 \Sigma_p^{(\ell)}\right) \times \mathcal{N}\left(\log s_{ip}^{(\ell+1)}, \varsigma^{(\ell)}\right), \tag{6.1}$$

Ainsi, la structure complète du modèle (dépendence des parties relativement aux niveaux supérieurs vient du choix des proportions du mélange $\tau_{cp}^{(\ell)}$. Si l'affectation des sous-parties aux parties est déterministe, les matrices $\tau^{(\ell)}$ ne contiennent que des zéros et des uns, et les mélanges de distributions définis en (6.1) deviennent de simples gaussiennes.

Le dernier niveau de la hiérarchie ne contient qu'une seule partie ($P^{(L)} = 1$). Cette partie est contrainte à être au centre de gravité de l'objet $\bar{x}_{i1}^{(L)} = \bar{\bar{x}}_i$ et donne l'échelle de référence : $s_1^{(L)} = 1$.

Le fait d'utiliser des affectations probabilistes pose un problème d'identifiabilité qui est résolu en contraignant les transformations entre une partie et ses parents à ne correspondre qu'à une seule et même transformation. Sans imposer cela, le modèle comporterait un grand nombre de paramètres inutiles pour la modélisation que l'on souhaite : la transformation entre une sous-partie et une partie ayant une faible proportion dans le mélange (6.1) ne serait presque jamais observée, et l'estimation de ses paramètres serait très instable. En ne considérant qu'une seule transformation possible, les parties sont affectées (de manière probabiliste) à des parents dont la position explique au mieux celle de la partie (les positions sont très corrélées sur l'ensemble d'apprentissage). Ainsi, à chaque partie p au niveau ℓ ne correspond qu'un seul vecteur de paramètres de transformation $\bar{x}_{ip'}^{(\ell+1)} - \bar{x}_{ip}^{(\ell)}$ représentant la

[26]Cela correspond aux transformations de la forme $\mathbf{T}_{pp'} = \begin{pmatrix} s & 0 & u \\ 0 & s & v \\ 0 & 0 & 1 \end{pmatrix}$ en coordonnées homogènes où s est l'échelle relative et (u, v) est la translation relative de la partie.

position moyenne relative dans le repère de la partie p'. Cela revient à imposer aux paramètres $\mu_p^{(\ell)}$ la contrainte $\mu_p^{(\ell)} = \sum_{p'} \tau_{pp'}^{(\ell)} \mu_{pp'}^{(\ell+1)}$ et $\mu_1^{(L)} = 0$. De cette manière, les positions des parties ont le même centre de gravité à tous les niveaux :

$$\mu_c^{(\ell)} = \sum_p \tau_{cp}^{(\ell)} \sum_{p'} \tau_{pp'}^{(\ell+1)} \mu_{pp'}^{(\ell+2)} = \cdots = \mu_1^{(L)} = 0. \tag{6.2}$$

D'un point de vue pratique, cette contrainte permet de travailler avec les variables de décalage $\bar{x}_{ip'}^{(\ell+1)} - \bar{x}_{i1}^{(L)}$ relativement au centre de gravité de l'objet.

Correspondance entre l'image et le modèle

Le premier niveau de la hiérarchie est composé de parties élémentaires contenant aussi une représentation de leur apparence sur l'image. Celle-ci est codée par des descripteurs locaux invariants par échelle, similaires à ceux utilisés dans les modèles de constellation ou de type « bag of features » [162, 45, 44, 38, 110, 118, 32]. Lorsque le modèle est appliqué sur une image, ces parties élémentaires se comportent comme des « points d'attraction » pour les primitives graphiques les plus proches et d'apparence similaire. En résumé, les parties terminales de la pyramide des parties sont caractérisées par leur localisation dans l'image (position et échelle) et leur vecteur d'apparence α. Dans les expériences qui suivent, le descripteur SIFT a été utilisé après détection des points d'intérêt invariants par échelle par une méthode de type Harris-Laplace [110, 118], mais la méthode resterait la même pour d'autres combinaisons détecteur[27]/descripteur de points d'intérêt[28]. Nous définissons l'ensemble des N_i paires apparence/localisation détectées sur l'image i par $\mathcal{S}_i = \{a_{ij}, x_{ij}\}_{j=1,\cdots,N_i}$. Pour chaque partie élémentaire p, l'apparence est modélisée par une distribution gaussienne de moyenne $\overline{\alpha}_p$ et de variance $\mathbf{Var}\,(\alpha_p)$. Ainsi, l'apparence et la localisation sont des instanciations de lois gaussiennes dont les paramètres dépendent de la position des parents et de l'index de la partie.

Le modèle peut accepter un grand nombre de parties élémentaires, bien qu'en pratique elles ne soient jamais toutes observées simultanément sur une image. Il est donc important de permettre à certaines parties d'être inobservées dans le modèle. En pratique, les parties sont toujours affectées à un point de l'image, même si l'affectation est peu probable. La distribution sur l'espace apparence/localisation devient donc un mélange entre :

– la distribution gaussienne définie précédemment modélisant une partie observée et

[27] Un descripteur local est une méthode qui génère les primitives graphiques locales de l'image.

[28] Le descripteur du point d'intérêt est un vecteur contenant des informations sur la forme locale de l'image autour d'un point d'intérêt

– une distribution uniforme modélisant une partie inobservée :

$$\pi \, \boldsymbol{p}(\bar{x}_c^{(1)}|\bar{x}^{(2)})\boldsymbol{p}(a_c) + (1-\pi)\mathcal{U}_c^{\text{app}}\mathcal{U}^{\text{sub}} \tag{6.3}$$

où $\boldsymbol{p}(\bar{x}_{ic}^{(1)}|\bar{x}_i^{(2)})$ est un mélange comportant $P^{(2)}$ composants déjà défini par l'équation (6.1) et $\boldsymbol{p}(a_c)$ est la distribution de l'apparence de la partie élémentaire c. Seules les positions $x_{\,ic}$ et les apparences a_{ic} des C parties élémentaires sont observées. La position, l'échelle des parties et celle du centre de gravité sont considérés comme des variables cachées et sont estimées pour chaque image.

Le modèle suppose que les sous-parties sont affectées à un seul parent, ce parent pouvant différer d'une image à l'autre. Considérer que les parties sont toujours associées au même parent correspondrait plus à la philosophie générale du modèle, mais les expériences réalisées montrent des difficultés d'estimation : le modèle n'est pas clairement divisé en parties, et la correspondance de parties élémentaires aux primitives graphiques est moins efficace.

Lors de la phase de test, les correspondances multiples entre parties élémentaires et primitives graphiques sont permises, pour des raisons de rapidité de calcul et parce que les résultats finaux sont visuellement équivalents. En revanche, durant la phase d'apprentissage, nous avons contraint les affectations à être uniques pour éviter que les parties proches en localisation et en apparence ne se confondent et correspondent de cette manière à une seule et même partie.

6.3 Apprentissage

Le modèle sur une image donnée mais aussi sur l'ensemble d'apprentissage tout entier en utilisant l'algorithme Expectation-Maximization. Les paramètres estimés lors de la phase d'apprentissage sont

$$\theta = (\mu, \tau, \pi, \Sigma, \varsigma, \alpha, \Sigma^{\alpha}).$$

Dans chaque image les variables cachées continues sont les positions et échelles des parties $\bar{x}$ et les variables cachées discrètes sont les affectations des parties élémentaires aux primitives graphiques η, la variable binaire O représentant si la partie est observée ou non, et l'association B des sous-parties aux parties à chaque niveau de la hiérarchie. La figure 6.2 donne le modèle graphique de la distribution de toutes ces variables pour un modèle à trois niveaux.

Les étapes EM sont implémentées de manière classique. Une fois initialisée, la méthode converge en 50 itération environ mais donne des résultats satisfaisant en 5–10 itérations. L'apprentissage prend environ 1 seconde par image en MATLAB, la plupart du temps de calcul étant due à l'étape de mise en correspondance des primitives graphiques. Il est important de remarquer que tous les paramètres sont appris par l'algorithme. Il n'y a donc, à part le nombre de parties à chaque niveau de la hiérarchie et les paramètres de bas niveau inclus dans le calcul des primitives graphiques, aucun seuil ou paramètre de régularisation à définir manuellement.

6.3.1 Instanciation du modèle dans une image

La fonction objectif (la log-vraisemblance négative) a de nombreux minima locaux, et un instanciation robuste est nécessaire. Nous utilisons une méthode de type transformée de Hough de manière hiérarchique. Cette méthode est basée sur des votes dans la pyramide des positions/échelles possibles pour chaque partie.

1. Pour chaque partie c au niveau le plus bas de la hiérarchie, chaque primitive graphique (avec une apparence a_f et une position $\bar{x}_f$) vote dans la pyramide pour une position $\bar{x}_p^{(2)}$, en calculant l'espérance du mélange de distributions sur les apparences et les positions générées par le niveau supérieur de la hiérarchie c. Une fois que la localisation majoritaire est trouvée, les parties ayant contribué à cette localisation sont affectées au parent c

$$\mathrm{Vote}_c(\bar{x}_c) \;=\; \sum_f \max_c \frac{\boldsymbol{p}_c(p)}{w_p}\, \boldsymbol{p}(a|\alpha)\, \boldsymbol{p}(\bar{x}|\mu) \tag{6.4}$$

$$w_p \;\equiv\; \sum_c \boldsymbol{p}_p(\alpha_f) \tag{6.5}$$

   La somme se fait sur la primitives graphiques f, et le maximum est choisi parmi les parties élémentaires dans le parent et q. Pour des raisons de vitesse de calcul, une primitive f ne peut voter que pour une seule partie p. Il faut noter que les votes sont pondérés pas le nombre de primitives appartenant à la même catégorie d'apparence : $w_p = \sum_f \mathbf{p}_p^{\mathrm{app}}(\alpha_f)$. Cela permet de diminuer l'influence des primitives appartenant au fond, ou à des parties texturées dont l'apparence se répète un grand nombre de fois.

2. L'arbre des positions spatiales est parcouru en transformant les votes des sous-parties en votes pour la posi-

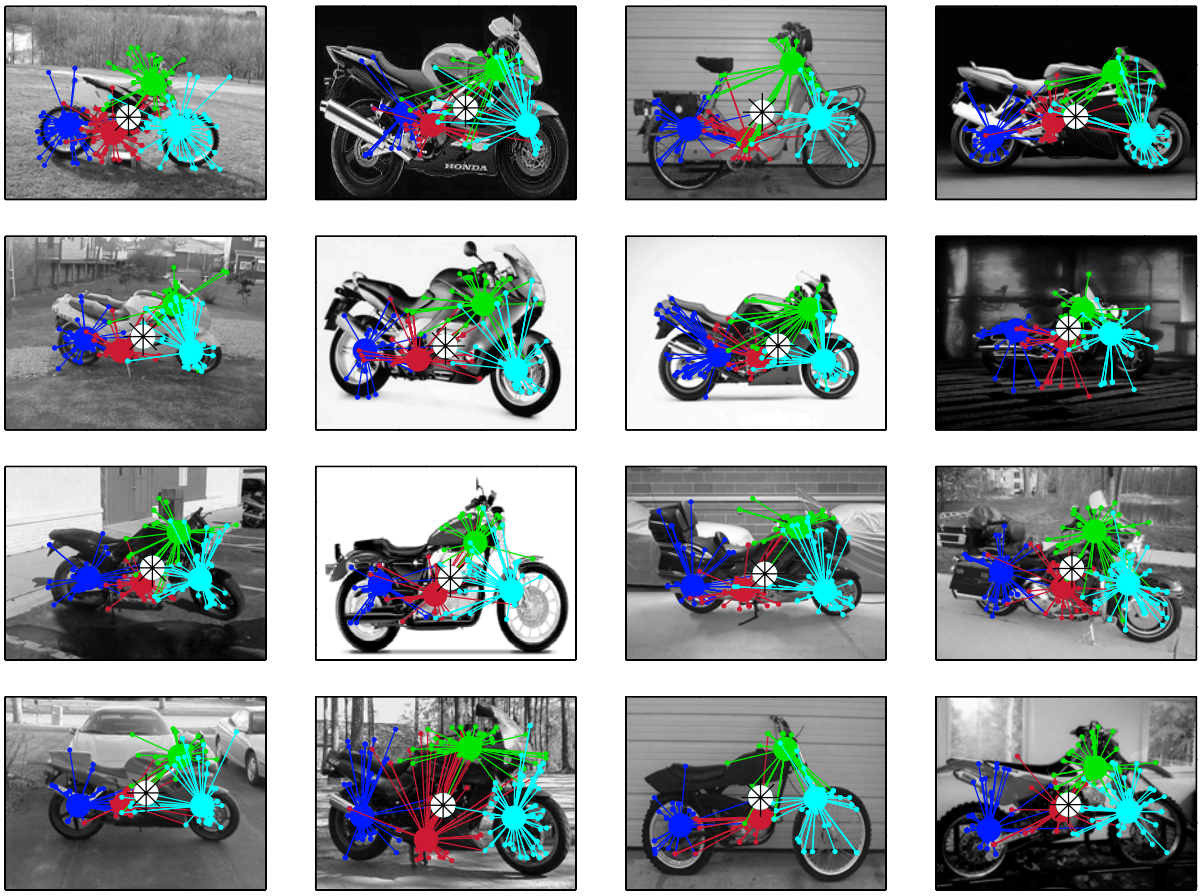


FIG. 6.3 – Quelques modèles appliqués à des images de test du jeu de données de motos. Les rectangles indiquent les positions et échelles des parties estimées.

tion de leurs parents en utilisant le décalage moyen des positions propres au modèle :

$$\mathrm{Vote}_{p'}^{(\ell+1)}(\bar{x}_{p'}) \;=\; S(\sum_{p} \log(1 + \mathrm{Vote}_{p}(s_p^{(\ell)}(\bar{x}_p^{(\ell)} - \mu_{pp'}^{(\ell+1)})) \tag{6.6}$$

Ici, la somme s'effectue sur les sous-parties p pour lesquelles p' est le parent le plus probable ($\arg\max_{p''} \boldsymbol{\tau}_p(p'') = p'$) — là encore, pour des questions de rapidité de calcul, les parties ne votent que pour un seul parent. La fonction non-linéaire $\log(1 + \ldots)$ permet de relativiser l'importance des pics présents dans des sous-parties erronées (par exemple sur le fond de l'image) par rapport à la contribution de plusieurs parties qui votent toutes pour un même centre. S est une fonction de lissage heuristique. Dans notre cas, une convolution gaussienne a été utilisée.

3. Les maxima dans la pyramide des votes pour la partie du plus haut niveau donnent les placements $\bar{x}^{\,(L)}$ et

les échelles $s^{(L)}$. potentielles des objets de la même catégorie présents dans l'image.

4. Pour le meilleur (ou éventuellement pour chaque) maximum, l'arbre est parcouru en sens inverse pour trouver les affectations des parties aux sous-parties. Si la pyramide des parties a un maximum satisfaisant autour de la position moyenne calculée grâce aux autres parties, cette partie est utilisée dans la suite. Sinon, elle est supposée inobservée et la position relative est égale au décalage par défaut $\mu_{pp'}^{(\ell+1)}$.

Cette procédure donne une position initiale de l'objet satisfaisante sur les jeux de données tels que Caltech, même lorsque la position de l'objet n'est pas évidente et est entourée d'une quantité modérée de détections sur le fond. Elle ne donne cependant pas une vraisemblance suffisamment grande aux images de la même catégorie, car certaines parties ne sont pas reconnues. Quelques itérations de EM permettent d'améliorer notablement la mise en correspondance.

6.3.2 Apprentissage — Initialisation du modèle

La procédure précédente initialise seulement la position des parties dans une image, mais elle suppose que les paramètres sont connus. Nous détaillons ici une manière d'initialiser ces paramètres, de manière à éviter un placement manuel des objets dans les images. La méthode suppose que chaque image d'apprentissage contient une instance de la classe d'objet, mais dont la position et l'échelle sont inconnus et des élément non uniformes dans le fond de l'image sont tolérés. Le nombre de parties à chaque niveau de la hiérarchie doit être déterminé à l'avance. La méthode d'initialisation suit les étapes suivantes :

1. Trier les images suivant un critère de qualité (précisé dans la suite) et utiliser la meilleure image pour initialiser les paramètres du modèle. Si la méthode échoue, utiliser la seconde image ou les suivante, mais dans les expériences proposées, la première a toujours suffit.

2. En utilisant l'algorithme des K-means, toutes les primitives graphiques de l'image initiale sont classées en n classes de localisation/apparence, où n est le nombre de parties élémentaires souhaitées. Ensuite, initialiser une partie élémentaire au centre de ces parties. Certaines des parties élémentaires vont correspondre à des points du fond. Celles-ci auront tendance à disparaitre durant l'apprentissage (leur probabilité d'observation devrait tendre vers zéro). Certaines parties auront la même apparence, ce qui est intentionnel : cela permet de définir des parties répétitives telles que des yeux ou des roues de voiture.

3. Pour $\ell = 2, \cdots, L$, la position des sous-parties est classée en $P^{(\ell)}$ groupes par K-means. Un parent est initialisé pour chaque centre de groupe. L'affectation aux clusters donne les affectations initiales des sous-parties aux parties. En réalité, la matrice des affectations $\boldsymbol{\tau}^{(\ell)}$ est légèrement lissée pour permettre des affectations floues.

L'utilisation d'une seule image dans la première étape peut être critiquée, mais nos tentatives d'initialisation à partir d'une moyenne sur plusieurs images ont donné de plus mauvais résultats. Le point critique est de trouver un modèle initial qui contienne des parties clairement séparées et d'apparences spécifiques, à partir desquelles un apprentissage dépourvu d'ambiguités peut être effectué. Le moyennage tend à rendre confus les apparences et produit une initialisation bien moins satisfaisante.

La méthode de note des images suivant leur qualité probable est la suivante :

1. Utiliser K-means sur les apparences seulement pour trouver les caractéristiques communes aux images d'apprentissage positives (*i.e.* contenant l'objet) en fixant $K = 500$ classes. Chaque image est décrite par un vecteur « signature » $\mathbf{S}$ de dimension 500 (le vecteur de proportion des classes).
2. Ordonner les classes d'apparence suivant leur mesure d'information (voir plus loin) et sélectionner les 30 (environ) classes les plus informatives. Ordonner les images suivant le nombre de points appartenant à au moins une des ces classes d'apparence (*i.e.* le nombre de classes c pour lesquelles $\mathbf{S}_c \neq 0$).

Pour mesurer l'information des classes d'apparence, deux méthodes ont été étudiées. Une méthode supervisée nécessite un ensemble d'apprentissage négatif, *i.e.* ne contenant pas l'objet à classer. Un classifieur linéaire est appris sur le vecteur signature binaire $(\mathbf{S} \neq 0)$. Les variables ayant les plus forts coefficients sont considérées comme étant les plus informatives. Toutes les méthodes de classification linéaires peuvent être utilisées : l'Analyse Discriminante Linéraire, les SVM linéaires, Relevance Vector Machines linéaire, la méthode LASSO, Least Angle Regression, etc.

Nous pouvons aussi utiliser une méthode purement non-supervisée qui semble fonctionner aussi bien que la méthode supervisée précédente, et ne nécessite pas d'images négatives. Pour chaque classe d'apparence, le score associé est le nombre d'images qui contiennent exactement une fois l'apparence en question (alternativement, on peut prendre 1–2 fois la même classe). Ce critère fonctionne car il sélectionne les apparences qui représentent une seule partie d'objet. De nombreuses classes d'objet contiennent de telles apparences alors que que les apparences

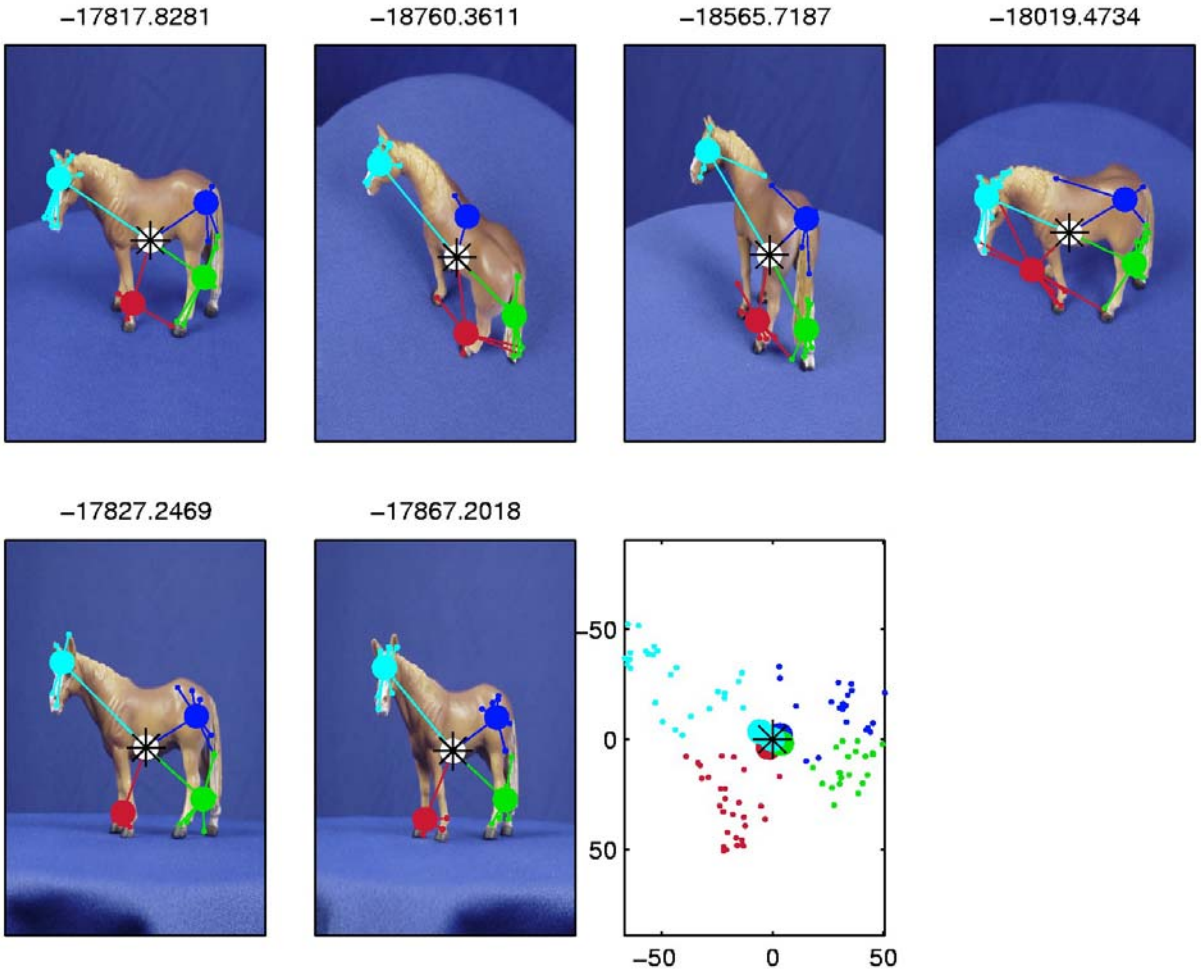


FIG. 6.4 – Estimation du modèle sur des images d'un cheval en plastique. Les points en bas à droite sont les positions moyennes des parties élémentaires.

du fond sont beaucoup plus variables et apparaissent rarement une fois par image. La méthode serait évidemment inefficace pour des objets qui ne contiennent que des motifs répétitifs.

6.4 Expériences

Dans ce chapitre nous ne considérons que des modèles à trois niveaux de hiérarchie. La Figure 6.4 montre la capacité du modèle à prendre en compte des déformations locales de l'image, pour lesquelles une mise en correspondance rigide échouerait. Le modèle a été appris à partir de 6 images contenant le même objet dans des prises de vue différentes, en utilisant cent parties élémentaires et quatre parties au second niveau. Le modèle a

	V.	L.	M.	A.	P.
Visages	198	12	5	1	1
Léopards	0	92	8	0	0
Motos	0	6	383	10	0
Avions	0	4	15	351	30
Profils de voitures	0	0	0	1	60

TAB. 6.2 – Matrice de confusions pour le classificateur sur les données de Caltech comportant 5 catégories.

ensuite été instancié sur cinq images de test avec la méthode de mise en correspondance décrite plus haut. Les changements d'angle de vue sont considérables, mais le modèle trouve tout de même les bonnes parties de l'objet, même lorsque très peu de points d'intérêt on été détectés sur une partie donnée.

Jeux de données : Afin de tester le modèle, nous avons utilisé cinq classes d'objet différentes issues de la base d'images « Caltech 101 Object Categories » [29] [44], qui contient de nombreux exemples d'images réparties en 101 catégories, comportant notamment des visages (435 images), des léopards (200), des motos (800), des avions (800) et des profils de voitures (123). Ces bases d'images ont déjà été utilisées dans différentes études [162, 46, 44, 38]. La moitié de ces images ont été utilisées pour tester les performances de classification.

Quelques exemples de modèles appris sont montrés sur la Figure 6.5.

Pour tester si les modèles apprennent vraiment les paramètres d'apparence et de position corrélés entre eux et s'ils sont suffisament sélectifs pour une catégorie donnée, nous avons validé leur performance en discrimination en apprenant différentes classes sur des images de test, en utilisant la vraisemblance comme critère de décision. Pour chaque classe, un seuil de décision est calculé en minimisant le taux d'erreur moyen. Nous avons utilisé dix itérations de EM pour l'apprentissage et cinq pour le test. Les matrices de confusion sont données dans la Table 6.2 pour les sept classes de Caltech décrite plus haut avec deux cents parties élémentaires, et dans la Table 6.3 pour le modèle appliqué aux aux cinq premières catégories des données Caltech. Dans ce dernier cas, les modèle à deux et trois niveaux sont comparés pour quatre vingts parties élémentaires. Le nombre d'erreurs dépend fortement de la classe considérée, mais les résultats semblent compétitifs par rapport aux meilleures techniques actuelles [37, 45]. Le modèle rigide de base est déjà très discriminatif pour ce jeu de données mais l'utilisation du modèle à trois

[29] Disponible à l'adresse `http://www.vision.caltech.edu/feifeili/Datasets.htm`

one-level model					
	Acc	Avi	Anc	Fou	Ton
Accordéons	18	0	0	9	0
Avions	0	359	6	35	0
Ancres	0	1	4	12	4
Fourmis	0	2	1	17	1
Tonneaux	0	3	1	9	10
Two-level, three part model					
Accordéons	25	0	1	1	0
Avions	1	384	0	12	0
Ancres	0	3	6	12	0
Fourmis	0	4	1	18	1
Tonneauxl	0	6	0	8	9

TAB. 6.3 – Matrice de confusion pour le classifieur basé sur 80 parties terminales sur les premières catégories du jeu de données de Caltech.

parties réduit le taux d'erreur en test d'un facteur de 2.

La figure 6.6 montre que les résultats ne sont pas très sensibles au nombre de parties considérées, bien qu'un phénomène de sur-apprentissage apparaisse à environ 8–10 parts. Un nombre relativement large de parties élémentaires est nécéssaire pour obtenir des résultats optimaux (environ 200 dans ce cas-ci).

Affectations déterministes ou probabilistes : La matrice τ codant pour la structure peut être contrainte à avoir seulement des uns et des zéros, de manière à ne permettre qu'à un seul parent de générer une sous-partie. Pour illustrer l'avantage d'une affectation floue, nous avons tester le classifieur binaire en utilisant 40 images d'apprentissage, 200 parties élémentaires et 4 parties. Une affectation déterministe donne un taux de bonne classification final de 83% alors que l'affectation floue (*i.e.* probabiliste) donne un taux de classification de 88%. Des résultats similaires sont obtenus sur d'autres jeux de données.

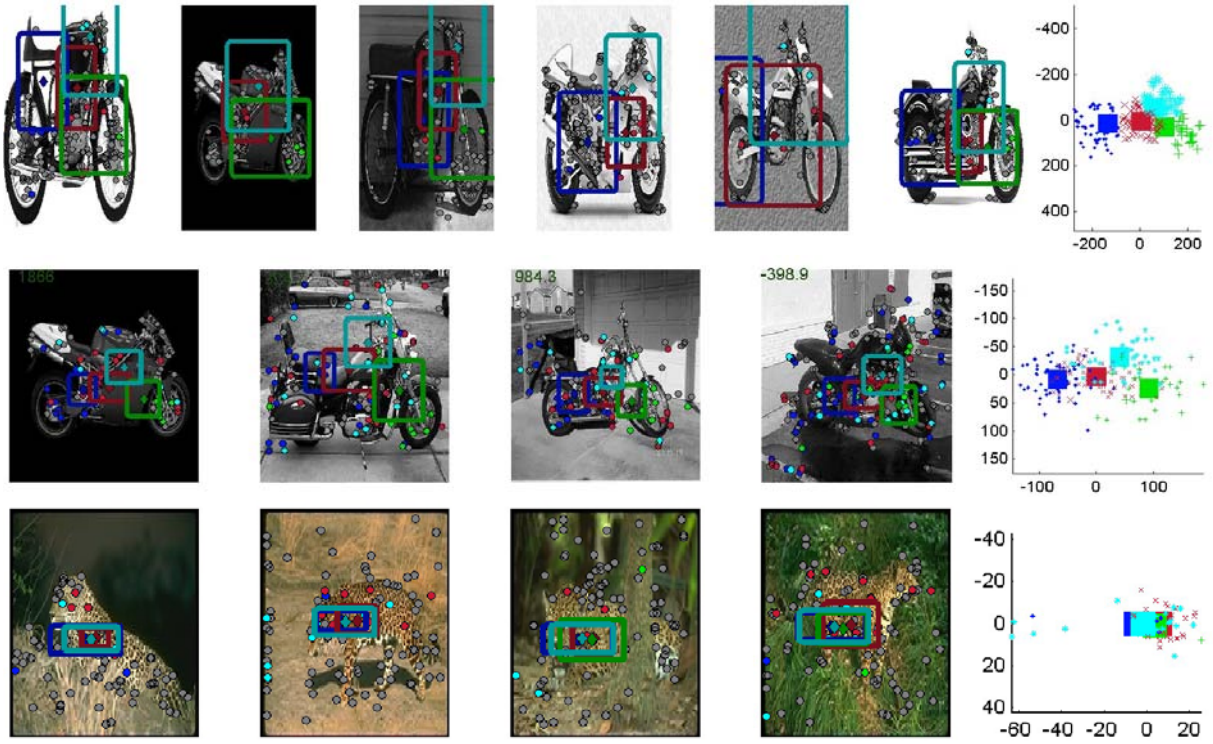


FIG. 6.5 – Exemples de modèles appris sur les images de moto et de léopards. La première ligne montre l'initialisation basée sur un vote en postion/échelle des classes d'apparence.

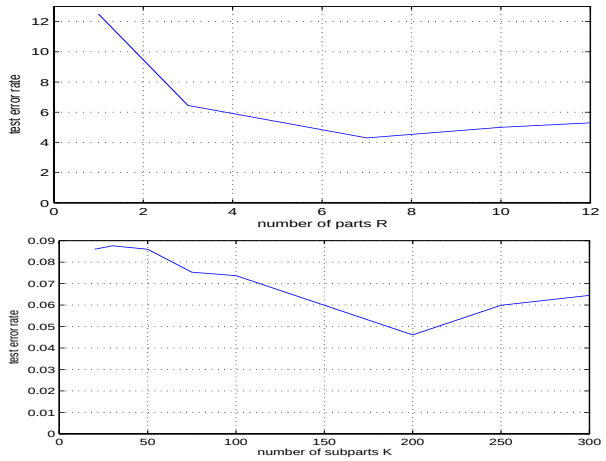


FIG. 6.6 – Les taux d'erreur en test sur les jeux de données de feuilles et de visage par rapport au nombre de parties de niveau 2 (haut) et de parties élémentaires (bas).

6.5 Conclusions et perspectives

Nous avons décrit un modèle génératif basé sur des parties pour classifier des catégories d'objets visuels. Ce modèle utilise un grand nombre de primitives graphiques locales et s'adapte très bien aux objets testés dans nos expériences de catégorisation. Les raisons de ce bon comportement sont la flexibilité spatiale organisée de manière hiérarchique et le fait qu'il peut incorporer efficacement un grand nombre de points d'intérêts, chacun comportant une information discriminante non-négligeable. Ce modèle permet de créer un classifieur multi-classes particulièrement efficace par rapport aux méthodes existantes. Nous avons aussi montré que tant que le modèle utilise suffisamment de points d'intérêts, la mise en correspondance des parties élémentaires n'a pas besoin d'être vraiment précise. Une hiérarchie comportant seulement trois niveaux de hiérarchie (objet, parties et sous-parties) donne des résultats à la fois visuellement satisfaisants et ayant une bonne précision probabiliste.

Perspectives : Le modèle s'applique à des transformations spatiales arbitraires entre parties et sous-parties. Bien que nous n'ayons considéré que des transformations de translation et de changement d'échelle, il serait naturel de considérer des transformations affines en ajoutant les rotations. Les expérience présentées n'incluent que trois niveaux de hiérarchie, et il serait intéressant de considérer plus de niveaux. La difficulté principale est de trouver une bonne initialisation de ces modèle plus complexes. Une autre voie de recherche prometteuse est l'apprentissage d'un mélange de ces modèles, pour la classification non-supervisée (clustering) d'images et pour prendre en compte des classes plus complexes telles que des objets tri-dimensionnels vus dans différentes directions.

Chapitre 7

Réactualisation bayésienne d'un modèle de dégradation

Après avoir construit un modèle probabiliste sur des données d'images numériques, nous proposons un deuxième exemple de modélisation générative. Dans un contexte de fiabilité, on souhaite prédire un état de défaillance ($Y = 1$) ou de non-défaillance ($Y = 0$) en fonction d'observations X d'un système. L'objectif est d'estimer au mieux la loi conditionnelle $\boldsymbol{p}(Y = 1|X)$. C'est un exemple typique d'apprentissage supervisé. La spécificité du problème est, qu'en général, la proportion de défaillances dans l'ensemble d'apprentissage est très faible.

Dans ce travail, la dégradation d'éléments métalliques intervenant dans le fonctionnement de centrales nucléaires est modélisée. A partir du modèle physique de dégradation existant, un modèle probabiliste $\boldsymbol{p}(X, Y; \theta)$ de paramètre θ sur les données jointes a été défini. La présence d'information *a priori* a justifié l'utilisation du paradigme bayésien. Le formalisme des modèles graphiques a été d'un intérêt considérable dans la mise au point d'un modèle en accord avec les experts du domaine. L'échantillonnage de Gibbs a permis d'estimer par simulations la loi *a posteriori* du modèle, c'est-à-dire $\boldsymbol{p}(\theta|\mathbf{x}, \mathbf{y})$ où $(\mathbf{x}, \mathbf{y})$ sont les données d'apprensissages (contrôles sur site). L'exploitation du modèle sur les données réelles répond aux objectifs d'anticipation des dégradation et de calcul du risque de défaillance.

Bien que spécifique du modèle étudié, la démarche utilisée peut être appliquée à de nombreux problèmes

industriels. Nous détaillons les principaux aspects novateurs de ce travail.

Sources d'information hétérogènes Au sein d'un même modèle, les données issues d'expériences en laboratoire, de mesures sur site, de connaissances d'expert et d'observations de défaillances sont prises en compte de manière cohérente dans un même modèle. Grâce aux indépendances conditionnelles contenues dans le modèle graphique, il a été possible d'incorporer au cas par cas un modèle pour chaque type de donnée : température, contrainte, sensibilité du matériau, cinétique de propagation et retour d'expérience. De plus, le choix de distributions de probabilité appropriées (par exemple les lois *a priori* conjuguées) permet de simplifier considérablement l'inférence.

Modélisation d'événements rares Une particularité du problème est que le nombre d'observations de défaillance est nul. Cela nécessite de définir des scénarios virtuels pour lesquels les observations sont en forte contradiction avec les connaissances *a priori*. Ce type d'approche permet d'anticiper des evénements rares mais aux lourdes conséquences.

Quantification de l'effet des contrôles négatifs Être capable d'observer comment un risque évolue en fonction des contrôles effectués est d'un intérêt majeur dans les applications industrielles. Ceci permet de pouvoir apprécier de manière quantitative l'effet des contrôles négatifs (c'est-à-dire sans observation de défaillance). Dans tous les domaines où toute défaillance est à proscrire, les statistiques classiques demeurent impuissantes puisqu'on dispose de très peu d'observations. L'utilisation des statistiques bayésiennes nous a permis de montrer que le risque de défaillance est significativement plus grand si on ne prend pas en compte ces contrôles négatifs.

Echantillonnage de Gibbs et calcul de risque structurel Un aspect particulièrement novateur est l'utilisation d'un calcul de risque structurel (calcul FORM) au cours de la simulation. En effet, ce type de modèle était généralement utilisé pour calculer un risque de défaillance en fonction de variables aléatoires, mais l'utilisation du retour d'expérience pour réactualiser la valeur de ce risque n'avait jamais été étudié.

7.1 Presentation du problème

7.1.1 Contexte

EDF se doit d'anticiper d'éventuelles dégradations de ses composants, surtout lorsque des événements passés les rendent plausibles. L'alliage 600 (alliage Fe-Ni-Cr), utilisé pour la fabrication de pièces particulièrement sensibles des tranches nucléaires, s'est avéré sujet à la fissuration par corrosion sous contrainte. Tous les composants en alliage 600, y compris ceux sur lesquels les modèles ne prévoient pas de fissuration avant longtemps, doivent faire l'objet d'inspections destinées à vérifier l'absence de fissures et de travaux d'anticipation d'EDF, afin de pouvoir réagir en cas de dégradation constatée.

Le temps au bout duquel l'amorçage de fissures dans l'alliage 600 est susceptible de se produire est estimé au moyen d'une équation dans laquelle interviennent trois indices : l'*indice matériau*, caractérisant la sensibilité du matériau à la corrosion, l'*indice contrainte*, dépendant de l'ensemble des contraintes subies par le composant (fabrication, assemblage, géométrie, fonctionnement), l'*indice température*, dépendant de la température de fonctionnement de la tranche.

L'estimation des trois indices est entachée d'une certaine imprécision, d'autant plus que leur connaissance est issue d'essais de laboratoire, ce qui fait du temps d'amorçage[30] une variable aléatoire. Les résultats du modèle sont donc entachés de nombreuses incertitudes. Son application à un composant particulier (un tuyau cylindrique épais dans lequel circule un fluide), présent à environ 2800 exemplaires sur l'ensemble des 58 tranches françaises, conduit à des temps d'amorçage très élevés, compris entre 100 et 500 ans.

L'approche développée consiste à tenir compte des résultats des contrôles (présence[31] ou absence de fissures) sur les composants en condition réelle d'exploitation afin de réactualiser les paramètres du modèle physique qui sont au nombre de trois : la température, la contrainte et l'indice matériau.

L'étude de l'influence des résultats de contrôle, désignés par la suite par Retour d'Expérience (REX), sur les paramètres du modèle permet non seulement de déterminer quels sont les composants critiques et avec quelle probabilité, mais aussi d'évaluer l'influence d'une stratégie d'inspection sur le risque global de détection d'une

[30] Le temps d'amorçage est la durée entre la construction de la pièce et l'apparition d'une fissure.

[31] Ne disposant pas de contrôle positif lors de l'étude, les contrôles sont inclus dans des scénarios virtuels. Cela permet de quantifier l'effet d'une politique de contrôles en cas de fissuration.

fissure.

Un modèle probabiliste a été utilisé afin de concilier des événements rares et très improbables (par exemple la détection d'une fissure sur un composant âgé de 30 ans), et l'ensemble des informations « normales » -tous les composants contrôlés sains).

7.1.2 Méthode

Le modèle décrit permet d'estimer la loi de probabilité du temps d'amorçage de fissures. Il met en œuvre une approche de type bayésien afin de prendre en compte les connaissances a priori. On note que le modèle est partiellement hiérarchique dans la mesure où certains paramètres définissant les lois a priori sont eux même aléatoires et ainsi leur valeur est attachée à celle d'autres paramètres. Ce type de construction hiérarchique permet d'atténuer l'influence de paramètres a priori dont les valeurs ne sont pas connues avec une grande précision. Les distributions de probabilité de la plupart des variables du modèle sont définies conditionnellement à d'autres variables.

Les spécificités de l'approche portent sur les quatre points suivants :

1. les données ne sont pas homogènes : elles comprennent différents types de matériau, d'environnement et de contrainte,
2. le temps à fissuration est donné par un modèle physique qui n'est pas remis en cause. Il repose sur des critères physiques, et on a considéré que les sources d'incertitude provenaient uniquement des paramètres du modèle,
3. la présence d'une forte connaissance a priori sur les paramètres du modèle, en fonction d'essais de laboratoire, donnant des temps d'amorçage extrêmement longs,
4. les informations issues du REX : ce sont essentiellement des données de survie, donc censurées ; il y a très peu d'observations de fissure.

Afin de rendre la modélisation cohérente, il a fallu prendre en compte des liens entre paramètres de même nature. Par exemple, nous savons que la température de fonctionnement de certaines installations est plus élevée que d'autres. On ne peut admettre une réactualisation de la température qui fasse que cette propriété ne serait plus

respectée : ces règles de cohérence ont été modélisées au moyen de variables de décalage communes à l'ensemble des paramètres.

L'estimation de la loi jointe de l'ensemble des variables aléatoires du modèle est réalisée par l'échantillonnage de Gibbs [55], qui consiste à simuler successivement les variables du modèle conditionnellement à toutes les autres. La distribution des variables ainsi obtenue suit, au bout d'un grand nombre d'itérations, la loi a posteriori recherchée.

L'utilisation de lois conjuguées (c'est-à-dire de lois qui conservent leur forme analytique après intégration des données aux connaissances a priori) permet de simplifier les simulations. Cependant la prise en compte de scénarios peu précis (par exemple 10 fissures sur n'importe lesquels des 2800 composants) requiert un calcul de risque structurel pour chaque composant. La méthode FORM (First Order Reliability Method) a été utilisée pour conduire ces calculs [116]. L'imbrication de ces deux techniques (calcul FORM et échantillonnage de Gibbs) constitue un intérêt technique majeur de ce travail.

7.2 Modèle de fissuration

7.2.1 Modèle physique

La fissuration des composants métalliques sous contrainte est un phénomène qui a fait l'objet de nombreuses études à EDF. Des études antérieures ont défini un modèle d'évaluation du temps de fissuration, c'est-à-dire le temps qui s'est écoulé entre la date de de mise en service du composant et l'amorçage d'une fissure.Une fissuration est divisée en deux phases distinctes : l'*amorçage* et sa *propagation* jusqu'à une profondeur jugée importante relativement au risque de fuite.

Modèle des indices L'amorçage représente le temps d'apparition d'une fissure telle que sa profondeur soit à la limite des performances techniques des contrôles (1mm de profondeur par exemple). Il est calculé par une formule appelée *modèle des indices* utilisé dans plusieurs études antérieures [41]. Il peut se résumer à la formule donnant le temps de fissuration théorique t_{fiss} :

$$t_{fiss} = \frac{t_{ref}}{I_{mat} I_\theta I_\sigma} \tag{7.1}$$

où t_{ref} est le temps de fissuration de référence, I_{mat} un indice déterminant la sensibilié du matériau, I_θ un indice déterminé par température de fonctionnement et I_σ un indice dépendant de la contrainte que subit le composant. Le modèle des indices cumule de manière multiplicative l'influence de trois facteurs favorisant la fissuration.

La quantité t_{ref} est une constante de normalisation et on a choisi de ne pas la remettre en cause par le retour d'expérience. Dans cette étude, il est fixé à 10000 heures.

L'indice I_{mat} n'a pas de forme explicite. Suite à plusieurs études on admet que sa loi de probabilité est une loi lognormale dont les paramètres dépendent directement du matériau (trois matériaux possibles) et de la présence ou non de détensionnement (traitement thermique de la pièce diminuant le contraintes de soudure lors de la construction de la tranche [41]). Il y a donc en tout six types de lois différentes en fonction du type de matériau. Nous travaillerons dans la suite avect la variable $\ell = \log(I_{mat})$ qui suit une loi normale.

L'indice I_θ est une fonction de la température de fonctionnement : $I_\theta = e^{-(\frac{1}{\theta}-\frac{1}{\theta_0})}$ avec $\theta_0 = 598\,^0K$ la température de référence. Dans les conditions d'exploitation, la température peut varier sensiblement d'un composant à l'autre, et elle n'est pas directement mesurable. Ainsi, θ est considéré aléatoire et est modélisé par une loi normale.

L'indice I_σ dépend de la contrainte exercée sur le composant : $I_\sigma = \left(\frac{\sigma}{\sigma_{ref}}\right)^4$ avec $\sigma_{ref} = 450\,MPa$ la contrainte de référence. Nous différencierons dans la suite σ et σ_{calc} la contrainte réelle de la contrainte estimée.

Propagation d'une fissure La propagation représente le temps que met la fissure pour atteindre une profondeur donnée. Dans cette étude, on a retenu la valeur de 6 mm. Remarquons que dans le cadre de cette étude, notre approche a été très conservative car il n'y a pas de perte fonctionnelle à cette profondeur. On justifie cependant une opération de maintenance à partir de cette profondeur, c'est-à-dire à la mi-épaisseur du composant.

On utilise la loi de cinétique suivante [41] :

$$\frac{da}{dt} = C.\left[q(a) - q_0\right]^m, \tag{7.2}$$

où

- a est la profondeur de fissure,
- q est le facteur d'intensité de contrainte en fond de fissure,
- q_0 est le seuil de propagation,
- C est un coefficient aléatoire,

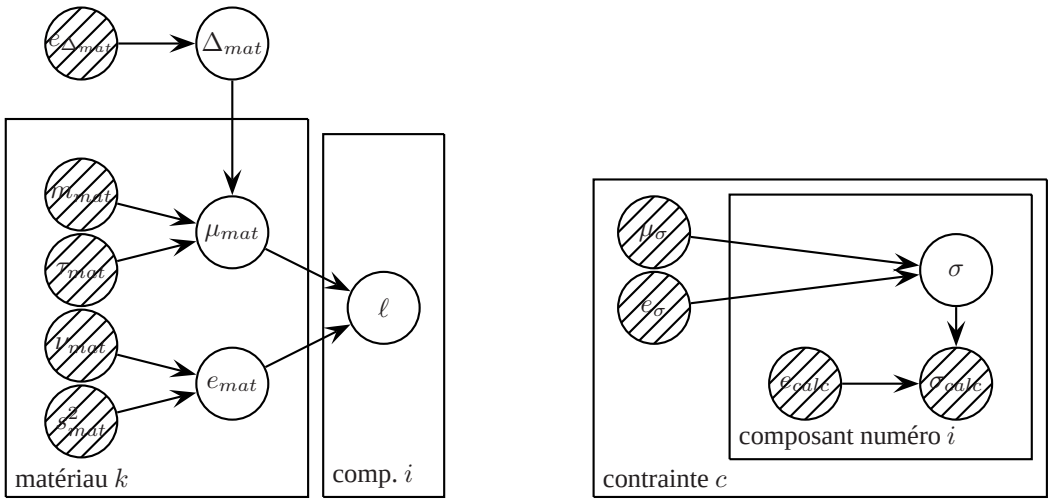


FIG. 7.1 – Exemples de modèles graphiques. Le modèle de droite représente la modélisation de la contrainte, celui de gauche la modélisation de la sensibilité du matériau. Des cadres délimitent les variables qui doivent être répétées plusieurs fois. Il faut par exemple remarquer que ℓ ne peut pas être à l'intérieur du cadre matériau car il y a N (et non $N \times K$) variables ℓ définies.

– m caractérise la vitesse de la cinétique.

La valeur m est considérée comme une constante pour tous les composants étudiés.

En intégrant l'équation 7.2, on peut exprimer le temps écoulé entre l'apparition de la fissure (date d'amorçage t_{fiss}) et l'instant t_{def} où la profondeur de la fissure a_{def} est suffisamment grande pour justifier une opération de maintenance.

$$t_{def} - t_{fiss} = \frac{1}{C} \int_{a_{fiss}}^{a_{def}} \frac{da}{[K(a) - K_{ISCC}]^m} = \frac{\lambda}{C} \tag{7.3}$$

où a_{fiss} est la profondeur de fissure à son amorçage supposé constant. On considère que m est une constante. Le facteur d'intensité $K(a)$ dépend des caractéristiques de chaque composant. On dispose de valeurs de $K(a_i)$ pour des profondeurs a_i allant de 1 mm à 11 mm. L'intégrale λ est calculée par intégration numérique pour chaque composant.

Le temps de fissuration est obtenu en sommant le temps d'amorçage et le temps de propagation. Nous donnons

la formule finale utilisée dans notre modèle :

$$t_{def} = \frac{t_{ref}}{e^{\ell} e^{-(\frac{1}{\theta}-\frac{1}{\theta_0})} \left(\frac{\sigma}{\sigma_{ref}}\right)^4} + \frac{\lambda}{e^{\chi}}, \tag{7.4}$$

où les quantités t_{ref}, θ_0, σ_{ref} et λ sont des constantes. Les paramètres aléatoires sont le logarithme de l'indice matériau ℓ, la température θ, la contrainte σ et le facteur de cinétique χ. Ils sont spécifiques à chaque composant et on suppose qu'ils suivent une loi normale. Le choix d'une loi normale est justifiée par des études antérieures.

– sensibilité du matériau ℓ

$$\ell \sim \mathcal{N}(\mu_{mat}(k), e^2_{mat}(k)), \tag{7.5}$$

– température θ

$$\theta \sim \mathcal{N}(\mu_{\theta}(j), e^2_{\theta}(j)), \tag{7.6}$$

– contrainte σ

$$\sigma \sim \mathcal{N}(\mu_{\sigma}(c), e^2_{\sigma}(c)), \tag{7.7}$$

– coefficient de cinétique C :

$$\chi \sim \mathcal{N}(\mu_{\chi}(c), e_{\chi}(c)). \tag{7.8}$$

Les paramètres sont indicés par le type de matériau k, le palier[32] j ou le type de contrainte c du composant considéré. En effet, les matériaux, même au sein d'une tranche nucléaire donnée, peuvent être différents, et obéissent à une loi de probabilité différente. De même, à chaque type de palier j correspond une température différente. Il n'y a que deux types de contrainte c : la paroi *interne* du tube et la paroi *externe*. A chaque paroi sont associées une contrainte σ et une cinétique de fissuration χ. Ainsi, tout composant a deux dates de fissuration : une sur la paroi interne et une sur la paroi externe. Ces dates dépendent de cinq paramètres ℓ, θ, σ^{int}, σ^{ext}, χ^{int}, χ^{ext}. par la formule (7.4). Nous noterons N le nombre de composants susceptibles de se fissurer, K le nombre de matériaux différents (quatre dans notre cas), et J le nombre de paliers différents (quatre dans cette étude). Nous avons constaté que la paroi interne donne de plus fortes probabilité de fissuration. Les résultats donnés à la fin ne portent que sur celle-ci bien que les calculs soient toujours effectués sur les deux parois.

[32] Un palier est un ensemble des centrales de même architecture. Dans ce problème, les températures moyennes sont différentes suivant le palier.

Estimation de σ Nous disposons d'une estimation σ_{calc} de la contrainte pour chaque composant. Cette valeur, censée approximer σ est entachée d'une erreur et σ_{calc} peut donc différer notablement de la vraie valeur, inconnue, σ. On considère donc que la contrainte calculée σ_{calc} suit une loi normale autour de la valeur réelle σ avec un écart-type e_{calc} donné et mis à 30 MPa.

$$\sigma_{calc} \sim \mathcal{N}(\sigma, e_{calc}^2). \tag{7.9}$$

La modélisation de σ est représentée sur la gauche de la figure 7.1. Les valeurs de μ_σ et de e_σ doivent être fixées de manière à représenter la distribution des contraintes si on n'avait pas de méthode de calcul pour les estimer. La connaissance de ces contraintes est très faible, ce qui est traduit dans la modèle par un écart-type e_σ assez élevé.

Modélisation bayésienne Les experts ont une connaissance relativement incertaine des paramètres μ et e de chacune des lois que nous avons introduites. Cette connaissance est modélisée par de nouvelles lois de probabilités : en utilisant les lois conjuguées des paramètres de la loi normale, les moyennes sont modélisées par des lois normales, et les variances par des lois inverse gamma :

$$\begin{aligned}
\mu_{mat}(k) &\sim \mathcal{N}(m_{mat}(k) + \Delta_{mat}, \tau_{mat}^2(k)), && (7.10)\\
e_{mat}^2(k) &\sim \mathcal{IG}(\frac{\nu_{mat}(k)}{2}, \frac{s_{mat}^2(k)}{2}), && (7.11)\\
\mu_\theta(j) &\sim \mathcal{N}(m_\theta(j) + \Delta_\theta, \tau_\theta^2(j)), && (7.12)\\
e_\theta^2(j) &\sim \mathcal{IG}(\frac{\nu_\theta(j)}{2}, \frac{s_\theta^2(j)}{2}), && (7.13)\\
\sigma_{calc}(i) &\sim \mathcal{N}(\sigma(i) - \Delta_\sigma, e_{calc}^2), && (7.14)\\
e_{calc}^2 &\sim \mathcal{IG}(\frac{\nu_{calc}}{2}, \frac{s_{calc}^2}{2}), && (7.15)\\
\mu_\chi &\sim \mathcal{N}(m_\chi, \tau_\chi^2), && (7.16)\\
e_\chi^2 &\sim \mathcal{IG}(\frac{\nu_\chi}{2}, \frac{s_\chi^2}{2}). && (7.17)
\end{aligned}$$

Les hyperparamètres $m_.$, $\tau_.$, $\nu_.$ et $s_.$ sont fixés par l'utilisateur du modèle de manière à refléter ses connaissances sur modèle physique. Les variables $\Delta_.$ ont été introduites pour refléter les corrélations qui existent entre les paramètres de moyennes de différentes catégories. Nous expliquons ce point dans le paragraphe suivant.

Cohérence des paramètres Supposons dans un premier temps que les valeurs $\Delta_{\cdot}$ soient nulles. Dans ce cas, si on observe des fissures uniquement sur un seul type de matériau, une seule catégorie de tranche ou un composant particulier, alors les paramètres réactualisés ne concernent que la catégorie du composant, et non l'ensemble des composants du parc. Ceci vient du fait que la modélisation suppose que chaque catégorie a des paramètres indépendants des autres catégories. On souhaiterait par exemple qu'une réactualisation de $\mu_{mat}(k)$ à une valeur supérieure engendre du même coup une augmentation de tous les $\mu_{mat}(k')$ avec $k' = 1, \ldots, k-1, k+1 \ldots, K$. Afin de prendre en compte le fait qu'entre deux matériaux différents les valeurs de μ_{mat} sont liées, nous introduisons une nouvelle variable commune à tous les types de matériau qui influent sur les différentes lois de μ_{mat}. Cette variable notée Δ_{mat} représente un *décalage* additif systématique de tous les paramètres $\mu_{mat}(k)$ pour $k = 1, \ldots, K$:

$$\mu_{mat}(k) = m_{mat}(k) + \Delta_{mat} \tag{7.18}$$

Ce décalage hypothétique vaut évidemment 0 *a priori* mais peut prendre une toute autre valeur suivant les données du REX. il est clair que tous les indices matériau seront affectés par un tel décalage. Précisons que cette variable n'a pas de sens physique, et que son introduction dans le modèle n'a pour justification *que* la cohérence des paramètres *a posteriori* et la volonté de l'exploitant de vouloir propager ou non un événement de REX à l'ensemble des tranches. Exactement de la même manière, nous introduisons des décalages systématiques Δ_{θ} et Δ_{σ} dans la mesure de la température et dans la mesure de la contrainte. Pour obtenir des décalages positifs, on soustrait le décalage par rapport à σ.

Les trois nouvelles variables aléatoires Δ_{mat}, Δ_{θ} et Δ_{σ}, requièrent une loi de probabilité. En moyenne, le décalage vaut 0, c'est-à-dire qu'il n'y a aucun décalage au vu des connaissances *a priori* que l'on a. Le choix naturel est de prendre la loi normale de moyenne nulle et d'écart-type e_{Δ} qui reste à définir, mais qui est fixe (i.e. c'est un hyperparamètre) :

$$\Delta_{mat} \sim \mathcal{N}(0, e^2_{\Delta_{mat}}), \tag{7.19}$$

$$\Delta_{\theta} \sim \mathcal{N}(0, e^2_{\Delta_{\theta}}), \tag{7.20}$$

$$\Delta_{\sigma} \sim \mathcal{N}(0, e^2_{\Delta_{\sigma}}). \tag{7.21}$$

Les valeurs $e_{\Delta_{mat}}$, $e_{\Delta_{\theta}}$ et $e_{\Delta_{\sigma}}$ sont fixées par une expert suivant sa connaissance sur la force des relations qui lient les indices matériau entre eux, les températures de tranches différentes ou les calculs de contraintes. Par,

exemple, si l'expert pense que les indices matériau n'ont aucun rapport entre eux, il donnera une petite valeur à $e_{\Delta_{mat}}$. Inversement, s'il pense qu'une réactualisation de μ_{mat} pour un certain type de sensibilité doit entraîner systématiquement un décalage comparable pour les autres indices matériau, alors il prendra une valeur de $e_{\Delta_{mat}}$ relativement grande par rapport à s_{mat}. En revanche, aucune dépendance dans la cinétique de propagation χ n'a été modélisée pour des raisons physiques : la propagation interne et la propagation externe sont totalement différentes.

Pas d'incertitude sur le modèle physique Le modèle physique donne l'instant de fissuration d'un composant en fonction des valeurs exactes des trois paramètres ℓ, θ et σ. De la même manière, la valeur exacte de χ donne le temps exact écoulé entre l'amorçage et l'atteinte d'un seuil de profondeur critique. On considère donc que ces deux modèles (amorçage et propagation) sont systématiquement justes et que la seule source d'erreur provient de la mauvaise évaluation des paramètres. Cette hypothèse est évidemment invérifiable puisque les paramètres ℓ, θ, σ et χ ne sont jamais observés avec certitude. Dans une approche différente, on aurait pu modéliser le temps de fissuration comme une variable aléatoire autour de la valeur donnée par le modèle. Cela conduirait à un niveau d'incertitude supplémentaire dans le modèle. Ainsi, une fissure aurait une probabilité non nulle de s'amorcer en un temps très court. Cependant, l'observation d'une seule fissure pourrait alors provenir de l'aléa du modèle physique, et la réactualisation des paramètres serait donc naturellement plus faible. C'est donc dans un souci de *conservatisme* que nous n'introduisons pas d'incertitude à ce niveau. En outre, il n'y a pas de raison physique pour que le temps de fissuration soit aléatoire : ce sont les conditions de fabrication et d'exploitation du composant qui ne sont pas constantes, et qui sont responsables de l'aléa sur le temps de fissuration.

7.2.2 Modèle graphique

Un grand nombre de variables entrent en jeu dans la modélisation de la fissuration. Pour représenter les dépendances entre ces variables, un modèle graphique est très utile (figure 7.4). Les variables entourées par des carrés sont fixes, c'est-à-dire qu'elles sont soit observées, soit fixées par l'utilisateur. Les variables entourées d'un cercle sont aléatoires et nous cherchons leur loi conditionnellement aux variables fixes. Les cadres sont définis de manière à représenter la multiplication des variables. Leur en-tête permet en effet de définir le nombre de fois qu'il faut les dupliquer pour représenter toutes les variables.

Répéter un grand nombre de fois :

- $\Delta_{mat}|. \sim \mathcal{N}(m'_{\Delta_{mat}}, e'^2_{\Delta_{mat}})$
- $\Delta_\theta|. \sim \mathcal{N}(m'_{\Delta_\theta}, e'^2_{\Delta_\theta})$
- **Pour chaque type de contrainte** c
  - $\Delta_\sigma|. \sim \mathcal{N}(m'_{\Delta_\sigma}, e'^2_{\Delta_\sigma})$
  - $e^2_{calc}|. \sim \mathcal{IG}(\frac{\nu'_{calc}}{2}, \frac{s^{2'}_{mes}}{2})$
  - $\mu_\chi|. \sim \mathcal{N}(m'_\chi, e'^2_\chi)$
  - $e^2_\chi|. \sim \mathcal{IG}(\frac{\nu'_\chi}{2}, \frac{s^{2'}_\chi}{2})$
- **Pour chaque sensibilité** k**,**
  - $\mu_{mat}|. \sim \mathcal{N}(m'_{mat}, e'^2_{mat})$
  - $e^2_{mat}|. \sim \mathcal{IG}(\frac{\nu'_{mat}}{2}, \frac{s^{2'}_{mat}}{2})$
- **Pour chaque type de tranche** j**,**
  - $\mu_\theta|. \sim \mathcal{N}(m'_\theta, e'^2_\theta)$
  - $e^2_\theta|. \sim \mathcal{IG}(\frac{\nu'_\theta}{2}, \frac{s^{2'}_\theta}{2})$
- **Pour chaque composant numéro** i**,**

  Répéter plusieurs fois et dans le désordre
  - $\ell|. \sim \mathcal{N}_{t2}(\mu_{mat}, e^2_{mat}, (K^c_l, d(c))_{c=\{int,ext\}})$
  - $\theta|. \sim \mathcal{N}_{t2}(\mu_\theta, e^2_\theta, (K^c_\theta, d(c))_{c=\{int,ext\}})$
  - **Pour chaque type de contrainte** c
    - $\sigma|. \sim \mathcal{N}_t(\mu'_\sigma, e'^{\,2}_\sigma, K_\sigma, d)$
    - $\chi|. \sim \mathcal{N}_t(\mu_\chi, e^2_\chi, K^c_\chi, d(c))$
- **Pour chaque type de contrainte** c
  - $d|. \sim \mathcal{M}_d(S, (p_i)_{i=1\ldots N})$

TAB. 7.1 – Plan d'échantillonnage de Gibbs

Nous considérerons que $d(i)$ prend la valeur 1 pour les composants fissurés et 0 pour les composants non fissurés, de façon à avoir $\sum_i d_i = S$. À ce stade de l'étude, il s'agit d'estimer, dans une optique d'inférence bayésienne, les lois *a posteriori* des variables du modèle sachant des scénarios pour les lois *a priori* définies dans

le modèle graphique de la figure 7.4. Ces lois seront obtenues par intégration de la loi jointe *a posteriori* du modèle. L'ensemble des variables simulées est :

$$X = (\Delta_{mat}, \Delta_\theta, \Delta_\sigma, \mu_{mat}, \mu_\theta, \mu_\chi, e_{mat}, \\ e_\theta, e_{calc}, e_\chi, \ell, \theta, \sigma, \chi, d)\,. \tag{7.22}$$

Il nous faut donc estimer la loi de X. Cela se fait, comme décrit ci-dessous, par l'échantillonnage de Gibbs [55]. Partant de là, les estimateurs bayésiens des variables sont obtenus comme sous-produit de l'échantillonnage de Gibbs par intégration de Monte-Carlo sur la loi *a posteriori* approximée.

Notons que σ_{calc} est fixe, car c'est une donnée observée. La loi de X est calculée *conditionnellement* à cette variable. Nous avons introduit sa modélisation précédemment uniquement pour pouvoir obtenir la loi conditionnelle de X. Intuitivement, on peut comprendre qu'à partir des valeurs de σ et σ_{calc}, on aura une idée de l'erreur de calcul e_{calc}, et ainsi de toutes les variables dont elle dépend.

Lois conditionnelles Soit v une variable aléatoire du modèle, alors

$$P(v|.) \propto P(X) \tag{7.23}$$

où $P(v|.)$ représente la probabilité conditionnelle à l'ensemble des variables de X sauf v. On exploite la structure du modèle graphique dirigé grâce à la formule suivante [55] :

$$P(v|V_{-v}) = P(v|par[v]) * \prod_{w \in enf[v]} P(w|par[w]). \tag{7.24}$$

où $par(x)$ désigne l'ensemble des parents d'une variable x, et $enf(x)$ l'ensemble de ses enfants. Cette formule découle de l'application en chaîne de la formule de Bayes $P(A|B) = \frac{P(A \bigcap B)}{P(B)}$. La convergence de l'échantillonnage de Gibbs est théoriquement assurée par l'ergodicité (irréductibilité et apériodicité) de la chaîne de Markov. Nous devons donc simuler un très grand nombre de fois des réalisations de la loi jointe de X, portant sur les variables inobservées. L'utilisation de lois conjuguées permet de calculer les lois conditionnelles analytiquement. Dans notre cas, le choix de lois *a priori* normales et gamma pour les moyennes et les variances permet d'obtenir *a posteriori* une loi normale dont il suffit de caractériser les paramètres. Par exemple, L'application de la formule (7.24) donne :

$$\mu_\theta|. \sim \mathcal{N}(m'_\theta, e'^2_\theta) \tag{7.25}$$

avec

$$\frac{1}{e'^2_\theta} = \frac{1}{\tau^2_\theta} + \frac{|T_j|}{e^2_\theta} \tag{7.26}$$

$$m'_\theta = e'^2_\theta \left[\frac{m_\theta + \Delta_\theta}{\tau^2_\theta} + \sum_{i \in T_j} \frac{\theta(i)}{e^2_\theta} \right]. \tag{7.27}$$

où $|T_j|$ est le nombre de composants dans le palier de type j. Le lois conditionnelles des autres variables $\mu_.$, $e_.$ et $\Delta_.$ sont identifiées à des lois normales de la même manière, et ne sont pas reportées ici pour ne pas alourdir le texte.

Les paramètres ℓ, θ, σ et χ sachant d suivent des lois normales tronquées. Nous étudions ici le cas de la loi conditionnelle de ℓ. Cette loi peut être calculée en considérant les différentes valeurs possibles de $d(int)$ et $d(ext)$. Supposons que $d(int) = 1$ et $d(ext) = 0$, c'est-à-dire que le composant est fissuré sur sa paroi interne mais pas sur sa paroi externe.

$$\begin{aligned}
&P(\ell | d(int = 1), d(ext) = 0, \cdots) \\
&\propto P(\ell | \mu_{mat}, e_{mat}) P(d(int) = 1|.) P(d(ext) = 0|.) \\
&\propto exp\left(-\frac{1}{2} \frac{(\ell - \mu_{mat})^2}{e^2_{mat}} \right) \times \\
&\mathbf{1}_{t_{théo(\ell,\theta,\chi,\sigma(int))} > t_{max}} \times \mathbf{1}_{t_{théo(\ell,\theta,\chi,\sigma(ext))} < t_{max}} \\
&\propto exp\left(-\frac{1}{2} \frac{(\ell - \mu_{mat})^2}{e^2_{mat}} \right) \mathbf{1}_{K^{int}_l < \ell < K^{ext}_l}
\end{aligned} \tag{7.28}$$

avec pour c = {int,ext},

$$K^c_\ell = \log\left(\frac{\frac{t_{ref}}{t_{max} - \frac{\lambda}{e^\chi}}}{e^{-E_a(\frac{1}{\theta} - \frac{1}{\theta_0})} (\frac{\sigma(c)}{\sigma_{ref}})^4} \right). \tag{7.29}$$

La densité de ℓ correspond exactement à la densité d'une loi normale tronquée en K_l^- à gauche et en K^+_l à droite. Pour les autres cas de défaillance, le raisonnement est le même. La simulation de ces lois tronquées se fait par un algorithme d'acceptation-rejet adapté[139].

Probabilités de fissuration et calcul FORM La variable d a cependant un statut particulier. En effet la valeur que prend d sachant ℓ, θ, σ et χ est déterministe, et on ne peut donc pas simuler suivant cette loi conditionnelle. Il faut considérer un niveau hiérarchique supplémentaire, et étudier la loi (notée $\mathcal{M}_d$) de d sachant S, μ_{mat}, μ_θ, μ'_σ, μ_χ, e_{mat}, e_θ, e'_σ et e_χ. Pour éviter d'alourdir les notations, nous désignons ces variables par "." (cf. par exemple le

tableau 7.1). μ'_σ et e'_σ sont l'espérance et l'écart-type de σ sachant e_{calc}, μ_σ et e_σ. La loi $\mathcal{M}_d$ s'écrit de la manière suivante :

$$P(d|S,.) \propto P(S|d)P(d|.) \tag{7.30}$$

par double application de la règle de Bayes, et par indépendance de S et des autres variables conditionnellement à d. Nous avons $P(S|d) = \mathbf{1}_{(\sum_i d_i = S)}$ et $P(d|.) = \prod_{i=1}^{N} P(d_i)$. Les probabilités $p_i = P(d_i = 1|.)$ valent :

$$p_i = \int\int\int\int_{t_{def}(\ell,\theta,\sigma,\chi) > t_{max}} dP_\ell dP_\theta dP_\sigma dP_\chi. \tag{7.31}$$

Les probabilités P_ℓ, P_θ, P_σ et P_χ sont les lois des variables ℓ, θ, σ et χ conditionnellement à leurs variables parentes μ et e. Le calcul analytique de cette intégrale est extrêmement difficile, voire impossible, mais c'est un problème d'intégration que l'on rencontre couramment en fiabilité. Puisque ce sont des lois normales, on peut avoir recours à des méthodes d'approximation appelées First and Second Order Reliability Methods (FORM et SORM)[116]. Ces méthodes approchent la frontière d'intégration $t_{théo(\ell,\theta,\sigma,\chi)} = t_{max}$ par une droite ou une fonction quadratique. Nous avons utilisé la méthode FORM, décrite en annexe, effectuant l'approximation linéaire.

Plan d'échantillonnage Nous avons donc obtenu les lois de chacune des variables conditionnellement aux autres. Le plan d'échantillonnage complet est donné dans le tableau 7.1. $\mathcal{N}_t$ et $\mathcal{N}_{t2}$ désignent les lois normales avec une et deux troncatures. Les simulations des variables ℓ, θ, σ, χ dépendent fortement de l'ordre dans lequel elles sont simulées. Ainsi, ces variables sont simulées à chaque fois dans un ordre différent, et ceci plusieurs fois par itération (quatre fois dans nos expériences) afin d'augmenter la vitesse de convergence de la chaîne de Markov vers sa loi stationnaire.

7.3 Résultats

Un programme en C++ avec une interface Matlab a été réalisé. Il permet d'analyser la sensibilité des paramètres *a priori* et de détecter les éventuelles situations critiques susceptibles de favoriser l'apparition d'une fissure.

7.3.1 Comportement des simulations

Comme dans toutes les techniques basées sur les simulations par chaîne de Markov, les valeurs successives des variables sont corrélées, et le nombre d'échantillons issus de l'échantillonnage de Gibbs doit être suffisamment important. Ceci est illustré par les graphes de gauche et de droite de la figure 7.2. Nous avons empiriquement utilisé 50000 itérations, où plus exactement 5000 échantillons espacés de 10 itérations chacun. Au préalable, 500 itérations sont effectuées sans échantillonnage pour éliminer la dépendance à l'initialisation. Pour chaque variable simulée, la moyenne des valeurs échantillonnées est choisie comme estimateur *a posteriori*. Pour toutes les applications considérées, la corrélation entre les variables μ_{mat}, μ_θ, Δ_σ et Δ_χ est calculée et est inférieure à 0.02. Nous considérons donc qu'elles sont *a posteriori* proche de l'indépendance. La figure 7.2, au milieu, illustre cette absence de corrélation entre μ_{mat} et μ_θ. L'utilisation des densités marginales pour prendre des décisions d'inspection est donc possible, sans risque de non conservatisme du risque calculé.

	1 fissure	1 fiss.+cn	10 fissures	10 fiss.+cn
matériau 1	3.15	2.83	4.35	3.91
matériau 2	3.19	2.82	4.42	3.38
matériau 3	3.12	2.74	4.48	4.40
matériau 4	3.10	2.82	4.32	3.81
température 1	0.160	0.110	0.332	0.272
température 2	0.551	0.112	1.31	-0.251
température 3	0.571	0.431	0.596	0.543
température 4	0.173	0.0928	0.3	0.291
contrainte	0.631	0.601	1.59	1.18
cinétique	1.10	0.962	2.45	2.13

TAB. 7.2 – Quantification de la réactualisation des paramètres en fonction du scénario. Chaque valeur représente le décalage normalisé par l'écart-type *a priori* entre les moyennes *a priori* et *a posteriori*. Ici, "cn" indique la présence de contrôles négatifs (sans fissure).

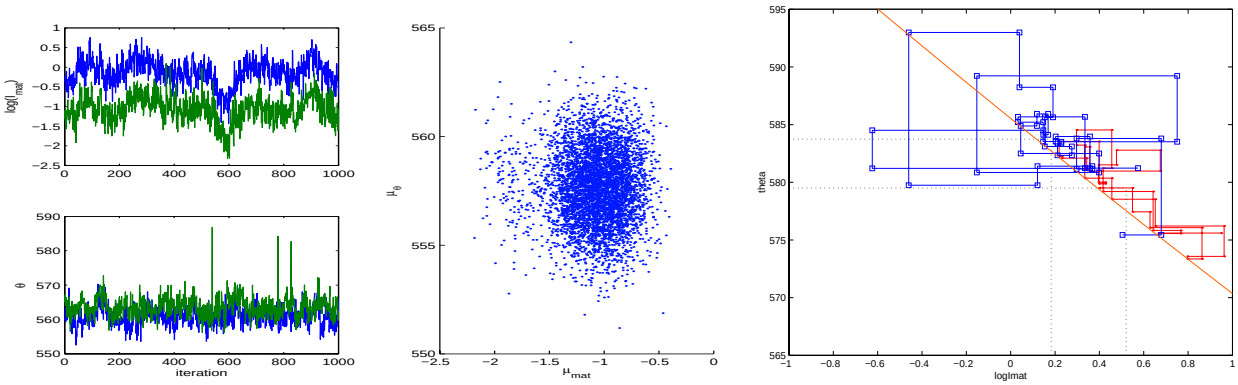


FIG. 7.2 – Illustrations de l'échantillonnage de Gibbs. A droite sont représentées 1000 valeurs de ℓ et θ simulées pour le composant le plus critique (en bleu, courbe supérieure) et un composant moins critique (vert). θ en fonction de ℓ est donné sur la figure du milieu, illustrant le fait qu'*a posteriori*, ces deux quantités sont proches de l'indépendance. La figure de droite montre la frontière de fissuration et quelques étapes de simulation pour le modèle simplifié aux paramètres ℓ et θ, les autres paramètres étants fixes. Les simulations correspondent à un scénario de 10 fissurations parmi 14 composants (bleu) et 14 fissurations parmi 14 (rouge). Ce dernier scénario n'ayant que des fissures, les valeurs sont simulées au dessus de la limite de fissuration.

7.3.2 Exploitation des résultats

Actuellement, aucune fissure n'a jamais été détectée lors des contrôles effectués. Les résultats présentés ici ne sont donc basés que sur des scénarios hypothétiques de fissuration, que nous pouvons modifier afin de bien appréhender le comportement du modèle.

Les hyperparamètres du modèle correspondent dans la plupart des cas à des moyennes et variances de grandeurs physiques connues (comme la température), ou de quantité étudiées (comme l'indice matériau). Ils ont donc été fixés par des experts à des valeurs plausibles. Certains paramètres comme l'écart-type du décalage commun Δ_{mat}, Δ_θ ou Δ_σ sont plus difficiles à interpréter. La valeur utilisée a donc été fixée après plusieurs essais du modèle, en fonction de la cohérence des valeurs *a posteriori*. Plusieurs essais ont montré qu'un décalage commum cohérent est obtenu en fixant $e_{\Delta_{mat}} = \frac{1}{2}\tau_{mat}$, $e_{\Delta_\theta} = \frac{1}{2}\tau_\theta$ et $e_{\Delta_\sigma} = \frac{1}{2}s_{calc}$. Afin d'illustrer l'intérêt de ce paramètre de cohérence, dans la première expérience, nous relachons la contrainte sur les températures en fixant $e_{\Delta_\theta} = \frac{1}{20}\tau_\theta$.

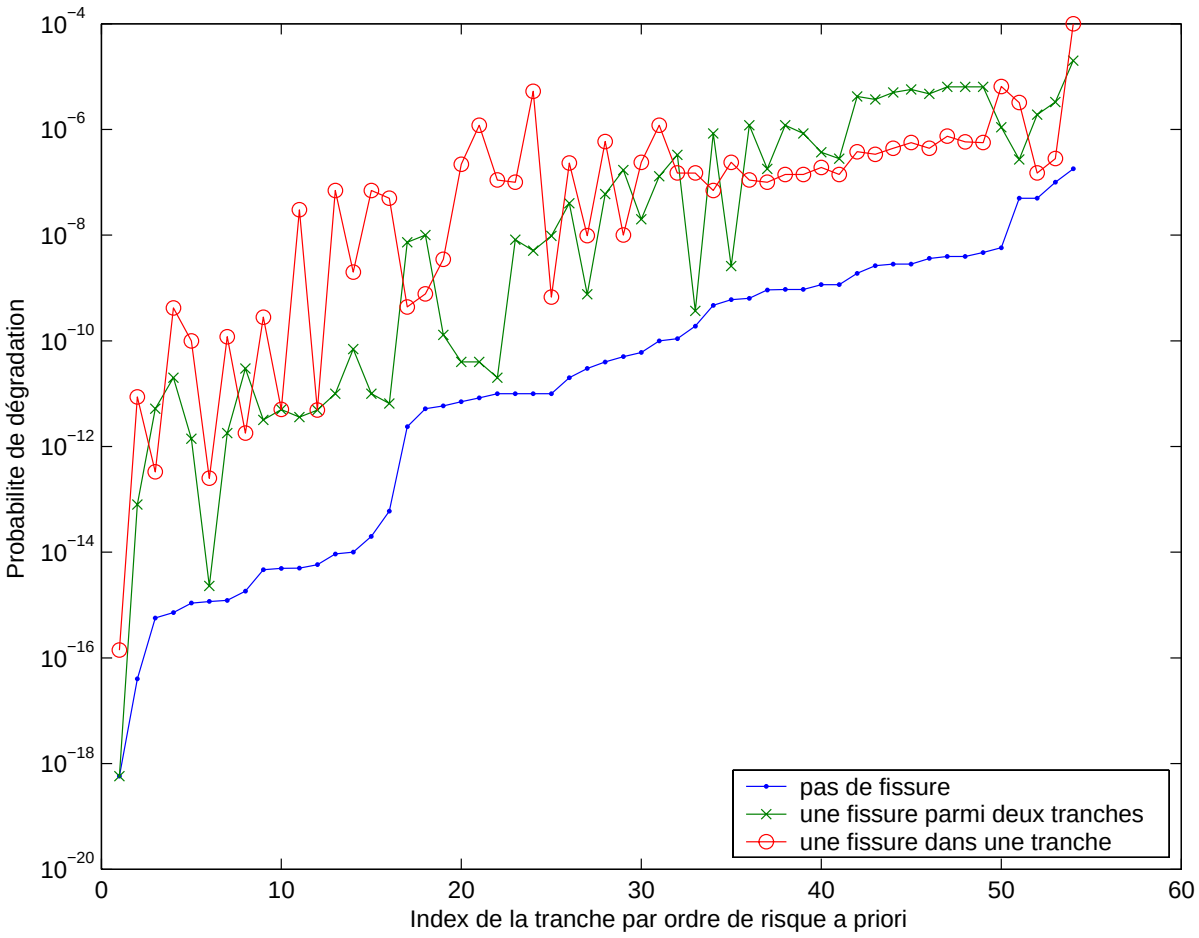


FIG. 7.3 – Risque de dégradation pour 54 tranches comportant chacune une cinquantaine de composants. Elles sont triées sur la première courbe (.) par risque *a priori*. La courbe verte (x) représente le scénario d'une fissure sur 2 tranches contrôlées et la courbe rouge (o) le scénario d'une fissure sur 1 tranche contrôlée.

Mesure de réactualisation des paramètres Si les connaissances *a priori* sont exactes, les simulations ne donnent aucune fissuration, confirmant le fait que les connaissances actuelles des experts donnent une très faible probabilité à l'amorçage des fissures. Elle est de l'ordre de 10^{-7} pour le composant le plus critique au bout de 30 ans de fonctionnement.

Les paramètres étant inchangés, différents scénarios de fissuration sont introduits. Nous avons choisis 108 composants contrôlés, sur lesquels nous observons soit une unique fissure, soit 10 fissures. Pour chaque cas, un contrôle éventuel donnant 0 fissure sur 689 autres composants est ajouté. Ces 689 composants sont tous de la catégorie de température 2. Pour les quatre scénarios ainsi obtenus, les lois *a posteriori* ont été estimées. La dates

des contrôles est fixée arbitrairement au 15 juin 2004.

Pour plus de clarté dans la présentation des résultats, une mesure de *décalage* entre l'*a priori* et *a posteriori* a été introduite : $dec = \frac{\mu_1 - \mu_0}{\sigma_0}$ où μ_0 et σ_0 sont les moyennes et les écarts-types *a priori* et μ_1 l'estimation de la moyenne *a posteriori*. Ces décalages reportés dans le tableau 7.2 quantifient les effets de la réactualisation pour différents scénarios avec les mêmes paramètres.

Le tableau 7.2 est assez riche en informations :

1. On peut constater qu'une seule fissure entraîne une très forte réactualisation des paramètres (de l'ordre de trois écarts-types). Ceci est dû au fait que l'observation d'une fissure est contradictoire avec les connaissances *a priori*.
2. Cette forte réactualisation se trouve diminuée par l'ajout des contrôles de non-fissuration. En effet, ces contrôles sont cohérents avec la loi *a priori* du modèle. Pouvoir quantifier de cette manière l'effet des contrôles négatifs est très important d'un point de vue applicatif.
3. Il y a une très forte différence de réactualisation entre les variables associées au matériau et les variables de température. En effet, la variable Δ_{mat}, en prenant de fortes valeurs permet d'obtenir *a posteriori* les plus grandes probabilités de fissuration. Ainsi, cette différence est essentiellement due au fait que e_{Δ_θ} est notablement différente de $e_{\Delta_{mat}}$. En intervenant sur ce paramètre *a priori*, on peut contrecarrer cette différence. Précisons qu'en plus des effets dus à $e_{\Delta_{mat}}$ et e_{Δ_θ}, les autres hyperparamètres ($\mu.$,$\tau.$,$\nu.$ et s), l'équation du modèle physique et les observations sont trois origines possibles des différences de réactualisation.
4. La valeur négative du décalage de la température numéro 2 n'est pas un aléa d'estimation, mais un comportement prévisible du modèle : tous les contrôles négatifs ayant été effectués pour ce type de température, lui donner une petite valeur est cohérent avec les observations, et dans le cas d'un scénario catastrophique (ici 10 fissures), la réactualisation des autres paramètres est si importante que l'absence de fissure ne peut être compensée que par une température plus faible qu'*a priori*.

Les résultats que nous obtenons sont purement virtuels puisqu'aucune fissuration n'a jamais été constatée, mais de tels scénarios permettent à un expert de fixer dynamiquement les paramètres *a priori*. Cette tâche, cruciale en analyse bayésienne [140] est en général délicate. Elle est ici réalisée en analysant la sensibilité du modèle face aux changements des différents paramètres. L'expert peut ainsi mesurer leur influence sur la prise de décision, et

déterminer par tâtonnements les valeurs qui correspondent à ses conjectures afin de définir un scénario de référence.

Probabilités de fissuration des tranches On a décidé ici d'étudier une réactualisation dans son ensemble, et donc de regrouper la totalité des composants d'une tranche. En effet, lorsqu'on effectue une inspection, l'ensemble des composants d'une tranche sont inspectés simultanément. C'est donc au niveau des tranches que se définit la stratégie d'inspection. Une tranche est dégradée si un de ses composants est fissuré. La probabilité de dégradation de chaque tranche a été calculée en utilisant les paramètres *a priori*, et les paramètres *a posteriori* pour le scénario précédent (détection d'une fissure parmi 108 composants, soit deux tranches contrôlées) et pour un scénario similaire portant sur la tranche pour laquelle le rique est plus faible : détection d'une fissure parmi 58 composants.

Les risques de dégradation *a priori* et *a posteriori* pour l'année 2010 ont été calculé pour 54 tranches, et représenté par ordre de risque de dégradation *a priori* sur la figure 7.3. On peut visualiser de quelle manière ce classement des tranches par risque de fissuration se trouve modifié par les deux scénarios.

On observe que pour toutes les tranches, le risque réactualisé est au moins égal au risque *a priori* : la détection des fissures ne peut que dégrader les estimations actuelles du modèle physique. On remarque aussi des fluctuations importantes pour certaines tranches, en raison de leur population spécifique (palier, sensibilité du matériau) qui les expose ou non à la réactualisation due a l'événement modélisé. Les valeurs de risque obtenues montrent que la réactualisation est importante : pour chaque composant, le risque de fissuration en 2010 augmente de plusieurs décades, en général trois, c'est-à-dire qu'il est multiplié par 10^3. Le modèle remplit donc son rôle qui est d'indiquer un risque nettement plus élévé si un événement de REX improbable survient.

À partir de scénarios de fissuration, il est possible avec notre modèle de décider des installations à contrôler en priorité : ce sont celles qui fournissent les plus fortes probabilités de fissuration a posteriori.

7.4 Discussion

Ce travail nous a permis d'incorporer à la fois

– des connaisances *a priori* sur un modèle physique,

– des données issues du REX.

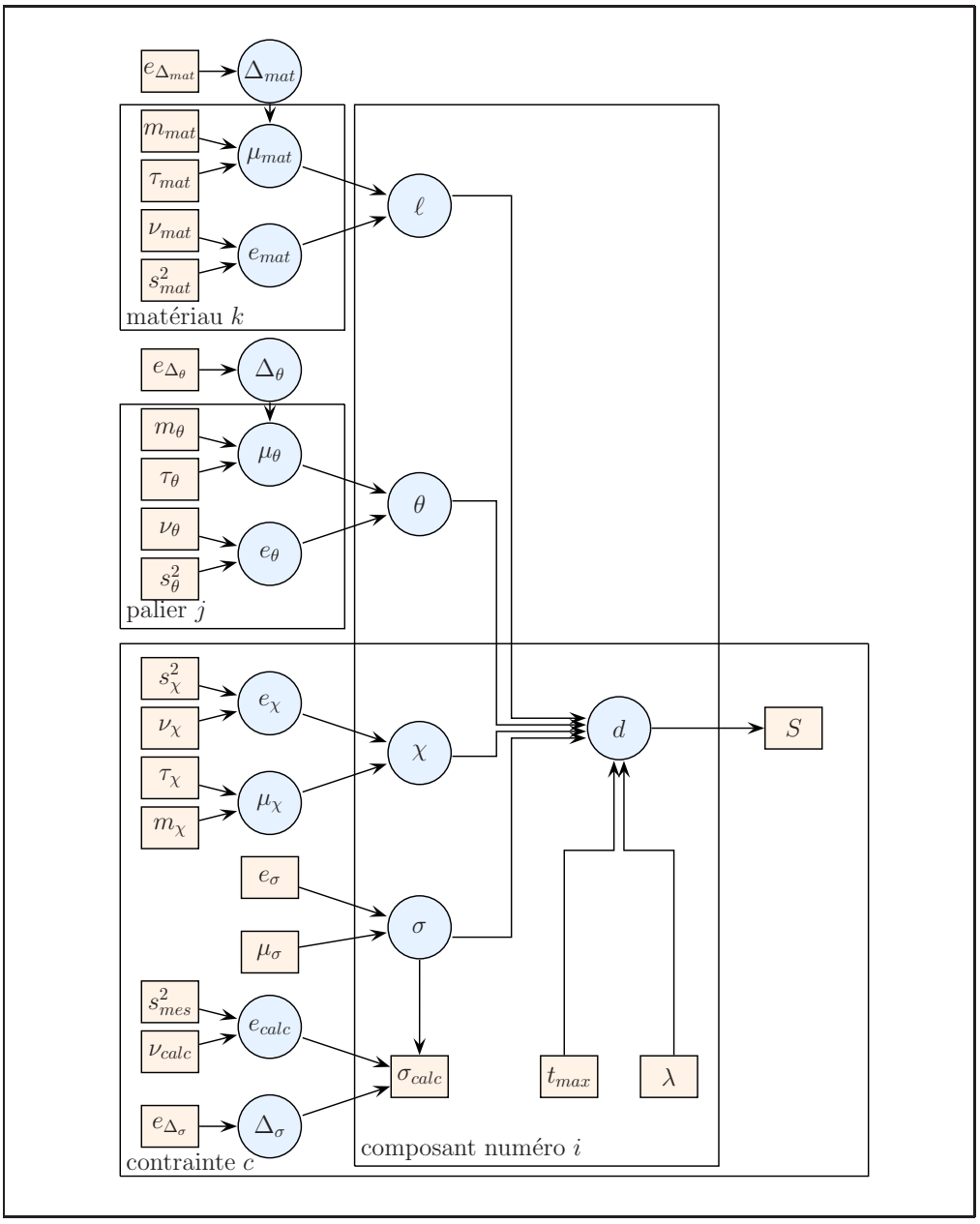


FIG. 7.4 – Modèle graphique complet. Le ronds représentent toutes les variables simulées par échantillonnage de Gibbs. La variable S désigne le nombre de fissures ($S = \sum_i d_i$).

Lorsque les connaissances *a priori* sont trop faibles pour garantir un niveau de risque donné, les données conformes au modèle (c'est à dire les contrôles négatifs) issues du REX peuvent s'avérer bénéfiques pour minimiser les risque.

L'étude effectuée modélise un très grand nombre de variables. La visualisation des dépendances par un modèle graphique a permis d'isoler chaque variable d'intérêt de construire le modèle de densité jointe par étapes suc-

cessives. L'analyse bayésienne est bien adaptée à la résolution du problème de réactualisation des paramètres en fonction du REX.

Le modèle peut cependant être trop simpliste, car il suppose des indépendances entre variables qui sont probablement loin de la réalité. De plus, on considère que le modèle de fissuration est parfaitement juste, ce qui n'est évidemment pas vérifié dans la réalité, mais modéliser un temps de fissuration aléatoire est par construction moins conservatif que notre approche.

La connaissance a priori est réactualisée par des mesures sur site, et l'étude de la distribution a posteriori permet de réévaluer les temps d'amorçage de chaque composant. Les résultats présentés montrent qu'en conjecturant la détection d'une seule fissure sur les 2800 composants, la réactualisation des paramètres du modèle des indices est significative : la rareté d'une information pénalisante est bien prise en compte et traduite quantitativement par le modèle statistique.

Annexe : Calcul FORM de fiabilité structurelle

Une étape spécifique de l'échantillonnage de Gibbs nécessite un calcul de fiabilité structurelle. Le but est de calculer l'intégrale suivante :

$$p = \int_{g(x)>0} \Phi(x; 0, Id), \tag{7.32}$$

où Φ représente la densité d'une loi gaussienne d-dimensionnelle et g une fonction dérivable quelconque.

Dans l'application présentée plus haut, $d = 4$ dimensions et la fonction g s'écrit :

$$g(x) = t_{def}(e_{mat}(x_1 + \mu_{mat}), e_\theta(x_2 + \mu_\theta), e_\sigma(x_3 + \mu_\sigma), e_\chi(x_4 + \mu_\chi)) - t_{max}.$$

où t_{def} est définie en (7.4).

L'équation (7.32) a une solution explicite lorsque $g(x)$ est une fonction linéaire : en notant β la projection orthogonale de l'origine sur le plan $\{x; g(x) = 0\}$, on a $p = F(\|\beta\|)$ où F est la fonction de répartition d'une loi normale centrée réduite.

Pour une fonction g dérivable quelconque, un développement au premier ordre de la frontière $g(x) = 0$ en un point particulier appelé *point de conception* permet de se ramener au cas linéaire précédent. C'est ainsi que se définit l'approximation FORM. Le point de conception β est le point de probabilité maximale appartenant au domaine $g(x) > 0$. Du fait de la sphéricité de la distribution normale,

$$\beta = \underset{g(x)\geq 0}{\operatorname{argmin}} \|x\|$$

On peut vérifier que β satisfait l'équation de point fixe $\frac{\partial}{\partial x} g(\beta) \propto \beta$ et un algorithme itératif [116] permet de trouver le point β en peu d'itérations. Ce calcul est illustré sur la Figure 7.5 : une approximation de la frontière $g(x) = 0$ au point β permet de calculer l'intégrale gaussienne sur le domaine réprésenté par des hachures.

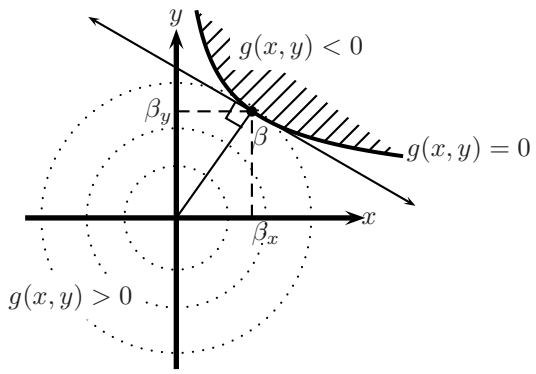


FIG. 7.5 – Approximation au premier ordre d'une probabilité de risque structurel. Les cercles symbolisent une distribution gaussienne sphérique. La partie grisée représente la zone de défaillance. Le caclul FORM correspond au calul du point de conception β.

Chapitre 8

Estimation de frontière par programmation linéaire

L'estimation de frontière en apprentissage statistique Nous proposons de nouvelles méthodes pour estimer la frontière d'un ensemble de points. Le problème d'estimation de frontière est relativement ancien, mais reste peu connu de la communauté de l'apprentissage statistique. Il consiste à estimer la valeur maximale d'une variable (profit, consommation, etc.) en fonction d'une ou plusieurs covariable(s). Etant données des entrées X et une sortie quantitative et bornée Y, le problème peut être défini sous la forme d'un problème de régression : la quantité à estimer est :

$$f(x) = \sup(Y|X = x),$$

où f est appelée fonction *frontière*. La spécificité du problème vient du fait qu'on ne cherche pas à estimer l'espérance conditionnelle ou un quantile conditionnel, mais la valeur maximale d'une quantité en fonction de covariables.

Au départ introduit dans des problèmes économétriques, ce problème consiste à estimer la valeur maximale d'une variable (par exemple la production maximale d'un certain produit) étant données plusieurs variables explicatives (énergie disponible, nombre d'ouvriers, etc.). Des applications de l'estimation de frontière apparaissent aussi en robotique et en vision par ordinateur lorsqu'on souhaite estimer une silhouette à partir d'un nombre fini

de points de l'espace. La figure 8.1 illustre ce type d'application.

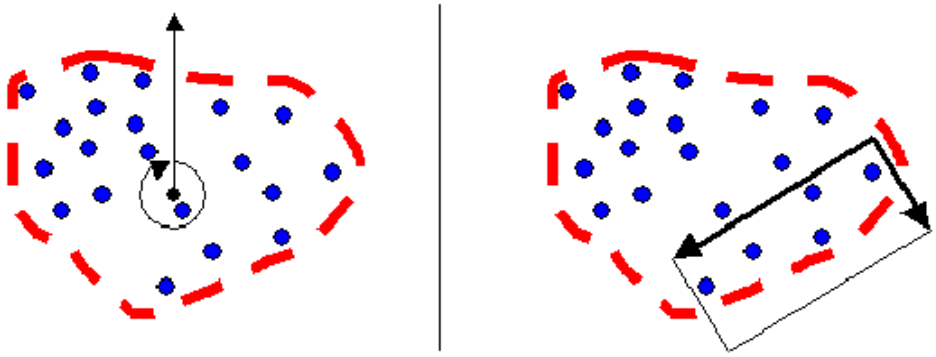


FIG. 8.1 – Illustrations de l'estimation de contour appliquée à des nuages de points connexes. Un pré-traitrement des données est nécessaire pour se ramener à un problème de régression. Il peut se faire par passage en coordonnées polaires (à gauche) ou par un traitement local des contours (à droite).

Nous avons vu en introduction qu'un ensemble important de méthodes d'apprentissage statistique consiste à définir un estimateur sous la forme d'une fonction noyau. Ce type de fonction est apprécié pour ses qualités d'approximation et de simplicité. Elles permettent d'exprimer un problème d'apprentissage non linéaire en un problème linéaire, sans définir explicitement la transformation des données. Dans le cas de l'estimation de frontière, l'introduction de tels estimateurs a permis de définir de manière très simple le problème d'estimation : Les estimateurs sont des fonctions noyaux contraintes à être au dessus de tous les points et dont le support de densité associé est de surface minimale. Ils sont définis comme des combinaisons linéaires de fonctions noyau appliquées à tous les points de l'échantillons d'apprentissage. Les poids de la combinaison linéaire sont ensuite obtenus comme solution d'un problème de programmation linéaire.

Pour des noyaux suffisament lisses, la solution du problème d'optimisation est parcimonieuse dans le sens où la plupart des poids obtenus sont nuls. Les points non nuls jouent le rôle de vecteurs supports. Dans le cas de densités uniformes dans $\mathbb{R}^2$, l'erreur L_1 entre la fonction estimée et la vraie frontière décroit presque sûrement vers 0. Sur des données simulées, nous illustrons le comportement de notre estimateur. Un avantage de la méthode proposée est qu'elle ne dépend des données d'entrées, qu'à travers le noyau défini pour des paires de points. Cet estimateur peut donc s'appliquer à d'autres types de données d'entrée que des valeur scalaires, moyennant la définition d'une

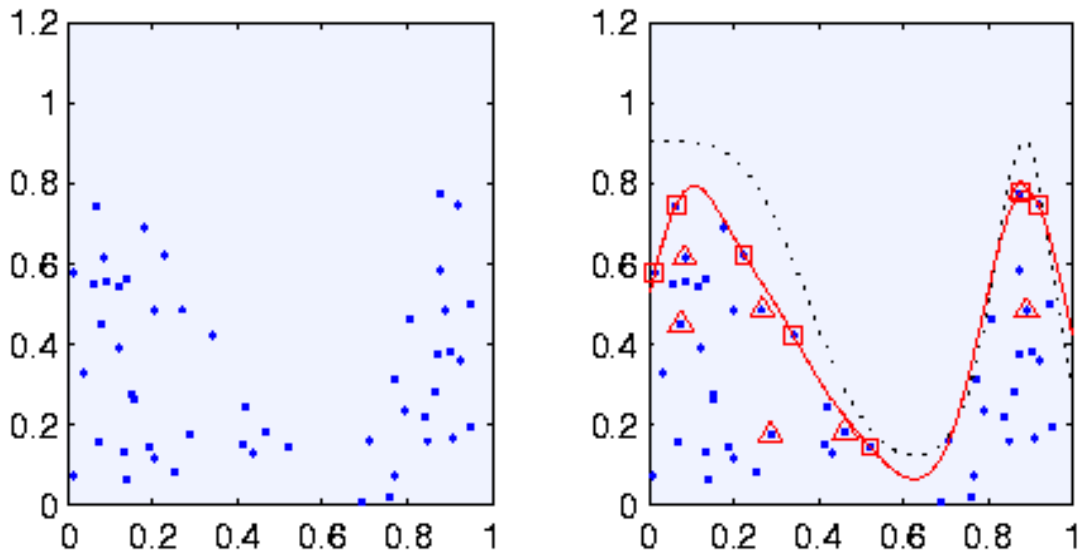


FIG. 8.2 – Illustration de la méthode d'estimation de frontière proposée dans ce chapitre. Sur la gauche sont représentées des données d'apprentissage simulées de manière uniforme sur l'aire délimitée par les axes et la frontière $f(x)$. Cette frontière est représentée en pointillés sur la figure de droite. Notre estimateur est représenté par la fonction en trait continu. Lors de l'apprentissage, un problème d'estimation sous contraintes (une contrainte par point) est résolu. Les points pour lesquels la contrainte est satisfaite sont représentés par des carrés. L'estimateur est parcimonieux : il ne s'exprime qu'en fonction des ordonnées des points représentés par des triangles.

fonction noyau adéquate. La figure 8.2 illlustre la méthode proposée dans ce chapitre.

8.1 Le problème de l'estimation de frontière

On trouve dans la littérature de nombreuses méthodes pour estimer un ensemble S à partir de l'observation d'un sous-ensemble de celui-ci généré aléatoirement. Ce problème d'estimation de frontière ou de support apparaît en classification [67], dans des méthodes de clustering [68], en analyse discriminante [14], et en détection de points aberrants. Diverses applications se retrouvent au niveau du diagnostic médical [102] et pour le monitoring des machines [112]. En analyse d'images, le problème de segmentation peut être considéré sous la forme d'un problème d'estimation de support, où le support est un ensemble connexe de $\mathbb{R}^2$ [97]. Ce problème apparaît aussi dans des applications économétriques [33].

Après renormalisation, le support à estimer peut s'écrire sous la forme :

$$S = \{(x, y) : \ 0 \leq \ x \leq \ 1 \ ; \ \ 0 \leq \ y \ \leq \ f(x)\}, \tag{8.1}$$

où $f : [0, 1] \rightarrow (0, +\infty)$ est une fonction inconnue. Ainsi, le problème se ramène à l'estimation de la fonction f appelée *fonction frontière* (voir par exemple [160]). Les données consistent en des paires (X, Y) où X représente les entrées (quantité de travail, d'énergie ou capital investi) et Y la variable de sortie (retour sur investissement, production maximale, etc.). La valeur $f(x)$ peut être interprétée comme le niveau maximal qui peut être atteint pour une configuration x. Dans [98] la fonction f est supposée croissante et concave, ce qui peut être justifié par des considérations économétriques, et mène à un type particulier d'estimation de frontière appelé *Data Envelopment Analysis* (DEA). C'est la plus petite fonction monotone et concave qui couvre tous les points de l'ensemble d'apprentissage. L'estimateur est donc linéaire par morceaux et à notre connaissance, c'est le premier estimateur basé sur une technique de programmation linéaire [1]. Sa distribution asymptotique est établie par [78].

Un article fondateur de Greffroy [79] propose un estimateur à partir d'observations identiquement distribuées sur le support. L'estimateur est une sorte d'histogramme basé sur les valeurs extrêmes de l'échantillon. Ce travail a été étendu de deux manières.

La première extension utilise des estimateurs polynomiaux locaux. Ils sont définis localement sous la forme du plus petit polynome couvrant les points dans la partie de l'histogramme considéré. Leur optimalité asymptotique au sens minimax est prouvée sous des hypothèses peu restrictives sur le taux de convergence α de la densité vers

0 [97, 159]. Des méthodes de valeurs extrêmes ont été proposées ensuite par Hall *et al* [66] ainsi que Gijbels et Peng pour en estimer les paramètres.

La seconde extension propose un lissage de l'estimateur de Geffroy's [56] et introduit des estimateurs à noyau ou à série orthogonale [57, 58]. La distribution asymptotique est obtenue en supposant que l'échantillon d'apprentissage est généré par un processus de Poisson. Dans le même esprit, Gardes [101] propose un estimateur de Faber-Shauer. Girard et Menneteau [59] généralisent ces approches en étudiant les estimateurs de support de la forme :

$$S = \{(x, y) : \ x \in E \ ; \ \ 0 \leq \ y \ \leq \ f(x)\},$$

où $f : E \rightarrow (0, +\infty)$ est une fonction inconnue et E un ensemble arbitraire. Dans chaque cas, la distribution asymptotique est établie. Enfin, d'autres auteurs, dont Abbar [3] et Jacob et Suquet [82] utilisent des approches similaires de lissage, mais les estimateurs ne sont pas basés sur des valeurs extrêmes ou des processus de Poisson.

L'estimateur proposé dans ce chapitre peut être considéré comme l'intersection de ces deux directions : il est défini comme un estimateur à noyau obtenu par lissage de points sélectionnés sur l'échantillon. Ces points sont obtenus automatiquement par résolution d'un problème de programmation linéaire en minimisant la surface délimitée par la fonction sous la contrainte d'être au dessus de tous les points d'apprentissage. Ses avantages sont les suivants : l'estimateur est obtenu par un algorithme d'optimisation standard (voir e.g. [43], chapitre 4), le lissage est directement lié à la forme de la fonction noyau choisie et bénéficie de propriétés théoriques intéressantes. Par exemple, nous prouvons qu'il est presque sûrement convergent au sens L_1. Cet estimateur est décrit dans la section suivante.

8.2 Résolution du problème d'estimation de frontière

8.2.1 Un problème de programmation linéaire

Nous considérons que toutes les variables aléatoires sont définies dans l'espace probabilisé $(\Omega, \mathcal{F}, P)$.
Le problème est d'estimer une fonction positive inconnue $f : [0, 1] \rightarrow (0, \infty)$ à partir d'observations $Z_N = (X_i, Y_i)_{i=1,\ldots,N}$. Ces dernières représentent une séquence i.i.d. de paires (X_i, Y_i) uniformément réparties dans l'ensemble S défini en (8.1). Pour simplifier, nous considérons dans la suite l'extension de f à l'espace réel $\mathbb{R}$ tout

entier en définissant $f(x) = 0$ pour tout $x \notin [0,1]$. Soit :

$$C_f \triangleq \int_0^1 f(u)\,du = \int_{\mathbb{R}} f(u)\,du,$$

chaque variable X_i est distribuée dans $[0,1]$ avec une densité $f(\cdot)/C_f$ et chaque Y_i suit une loi uniforme conditionnellement à X_i dans l'intervalle $[0, f(X_i)]$.

L'estimateur considéré de la frontière est choisi dans la famille de fonctions :

$$\begin{cases} \widehat{f}_N(x) = \sum_{i=1}^{N} K_h(x - X_i)\alpha_i\,, \qquad K_h(t) = h^{-1}K(t/h), \\ \alpha_i \geq 0, \qquad i = 1, \ldots, N, \end{cases} \tag{8.2}$$

où K est une fonction noyau $K : \mathbb{R} \to [0, \infty)$ d'intégrale unité et de paramètre de lissage $h > 0$. Chaque coefficient α_i représente l'importance du point (X_i, Y_i) dans l'estimation. En particulier, si $\alpha_i \neq 0$, le point correspondant (X_i, Y_i) peut être appelé un point support par analogie avec les Support Vector Machines (SVM). Nous renvoyons à l'ouvrage de Cristianini et Shawe-Taylor [30] pour un aperçu de ces méthodes ainsi qu'au chapitre 8 de Schölkopf et Smola [146] pour des application des SVM à l'estimation de quantile. L'estimation de frontière peut être vue sous la forme d'un quantile d'ordre 0, mais la méthode proposée dans [146] donne un estimateur sous la forme d'une fonction implicite, alors que l'estimateur que nous proposons est explicite, *i.e.* la valeur de f est déduite directement de x. La contrainte $\alpha_i \geq 0$ pour tout $i = 1, \ldots, N$ assure que $\widehat{f}_N(x) \geq 0$ pour tout $x \in \mathbb{R}$ et évite que l'estimateur ne soit trop irrégulier. Remarquons de plus que la surface du support estimé est donnée par :

$$\int_{\mathbb{R}} \widehat{f}_N(x)\,dx = \sum_{i=1}^{N} \alpha_i.$$

Cela suggère de définir le vecteur de paramètres $\alpha = (\alpha_1, \ldots, \alpha_N)^T$ sous la forme d'un problème de programmation linéaire de la manière suivante :

$$J_P^* \quad \triangleq \quad \min_{\alpha} \mathbf{1}^T \alpha \tag{8.3}$$

vrifiant :

$$A\alpha \geq Y \tag{8.4}$$

$$\alpha \geq 0. \tag{8.5}$$

La notation suivante a été introduite :

$$\begin{aligned} \mathbf{1} &\triangleq (1,1,\ldots,1)^T \in \mathbb{R}^N \\ A &\triangleq \{K_h(X_i - X_j)\}_{i,j=1,\ldots,N} \\ Y &\triangleq (Y_1,\ldots,Y_N)^T. \end{aligned}$$

Ainsi, $A\alpha = (\widehat{f}_N(X_1),\ldots,\widehat{f}_N(X_N))^T$, et le vecteur de contraintes (8.4) correspond à $\widehat{f}_N(X_i) \geq Y_i$, $i = 1,\ldots,N$. En d'autres termes, $\widehat{f}_N$ définit l'estimateur à noyau du support recouvrant tous les points et de surface minimale. En pratique (voir Section 8.4 pour une illustration) la solution du programme linéaire est parcimonieuse au sens où $n(\alpha) = \#\{\alpha_i \neq 0\}$ est petit (pour une valeur de h convenable, voir le paragraphe 8.2.3) et ainsi l'estimateur résultant est rapide à calculer, même pour des échantillons de grande taille.

Notons que l'estimateur que nous venons de décrire (8.2)–(8.5) pourrait être obtenu par maximisation de la vraisemblance relativement à la famille d'approximation (8.2). En effet, la densité jointe des observations Z_N étant donnés les paramètres de la fonction $f(x)$ peut être écrite :

$$p(Z_N \mid f) = \prod_{i=1}^{N} \frac{f(X_i)}{C_f} \cdot \frac{1}{f(X_i)} \mathbf{1}\{0 \leq Y_i \leq f(X_i)\}.$$

De plus,

$$C_f\Big|_{f=\widehat{f}_N} = \sum_{i=1}^{N} \alpha_i,$$

et ainsi, la log-vraisemblance est :

$$L(\alpha) \triangleq \log p(Z_N \mid \widehat{f}_N) = -N \log \sum_{i=1}^{N} \alpha_i + \sum_{i=1}^{N} \log \mathbf{1}\{Y_i \leq \widehat{f}_N(X_i)\},$$

et sa maximisation (sur l'espace des paramètres α_i positifs) est équivalente aux problèmes (8.3)–(8.5).

8.2.2 Correction de bord

Dans cette définitions de base, la frontière estimée ne s'annule pas aux limites du support. Les estimateurs à noyau sont connus pour avoir des effets indésirables aux bords d'un support fermé. En effet, les fonction de base décroissent généralement vers zéro de chaque côté. On peut aisément vérifier que l'estimateur décroit aussi aux bords du support. Des corrections de bord peuvent être appliquées pour corriger cet effet. Nous avons utilisé une technique de *pseudo-données* [29]. De chaque côté du support, les données sont mises en miroir de manière à

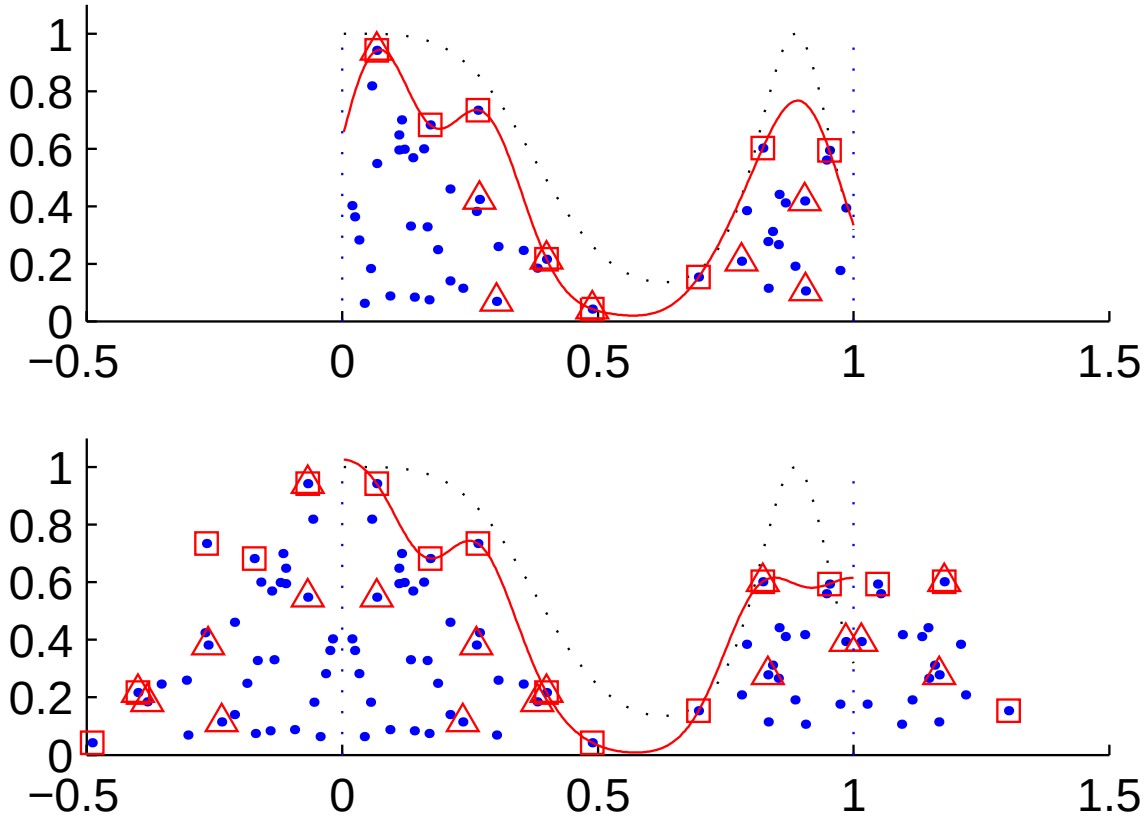


FIG. 8.3 – Illustration de la correction de bord par introduction de *pseudo-données*. Dans la méthode originale (haut), la fonction estimée décroit aux points $x = 0$ et $x = 1$. Les données sont dupliquées de chaque coté du support et la fonction estimée est plus proche de la fonction réelle (ligne en pointillés).

définir de nouveaux points à l'exterieur du domaine. Ces point sont utilisés dans le problème d'optimisation. Ils permettent à l'estimateur d'être croissant aux abords des limites du support. La Figure 8.3 illustre cette méthode.

8.2.3 Choix du paramètre de lissage

Nous définissons ici une méthode pour choisir le paramètre de lissage h. Les résultats asymptotiques de la section 8.3 (voir Corollaires 1 et 2) ne donnent qu'une vitesse de convergence asymptotique qui n'est pas utilisable dans des situations pratiques. Supposant que (X, Y) est uniformément distribué sur S, nous obtenons deux estimations différentes de $E[Y]$:

– $\hat{m}_1 = \frac{1}{N} \sum_{i=1}^{N} Y_i$

– $\hat{m}_2 = \frac{1}{N}\sum_{i=1}^{N} \frac{\hat{f}_N(X_i;h)}{2}$

Ces deux estimations sont supposées être égales lorsque la frontière estimée $\hat{f}_N$ est proche de f. Nous proposons donc de sélectionner la valeur $\hat{h}_N$ minimisant la quantité

$$D(h) = \frac{1}{N}\left|\sum_{i=1}^{N} Y_i - \frac{1}{2}\sum_{i=1}^{N} \hat{f}_N(X_i;h)\right|.$$

Ce critère est testé sur des simulations dans la section 8.4.

8.2.4 Comparaison avec les autres méthodes

Remarquons que des études antérieures proposent des solutions pour estimer α dans l'expression (8.2). Girard et Menneteau [59] considèrent une partition $\{I_r : \ 1 \leq r \leq k\}$ de $[0,1]$, avec $k \to \infty$. Pour tout $1 \leq r \leq k$, ils introduisent :

$$D_r = \{(x,y) : x \in I_r,\ 0 \leq y \leq f(x)\},$$

la partie de S définie dur I_r, $Y_r^* = \max\{Y_i; (X_i,Y_i) \in D_r\}$, et les estimateurs :

$$\widehat{\alpha_i} = \begin{cases} \lambda(I_r)Y_r^* & \text{if } \exists\, r \in \{1,\ldots,k\}\,;\, Y_i = Y_r^* \\ 0 & \text{sinon,} \end{cases}$$

où λ est la mesure de Lebesgue. Ils proposent l'estimateur suivant :

$$\check{f}_N(x) = \sum_{r=1}^{k} K_h(x - x_r)\lambda(I_r)Y_r^*,$$

où x_r est le centre de I_r. Cette approche a certains aspects pratiques difficiles, comme le choix de la partition, et plus précisément le choix de k. Dans notre cas, la résolution du problème d'optimisation linéaire (8.3)–(8.5) donne directement les vecteurs supports, *i.e.* la partition du domaine une fois que h a été estimé (par exemple par la méthode de la section 8.2.3).

Dans ce sens, l'estimateur proposé par Barron *et al.* [13] est similaire à $\widehat{f}_N$. Il est défini par une série de Fourier :

$$\widehat{g}_N(x) = c_0 + \sum_{k=1}^{M} a_k \cos(2\pi k x) + \sum_{k=1}^{M} b_k \sin(2\pi k x),$$

Le vecteur de paramètres $\beta = (c_0, a_1, \ldots, a_M, b_1, \ldots, b_M)^T$ est solution du problème d'optimisation linéaire :

$$\min c_0 \quad \left(= \int_0^1 \widehat{g}_N(x)dx\right) \tag{8.6}$$

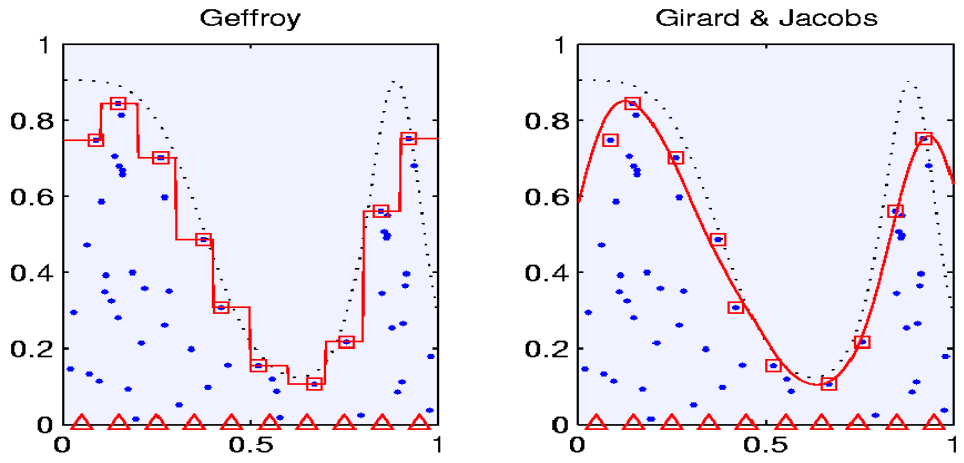


FIG. 8.4 – Illustration des méthodes d'estimation de frontière existantes. L'estimateur de Geffroy (gauche) n'est pas continu. La version noyau de cette méthode (droite) donne un estimateur dérivable mais permet à certains points d'apprentissage d'être au dessus de la fonction estimée. Les carrés représentent les points qui contribuent à l'estimation et les triangles sont les centres des fonctions noyau.

sous les contraintes :

$$\widehat{g}_N(X_i) \geq Y_i, \quad i = 1, \ldots, N \tag{8.7}$$

$$\sum_{k=1}^{M} k(|a_k| + |b_k|) \leq L/(2\pi). \tag{8.8}$$

Ainsi, $\widehat{g}_N$ définit l'estimateur de Fourier du support couvrant tous les points (équation (8.7)), L-Lipschitzienne (équation (8.8)) et avec la plus faible surface (équation (8.6)). D'un point de vue théorique, cet estimateur bénéficie d'une optimalité minimax. Il est comparé à $\widehat{f}_N$ dans une situation pratique en Section 8.4 pour différents choix des paramètres M, L et h.

8.3 Résultats théoriques

Dans cette section, nous donnons des justifications théoriques de la consistance de l'estimateur. Le résultat principale est la convergence presque sûre de l'estimateur $\widehat{f}_N$ vers la fonction réelle pour la norme L_1 sur $[0, 1]$. Ce résultat est valide sous des hypothèses basiques sur la fonction frontière :

A1. $0 < f_{\min} \leq f(x) < f_{\max} < \infty$, pour tout $x \in [0, 1]$,

A2. $|f(x) - f(y)| \leq L_f|x - y|$, pout tout $x, y \in [0, 1], \quad L_f < \infty$.

De plus, les hypothèses suivantes sur la fonction noyau sont considérées :

B1. $K(t) = K(-t) \geq 0$,

B2. $\int K(t)\, dt = 1$,

B3. $|K(s) - K(t)| \leq L_K|s - t|,\ L_K < \infty$,

B4. $\int K^2(t)\, dt < \infty$ et $\int t^2\, K(t)\, dt < \infty$.

Dans la suite, nous noterons $\|\hat{f}_N - f\|_1 = \int |f\hat{(x)}_N - f(x)|dx$. Il est maintenant possible d'écrire le théorème de convergence principal ([54]) :

Theorème 2 *Soient $h \to 0$ et $\log N/(Nh^2) \to 0$ lorsque $N \to \infty$. Supposons que les suppositions A et B précédentes sont vérifiées. Alors, l'estimateur (8.2)–(8.5) a les propriétés asymptotiques suivantes :*

$$\limsup_{N\to\infty} \varepsilon_1^{-1}(N)\|\hat{f}_N - f\|_1 \leq C < \infty \quad \text{a.s.}$$

avec $\varepsilon_1(N) \triangleq \max\left\{h, \sqrt{\log N/(Nh^2)}\right\}$.

Ces résultats mênent naturellement vers une borne sur la vitesse de convergence de l'estimateur :

Corollary 2 *La vitesse de convergence maximale garantie par le théorème 2*

$$\|\hat{f}_N - f\|_1 = O_p\left((\log N/N)^{1/4}\right)$$

est atteinte pour $h \asymp (\log N/N)^{1/4}$.

Ce taux de convergence peut être amélioré au prix d'une légère modification de l'estimateur afin d'éviter de trop fortes variations locales de l'estimateur. Dans la suite, une contrainte supplémentaire est imposée aux paramètres α_i pour qu'ils soient tous de l'ordre de $1/N$. La contrepartie de cette modification est qu'un estimateur nouveau $\tilde{f}_N$ sera basé habituellement sur plus de vecteurs supports que $\widehat{f}_N$.

Nous modifions donc l'estimateur (8.2)–(8.5) de la façon suivante :

$$\tilde{f}_N(x) = \sum_{i=1}^{N} K_h(x - X_i)\alpha_i \tag{8.9}$$

où le vecteur $\alpha = (\alpha_1, \ldots, \alpha_N)^T$ est défini grâce au problème de programmation linéaire modifié :

$$J^*_{MP} \quad \triangleq \quad \min_{\alpha} \mathbf{1}^T \alpha \tag{8.10}$$

$$\text{sous la contrainte :}$$

$$A\alpha \geq Y \tag{8.11}$$

$$0 \leq \alpha \leq C_\alpha / N \tag{8.12}$$

avec la constante :

$$C_\alpha > f_{\max}. \tag{8.13}$$

Remarque. En réalité, nous devons aussi nous assurer que $C_\alpha > C_f$ ce qui est impliqué par (8.13).

L'estimateur modifié (8.9)–(8.13) est différent de celui de (8.2)–(8.5) grâce à la borne autour de chaque valeur α_i précédentes (voir les contraintes (8.12)).

Theorème 3 *Soit $h \to 0$ et $\log N/(Nh) \to 0$ alors que $N \to \infty$. Soit la fonction $K(\cdot)$ a un support fini, c'est-à-dire $K(t) = 0 \; \forall |t| \geq 1$, et les suppositions A et B sont vraies. Ensuite, l'estimateur (8.9)–(8.13) a les propriétés asymptotiques suivantes :*

$$\limsup_{N \to \infty} \varepsilon_2^{-1}(N) \|\tilde{f}_N - f\|_1 \leq C < \infty \quad \text{a.s.}$$

avec $\varepsilon_2(N) \triangleq \max\left\{h, \sqrt{\log N/(Nh)}\right\}$.

Remarque. Le support de $K(\cdot)$ est contraint à être dans l'intervale $[-1, 1]$ sans perte de généralité.

Corollaire 1 *Le taux de convergence maximal garanti par le Théorème 3*

$$\|\tilde{f}_N - f\|_1 = O_p\left((\log N/N)^{1/3}\right)$$

est obtenu pour $h \asymp (\log N/N)^{1/3}$.

Notons que les taux de convergence des méthodes DEA et FDH sont en $O_p(N^{-2/3})$ [2] et $O_p(N^{-1/2})$ [130], mais des hypothèse plus fortes (monotonicité ou concavité de la fonction frontière).

8.4 Expériences numériques

Nous présentons des simulations qui illustrent le comportement de l'estimateur à noyau $\widehat{f}_N$ et le comparons à l'estimateur basé sur les séries de Fourier $\widehat{g}_N$ proposé par Barron *et al* [13]. Comme ce dernier nécessite que la fonction $\widehat{g}_N$ soit périodique, nous choisissons f de manière à ce que $f(0) = f(1)$. En outre, de manière à éviter les effets de bords liés à l'espace des covariables x, nous considérons des fonctions proches de zéro lorsque x est proche de 0 ou de 1. La fonction choisie s"ecrit :

$$\begin{aligned} f(x) = 0.1 \quad &+ \quad 5(x-0.1)\mathbf{1}_{\{x>0.1\}} \\ &- \quad 5(x-0.2)\mathbf{1}_{\{x>0.2\}} \\ &+ \quad 1(x-0.5)\mathbf{1}_{\{x>0.5\}} \\ &- \quad 9(x-0.8)\mathbf{1}_{\{x>0.8\}} \\ &+ \quad 8(x-0.9)\mathbf{1}_{\{x>0.9\}}. \end{aligned}$$

Elle est linéaire par morceaux et lipchitzienne de constante de Lipschitz $L_f = 8$. Son graphe est donné dans la partie de droite de la figure 8.5 (ligne en pointillés). $N = 50$ points ont été générés uniformément sur le domaine S borné supérieurement par f (part de droite de la Figure 8.5). Le paramètre de lissage h est choisi par la méthode proposée Section 8.2.3. A droite, l'estimateur à noyau (ligne continue) est superposé avec la fonction inconnue f. Les carrés représentent les points pour lesquels $\widehat{f}_N(X_i) = Y_i$. Les triangles représentent les vecteurs support (*i.e.* les points pour lesquels $\alpha_i > 0$). Sur cet exemple, l'estimateur $\widehat{g}_N$ est notablement différent de $\widehat{f}_N$.

Une comparaison plus précise necessite d'être effectuée. Pour chaque échantillon d'apprentissage simulé, l'erreur L_1, la valeur Δ_N et le nombre de paramètres effectifs np (c'est-à-dire n_α et $n_\beta = \#\{\beta_i \neq 0\}$) sont évalués pour $N = 25$ et $N = 100$. La valeur moyenne et l'écart-type de ces quantités sont évalués sur 1000 réplications de l'expériences. L'estimation est obtenue avec différentes valeurs des paramètres, à savoir h pour l'estimateur à noyau, L et M pour les estimateurs de Fourier. Pour chaque valeur h, le critère $D(h)$ de la Section 8.2.3 est évalué. Le choix adaptatif de L et M n'est pas implémenté dans cette expérience. Les résultats sont présentés sur les tableaux 8.1 et 8.2. L'erreur la plus faible est mise en gras pour chaque estimateur. Notons que l'erreur moyenne des deux estimateurs est plutôt similaire. En fait, l'estimateur à noyau donne une erreur légèrement plus faible pour des échantillons de petites taille alors que l'estimateur de Fourier est meilleur sur des échantillons de grande taille,

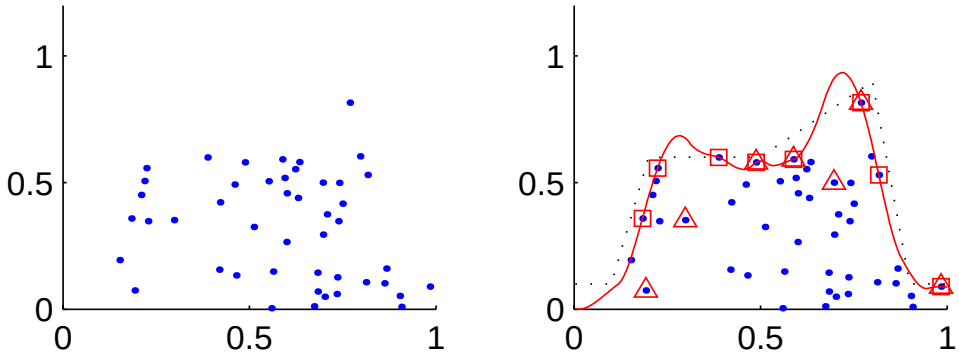


FIG. 8.5 – Illustration de l'estimateur à noyau sur les données simulées de la section 8.4. Comme pour les figures précédentes, les triangles représentent les centres des fonction noyaux (coefficients α_i non nuls), les carrés sont les vecteurs supports, c'est-à-dire les points pour lesquels la contrainte $\hat{f}_N(x_i) \geq y_i$ est activée.

confirmant son optimalité asymptotique. Remarquons que l'écart-type de l'erreur L_1 est en général plus faible avec l'estimateur à noyau. Par rapport au nombre de paramètres, l'estimateur à noyau semble plus parcimonieux que l'estimateur de Fourier. Enfin, on peut noter que le critère $D(h)$ donne un choix raisonnable du paramètre de lissage, avec une légère tendance à sur-estimer sa valeur, c'est-à-dire à lisser plus que nécessaire.

8.5 Discussion

Nous avons défini dans ce chapitre une nouvelle méthode d'estimation de frontière basée sur un estimateur à noyau. Cet estimateur est à la fois simple à estimer, parcimonieux, et relativement efficace sur les simulations que nous avons effectuées.

Cependant, comme il a été remarqué dans la section 8.3, le taux de convergence de l'erreur L_1 n'est pas optimal. Ce problème peut être résolu en modifiant l'estimateur par des contraintes supplémentaires de manière à le forcer à avoir la même constante de Lipschitz que la fonction qu'il estime. Les preuves pourraient aussi être adaptées à des distributions plus réalistes que la loi uniforme, et sur des domaines de dimension supérieure, la méthode ne dépendant que de la fonction noyau définie entre deux points. Dans le cas de lois non-uniformes, il est nécessaire de faire des hypothèses sur le comportement de la distribution au voisinage de la fontière. D'un point de vue plus pratique, la procédure de sélection pourrait être améliorée de manière à corriger le biais positif

systématique observé sur nos simulations.

estimateur	h	L	M	Δ_N	np	$D(h)$
kernel	0.120			0.113 (0.033)	4.476 (0.840)	43.692
	0.150			**0.111 (0.032)**	**3.613 (0.649)**	**38.232**
	0.180			0.122 (0.027)	3.093 (0.623)	34.362
	0.210			0.137 (0.023)	2.702 (0.590)	32.258
	0.240			0.153 (0.023)	2.434 (0.557)	33.793
	0.270			0.163 (0.028)	2.184 (0.403)	37.642
Fourier		3	4	0.142 (0.036)	4.534 (0.741)	
		5	**4**	**0.116 (0.043)**	**5.532 (1.008)**	
		7	4	0.124 (0.040)	6.636 (1.193)	
		9	4	0.135 (0.040)	7.312 (1.235)	
		11	4	0.143 (0.041)	7.690 (1.219)	
		13	4	0.150 (0.041)	7.870 (1.172)	
Fourier		3	8	0.143 (0.036)	4.545 (0.757)	
		5	**8**	**0.117 (0.043)**	**5.574 (1.055)**	
		7	8	0.126 (0.041)	6.771 (1.321)	
		9	8	0.139 (0.042)	7.691 (1.584)	
		11	8	0.150 (0.042)	8.376 (1.756)	
		13	8	0.159 (0.042)	8.877 (1.900)	

TAB. 8.1 – Résultats sur 1000 simulations avec $N = 25$ points. La valeur moyenne de Δ_N et np (nombre de paramètres effectifs) est donnée avec l'écart-type entre parenthèses.

estimateur	h	L	M	Δ_N	np	$D(h)$
kernel	0.050			0.072 (0.015)	13.777 (1.351)	31.972
	0.070			0.058 (0.013)	9.976 (1.249)	23.200
	0.090			**0.058 (0.011)**	**7.558 (1.104)**	**17.611**
	0.110			0.061 (0.011)	5.696 (0.887)	15.210
	0.130			0.073 (0.011)	4.725 (0.714)	15.881
	0.150			0.082 (0.012)	3.964 (0.524)	17.337
Fourier		3	4	0.123 (0.021)	5.080 (0.716)	
		5	4	0.073 (0.019)	5.792 (0.781)	
		7	**4**	**0.060 (0.011)**	**7.850 (0.979)**	
		9	4	0.063 (0.012)	8.751 (0.579)	
		11	4	0.067 (0.014)	8.896 (0.362)	
		13	4	0.069 (0.015)	8.955 (0.235)	
Fourier		3	8	0.124 (0.021)	5.117 (0.752)	
		5	8	0.073 (0.020)	5.889 (0.863)	
		7	8	0.057 (0.012)	8.315 (1.470)	
		9	**8**	**0.057 (0.013)**	**10.620 (1.672)**	
		11	8	0.061 (0.014)	12.454 (1.858)	
		13	8	0.067 (0.014)	13.890 (1.879)	

TAB. 8.2 – Résultats sur 1000 simulations avec $N = 100$ points. La valeur moyenne de Δ_N et np est donnée avec l'écart-type entre parenthèses.

Annexe : Preuves

Nous donnons ici les preuves des théorèmes 2 et 3 donnés Section 8.5. Ces théorèmes sont obtenus à partir de bornes supérieures et inférieures de l'estimateur. Nous commençons par expliciter ces bornes.

Borne supérieure de $\widehat{f}_N$

Lemma 3 *Soient $h \to 0$ et $\log N/(Nh) \to 0$ lorsque $N \to \infty$. Supposons que les hypothèses A et B définies plus haut sont vérifiées. Alors, pour presque tout $\omega \in \Omega$ il existe un nombre fini $N_0(\omega)$ tel que :*

$$J_P^* \leq C_f + O(h) + O\left(\sqrt{\log N/(Nh)}\right), \qquad \forall\, N \geq N_0(\omega), \tag{8.14}$$

où $O(h)$ et $O\left(\sqrt{\log N/(Nh)}\right)$ sont des quantités non aléatoires.

Preuve du Lemme 3. 1. Comme la fonction noyau $K(.)$ est supposée paire, la matrice A est symétrique, et le problème dual associé à (8.3) – (8.5) s'écrit :

$$J_D^* \triangleq \max_{\lambda} Y^T \lambda \tag{8.15}$$

$$\text{subject to}$$

$$A\lambda \leq \mathbf{1} \tag{8.16}$$

$$\lambda \geq 0. \tag{8.17}$$

Remplaçons le vecteur Y dans (8.15) par :

$$F \triangleq (f(X_1), \ldots, f(X_N))^T \tag{8.18}$$

et modifions la contrainte (8.16) par un scalaire obtenu par sommation des N lignes de (8.16). Nous obtenons ainsi le problème dual :

$$J_{MD}^* \triangleq \max_{\lambda} F^T \lambda \tag{8.19}$$

$$\text{subject to}$$

$$\mathbf{1}^T A\lambda \leq N \tag{8.20}$$

$$\lambda \geq 0. \tag{8.21}$$

Comme $F \geq Y$ et en utilisant le théorème Dual (voir e.g. HIRIART-URRUTY & LEMARÉCHAL [74], chapitre 7) :

$$J_P^* = J_D^* \leq J_{MD}^*. \tag{8.22}$$

Nous pouvons maintenant obtenir une borne supérieure pour J_{MD}^* .

2. Fixons un vecteur arbitraire λ qui obéit aux contraintes (8.20), (8.21) et écrivons l'inégalité (8.20) de manière équivalente sous la forme :

$$\frac{1}{N}\sum_{j=1}^{N}\lambda_j\left(K_h(0)+\sum_{i\neq j}^{N}K_h(X_i-X_j)\right)\leq 1, \tag{8.23}$$

ce qui revient à :

$$\frac{1}{N}\sum_{j=1}^{N}\lambda_j\left(\frac{1}{h}K(0)+\sum_{i\neq j}^{N}E\left\{K_h(X_i-X_j)\mid X_j\right\}+\sum_{i\neq j}^{N}\xi_{ij}\right)\leq 1, \tag{8.24}$$

avec :

$$\xi_{ij}\triangleq K_h(X_i-X_j)-E\left\{K_h(X_i-X_j)\mid X_j\right\}.$$

Appliquons maintenant la borne supérieure (8.85), obtenue par le Lemme 7 à l'inégalité (8.24), en prenant en compte le fait que $K(0)>0$ et :

$$\begin{aligned} E\left\{K_h(X_i-X_j)\mid X_j\right\} &= \frac{1}{h}\int_0^1 K\left(\frac{u-X_j}{h}\right)\frac{f(u)}{C_f}\,du && (8.25)\\ &= \frac{1}{C_f}\int_{\mathbb{R}}K(t)f(X_j+ht)\,dt \\ &= \frac{1}{C_f}\left(f(X_j)+O(h)\right), && (8.26)\end{aligned}$$

avec un $O(h)$ non aléatoire. Ainsi :

$$\frac{N-1}{C_f N}\sum_{j=1}^{N}\lambda_j\left(f(X_j)+O(h)-C\sqrt{\frac{\log N}{Nh}}\right)\leq 1,\qquad \forall\, N\geq N_2(\omega), \tag{8.27}$$

où C est une constante donnée. En premier lieu, l'inégalité (8.27) implique que :

$$\sum_{j=1}^{N}\lambda_j\leq\frac{2C_f}{f_{\min}}<\infty,\qquad \forall\, N\geq N_3(\omega), \tag{8.28}$$

avec $N_3(\omega)\geq N_2(\omega)$ presque sûrement fini. En second lieu, (8.28) et (8.27) impliquent la borne supérieure (8.14) et le lemme 3 est prouvé. □

Borne inférieure pour $\widehat{f}_N$

Lemma 4 *D'après les hypothèses du Theoréme 2, pour presque tout $\omega \in \Omega$ on peut trouver un entier $N_1(\omega)$ tel que pour tout $x \in (0,1)$*

$$\widehat{f}_N(x) \geq f(x) - O\left(\sqrt{\log N/(Nh^2)}\right), \qquad \forall\, N \geq N_1(\omega), \tag{8.29}$$

om $O(\cdot)$ ne dépend pas de x.

Preuve du Lemme 4. 1. Supposons que pour une constante $\delta_x > 0$ donnée il existe (avec une probabilité 1) un entier $i_k \in \{1, \ldots, N\}$ tel que :

$$|x - X_{i_k}| \leq \delta_x. \tag{8.30}$$

Ainsi l'erreur d'estimation au point $x \in (0,1)$ peut être decomposée de la manière suivante :

$$\begin{aligned} f(x) - \widehat{f}_N(x) &= [f(x) - f(X_{i_k})] && (8.31)\\ &+ \left[f(X_{i_k}) - \widehat{f}_N(X_{i_k})\right] && (8.32)\\ &+ \left[\widehat{f}_N(X_{i_k}) - \widehat{f}_N(x)\right]. && (8.33) \end{aligned}$$

Le terme de droite (8.31) peut être borné :

$$|f(x) - f(X_{i_k})| \leq L_f\, |x - X_{i_k}| \leq L_f \delta_x, \tag{8.34}$$

tout comme le terme (8.33) :

$$\left|\widehat{f}_N(X_{i_k}) - \widehat{f}_N(x)\right| \leq L_{\widehat{f}_N}\, |x - X_{i_k}| \leq L_{\widehat{f}_N}\, \delta_x, \tag{8.35}$$

où $L_{\widehat{f}_N}$ est la constante Lipschitz de la fonction $\widehat{f}_N(x)$. Nous cherchons maintenant à borner cette dernière. Pour borner (8.32), supposons qu'une certaine valeur constante $\delta_y > 0$ vérifie :

$$Y_{i_k} \geq f(X_{i_k}) - \delta_y \text{ a.s.} \tag{8.36}$$

Rappelons que $\widehat{f}_N(X_{i_k}) \geq Y_{i_k}$ à cause de (8.4). Ainsi,

$$f(X_{i_k}) - \widehat{f}_N(X_{i_k}) \leq (Y_{i_k} + \delta_y) - Y_{i_k} = \delta_y. \tag{8.37}$$

En combinant ces bornes nous obtenons à partir de (8.31) que pour tout $N \geq N_0(\omega)$,

$$f(x) - \widehat{f}_N(x) \leq \delta_y + \left(L_f + L_{\widehat{f}_N}\right)\delta_x. \tag{8.38}$$

2. Notons qu'un estimateur de la constante de Lipschitz de l'estimateur est tout simplement :

$$\begin{aligned} |\widehat{f}_N(u) - \widehat{f}_N(v)| &\leq \sum_{i=1}^{N} \alpha_i \, |K_h(u - X_i) - K_h(v - X_i)| & (8.39)\\ &\leq \frac{L_K}{h^2}\left(\sum_{i=1}^{N}\alpha_i\right)|u - v|. & (8.40) \end{aligned}$$

Ainsi, à partir de (8.14), nous obtenons presque sûrement :

$$L_{\widehat{f}_N} = \frac{L_K}{h^2}\, C_f(1 + o(1)), \qquad \forall\, N \geq N_0(\omega), \tag{8.41}$$

avec $N_0(\omega)$ presque sûrement fini.

3. Maintenant, nous pouvons démontrer qu'une définition appropriée de δ_x et δ_y comme fonctions de h et N, il existe un entier aléatoire $N_0(\omega)$ presque sûrement fini tel que :

$$\forall\, N \geq N_0(\omega), \qquad \exists\, i_k \in \{1, \ldots, N \,:\, (X_{i_k}, Y_{i_k}) \in \Delta(x)\}, \tag{8.42}$$

avec :

$$\Delta(x) \triangleq \{(u, v) \,:\, |x - u| \leq \delta_x,\ f(x) - \delta_y \leq v \leq f(u)\}. \tag{8.43}$$

En effet, en introduisant :

$$\delta_y \triangleq \left(\frac{q \log N}{N h^2}\right)^{1/2}, \tag{8.44}$$

et :

$$\delta_x = h^2 \delta_y. \tag{8.45}$$

Ainsi,

$$\begin{aligned} P\{(X_i, Y_i) \notin \Delta(x) \quad \forall\, i = 1, \ldots, N\} &= \left(1 - \frac{1 + o(1)}{C_f}\delta_x \delta_y\right)^N \\ &= \left(1 - \frac{1 + o(1)}{C_f} h^2 \delta_y^2\right)^N \\ &\leq \exp\left\{-\frac{1 + o(1)}{C_f} N h^2 \delta_y^2\right\} \\ &\leq N^{-q/(2C_f)}. & (8.46) \end{aligned}$$

Finalement, en fixant :

$$q > 2C_f \tag{8.47}$$

on obtient la convergence de la série :

$$\sum_{N=1}^{\infty} P\{(X_i, Y_i) \notin \Delta(x) \quad \forall\, i = 1, \ldots, N\} < \infty, \tag{8.48}$$

qui, par le lemme de Borel–Cantelly, implique l'existence d'un entier presque sûrement fini $N_0(\omega)$ tel que la relation (8.42) soit vraie.

4. En substituant les relations (8.41), (8.44), et (8.45) dans (8.38), on obtient la borne inférieure souhaitée :

$$\begin{aligned} \widehat{f}_N(x) &\geq f(x) - \delta_y - O\left(h^{-2}\right)\delta_x \\ &= f(x) - O\left(\sqrt{\frac{\log N}{Nh^2}}\right), \end{aligned} \tag{8.49}$$

où $O(\cdot)$ est un terme non aléatoire indépendant de x. □

Preuve du Théorème 2

1. Comme $|u| = u - 2u\mathbf{1}\{u < 0\}$, la norme L_1 de l'erreur d'estimation peut être décomposée de la manière suivante :

$$\|\hat{f}_N - f\|_1 = \int_0^1 \left[\hat{f}_N(x) - f(x)\right] dx \tag{8.50}$$

$$+2\int_0^1 \left[f(x) - \hat{f}_N(x)\right] \mathbf{1}\left\{\hat{f}_N(x) < f(x)\right\} dx. \tag{8.51}$$

2. L'application du Lemme 3 au terme de droite (8.50) donne :

$$\limsup_{N\to\infty} \varepsilon_{UB}^{-1}(N) \left(\int_0^1 \left[\hat{f}_N(x) - f(x)\right] dx\right) \leq \text{const} < \infty \quad \text{a.s.} \tag{8.52}$$

avec :

$$\varepsilon_{UB}(N) \triangleq \max\left\{h, \sqrt{\log N/(Nh)}\right\}. \tag{8.53}$$

3. Pour obtenir un résultat similaire avec le terme (8.51), il faut remarquer que le Lemme 4 implique :

$$\zeta_N(x) \triangleq \varepsilon_{LB}^{-1}(N)\left[f(x) - \hat{f}_N(x)\right] \leq C(\omega) < \infty \quad \text{a.s.}$$

uniformément par rapport à x et N, avec :

$$\varepsilon_{LB}(N) \triangleq \sqrt{\log N/(Nh^2)}. \tag{8.54}$$

Ainsi, on peut appliquer le lemme de Fatou, en utilisant le fait que $u\mathbf{1}\{u > 0\}$ est une fonction continue et monotone :

$$\limsup_{N\to\infty} \varepsilon_{LB}^{-1}(N) \int_0^1 \left[f(x) - \hat{f}_N(x)\right] \mathbf{1}\left\{\hat{f}_N(x) < f(x)\right\} dx \tag{8.55}$$

$$\leq \int_0^1 \limsup_{N\to\infty} \zeta_N(x)\, \mathbf{1}\{\zeta_N(x) > 0\}\, dx \tag{8.56}$$

$$\leq C(\omega) < \infty \quad \text{a.s.} \tag{8.57}$$

4. Ainsi, les relations obtenues associées à (8.50) et (8.51) prouvent le théorème 2. □

La preuve du Théorème 3 (page 206) est basée sur une démonstration très similaire à la précédente.

Borne supérieure de $\tilde{f}_N$

Puisque l'ensemble admissible (8.11), (8.12) est plus petit par rapport à (8.4), (8.5), ils est important de démontrer que la borne supérieure reste au moins équivalente à la précédente.

Lemma 5 *Supposons que les hypothèses du théorème 3 sont vérifiées. Alors, pour presque tout $\omega \in \Omega$ il existe un nombre fini $N_0(\omega)$ tel que*

$$J_{MP}^* \leq C_f + O(h) + O\left(\sqrt{\frac{\log N}{Nh}}\right), \qquad \forall\, N \geq N_0(\omega), \tag{8.58}$$

ou $O(h)$ et $O\left(\sqrt{\log N/(Nh)}\right)$. sont des quantités non aléatoires.

Preuve du Lemme 5. 1. Comme la fonction noyau $K(t)$ est supposée paire, la matrice A est symétrique et d'après (8.10)–(8.12), le problème dual s'écrit sous la forme :

$$J_{MD}^* \triangleq \max_{\lambda,\,\nu} \left(Y^T\lambda - C_\alpha N^{-1}\mathbf{1}^T\nu\right) \tag{8.59}$$

subject to

$$A\lambda - \nu \leq \mathbf{1} \tag{8.60}$$

$$\lambda \geq 0 \tag{8.61}$$

$$\nu \geq 0. \tag{8.62}$$

Remplaçons le vecteur Y dans (8.59) par

$$F \triangleq (f(X_1), \ldots, f(X_N))^T. \tag{8.63}$$

Remplaçons aussi la contrainte (8.60) par un scalaire, obtenu en sommant les N lignes de (8.60). Nous obtenons le problème dual modifié :

$$J^*_{MMD} \triangleq \max_{\lambda,\,\nu} \left(F^T\lambda - C_\alpha N^{-1}\mathbf{1}^T\nu\right) \tag{8.64}$$

subject to

$$\mathbf{1}^T A\lambda - \mathbf{1}^T\nu \leq N \tag{8.65}$$

$$\lambda \geq 0 \tag{8.66}$$

$$\nu \geq 0. \tag{8.67}$$

Comme $F \geq Y$ et en utilisant le théorème Dual,

$$J^*_{MP} = J^*_{MD} \leq J^*_{MMD}. \tag{8.68}$$

Ensuite, nous obtenons une borne supérieure pour J^*_{MMD}.

2. Fixons arbitrairement (λ, ν) satisfiant les contraintes (8.65)–(8.67) et écrivons l'inégalité (8.65) sous la forme :

$$\frac{1}{N}\sum_{j=1}^{N}\lambda_j\left(K_h(0) + \sum_{i\neq j}^{N} K_h(X_i - X_j)\right) \leq 1 + \frac{1}{N}\mathbf{1}^T\nu, \tag{8.69}$$

ou, de manière équivalente,

$$\frac{1}{N}\sum_{j=1}^{N}\lambda_j\left(\frac{1}{h}K(0) + \sum_{i\neq j}^{N} E\left\{K_h(X_i - X_j) \mid X_j\right\} + \sum_{i\neq j}^{N}\xi_{ij}\right) \leq 1 + \frac{1}{N}\mathbf{1}^T\nu, \tag{8.70}$$

avec :

$$\xi_{ij} \triangleq K_h(X_i - X_j) - E\left\{K_h(X_i - X_j) \mid X_j\right\}. \tag{8.71}$$

Appliquons maintenant la borne supérieure (8.85), obtenue dans le lemme 7, à l'inégalité (8.70), prenant en compte le fait que $K(0) > 0$ et (8.25)–(8.26). Ainsi,

$$\frac{N-1}{C_f N}\sum_{j=1}^{N}\lambda_j\left(f(X_j) + O(h) - C\sqrt{\frac{\log N}{Nh}}\right) \leq 1 + \frac{1}{N}\mathbf{1}^T\nu, \qquad \forall\, N \geq N_2(\omega), \tag{8.72}$$

où C est une constante donnée. Tout d'abord, l'inégalité (8.72) permet d'obtenir :

$$\sum_{j=1}^{N} \lambda_j \leq \frac{C_f}{f_{\min}} \left(2 + \frac{1}{N} \mathbf{1}^T \nu\right), \qquad \forall\, N \geq N_3(\omega), \tag{8.73}$$

avec $N_3(\omega) \geq N_2(\omega)$ un entier presque sûrement fini. Ainsi, en utilisant (8.72), pour presque tout $\omega \in \Omega$ et un N suffisament grand,

$$\begin{aligned} F^T \lambda - \frac{C_\alpha}{N} \mathbf{1}^T \nu &\leq C_f \left(1 + O(h) + O\left(\sqrt{\frac{\log N}{Nh}}\right)\right) && (8.74) \\ &\quad -(C_\alpha - C_f(1 + o(1)))\mathbf{1}^T \nu, && (8.75) \end{aligned}$$

avec $O\left(\sqrt{\log N/(Nh)}\right)$ non aléatoire. Ainsi, (8.68) et (8.74) prouvent la borne supérieur (8.58), puisque (8.12) implique $C_\alpha > C_f$. □

Borne inférieure pour $\tilde{f}_N$

Lemma 6 *Sous les hypothèses du Théorème 3, pour presque tout $\omega \in \Omega$ il existe un nombre fini $N_1(\omega)$ tel que pout tout $x \in (0,1)$*

$$\tilde{f}_N(x) \geq f(x) - O\left(\sqrt{\log N/(Nh)}\right), \qquad \forall\, N \geq N_1(\omega), \tag{8.76}$$

où $O(\cdot)$ ne dépend pas de x.

La **preuve du Lemme 6** est donnée de la même manière que pour le Lemme 4. La différence essentielle est une meilleure constante de Lipschitz pour $\tilde{f}_N(x)$. En effet, pour tout $u, v \in (0,1)$:

$$\begin{aligned} \left|\tilde{f}_N(u) - \tilde{f}_N(v)\right| &\leq \sum_{i=1}^{N} \alpha_i \left|K_h(u - X_i) - K_h(v - X_i)\right| && (8.77) \\ &\leq \frac{L_K}{h^2} \left(\sum_{i \in I(u)} \alpha_i + \sum_{i \in I(v)} \alpha_i\right) |u - v|, && (8.78) \end{aligned}$$

avec :

$$I(\cdot) \triangleq \{i \mid K_h(\cdot - X_i) \neq 0\}. \tag{8.79}$$

D'après la loi forte des grands nombres,

$$\operatorname{Card} I(\cdot) = \frac{f(\cdot)}{C_f} Nh(1 + o(1)) \quad \text{a.s.} \tag{8.80}$$

et ainsi,

$$L_{\tilde{f}_N} = \frac{L_K}{h^2}\,\frac{C_\alpha}{N}\,\frac{2f_{\max}}{C_f}\,Nh = O\left(\frac{1}{h}\right) \tag{8.81}$$

par le borne supérieure (8.12) de α. □

Preuve du Théorème 3

Le Théorème 3 est prouvé de la même manière que le Théorème 2, en se basant sur les lemmes 3 et 4. Notons que la borne inférieure du Lemme 4 n'est pas pire que celle la borne supérieure, grâce à la modification introduite dans l'estimateur.

Note : Le résultatt du Théorème 3 pourrait aussi être prouvé pour des noyaux dérivables sur un support infini qui satisfait à l'hypothèse suivante :

$$|K'(t)| \leq \mu K(t), \quad \forall\, t \in \mathbb{R}, \tag{8.82}$$

où μ est une constante donnée. En effet, (8.82) implique

$$\left|\tilde{f}'_N(x)\right| \leq \frac{1}{h^2}\sum_{i=1}^{N}\alpha_i\left|K'\left(\frac{x-X_i}{h}\right)\right| \leq \frac{\mu}{h}\tilde{f}_N(x). \tag{8.83}$$

En conséquence, l'estimation de la fonction $\tilde{f}_N(x)$ est bornée par valeur supérieure, sa constante de Lipschitz est de l'ordre de $O\left(h^{-1}\right)$, c'est-à-dire identique à celle de (8.81).

Preuve du Lemme 7

Lemma 7 *Supposons que les hypothèses A et B sont vraies et que la constante C est suffisament grande. Définissons les variables aléatoires :*

$$\xi_{ij} \triangleq K_h(X_i - X_j) - E\left\{K_h(X_i - X_j) \mid X_j\right\}, \quad i \neq j. \tag{8.84}$$

Alors, pour presque tout $\omega \in \Omega$ il existe un entier fini$N_2(\omega)$ tel que :

$$\max_{j=1,\ldots,N}\left|\frac{1}{N-1}\sum_{i\neq j}^{N}\xi_{ij}\right| \leq C\sqrt{\log N/(Nh)} \qquad \forall\, N \geq N_2(\omega). \tag{8.85}$$

Preuve du Lemme 7. Notons que pour tout $j = 1, \ldots, N$ les variables i.i.d. non biaisées $(\xi_{ij})\,|_{i \neq j}$ ont les propriétés suivantes :

$$|\xi_{ij}| \leq \frac{2}{h} K_{\max} \triangleq a, \tag{8.86}$$

et

$$\begin{aligned} E\left\{\xi_{ij}^2 \mid X_j\right\} &\leq \frac{1}{h^2 C_f} \int_0^1 K^2\left(\frac{u - X_j}{h}\right) f(u)\, du \\ &\leq \frac{1}{h C_f} \int_{\mathbb{R}} K^2(t)\, f(X_j + ht)\, dt \\ &\leq \frac{C_0(K)}{h C_f} f_{\max} \triangleq \sigma_1^2. \end{aligned} \tag{8.87}$$

Ainsi, on peut appliquer l'inégalité de Bernstein (voir e.g., BIRGÉ & MASSART [18] ou BOSQ [19], Théorème 2.6) donne :

$$P\left\{\left|\frac{1}{N-1}\sum_{i \neq j}^{N} \xi_{ij}\right| > \mu \quad \middle| \, X_j \right\} \leq 2\exp\left(-\frac{(N-1)\mu^2}{2(\sigma_1^2 + a\mu/3)}\right).$$

Définissons :

$$\mu = \sqrt{\frac{q \log N}{Nh}}, \tag{8.88}$$

où q est défini plus loin et est supposé suffisament grand. Ainsi, pour tout $N \geq N_1$, N_1 étant une constante entière suffisament grande,

$$P\left\{\left|\frac{1}{N-1}\sum_{i \neq j}^{N} \xi_{ij}\right| > \sqrt{\frac{q \log N}{Nh}} \quad \middle| \, X_j \right\} \leq 2N^{-q_1},$$

avec :

$$q_1 \triangleq \frac{q\, C_f f_{\max}}{3 C_0(K)}. \tag{8.89}$$

Ainsi,

$$P\left\{\max_{j=1,N}\left|\frac{1}{N-1}\sum_{i \neq j}^{N} \xi_{ij}\right| > \sqrt{\frac{q \log N}{Nh}} \quad \middle| \, X_j \right\} \tag{8.90}$$

$$\leq \sum_{j=1}^{N} P\left\{\left|\frac{1}{N-1}\sum_{i \neq j}^{N} \xi_{ij}\right| > \sqrt{\frac{q \log N}{Nh}} \quad \middle| \, X_j \right\} \tag{8.91}$$

$$\leq 2N^{1-q_1}. \tag{8.92}$$

En conséquence, pour toute toute valeur du paramètre, on a :

$$q > \frac{6 C_0(K)}{C_f f_{\max}} \tag{8.93}$$

implique $q_1 > 2$, prouvant la convergence de la série $\sum^{\infty} N^{1-q_1}$ et, grâce au lemme de Borel–Cantelli, le résultat souhaité (8.85) est obtenu. □

Chapitre 9

Conclusion

Nous récapitulons les principaux résultats et contributions propres à ce travail de thèse, tout en proposant des directions de recherche futures.

9.1 Synthèse des travaux

9.1.1 Approche générative

Nous avons considéré l'approche générative comme une démarche de modélisation plutôt qu'une méthode d'apprentissage « clé en main ». La définition du classifieur génératif nécessite un travail de modélisation de la densité jointe, alors qu'une fois le modèle appris, seule la loi conditionnelle est nécessaire pour effectuer la prédiction sur des données de test. Le parti pris de modéliser « plus que nécessaire » se révèle être avantageux dans plusieurs applications où la structure des données est non triviale.

Les modèles génératifs ont longtemps été considérés comme sous-optimaux par la communauté du Machine Learning, au profit de méthodes purement discriminatives telles que les réseaux de neurones et les SVM. Cela s'explique en partie par la recherche de méthodes « clés en mains » pour lesquelles des données d'apprentissage sont fournies en grand nombre pour en déduire une règle de classification sans intervention humaine. Ce point de vue idéaliste est valide sur des données de faible dimension, mais dès que la tâche à résoudre se complexifie, la quantité de données nécessaire à capturer de manière fiable les règles de décision devient astronomique.

La qualité des résultats d'une méthode d'apprentissage dépend de celui qui l'utilise à travers

1. le choix judicieux des variables explicatives,
2. l'incorporation d'information additionnelle (différente des données d'apprentissage).

Le premier point est évident car il est naturel de choisir des variables en accord avec le problème à traiter. C'est sur le deuxième point que l'utilité des modèles génératifs se révèle. En effet, lorsque la taille de l'échantillon d'apprentissage est petite relativement à la complexité du problème à traiter (c'est souvent le cas en pratique), les performances de toute méthode d'apprentissage sont limitées. Ces bornes, liées à la complexité de la règle de décision recherchée, sont des limites théoriques infranchissables. Elles se retrouvent sous différentes formes dans plusieurs domaines (borne de Shanon en thorie de l'information [28], borne de Cramer-Rao en statistiques [99], bornes de Vapnik en théorie de l'apprentissage [158]). Ainsi, la seule chance de pouvoir améliorer une méthode d'apprentissage est d'ajouter des informations liées au contexte. Dans les méthodes génératives, le modèle de distribution des entrées revient à faire des hypothèses supplémentaires par rapport à une approche discriminative, et peut être vu comme un ajout d'information. Le simple fait de donner une *structure* aux covariables restreint le domaine de recherche des solutions, ce qui peut mener à une réduction importante du taux d'erreur. Différents formalismes de modélisation des distributions sont possibles, tels que les modèles linéaires gaussiens, les modèles graphiques ou les copules.

Un moyen d'inclure des informations additionnelles est d'utiliser des données non labellisées dans l'apprentissage [11]. Cela ne peut évidemment avoir de sens que si on fournit un modèle pour la densité des covariables, illustrant une fois de plus l'intérêt de l'approche générative.

Aujourd'hui, quelques problèmes d'apprentissage statistique réputés difficiles sont résolus grâce à des approches génératives. En robotique, on peut construire automatiquement la carte des alentours en utilisant les *processus de décision à base de modèles de Markov partiellement observables* (POMDP) dont l'équivalence avec des réseaux bayésiens dynamiques a été demontrée[125], ce qui permet d'estimer les paramètres en un temps linéaire avec la durée d'observation. En vision par ordinateur, des modèles probabilistes estimés par des méthodes de Monte Carlo séquentielles [39, 12] permettent d'estimer la distribution *a posteriori* de la pose d'une personne à partir d'images numériques [150]. Dans cette thèse, le modèle hiérarchique des parties proposées a une distribution qui se factorise sous la forme d'un modèle graphique, et c'est celui-ci qui a permis d'effectuer une estimation

cohérente des paramètres.

De nombreux modèles génératifs incluent des données inobservées ou latentes afin de modéliser au mieux les distributions. Nous avons montré l'efficacité et l'intérêt des modèles à classe latente pour la discrimination et la régression.

Afin d'optimiser les performances des approches génératives, nous avons introduit BEC, un critère de sélection de modèle, ainsi qu'un nouveau type d'estimateur, GDT. Les méthodes de classification basées sur des modèles génératifs restent en général cantonnées à une estimation de la densité jointe des données ou de leur distribution conditionnelle. Entre ces deux extrêmes, l'estimateur GDT que nous avons proposé équilibre le biais et la variance du taux d'erreur. Nous avons effectué quelques expériences sur des classifieurs linéaires, qui montrent un intérêt potentiel si la distribution des données est gaussienne.

9.1.2 Les modèles discriminatifs

L'estimation de frontière a illustré la souplesse des modèles conditionnels ou discriminatifs. Cette approche permet d'estimer une fonction avec un minimum d'hypothèses sur la forme de la distribution des entrées. L'estimateur que nous avons proposé partage plusieurs points communs avec les méthodes de type SVM : la frontière estimée est un estimateur à noyaux et la solution est compacte (*i.e.* la fonction estimée ne depend que d'un petit nombre de données d'apprentissage), ce qui est avantageux en terme de mémoire et de rapidité de prédiction. Ce travail renforce l'idée couramment admise que les modèles non paramétriques, grâce à leur qualité d'approximateur universel, sont adaptés aux approches discriminatives.

9.1.3 Fonctions de coût

Tout au long de la thèse, nous avons pu observer l'importance des fonctions de coût, ou contrastes, que ce soit pour l'apprentissage, la sélection de modèle, ou la validation des performances. Ces fonctions de coût sont résumées dans le Tableau 9.1.

Dans la cadre des méthodes génératives, nous avons introduit le critère BEC pour sélectionner d'un classifieur et la fonction GDT pour estimer les paramètres. Pour l'estimation de frontière, la fonction objectif a une interprétation plus géométrique que probabiliste puisqu'elle correspond à la surface située sous la courbe estimée.

	Fonction objectif		
Chapitre	apprentissage	selection de modèle	validation des performances
2	gen/disc	BIC/CV	taux d'erreur
3	gen/disc	BIC/AIC	erreur quadratique
4	gen	BEC	taux d'erreur
5	gen/GDT/disc	CV	taux d'erreur
6	gen	CV	taux d'erreur
7	–	–	interprétation des experts
8	aire sous la courbe	spécifique au modèle	erreur absolue

TAB. 9.1 – Fonctions de coût considérées dans les différents chapitres de la thèse.

Ces exemples montrent que le choix de la fonction de coût est complémentaire de la modélisation, et peut avoir une influence importante sur les performances d'une méthode d'apprentissage statistique. D'un point de vue général, pour déterminer cette fonction, il semble nécessaire de prendre en compte

– les aspects théoriques (unicité du minimum, comportement asymptotique de la solution),

– l'efficacité (réponse aux objectifs de l'utilisateur, validation expérimentale),

– la simplicité (facilité de calcul, algorithmes de minimisation efficaces).

Ces deux premiers points correspondent aux motivations principales du critère BEC et de l'estimation GDT que nous avons proposées, mais dans les application réelles, c'est la simplicité qui est généralement privilégiée : l'algorithme EM est très efficace pour estimer le maximum de vraisemblance des modèles à structure cachée, et l'échantillonnage de Gibbs est particulièrement adapté à la modélisation de systèmes complexes. Les estimations nécessitant une descente de gradient restent difficiles à utiliser pour des modèles probabilistes incluant plusieurs centaines de paramètres.

On peut aussi remarquer que nous n'avons pas défini de fonction objectif pour le modèle bayésien du chapitre 7, puisque l'obtention de la loi *a posteriori* se fait par simulation[33].

[33]Certaines méthodes bayésiennes minimisent une fonction de contraste, comme la recherche du mode *a posteriori* (MAP) ou l'estimation par méthode variationnelle. Cette dernière correspond à la minimisation d'un contraste entre la fonction *a posteriori* et une famille de lois paramétriques.

9.1.4 Vision par ordinateur

Nous avons proposé un nouveau modèle probabiliste qui modélise la distribution des points d'intérêts. En considérant qu'un objet est composé d'une hiérarchique de parties rigides, le modèle caractérise de manière intéressante une catégorie d'image, en séparant les variabilités dues à la détection de celles dues à la forme. Il donne des résultats de classification relativement performants par rapport aux approches existantes, mais de nombreux travaux restent à faire. En particulier, prendre en compte plus de déformations et plus de niveaux hierarchiques devrait permetre d'obtenir des modèles réalistes pour un grand nombre de catégories d'images : lorsque le nombre de parties et de sous-parties grandit, la déformation devient puremenent locale. Cela tend vers des modèles complètement déformables avec un niveau de rigidité donné.

9.2 Perspectives

9.2.1 Court terme

Amélioration des critères de validation et de sélection de modèle Les méthodes proposées dans les chapitres 4 et 5 nécessitent des calculs lourds qui devraient pouvoir se simplifier.

Le critère BEC, tel qu'il est proposé pour sélectionner un classifieur nécessite l'estimation du maximum de vraisemblance de la distribution marginale des données. Cette estimation d'un mélange de distribution n'est pas toujours satisfaisante, dans la mesure où elle correspond à un apprentissage supplémentaire. Un critère plus simple ou nécessitant moins de calcul serait le bienvenu. En outre, l'extension du critère BEC à l'estimation discriminative ou GDT devrait être envisagée, car ces méthodes se cantonnent à la validation croisée pour sélectionner un modèle.

De même, le problème théorique majeur de l'estimation GDT est le choix du paramètre λ de pondération génératif/discriminatif. La validation croisée est particulièrement lourde dans ce cas, alors que les premiers résultats théoriques laissent présager des possibilité d'évaluation du λ optimal. Le potentiel de l'estimation GDT apparaît lorsque le nombre de données est de l'ordre du nombre de paramètres. Il serait donc intéressant d'obtenir des résultats n'utilisant pas le théorème central limite, contrairement à ce qui est proposé dans cette thèse. Le développement récent des bornes sur le risque empirique pourrait permettre d'illustrer le phénomène de régularisation qui caractérise GDT et qui permet d'améliorer les performances en classification.

9.2.2 Moyen terme

Un intermédiaire entre génératif paramétrique et discriminatif non paramétrique Les estimations discriminatives et GDT ont montré des améliorations potentielles des taux de classification, mais ces gains restent limités par la structure « rigide » propre aux modèles paramétriques. En réalité, on peut se demander si les bénéfices liés à la modification de la fonction de coût sont importants. L'introduction d'une partie non paramétrique qui modélisent des données « sans modèle » pourrait être plus judicieux.

Par exemple, il peut arriver que la distribution paramétrique soit adaptée aux données sauf pour quelques valeurs des données d'entrées dont le comportement est anormal vis à vis du modèle. Comme on ne dispose pas d'information structurelle pour ces données « hors-modèle », une estimation non paramétrique de la frontière de décision permettrait de corriger localement le biais de la frontière paramétrique. Nous décrivons brièvement un modèle de ce type.

Notons que la fonction de discrimination appliquée à une donnée d'entrée x peut être la même pour les approches génératives paramétriques et discriminatives non paramétriques :

$$\boldsymbol{p}(Y=1|X=x;\theta)=\frac{1}{1+e^{-\delta_{GEN}(x;\theta)}} \quad \text{avec} \quad \delta_{GEN}(x;\theta)=\log\frac{\pi f_1(x;\theta)}{(1-\pi)f_{-1}(x;\theta)} \tag{9.1}$$

pour un modèle génératif modélisant les densités des classes par f_1 et f_{-1} et et

$$\boldsymbol{p}(Y=1|X=x;\alpha)=\frac{1}{1+e^{-\delta_{DISC}(x;\alpha)}} \quad \text{avec} \quad \delta_{DISC}(x;\alpha)=\sum_{j=1}^{n}\alpha_j K(x,x_j) \tag{9.2}$$

pour un modèle logistique à noyau, aujourd'hui très répandu. Ainsi, il semble possible de fusionner ces deux fonctions dans un modèle que l'on peut qualifier de *Generative-Discriminative Tradeoff* semi-paramétrique :

$$\boldsymbol{p}(Y=1|X=x;\theta,\alpha)=\frac{1}{1+e^{-\delta_{GEN}(x;\theta)-\delta_{DISC}(x;\alpha)}} \tag{9.3}$$

où $\delta_{GEN}(x;\theta)$ est la fonction de discrimination paramétrique définie dans (9.1) et δ_{DISC} la fonction non paramétrique définie dans (9.2). Nous proposons deux manières pour estimer les paramètres :

- soit par une optimisation en deux étapes : θ est estimé par maximisation de la vraisemblance, puis α_i par maximisation de la vraisemblance conditionnelle pénalisée,
- soit par une maximisation directe de la vraisemblance pénalisée :

$$(\hat{\theta},\hat{\alpha})=\underset{\alpha,\theta}{\operatorname{argmax}}\sum_{i=1}^{n}\log\boldsymbol{p}(Y=y_i|X=x_i;\theta,\alpha)-\|\alpha\|^2.$$

Ce modèle relativement simple devrait permettre d'associer la stabilité du modèle paramétrique à la souplesse de l'estimation discriminative. Les aspects de parcimonie (nombreux α nuls) peuvent être obtenus de plusieurs manières, soit par des techniques d'optimisation [145], soit en ajoutant un *a priori* sur les paramètres α_i favorisant les valeurs nulles, dans l'esprit des méthodes d'*Automatic Relevance Determination* [126, 153, 131]. Cette approche sera étudiée dans des travaux ultérieurs.

9.2.3 Long terme

Le développement des méthodes d'apprentissage Les méthodes d'apprentissage statistique vont naturellement s'orienter vers des problèmes de plus en plus complexes sur des données de grande taille. Cependant, le fait que le volume des données augmente ne veut pas dire que leur nombre va augmenter, et la création de programmes intelligents dépendra de leur faculté à « comprendre » (*i.e.* construire des règles) leur environnement plutôt que de « reconnaître » (reproduire des événements passés). De cette manière, la reconnaissance des formes évoluera vers la *compréhension des formes*, on ne verra plus un « losange » mais « quatre segments de même longueur ». Jusqu'à présent, la théorie de l'apprentissage a répondu à la question difficile :

« Comment généraliser un phénomène ? »

mais nous aimerions répondre à la question :

« Comment modéliser un phénomène ? »

En effet, lorsqu'un problème d'apprentissage doit être résolu, la modélisation est en général une partie assez délicate, souvent négligée au profit de méthodes plus générales (qui naturellement nécessitent plus de données d'apprentissage). Ainsi, dans les années à venir, il serait utile de définir un formalisme de description des modèles probabilistes qui répond à trois crières :

– décrit la distribution des données « réelles »,

– peut être compris facilement par un humain,

– peut être estimé de manière probabiliste.

Les modèles graphiques permettent de modéliser les distributions de données réelles de manière très précise, sont faciles à comprendre (car leur représentation visuelle est assez intuitive) et de nombreuses méthodes existent pour en estimer les paramètres. Ils répondent donc à tous ces critères. Cependant, le formalisme ne définit pas

la forme des distributions conditionnelles, celles-ci étant généralement limitées à des multinomiales, gaussiennes et mélanges de gaussiennes. De plus, de nombreuses distributions restent réfractaires à un apprentissage rapide. Plusieurs travaux récents (filtres à particules, apprentissage variationnel) laissent présager d'importantes avancées dans ce domaine.

Bibliographie

[1] W. C. A. Charnes and E. Rhodes. Measuring the inefficiency of decision making unit. *European Journal of Operational Research*, 2 :429–444, 1978.

[2] B. P. A. Kneip and L. Simar. A note on the convergence of nonparametric dea estimators for production efficiency scores. *Econometric Theory*, 14 :783–793, 1998.

[3] H. Abbar. Un estimateur spline du contour d'une répartition ponctuelle aléatoire. *Statistique et analyse des données*, 15(3) :1–19, 1990.

[4] A. Agarwal and B. Triggs. 3d human pose from silhouettes by relevance vector regression. In *International Conference on Computer Vision & Pattern Recognition*, pages II 882–888, Washington, June 2004.

[5] A. Agarwal and B. Triggs. Learning methods for recoverng 3d human pose from monocular images. Research Report 5333, INRIA Rhone Alpes, 655 ZIRST, Avenue de l'Europe, 38330 Montbonnot, France, October 2004.

[6] S. Agarwal and D. Roth. Learning a sparse representation for object detection. In *Proceedings of the 7th European Conference on Computer Vision, Copenhagen, Denmark*, pages 113–128, 2002.

[7] A. Agresti. *An Introduction to Categorical Data Analysis*. John Wiley and Sons Inc., 1996.

[8] H. Akaike. A new look at statistical model identification. *IEEE Transactions on Automatic Control*, 19 :716–723, 1974.

[9] Y. Altun and T. Hofmann. Large margin methods for label sequence learning. In *8th European Conference on Speech Communication and Technology (EuroSpeech)*, 2003.

[10] Y. Altun, A. Smola, and T. Hofmann. Exponential families for conditional random fields. In *20th Conference on Uncertainty in Artificial Intelligennce*, 2004.

[11] C. Ambroise and G. Govaert. Em algorithm for partially known labels, data analysis, classification, and related methods. In *Proceedings of the 7th Conference of the International Federation of Classication Societies (IFCS-2000)*, pages 161–166, University of Namur, Belgium, 2000.

[12] C. Andrieu, N. D. Freitas, A. Doucet, and M. I. Jordan. An introduction to mcmc for machine learning. *Machine Learning*, 50 :5–43, 2003.

[13] L. B. A.R. Barron and P. Massart. Risk bounds for model selection via penalization. *Probability Theory and Related Fields*, 113 :301–413, 1999.

[14] P. Baufays and J. Rasson. A new geometric discriminant rule. *Computational Statistics Quaterly*, 2 :15–30, 1985.

[15] H. Bensmail and G. Celeux. Regularized gaussian discriminant analysis through eigenvalue decomposition. *Journal of the American Statistical Association*, 91 :1743–48, 1996.

[16] J. M. Bernardo and A. F. M. Smith. *Bayesian Theory*. John Wiley & Sons, 1st edition, 1994.

[17] 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.

[18] L. Birgé and P. Massart. Minimum contrast estimators on sieves. Technical report, Université Paris Sud, France, 1995.

[19] D. Bosq. Linear processes in function spaces. theory and applications. *Lecture Notes in Statistics*, 149, 2000.

[20] L. Bottou. *Une Approche théorique de l'Apprentissage Connexionniste : Applications à la Reconnaissance de la Parole*. PhD thesis, Université de Paris XI, Orsay, France, 1991.

[21] G. Bouchard and G. Celeux. Supervised classification with spherical Gaussian mixtures. In *Proceedings of CLADAG 2003*, pages 75–78, 2003.

[22] G. Bouchard and B. Triggs. Hierarchical part-based visual object categorization. Submitted to CVPR'05, October 2004.

[23] J. G. Bryan. The generalized discriminant function : mathematical foundations and computational routine. *Harvard Educational Review*, 21 :90–95, 1951.

[24] O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee. Choosing multiple parameters for support vector machines. *Machine Learning*, 46(1) :131–159, 2002.

[25] P. Cheeseman and J. Stutz. Bayesian classification (AUTOCLASS) : Theory and results. In U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, editors, *Advances in Knowledge Discovery and Data Mining*, pages 153–180. AAAI Press/MIT Press, 1996.

[26] M. Collins. Discriminative reranking for natural language parsing. In *Proc. 17th International Conf. on Machine Learning*, pages 175–182. Morgan Kaufmann, San Francisco, CA, 2000.

[27] M. Collins. Discriminative training methods for hidden markov models : Theory and experiments with perceptron algorithms. In *Proc. of the Conference on Empirical Methods in Natural Language Processing*, 2002.

[28] T. Cover and J. Thomas. *Elements of Information Theory*. Series in Telecommunications. John Wiley & Sons, 1st edition, 1991.

[29] A. Cowling and P. Hall. On pseudodata methods for removing boundary effects in kernel density estimation. *Journal of the Royal Statistical Society B*, pages 551–563, 1996.

[30] N. Cristianini and J. Shawe-Taylor. *An introduction to support vector machines*. Cambridge University Press, 2000.

[31] G. Csurka, C. Bray, C. Dance, and L. Fan. Visual categorization with bags of keypoints. In *Proceedings of the 8th European Conference on Computer Vision, Prague*, pages 59–74, 2004.

[32] G. Csurka, C. Dance, L. Fan, J. Williamowski, and C. Bray. Visual categorization with bags of keypoints. In *ECCV'04 workshop on Statistical Learning in Computer Vision*, pages 59–74, Prague, 2004.

[33] L. S. D. Deprins and H. Tulkens. Measuring labor efficiency in post offices. In P. P. M. Marchand and N. H.

H. Tulkens, editors, *The Performance of Public Enterprises : Concepts and Measurements*, Amsterdam, 1984.

[34] A. F. M. S. D. M Titterington and U. E. Makov. *Statistical analysis of finite mixture distributions*. John Wiley, New York, 85.

[35] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, B 39 :1–38, 1977.

[36] P. Domingos and M. J. Pazzani. Beyond independence : Conditions for the optimality of the simple bayesian classifier. In *International Conference on Machine Learning*, pages 105–112, 1996.

[37] G. Dorko and C. Schmid. Selection of scale-invariant parts for object class recognition. In *Proceedings of the 9th International Conference on Computer Vision, Nice, France*, pages 634–640, 2003.

[38] G. Dorko and C. Schmid. Object class recognition using discriminative local features. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 2004. submitted.

[39] A. Doucet, N. de Freitas, and N. Gordon. *Sequential Monte Carlo methods in practice*. Springer-Verlag, New York, 2001.

[40] R. Duda and P. Hart. *Pattern Classification and Scene Analysis*. New York : John Wiley and Sons, 1973.

[41] EDF. Zones inconel du circuit primaire principal., 1999. rapport interne.

[42] B. Efron. The efficiency of logistic regression compared to normal discriminant analysis. *Journ. of the Amer. Statist. Assoc.*, 70 :892–898, 1975.

[43] C. L. F. Bonnans, J.C. Gilbert and C. Sagastizábal. Optimisation numérique. aspects théoriques et pratiques. *Mathématiques & Applications*, 27, 1997.

[44] L. Fei-Fei, R. Fergus, and P. Perona. A Bayesian approach to unsupervised one-shot learning of object categories. In *Proceedings of the 9th International Conference on Computer Vision, Nice, France*, pages 1134–1141, Nice, France, 2003.

[45] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 264–271, June 2003.

[46] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Madison, Wisconsin, USA*, 2003.

[47] R. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7 :179–188, 1936.

[48] C. Fraley and A. E. Raftery. Model-based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association*, 97 :611–631, 2002.

[49] J. Friedman. Regularized discriminant analysis. *Journal of the American Statistical Association*, 84 :165–175, 1989.

[50] N. Friedman, D. Geiger, and M. Goldszmidt. Bayesian network classifiers. *Machine Learning*, 29(2-3) :131–163, 1997.

[51] J. Fritsch. Modular neural networks for speech recognition. Master's thesis, Carnegie Mellon University & University of Karlsruhe, 1996.

[52] J. Fritsch, M. Finke, and A. Waibel. Adaptively growing hierarchical mixtures of experts. In M. I. J. In M. C. Mozer and T. Petsche, editors, *Advances in Neural Informations Processing Systems 9*. MIT Press, 1997.

[53] R. Fry, editor. *Bayesian Inference and Maximum Entropy Methods in Science and Engineering*. AIP Conf. Proc. Amer. Inst. Phys., Melville, NY, 2002.

[54] A. I. G. Bouchard, S. Girard and A. Nazin. Linear programming problems for frontier estimation. Technical Report RR-4717, INRIA, `http ://www.inria.fr/rrrt/rr-4717.html`, 2003.

[55] W. R. Gilks, S. Richardson, and D. J. Spiegelhalter. *Markov Chain Monte Carlo in Practice*. Chapman and Hall, London, 1996 (ISBN : 0-412-05551-1).

This book thoroughly summarizes the uses of MCMC in Bayesian analysis. It is a core book for Bayesian studies.

[56] S. Girard and P. Jacob. Extreme values and kernel estimates of point processes boundaries. *Technical report ENSAM-INRA-UM2*, pages 01–02, 2001.

[57] S. Girard and P. Jacob. Extreme values and haar series estimates of point processes boundaries. *Scandinavian Journal of Statistics*, 30 :369–384, 2003.

[58] S. Girard and P. Jacob. Projection estimates of point processes boundaries. *Journal of Statistical Planning and Inference*, 116 :1–15, 2003.

[59] S. Girard and L. Menneteau. Limit theorems for smoothed extreme values estimates of point processes boundaries. *Journal of Statistical Planning and Inference*, 2003. to appear.

[60] T. Gonçalves and P. Quaresma. Using ir techniques to improve automated text classification. In *Proc. of the 9th International Conference on Applications of Natural Language to Information Systems*, pages 374–379, Salford, 2004.

[61] G. L. Goodman and D. W. McMichael. Objective functions for maximum likelihood classifier design. In R. Evans, L. White, D. McMichael, and L. Sciacca, editors, *Proceedings of Information Decision and Control 99*, pages 585–589, Adelaide, Australia, February 1999. Institute of Electrical and Electronic Engineers, Inc.

[62] C. Goutte, E. Gaussier, N. Cancedda, and H. Déjean. Generative vs discriminative approaches to entity recognition from label deficient data. In *Proc. of the 7èmes Journées internationales Analyse statistique des Données Textuelles*, Louvain-la-Neuve, Belgium, 2004.

[63] P. Green and B. Silverman. *Nonparametric Regression and Generalized Linear Models*. Monographs on Statistics and Probability. Chapman & Hall, 1994.

[64] R. Greiner and W. Zhou. Structural extension to logistic regression : Discriminant parameter learning of belief net classifiers. In *Proc. of the Eighteenth Annual National Conference on Artificial Intelligence*, pages 167–173, Edmonton, 2002.

[65] A. Grigor'yan and M. Noguchi. The heat kernel on hyperbolic space. *Bulletin of the London Mathematical Society*, 30 :643–650, 1998.

[66] Hall, P., Nussbaum, M. and Stern, S.E. On the estimation of a support curve of indeterminate sharpness. *Journal of Multivariate Analysis*, 62 :204–232, 1997.

[67] A. Hardy and J. Rasson. Une nouvelle approche des problèmes de classification automatique. *Statistique et Analyse des données*, 7 :41–56, 1982.

[68] J. Hartigan. *Clustering Algorithms*. Wiley, Chichester, 1975.

[69] T. Hastie and R. Tibshirani. Generalized additive models. *Statistical Science*, 1 :297–318, 1986.

[70] T. Hastie and R. Tibshirani. Discriminant analysis by Gaussian mixtures. *Journal of the Royal Statistical Society series B*, 58 :158–176, 1996.

[71] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, 2001. HAS t 01 :1 1.Ex.

[72] C. Hennig. Models and methods for clusterwise linear regression. *Classification in the Information Age*, pages 179–187, 1999.

[73] C. Hennig. Identifiability of models for clusterwise linear regression. *Journal of Classification*, 17 :273–296, 2000.

[74] J. Hiriart-Urruty and C. Lemaréchal. Convex analysis and minimization algorithms. part 1 : Fundamentals. *Grundlehren der Mathematischen Wissenschaften*, 305, 1993.

[75] J. A. Hoeting, D. D. Madigan, A. E. Raftery, and C. T. Volinsky. Bayesian model averaging : A tutorial (with discussion). *Statistical Science*, 14 :382–417, 1999.

[76] P. J. Huber. The behavior of maximum likelihood estimates under nonstandard conditions. In *Proc. Fifth Berkeley Symp. Math. Statist. Probab. 1 (Univ. of Calif. Press*, pages 221–233, Berkeley and Los Angeles, 1967.

[77] M. A. Hurn, A. Justel, and R. C. P. Estimating mixtures of regressions. Technical report, CREST, France, 2000.

[78] B. P. I. Gijbels, E. Mammen and L. Simar. On estimation of monotone and concave frontier functions. *Journal of the American Statistical Association*, 94 :220–228, 1999.

[79] J. Geffroy. Sur un problème d'estimation géométrique. *Publications de l'Institut de Statistique de l'Université de Paris*, XIII :191–200, 1964.

[80] T. Jaakkola, M. Diekhans, and D. Haussler. A discriminative framework for detecting remote protein homologies. *Journal of Computational Biology*, 7(1,2) :95–114, 2000.

[81] T. S. Jaakkola and D. Haussler. Exploiting generative models in discriminative classifiers. In *Proc. of Tenth Conference on Advances in Neural Information Processing Systems*, 1999.

[82] P. Jacob and P. Suquet. Estimating the edge of a poisson process by orthogonal series. *Journal of Statistical Planning and Inference*, 46 :215–234, 1995.

[83] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton. Adaptive mixture of local experts. *Neural Computation*, 3(1) :79–87, 1991.

[84] G. Jarrad and D. McMichael. Shared mixture distributions and shared mixture classiers. In *Proc. of the Information, Decision and Control Conference*, pages 335–340, Adelaide, Australia, 1999.

[85] T. Jebara. *Discriminative, Generative and Imitative Learning*. PhD thesis, Media Laboratory, MIT, 2001.

[86] T. Jebara, R. Kondor, and A. Howard. Probability product kernels. *Journal of Machine Learning Research, JMLR*, pages 819–844, 2004. Special Topic on Learning Theory.

[87] T. Jebara and A. Pentland. The generalized cem algorithm, 1999.

[88] W. Jiang and M. Tanner. Hierarchical mixtures-of-experts for exponential family regression models, approximation and maximum likelihood estimation. *Ann. Statistics*, 27 :987–1011, 1999.

[89] T. Joachims. Text categorization with support vector machines : learning with many relevant features. In C. Nédellec and C. Rouveirol, editors, *Proceedings of ECML-98, 10th European Conference on Machine Learning*, volume 1398, pages 137–142, Chemnitz, DE, 1998. Springer Verlag, Heidelberg, DE.

[90] M. I. Jordan and R. A. Jacobs. Hierarchical mixtures of experts and the em algorithm. *Neural Computation*, 6 :181–214, 1994.

[91] R. Kass and A. Raftery. Bayes factors. *Journal of the American Statistical Association*, 90 :773–795, 1995.

[92] W. T. F. Kevin Murphy, Antonio Torralba. Using the forest to see the trees : A graphical model relating features, objects, and scenes. In *Neural Info. Processing Systems*, 2003.

[93] N. M. Kiefer. Discrete parameter variation. efficient estimation of a switching regression model. *Econometrica*, 46 :427–434, 1978.

[94] J.-H. Kim, K. K. Kim, and C. Y. Suen. An hmm-mlp hybrid model for cursive script recognition. *Pattern Anal. Appl.*, 3(4) :314–324, 2000.

[95] T. Kohonen. Learning vector quantization. *Neural Networks*, 1(suppl 1) :303, 1988.

[96] P. Kontkanen, P. Myllymäki, and H. Tirri. Classifier learning with supervised marginal likelihood. In J.Breese and D. M. K. Publishers, editors, *Proceedings of the 17th International Conference on Uncertainty in Artificial Intelligence*, pages 277–284, 2001.

[97] A. Korostelev and A. Tsybakov. Minimax theory of image reconstruction. *Lecture Notes in Statistics*, 82, 1993.

[98] Korostelev, A.P., Simar, L. and Tsybakov, A. B. Efficient estimation of monotone boundaries. *The Annals of Statistics*, 23 :476–489, 1995.

[99] S. Kullback. *Information Theory and Statistics*. John Wiley & Sons, 1959.

[100] L. G. L. Devroye and L. Lugosi. *A probabilistic Theory of Pattern Recognition*. Springer-Verlag, New York, 1997.

[101] L. Gardes. Estimating the support of a poisson process via the Faber-Shauder basis and extreme values. *Publications de l'Institut de Statistique de l'Université de Paris*, XXXXVI :43–72, 2002.

[102] N. C. L. Tarassenko, P. Hayton and M. Brady. Novelty detection for the identification of masses in mammograms. In *fourth IEE International Conference on Artificial Neural Networks*, pages 442–447, Cambridge, 1995.

[103] S. Lacoste-Julien. An introduction to max-margin markov networks. UC Berkeley cs281a project report, December 2003.

[104] J. Lafferty and G. Lebanon. Information diffusion kernels. In *Advances in Neural Information Processing, 15*. MIT press, 2003.

[105] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields : Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the ICML*, pages 282–289, 2001.

[106] P. Langley and S. Sage. Tractable average-case analysis of naive bayesian classifiers. In M. Kaufmann, editor, *Sixteenth International Conference on Machine Learning*, pages 220–228, Bled, Slovenia, 1999.

[107] M. W. J. Layard. Large sample tests for equality of two covariances matrices. *Annals of Mathematical Statistics*, 43 :123–141, 1972.

[108] Q. Le and S. Bengio. Hybrid generative-discriminative models for speech and speaker recognition. Technical Report IDIAP-RR 02-06, IDIAP, 2002.

[109] B. Leibe, A. Leonardis, and B. Schiele. Combined object categorization and segmentation with and implicit shape model. In *ECCV'04 workshop on Statistical Learning in Computer Vision*, pages 17–32, Prague, 2004.

[110] D. G. Lowe. Local feature view clustering for 3D object recognition. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Kauai, Hawaii, USA*, pages 682–688, December 2001.

[111] D. G. Lowe. Distinctive image features from scale-invariant keypoint. *International Journal of Computer Vision*, 60(2) :91–110, 2004.

[112] L.P. Devroye and G.L. Wise. Detection of abnormal behavior via non parametric estimation of the support. *SIAM Journal of Applied Mathematics*, 38 :448–480, 1980.

[113] P. McCullach and J. Nelder. *Generalized Linear Models*. Number 37 in Monographs on Statistics and Applied Probability. Chapman & Hall, 1st edition, 1983.

[114] G. J. McLachlan. *Discriminant Analysis and Statistical Pattern Recognition*. Wiley, New York, 1992.

[115] G. J. McLachlan and D. Peel. *Finite Mixture Models*. Wiley, New York, 2000.

[116] R. E. MELCHERS. *Integration and Simulation Methods*, chapter Second-Moment and Transformation Methods, pages 64–93. Wiley, 2001.

[117] S. Mika, G. Rätsch, J. Weston, B. Schölkopf, and K.-R. Müller. Fisher discriminant analysis with kernels. In Y.-H. Hu, J. Larsen, E. Wilson, and S. Douglas, editors, *Neural Networks for Signal Processing IX*, pages 41–48. IEEE, 1999.

[118] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Madison, Wisconsin, USA*, June 2003.

[119] D. Mladenic, J. Branka, M. Grobelnik, and N. Milic-Frayling. Feature selection using linear classifier weights : interaction with classification models. In *Proc. of SIGIR*, pages 234–241, 2004.

[120] P. Moerland. Classification using localized mixtures of experts. In *proc. of the International Conference on Artificial Neural Networks*, 1999.

[121] P. Moerland. A comparison of mixture models for density estimation. In *Proceedings of the International Conference on Artificial Neural Networks (ICANN'99)*, volume 1, pages 25–30. London : IEE, 1999. (IDIAP-RR 98-14).

[122] A. Mohammad-Djafari, editor. *Bayesian Inference and Maximum Entropy Methods in Science and Engineering*, volume 568 of *AIP Conf. Proc.* Amer. Inst. Phys., Melville, NY, 2001.

[123] P. J. Moreno, P. P. Ho, and N. Vasconcelos. A Kullback-Leibler divergence based kernel for SVM classification in multimedia applications. In *Advances in Neural Information Processing Systems 16*, 2004.

[124] N. Murata, S. Yoshizawa, and S.-I. Amari. Network Information Criterion—determining the number of hidden units for an artificial neural network model. *IEEE Transactions on Neural Networks*, 5(6) :865–872, November 1994.

[125] K. P. Murphy and M. A. Paskin. Linear-time inference in hierarchical hmms. In *NIPS*, pages 833–840, 2001.

[126] R. Neal. Assessing relevance determination methods using delve. *Generalization in Neural Networks and Machine Learning*, 1998.

[127] A. Y. Ng and M. I. Jordan. On discriminative vs. generative classifiers : A comparison of logistic regression and naive bayes. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, pages 609–616, Cambridge, MA, 2002. MIT Press.

[128] T. O'Neil. A general distribution of the error rate of a classification procedure with application to logistic regression discrimination. *JASA*, 75 :154–160, 1980.

[129] A. Opelt, M. Fussenegger, A. Pinz, and P. Auer. Weak hypotheses and boosting for generic object detection and recognition. In *Proceedings of the 8th European Conference on Computer Vision, Prague*, volume 2, pages 71–84, 2004.

[130] B. Park, L. Simar, and C. Wiener. The fdh estimator for productivity efficiency scores : asymptotic properties. *Econometric Theory*, 16 :855–877, 2000.

[131] Y. A. Qi, T. P. Minka, R. W. Picard, and Z. Ghahramani. Predictive automatic relevance determination by expectation propagation. In *Proceedings of the 21th International Conference on Machine Learning*, 2004.

[132] R. E. Quandt. A new approach to estimating switching regressions. *Journal of the American Statistical Association*, 67 :306–310, 1972.

[133] R. E. Quandt and J. B. Ramsey. Estimating mixtures of normal distributions and switching regressions. *JASA*, 73 :730–752, 1978.

[134] A. E. Raftery. Bayesian model selection in social research (with discussion). *Sociological Methodology*, pages 111–196, 1995.

[135] R. Raina, Y. Shen, A. Y. Ng, and A. McCallum. Classification with hybrid generative/discriminative models. In S. Thrun, L. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA, 2004.

[136] C. Rao. *Linear Statistical Inference and its applications*. Wiley, 2 edition, 2001.

[137] B. D. Ripley. *Pattern Recognition and Neural Networks*. University Press, Cambridge, 1996.

[138] C. Robert. Intrisic loss functions. *Theory and Decision*, 40(2) :191–214, 1996.

[139] C. P. ROBERT. Simulation of truncated normal variables. *Statistics and computing*, 5 :121–125, 1995.

[140] C. P. ROBERT. *The Bayesian Choice : from Decision-Theoretic Motivations to Computational Implementation (2001)*. Springer-Verlag, NY, 2001.

[141] K. Roeder and L. Wasserman. Practical Bayesian density estimation using mixtures of normals. *Journal of the American Statistical Association*, 92 :894–902, 1997.

[142] Y. D. Rubinstein and T. Hastie. Discriminative vs. informative learning. In A. Press, editor, *In Proc. of the Third International Conference on Knowledge and Data Mining*, pages 49–53, 1997.

[143] M. Saerens. Building cost functions minimizing to some summary statistics. *IEEE Transactions on Neural Networks*, 11(6) :1263–1271, 2000.

[144] M. Sato and S. Ishii. On-line em algorithm for the normalized gaussian network. *Neural Computation*, 12(2) :407–432, 2000.

[145] B. Schölkopf and A. Smola. *Learning with Kernels*. MIT Press, 1st edition, 2002.

[146] B. Schölkopf and A. Smola. *Learning with kernels*. MIT University Press, Cambridge, 2002.

[147] G. Schwartz. Estimating the dimension of a model. *The Annals of Statistics*, 6(2) :461–464, 1978.

[148] B. Schölkopf and A. J. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, 2002.

[149] F. Sha and F. Pereira. Shallow parsing with conditional random fields. In *Proceedings of HLT-NAACL*, pages 213–220, Association for Computational Linguistics, 2003.

[150] L. Sigal, S. Bhatia, S. Roth, M. J. Black, and M. Isard. Tracking loose-limbed people. In *CVPR (1)*, pages 421–428, 2004.

[151] M. M. T. Jaakkola and T.Jebara. Maximum entropy discrimination. In *Advances in Neural Information Processing Systems 11*, 1999.

[152] B. Taskar, C. Guestrin, and D. Koller. Max-margin markov networks. In S. Thrun, L. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16*, Cambridge, MA, 2004. MIT Press.

[153] M. Tipping. The relevance vector machine. In *Advances in Neural Information Processing Systems*, San Mateo, 2000. Morgan Kaufmann.

[154] M. Tipping and C. Bishop. Probabilistic principal component analysis. Technical report, Aston University, 1997.

[155] M. Titsias and A. Likas. Mixture of experts classification using a hierarchical mixture model. *Neural Computation*, 14(9) :2221–2244, 2002.

[156] K. Tsuda, M. Kawanabe, G. Rätsch, S. Sonnenburg, and K. Müller. A new discriminative kernel from probabilistic models. *Neural Computation*, 14(10) :2397–2414, October 2002.

[157] S. Ullman, E. Sali, and M. Vidal-Naquet. A fragment-based approach to object representation and classification. In *4th International Workshop on Visual Form, Capri, Italy*, May 2001.

[158] V. N. Vapnik. *Statistical Learning Theory*. Wiley, 1st edition, 1998.

[159] B. P. W. Härdle and A. Tsybakov. Estimation of a non sharp support boundaries. *Journal of Multivariate Analysis*, 43 :205–218, 1995.

[160] P. H. W. Härdle and L. Simar. Iterated boostrap with application to frontier models. *Journal of Productivity Analysis*, 6 :63–76, 1995.

[161] S. Waterhouse. *Classification and regression using mixtures of experts*. PhD thesis, Department of Engineering, Cambridge University, 1997.

[162] M. Weber, M. Welling, and P. Perona. Unsupervised learning of models for recognition. In *Proceedings of the 6th European Conference on Computer Vision, Dublin, Ireland*, pages 18–32, 2000.

[163] H. Wettig, P. Grünwald, T. Roos, P. Myllymäki, and H. Tirri. When discriminative learning of bayesian network parameters is easy. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*, pages 491–498, 2003.

[164] L. Xu, G. Hinton, and M. I. Jordan. An alternative model for mixtures of experts. In G. T. et al., editor, *Advances in Neural Information Processing Systems*, volume 7, pages 633–640, Cambridge MA, MIT Press, 1995.

[165] L. Xu and M. I. Jordan. On convergence properties of the EM algorithm for gaussian mixtures. *Neural Computation*, 8(1) :129–151, 1996.

[166] Y. Yang. An evaluation of statistical approaches to text categorization. *Information Retrieval*, 1(1/2) :69–90, 1999.

[167] J. Zhu and T. Hastie. Kernel logistic regression and the import vector machine. In *NIPS*, pages 1081–1088, 2001.