Apprentissage dans les espaces de grande dimension : Application à la caractérisation de tumeurs noires de la peau à partir d'images

Arthur Tenenhaus

UNIVERSITE PIERRE & MARIE CURIE
LA SCIENCE A PARIS

THESE DE DOCTORAT DE L'UNIVERSITÉ PARIS 6

Spécialité

Statistique

Présentée par
Arthur Tenenhaus

Pour obtenir le grade de

DOCTEUR de L'UNIVERSITÉ PARIS 6

Sujet de la thèse :
Apprentissage dans les espaces de grande dimension. Application à la caractérisation des tumeurs noires de la peau à partir d'images.

Soutenue le 8 Décembre 2006

Devant le jury composé de :

Paul DEHEUVELS	Professeur, Université Paris VI	*Directeur de thèse*
Bernard FERTIL	Directeur de recherche CNRS, Paris	*Directeur de thèse*
Gilbert SAPORTA	Professeur, CNAM	*Directeur de thèse*
Jean-Michel POGGI	Professeur, Université Paris V	*Rapporteur*
Michel VERLEYSEN	Professeur, Université Catholique de Louvain	*Rapporteur*
Patrick GALLINARI	Professeur, Université Paris VI	*Président du jury*
Michel BÉRA	Directeur du pôle universitaire Léonard de Vinci	*Examinateur*

Université Pierre & Marie Curie – Paris
Bureau de la scolarité Doctorat
Esc G, 2 ème Etage
15, rue de l'école de médecine
75270 Paris Cedex 06

Tél Secrétariat : 01 42 34 68 35
Fax : 01 42 34 68 40
Tél pour les étudiants de A à ME : 01 42 34 68 41
Tél pour les étudiants de MG à Z : 01 42 34 68 51
E-mail : scolarite.doctorat@upmc.fr

À ma grand-mère, Germaine Tenenhaus

Remerciements

Je remercie tout d'abord Alain Herment, directeur de l'unité INSERM 678 de m'avoir acceuilli dans son laboratoire.

J'ai passé mes trois années de thèse dans un environnement exceptionnel grâce à Bernard Fertil, mon directeur. Bernard m'a acceuilli chaleureusement dans son équipe, m'a fait découvrir son univers avec enthousiasme. Mon avenir professionnel sera nécessairement influancé par sa vision de la recherche.

Je remercie Gilbert Saporta pour sa disponibilité et ses conseils avisés.

Enfin, je remercie Paul Deheuvels pour les précieux conseils qu'il m'a prodigués (dont le principal a été de toujours plus travailler).

Je remercie Jean-Michel Poggi et Michel Verleysen de m'avoir fait l'honneur d'accepté d'être rapporteurs de cette thèse.

Je remercie Patrick Gallinari d'avoir accepté de présider mon jury de thèse.

Merci à Roger Haddad, Directeur de KXEN, d'avoir accepté de financer mon travail.

Je souhaite également remercier Michel Béra, bouillonant d'idées et d'enthousiasmes d'avoir accepté de me suivre tout au long de la thèse et d'avoir acceptè de faire parti des membres du jury.

Je remercie Alain Giron. J'ai beaucoup apprécié sa compagnie et espère avoir l'occasion de le découvrir plus qu'il ne veut bien se dévoiler.

Merci à Philippe Dumée qui m'a toujours protégé des d'aléas des caprices de l'informatique.

Je remercie les chercheurs de l'équipe 1 dirigée de main de maître par Habib Benali, Odile, Mélanie pour sa participation active à l'arrivée de mon bébé, Vincent, Saad, Pierre, Jean. J'ai rarement cotoyé des gens si brillants.

Je remercie les chercheurs de l'équipe 2 et 3. Frédérique, Alain de Césare avec qui j'ai partagé d'agréables pauses dans la cuisine. Merci à Sébastien, Nadjia et Daniel pour avoir partagé une multitude de pauses et repas avec moi. C'était super !

Enfin je remercie l'équipe 4 avec qui j'ai partagé l'espace et la plus grande partie de

mon temps. Jean-François Caroline, Alex, Marie-Odile, Philippe que j'espère ne pas avoir trop quotidiennement déranger. et Clara. Merci pour votre présence, votre soutien, votre gentillesse et tant d'autres choses.

Je remercie Philippe Bastien, une des premières personnes à m'avoir fait découvrir le monde de la recherche.

Je remercie Michel Tenenhaus, pour les conseils qu'il m'a prodigué. Son enthousiasme a toujours été source de motivation.

Je remercie ma famille et mes amis pour leur soutien constant et les joies qu'ils m'apportent.

J'adresse mes plus intimes remerciements à Fanny qui m'a toujours soutenu et encouragé. La vie nous a réservé de sacrés surprises. Nous avons traversé les joies et les tristesses main dans la main. La petite merveille, Élie Tenenhaus, est né le 06 octobre 2006 à 22h26, voilà encore une belle page de notre histoire qui s'ouvre...

Table des matières

Table des figures

Liste des tableaux

Introduction

La statistique appliquée a beaucoup évolué ces dernières années. Bien que ses objectifs soient restés les mêmes, à savoir la description, le traitement et l'analyse de données, elle doit répondre à une nouvelle problématique. En effet, l'explosion de la quantité d'information, due à l'apparition de nouvelles technologies (l'image, les puces à ADN, Internet, les données commerciales et financières, ...) nécessite le développement de nouvelles méthodes d'analyse ou du moins, l'adaptation des méthodes existantes.

Nous nous intéressons dans ce travail à des problématiques dites «supervisées» : l'information dont nous disposons est représentée par un tableau individus $\times$ variables $[X, y]$. Chaque individu i est décrit par le vecteur ligne (x_i, y_i) où x_i est formé de p caractéristiques (correspondant aux différentes mesures prélevées sur chacun des individus) et y_i une valeur discrète ou continue. Les n couples $(x_i, y_i)_{i=1,\dots,n}$ forment l'échantillon d'apprentissage, noté D_n. L'apprentissage supervisé a pour objectif de prédire la valeur y d'un individu x en fonction de l'information dont on dispose sur cet individu. Le résultat d'un tel algorithme est une fonction $\hat{f}_n$ estimée à partir de l'échantillon d'apprentissage D_n. On construira un modèle de régression si $y_i \in \mathbb{R}$ et de classification si $y_i \in \mathbb{N}$. Or, les données d'apprentissage ne sont que des exemples qui comportent leur part d'incertitude (liée au bruit, à l'imprécision des mesures). Si la fonction $\hat{f}_n$ la prend en compte, elle accorde trop de confiance aux données d'apprentissage et ses capacités de généralisation en seront affectées : c'est le sur-apprentissage (du terme anglais *overfitting*). Cependant, si $\hat{f}_n$ n'est pas assez fidèle aux données, les capacités de généralisation en seront également affectés : c'est le sous-apprentissage (du terme anglais *underfitting*). La fonction $\hat{f}_n$ issue de l'algorithme d'apprentissage doit donc rester fidèle à D_n sans donner trop d'importance aux données prises individuellement. Il s'agit donc de régler la complexité du modèle de sorte à obtenir le «meilleur» compromis entre biais du modèle ajusté et variance. Dans ce contexte, pour mesurer la qualité du compromis, l'idéal est de disposer d'un échantillon indépendant de D_n, dit échantillon de test, sur lequel se mesure la précision des prédictions de $\hat{f}_n$. On parle alors de validation.
Ce compromis prend en compte les notions de régularisation, de réduction de dimension ou encore de minimisation du risque structurel. Son étude constitue le fil conducteur de notre travail.

Dans les multiples domaines d'application de l'apprentissage, on rencontre souvent des données de grande dimension ($n << p$), dont voici trois exemples typiques :

- Aide au diagnostic et imagerie médicale : chaque image est caractérisée par les p pixels qui la composent.

- Bio-informatique et puces à ADN : cette technologie permet de mesurer l'expression de plusieurs dizaines de milliers de gènes sur un individu. On récupère alors une quantité d'information conséquente. Cette information ne peut être actuellement collectée que sur un petit

nombre d'individus, car chaque expérience de puce à ADN est coûteuse.

- Chimiométrie et Spectroscopie : cette technologie permet de décrire un individu par une courbe. La courbe est discrétisée pour fournir un ensemble de descripteurs.

Ces exemples ont comme particularité commune d'être composés d'un grand nombre de variables explicatives fortement corrélées et mesurées sur un petit nombre d'individus. Or dans de tels espaces, on rencontre souvent des phénomènes de sur-apprentissage. Il est donc fondamental de maîtriser ce compromis. Quelles sont les outils à disposition pour gérer ce type de problématique ? Quelles sont les principes sur lesquelles doivent s'appuyer le choix de modèles ?

Les deux premiers chapitres de ce document parcourent un horizon des méthodes adaptées à la gestion des données de grande dimension et permettent de dégager les grands principes mises en oeuvre dans ce but.

Le premier principe est d'ordre algorithmique : la majorité des algorithmes classiques de modélisation nécessite la plupart du temps l'inversion de matrices de dimension $p \times p$. Il s'avère qu'un grand nombre de ces algorithmes peuvent s'absoudre de la gestion de matrices $p \times p$ en passant d'une forme primale à une représentation duale. La dynamique primale/duale est à la base des méthodes à noyau (du terme anglais «Kernel Method») et implique alors la gestion de matrice $n \times n$, configuration particulièrement bien adaptée au contexte $n << p$. Ces méthodes présentent l'avantage de s'étendre sans difficulté au contexte non linéaire via les propritétés des espaces de Hilbert à noyau reproduisant. L'extension au cadre non linéaire est intéressante lorsque l'on cherche à maximiser la qualité de prédiction. Notons que cette dernière s'obtient au détriment d'une perte de lisibilité et d'interprétation des modèles.

Le second principe est d'ordre conceptuel : il s'agit de définir les fondements permettant de gérer les données de grande dimension. L'idée générale consiste à réduire la complexité des modèles en introduisant des contraintes sur la nature des solutions issues de l'apprentissage. Les méthodes de régularisation issues de l'analyse numérique pour la résolution de problèmes inverses ou de problèmes mal posés répond à ce type de problématique. Il est intéressant de constater que la majorité des méthodes à noyau supervisées de la littérature repose sur des principes de régularisation.

Ainsi, le premier chapitre de la thèse décrit les approches s'appuyant sur des techniques explicites de régularisation de type Tikhonov dans laquelle s'inscrivent, par exemple, la Ridge Regression, et plus spécifiquement en classification, les Support Vector Machines ou la Kernel Logistique Regression.
Le deuxième chapitre se focalise sur les méthodes implicites de régularisation basées sur des approches de réduction de dimension. En effet, comme nous le verrons, la réduction de dimension lorsqu'elle précède une étape de construction de modèle se comporte comme un outil de «régularisation». C'est le cas, par exemple, de la régression sur composantes principales et des moindres carrés partiels ou régression PLS (du terme anglais Partial Least Squares Regression). Ces deux méthodes ont l'intérêt algorithmique de pouvoir bénéficier de la dynamique primale-duale et ne requièrent que la gestion de matrices de dimension $n \times n$. Dans ce deuxième chapitre, nous nous focalisons sur les méthodes de réduction de dimension supervisée et notamment sur la régression PLS plus adaptée au contexte supervisé.

Partant du constat que la régression PLS n'a historiquement pas été conçue pour répondre à

des tâches de classification, nous présentons dans le troisième chapitre une extension dédiée à la classification : la régression logistique PLS et nous en proposons une version non linéaire : la Kernel Logistique PLS. Cette nouvelle approche est basée sur des méthodes de réduction de dimension supervisée et des transformations de type Empirical Kernel Map.
Le quatrième chapitre se concentre sur des problèmes de classification multiclasses. Après une brève revue des Support Vector Machines multiclasses, nous proposons des extensions linéaire et non linéaire de la régression logistique PLS au cas où la variable à prédire est catégorielle à plus de deux modalités : la Régression Logistique Multinomiale PLS et la Kernel Logistique Multinomiale PLS.
L'ensemble des méthodes décrites dans ces premiers chapitres est évalué empiriquement sur benchmarks dans le cinquième chapitre.
Le sixième chapitre décrit une application de la Kernel Logistique PLS à un problème complexe de classification d'images médicales. Il s'agit de concevoir un système de caractérisation automatique des tumeurs noires de la peau établissant, à partir d'images numériques, un diagnostic argumenté sur la nature de la lésion (lésion bénigne vs. lésion maligne).

Chapitre 1

Minimisation du risque structurel, méthodes à noyau et régularisation

Nous nous plaçons dans le cas d'un apprentissage supervisé à partir d'une base D de taille finie. On considère que les n observations de la base d'apprentissage sont indépendantes, identiquement distribuées, de loi P inconnue et définies dans $\mathcal{X} \subset \mathbb{R}^p$. À chaque observation est associée une valeur $y \in \mathcal{Y}$ de loi $P(y/x)$ inconnue qu'il s'agit de prédire. L'échantillon d'apprentissage D_n est alors défini par : $D_n = \{(x_1, y_1), ..., (x_n, y_n)\}$ selon la loi jointe $P(x, y) = P(x)P(y/x)$. Imaginons que l'on dispose de «beaucoup» de données dans le sens où de nombreuses mesures ont pu être prises sur chacun des individus (p grand). Chaque observation est alors décrite par un grand nombre de variables et il est fort probable que les différentes mesures apportent de l'information redondante (multicolinéarité) mais également du «bruit» (provenant par exemple d'imprécisions dans les mesures). Le grand nombre de variables (grande dimension), la redondance (multicolinéarité) et le bruit peuvent masquer la structure reliant les individus à la variable à prédire. Dans de telles situations, lorsque l'on cherche à modéliser le lien entre une variable à expliquer et p variables explicatives (régression, classification), la tâche peut s'avérer délicate et on parle généralement de problèmes mal-conditionnés impliquant principalement instabilité numérique et sur-apprentissage. La théorie de Vapnik [Vapnik (1998)] apporte des vues éclairantes sur ce qu'on appelle la capacité de généralisation d'un modèle, c'est-à-dire sa faculté à prédire correctement de nouvelles valeurs et pas seulement à rendre compte du passé. La première partie de ce chapitre est donc dédiée à la présentation de cette théorie : la Minimisation du Risque Structurel (SRM : Structural Risk Minimization). La mise en oeuvre pratique du SRM est un problème difficile mais les connections avec la théorie de la régularisation, que nous présentons dans une seconde partie, vont nous amener à considérer une panoplie d'algorithmes (e.g. SVM, Ridge Regression, ...) pour réaliser cette tâche.

1.1 La Minimisation du Risque Structurel (SRM)

La modélisation dans les espaces de grande dimension a donné lieu à l'essor d'une communauté à la frontière de l'informatique et des statistiques : la communauté de l'apprentissage. Vapnik est sans doute l'un des initiateurs de ce domaine de recherche fructueux et en constante évolution. La question centrale de cette communauté est d'évaluer le degré de généralisation

d'un modèle f, c'est-à-dire sa capacité à prédire correctement la classe d'un nouvel individu (qui n'a pas servi à l'estimation de la fonction de décision). L'apprentissage consiste donc à déterminer une fonction $\hat{f} \in \mathcal{F}$ qui minimise le risque fonctionnel (1.1)

$$R_{réel}[f] \quad = \quad \int_{\mathcal{X}\times\mathcal{Y}} V(y, f(x))P(x, y)dxdy. \tag{1.1}$$

où la fonction de coût V mesure la concordance entre y_i et $f(x_i)$ (par exemple, $V(y, f(x)) = (y - f(x))^2$). P étant inconnu, on définit le risque empirique (1.2).

$$R_{emp}[f, n] \quad = \quad \frac{1}{n}\sum_{i=1}^{n} V(y_i, f(x_i)) \tag{1.2}$$

À partir de n observations, $(x_i, y_i)_{i=1}^n \in \mathcal{X} \times \mathcal{Y}$, on cherche alors à approcher la fonction $\hat{f}$ minimisant (1.1) par la fonction $\hat{f}_n$ minimisant le risque empirique R_{emp}. La question centrale est de savoir si la fonction de décision $\hat{f}_n$ qui minimise R_{emp} minimise également $R_{réel}$. Les travaux de Vladimir Vapnik répondent à cette interrogation [Vapnik (1995) ; Vapnik (1998)]. Ils établissent un lien entre R_{emp} et $R_{réel}$.

Soit $\mathcal{F} = \{f_\lambda, \lambda \in \Lambda\}$, l'espace de recherche de la fonction de décision $\hat{f}_n$ (appelé couramment espace d'hypothèse). Si $\mathcal{F}$ est un espace «riche» (par exemple, $\mathcal{F} = \{$polynôme d'ordre $p\}$, p grand), il est simple de construire une fonction de décision $\hat{f}_n$ annulant R_{emp} ; une telle fonction ne minimisera pas, en général, $R_{réel}$. A contrario, si on choisit un espace $\mathcal{F}$ trop restreint (par exemple, $\mathcal{F} = \{$ polynôme d'ordre 1$\}$), il peut être impossible d'y trouver une bonne solution. [Vapnik (1995)] propose donc une méthode d'estimation de $\hat{f}_n$ basée sur l'obtention de bornes sur $R_{réel}$. Il démontre que $R_{réel}$ dépend des données et de la complexité de l'espace $\mathcal{F}$. Cette notion informelle de richesse des espaces d'hypothèses a été conceptualisée par Vapnik et Chervonenkis [Vapnik (1995)] dans le cadre de la classification par la dimension de Vapnik-Chervonenkis ou VC dimension et par la V_γ dimension dans le cadre de la régression (voir par exemple [Evgeniou and Pontil (1999)]). Par souci de simplicité de la présentation, nous nous focalisons dans la suite sur la VC-dimension.

Définition 1 *(VC dimension) Soit un ensemble de n observations, à chacune desquelles est associée une étiquette* ± 1 *donnant sa classe d'appartenance. Il existe* 2^n *étiquetages possibles. Si pour chaque étiquetage, il existe dans* $\mathcal{F}$ *une fonction f associant à chaque point la bonne étiquette, on dit que l'ensemble des n points est éclaté par* $\mathcal{F}$*. La VC-dimension de* $\mathcal{F}$ *notée h, est définie comme le nombre maximum de points pouvant être éclatés par* $\mathcal{F}$*.* □

Vapnik [Vapnik (1995)] montre que la VC-dimension, h, permet de relier $R_{réel}$ à R_{emp} mesuré sur l'ensemble d'apprentissage. On a, avec une probabilité $1 - \eta$,

$$R_{réel}(f) \quad \leq \quad \underbrace{R_{emp}(f) + \underbrace{\sqrt{\frac{h\left(log(\frac{2n}{h}) + 1\right) - log\left(\frac{\eta}{4}\right)}{n}}}_{= \textit{intervalle de confiance}}}_{= \textit{risque structurel}} \tag{1.3}$$

Remarque 1

1. La différence entre $R_{réel}$ *et* R_{emp} *est inversement proportionnelle à la valeur de n. Cela signifie que la qualité du classifieur est proportionnelle au nombre d'observations dont*

on dispose : plus n est grand, plus le modèle est précis. D'un autre côté, une fonction de décision choisie parmi un espace d'hypothèse de VC dimension h élevée, peut engendrer un sur-apprentissage c'est-à-dire donner lieu à un risque $R_{réel}$ plus élevé que celui mesuré sur R_{emp}.

2. L'inégalité (1.3) est une inégalité universelle dans le sens où elle ne suppose pas d'hypothèses sur la distribution des données.

3. La majoration qu'elle fournit peut-être très conséquente (notamment pour de grandes valeurs de h/n) et de nombreux travaux sont consacrés à la recherche de bornes plus fines. □

En observant l'inégalité (1.3), il est clair qu'une faible valeur du risque empirique n'implique pas nécessairement une faible valeur du risque réel. Puisque qu'on ne peut pas minimiser $R_{réel}$, on va alors chercher à minimiser la borne supérieure de l'inégalité (1.3) : le «risque structurel». Le risque structurel est composé de deux termes : le risque empirique et l'intervalle de confiance. Pour obtenir une solution $\hat{f}_n$ fournissant un risque réel faible, c'est-à-dire un fort pouvoir généralisant, il est fondamental de minimiser à la fois le risque empirique et le ratio entre la VC-dimension et le nombre de points. De surcroît, le risque empirique est habituellement une fonction décroissante de h impliquant que pour un n fixé, on peut trouver une valeur optimale de h (cf. figure 1.1). Le choix de h est donc crucial pour obtenir de bonnes performances de généralisation, particulièrement quand le nombre d'observations est faible. Ainsi, en minimisant la borne supérieure de l'inégalité (1.3), on établit un compromis entre la minimisation du risque empirique et la complexité de l'espace d'hypothèse $\mathcal{F}$. Ce compromis conduit Vapnik à proposer une nouvelle approche dite de «*Minimisation du Risque Structurel*» (SRM : **S**tructural **R**isk **M**inimization) [Vapnik (1982)]. L'objectif principal du SRM consiste à limiter la complexité de l'espace $\mathcal{F}$ des fonctions admissibles au travers d'une faible valeur de h. Il est malheureusement difficile de calculer h si bien que la plupart des implémentations en contrôle seulement la valeur. Finalement, le SRM recherche le meilleur compromis entre biais (premier terme de l'inégalité (1.3)) et variance (second terme de l'inégalité (1.3)) : plus la classe $\mathcal{F}$ est complexe (de VC-dimension élevée), plus la variance est grande et plus le biais est petit. La mise en oeuvre du SRM consiste alors à établir un compromis en considérant une structure sur l'ensemble $\mathcal{F} = \{f(x, \lambda) \ / \ \lambda \in \Lambda\}$ des fonctions admissibles.

Définition 2 *Une structure sur $\mathcal{F}$ est une suite de sous-ensembles $\mathcal{F}_i = \{f(x, \lambda) \ / \ \lambda \in \Lambda_i\}$ emboîtés*

$$\mathcal{F}_1 \subset \mathcal{F}_2 \subset ... \subset \mathcal{F}_M$$

L'ordre des $\mathcal{F}_i$ étant lié à l'ordre des h_i, $h_1 \leq h_2 \leq ... \leq h_M$ où chaque h_i définit la VC-dimension de $\mathcal{F}_i$. □

Le SRM consiste alors à estimer pour chaque $\mathcal{F}_m$, $m = 1, ..., M$ une fonction $\hat{f}_n^m$ puis de choisir parmi ces M fonctions celle qui minimise le deuxième terme de l'inégalité (1.3). Ce processus de sélection de modèles revient à choisir la fonction solution $\hat{f}_n$ combinant la qualité d'approximation à travers la minimisation du risque empirique à la complexité de la fonction solution.

Le SRM a de solides bases mathématiques mais est difficilement utilisable en pratique pour les raisons suivantes :

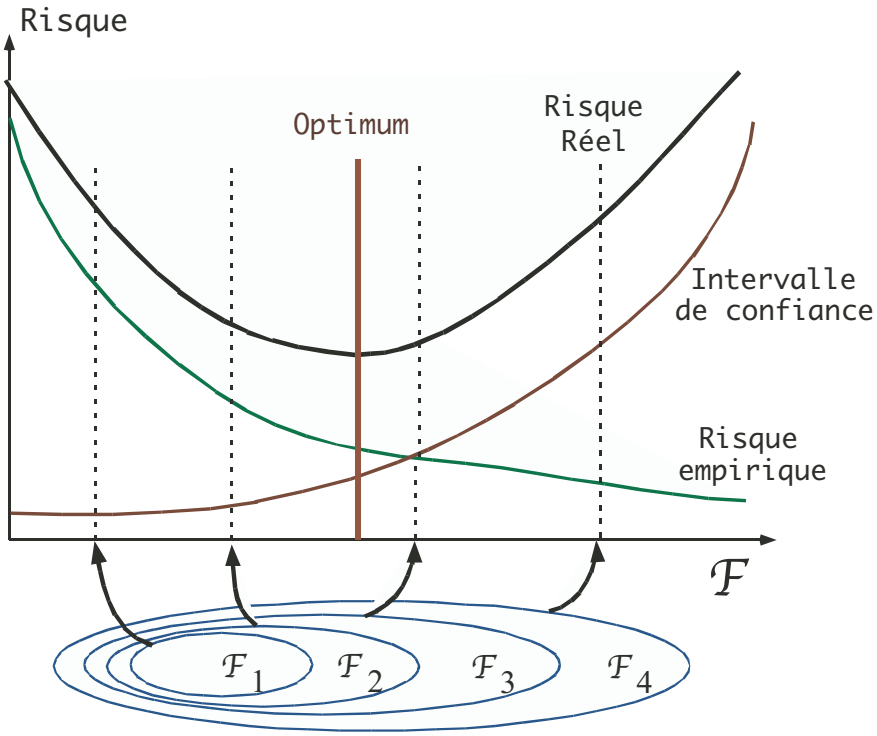

FIG. 1.1: Minimisation du risque structurel : schéma de mise en oeuvre

- La VC-dimension des $\mathcal{F}_i$ est difficile à calculer.

- Même en supposant la VC-dimension des $\mathcal{F}_i$ connue, minimiser le membre de droite de l'inégalité (1.3) sur chacun des $\mathcal{F}_i$ peut s'avérer être un problème délicat et coûteux.

La section suivante fournit donc la stratégie et les outils qui permettent de contourner chacune de ces deux difficultés. La stratégie repose sur deux idées fondamentales : les fonctions à estimer appartiennent à des Espaces de Hilbert à Noyau Reproduisant et sont obtenues par des méthodes de régularisation. Nous introduisons ces concepts dans la suite du chapitre.

1.2 Espaces de Hilbert à Noyau Reproduisant (RKHS)

Commençons par définir formellement les espaces de Hilbert à Noyau reproduisant (RKHS : Reproducing Kernel Hilbert Space).

Définition 3 *Un espace de Hilbert à noyau reproduisant $\mathcal{H}$ (RKHS) est un espace de Hilbert de fonctions pour lequel il existe une fonction $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ (appelée noyau auto-reproduisant) vérifiant :*

- *$\mathcal{H}$ contient toutes les fonctions $k(x, .)$ pour $x \in \mathcal{X}$*
- *La propriété reproduisante est satisfaite*

$$\forall f \in \mathcal{H},\ \forall x \in \mathcal{X}\ \ < f, k(x, .) >_{\mathcal{H}} = f(x)$$

□

Définition 4 *Une fonction $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ est un noyau défini positif si et seulement si :*

- *elle est symétrique : $\forall x_1, x_2 \in \mathcal{X}\ k(x_1, x_2) = k(x_2, x_1)$*
- *et définie positive :*

$$\forall n \geq 1,\ \forall (x_1, ..., x_n) \in \mathcal{X}^n,\ \forall (a_1, ..., a_n) \in \mathbb{R}^n,\ \sum_{i,j=1}^{n} a_i a_j k(x_i, x_j) \geq 0$$

□

Le théorème de Moore-Aronszajn stipule qu'à tout noyau défini positif correspond un unique RKHS [Aronszajn (1950)]. Notons que ce résultat ne suppose aucune condition sur l'espace d'origine $\mathcal{X}$. Pour renforcer cette idée d'unicité, dans la suite du document, nous noterons $\mathcal{H}_K$, le RKHS associé à la fonction noyau k.

Afin de mieux comprendre la définition formelle du RKHS. Nous introduisons le théorème de Mercer [Mercer (1909)] duquel vont découler de nombreuses et remarquables implications pratiques.

Théorème 1 *Théorème de Mercer*
Toute fonction définie positive $k(x, y)$ définie sur un domaine compact $\mathcal{X} \times \mathcal{X}$ peut s'écrire sous la forme d'une série uniformément convergente (1.4)

$$k(x, y) = \sum_{i=1}^{M} \lambda_i \psi_i(x) \psi_i(y) \qquad , M \leq \infty \tag{1.4}$$

où les $\{\lambda_i\}_{i=1}^{M}$ et les $\{\psi_i\}_{i=1}^{M}$ sont les valeurs propres et les fonctions propres orthonormales de l'opérateur (linéaire auto-adjoint compact donc diagonalisable) $T_K : L_2(\mathcal{X}) \to L_2(\mathcal{X})$,

$$(T_k f)(x) = \int_{\mathcal{X}} k(x, y) f(y) dy \quad \forall f \in L_2(\mathcal{X}) \tag{1.5}$$

Dans le cas où $M = \infty$, la série est alors presque partout absolument et uniformément convergente. □

Remarque 2 *La séquence $\{\psi_i\}_{i=1}^{M}$ est une base orthonormale de $\mathcal{H}_K$. Il s'ensuit que toutes fonctions $f \in \mathcal{H}_K$ s'expriment sous la forme (1.6).*

$$f(x) = \sum_{i=1}^{M} a_i \psi_i(x) \tag{1.6}$$

□

Le Théorème 1 nous apprend que tout noyau défini positif admet la représentation suivante :

$$k(x, y) = \sum_{i=1}^{M} \lambda_i \psi_i(x) \psi_i(y) = \sum_{i=1}^{M} \sqrt{\lambda_i} \psi_i(x) \sqrt{\lambda_i} \psi_i(y) = < \Phi(x), \Phi(y) > \tag{1.7}$$

Nous pouvons alors définir explicitement l'espace de redescription engendré par la fonction noyau k

$$\begin{array}{rcl} \Phi & : & \mathcal{X} \rightarrow \mathcal{H}_K \\ & & x \rightarrow (\sqrt{\lambda_1}\psi_1(x), \sqrt{\lambda_2}\psi_2(x), ..., \sqrt{\lambda_M}\psi_M(x)) \end{array} \tag{1.8}$$

M définit la dimension de $\mathcal{H}_K$ et peut être de dimension infinie.

À tout RKHS $\mathcal{H}_K$, on associe un produit scalaire et une norme, définis par les équations (1.9) et (1.10) : posons $f(x) = \sum_{i=1}^{M} a_i\psi_i(x)$ et $h(x) = \sum_{i=1}^{M} b_i\psi_i(x)$,

Le produit scalaire dans $\mathcal{H}_K$ est défini par

$$< f(x), h(x) >_{\mathcal{H}_K} = \left\langle \sum_{i=1}^{M} a_i\psi_i(x), \sum_{i=1}^{M} b_i\psi_i(x) \right\rangle_{\mathcal{H}_K} = \sum_{i=1}^{M} \frac{a_i b_i}{\lambda_i} \tag{1.9}$$

et la norme dans $\mathcal{H}_K$ est définie par

$$\|f\|^2_{\mathcal{H}_K} =< f(x), f(x) >_{\mathcal{H}_K} = \sum_{i=1}^{M} \frac{a_i^2}{\lambda_i} \tag{1.10}$$

De l'équation (1.7), on peut voir que $k(x, y)$ correspond à l'évaluation d'un produit scalaire dans le RKHS $\mathcal{H}_K$ induit par k. Cette constatation a de grandes conséquences pratiques que nous explicitons au travers de la proposition 1.

Proposition 1 *«L'astuce du noyau»(«Kernel Trick»)*
Tout modèle ne nécessitant dans sa construction que la manipulation de produits scalaires entre observations (et non leur cordonnées explicites) peut être construit implicitement dans un espace de Hilbert en remplaçant chaque produit scalaire par l'évaluation d'un noyau défini positif sur un espace quelconque. Ce procédé de substitution a pris le nom d' «astuce du noyau». □

Le tableau 1.1 présente trois fonctions noyaux fréquemment utilisées dont on sait qu'elles vérifient les hypothèses du théorème de Mercer.

TAB. 1.1: Fonctions noyaux vérifiant les conditions du théorème de Mercer

Noyau linéaire :	$k(x, y) =< x, y >$
Noyau polynomial :	$k(x, y) = (< x, y > +c)^d$
Noyau gaussien :	$k(x, y) = exp(-\gamma\|x - y\|^2)$

Nous disposons maintenant des outils qui vont nous permettre de décrire la mise en oeuvre pratique du SRM proposée par Evgeniou et al. [Evgeniou et al. (2000)] et Wahba [Wahba (1999)].

1.3 RKHS et Minimisation du Risque Structurel (SRM)

Nous présentons, dans cette section, une stratégie de mise en oeuvre du SRM établie par Evgeniou et al. [Evgeniou et al. (2000)] et Wahba [Wahba (1999)]. Soit un RKHS $\mathcal{H}_K$ muni

d'une norme $\|.\|_{\mathcal{H}_K}$ et d'un produit scalaire associé $< .,. >_{\mathcal{H}_K}$. Soient une séquence de réels $a_1 \leq a_2 \leq ... \leq a_M$ et $\mathcal{F}_j = \{f \in \mathcal{H}_K : \|f\|_{\mathcal{H}_K} \leq a_j\}$. On a alors :

$$\mathcal{F}_1 \subset \mathcal{F}_2 \subset ... \subset \mathcal{F}_M \tag{1.11}$$

Par construction, la VC dimension de $\mathcal{F}_j$ est une fonction croissante des a_j et Evgeniou et al. [Evgeniou et al. (2000)] suggèrent de minimiser le risque empirique sur chaque $\{\mathcal{F}_j\}_{j=1}^M$. Cela revient à résoudre le problème (1.12) de minimisation d'Ivanov [Ivanov (1962)] pour chaque $j = 1, ..., M$.

$$\begin{array}{l} \underset{f \in \mathcal{H}_K}{min} \ \frac{1}{n} \sum_{i=1}^n V(y_i, f(x_i)) \\ \text{sous la contrainte : } \|f\|^2_{\mathcal{H}_K} \leq a_j^2 \end{array} \tag{1.12}$$

Le problème d'optimisation (1.12) peut-être résolu à l'aide des multiplicateurs de Lagrange conduisant à la minimisation par rapport f et maximisation par rapport aux multiplicateurs de Lagrange $\lambda_j \geq 0$ du problème (1.13)

$$\frac{1}{n} \sum_{i=1}^n V(y_i, f(x_i)) + \lambda_j(\|f\|^2_{\mathcal{H}_K} - a_j^2) \quad \forall j = 1, ..., M \tag{1.13}$$

Pour chaque $\mathcal{F}_j$, $j = 1, ..., M$ est obtenu un couple solution (f_j^*, λ_j^*). La solution optimale $\hat{f}_n$ extraite de toutes les solutions $\{f_j^*\}_{j=1}^M$ doit fournir un compromis entre minimisation du risque empirique et complexité de $\mathcal{F}_j$; c'est-à-dire fournir une faible valeur de la partie droite de l'inégalité (1.3). Ainsi, en proposant une stratégie de contrôle de la valeur de la VC dimension de chacun des $h_{\mathcal{F}_k}$, il n'est plus nécessaire de la calculer. Nous venons donc de contourner la première difficulté évoquée précédemment. Plutôt que de résoudre ces M programmes d'optimisation sous contrainte, Evgeniou et al. [Evgeniou et al. (2000)] proposent de rechercher le minimum de (1.14) qui correspond au problème de régularisation de Tikhonov [Tikhonov and Arsenin (1977)].

$$\frac{1}{n} \sum_{i=1}^n V(y_i, f(x_i)) + \lambda \|f\|^2_{\mathcal{H}_K} \tag{1.14}$$

Remarque 3 *Les problèmes d'optimisation d'Ivanov et de Tikhonov sont équivalents dans le sens où pour λ fixé, si f_0 est solution de (1.14) alors f_0 est également solution de (1.12).* □

(1.14) établit un compromis entre complexité (ou régularité) de f (appartenant à un espace de Hilbert à noyau reproduisant) et minimisation du risque empirique. Ce compromis étant contrôlé par le paramètre de régularisation λ. En pratique, la résolution de (1.14) s'effectue par λ-validation croisée. Reste enfin à proposer une stratégie efficace de résolution du problème d'optimisation (1.14). Pour ce faire, nous allons introduire le résultat fondamental de la théorie de l'apprentissage dans les espaces de Hilbert à noyau reproduisant : le «théorème du représentant» ou théorème de Kimeldorf et Wahba [Kimeldorf and Wahba (1971)] .

Théorème 2 *Théorème de Kimeldorf et Wahba*
Soient un ensemble compact $\mathcal{X} \in \mathbb{R}^p$, k un noyau défini positif, $\mathcal{H}_K$ le RKHS associé, $D = \{x_1, ..., x_n\} \in \mathcal{X}$ un ensemble fini d'individus, y_i les sorties observées pour chacun des n individus et V une fonction de coût mesurant la concordance entre y_i et $f(x_i)$. Toute solution au problème

$$\underset{f \in \mathcal{H}_K}{min} \frac{1}{n} \sum_{i=1}^n V(y_i, f(x_i)) + \lambda \|f\|^2_{\mathcal{H}_K} \tag{1.15}$$

admet une représentation de la forme :

$$\forall x \in \mathcal{X}, \quad \hat{f}_n(x) = \sum_{i=1}^{n} \alpha_i k(x_i, x) \tag{1.16}$$

où $\{\alpha_i\}_{i=1}^n \in \mathbb{R}$ □

Le théorème de Kimeldorf et Wahba implique que la norme $\|\hat{f}_n\|_{\mathcal{H}_K}$ de la solution $\hat{f}_n$ de l'équation (1.15) est contrainte à de faibles valeurs, ce qui peut être bénéfique si on recherche des fonctions régulières. Ainsi, dans cette formulation, λ est un réel positif établissant un compromis entre complexité de $f \in \mathcal{H}_K$ et qualité d'approximation.

Notons que la solution $\hat{f}_n$, fournie par le théorème du représentant, évolue dans un sous-espace de dimension n. Ainsi, le problème consiste à estimer n paramètres $\{\alpha_i\}_{i=1}^n$ bien que $\mathcal{H}_K$ puisse être de dimension infinie.

1.4 Mise en oeuvre du SRM : la régularisation

La mise en oeuvre pratique du SRM se résume finalement au choix de la fonction de coût intervenant dans l'équation (1.15). Nous étudions dans cette section, l'impact de ce choix sur la solution finale.

1.4.1 La Kernel Ridge Regression : la Square loss

Dans un premier temps, nous présentons la Ridge Regression comme une alternative à la méthode des moindres carrés lorsque les données sont mal conditionnées.

L'estimateur des moindres carrés noté $\hat{\beta}^{OLS}$ est défini par

$$\hat{\beta}^{OLS} = \underset{\beta}{\operatorname{argmin}} \|y - X\beta\|_2^2 \tag{1.17}$$

et s'obtient en résolvant l'équation normale :

$$\hat{\beta}^{OLS} = (X^T X)^{-1} X^T y \tag{1.18}$$

Dans le cas de p variables, les éléments diagonaux de $C = (X^T X)^{-1}$ sont définis par :

$$C_{jj} = \frac{1}{1 - R_j^2}, \qquad j = 1, ..., p$$

où R_j^2 est le carré du coefficient de corrélation multiple de la variable x_j avec les $p-1$ autres variables explicatives. S'il existe une forte multicollinéarité entre x_j et les $p-1$ variables, alors $R_j^2 \to 1$, ce qui implique instabilité et explosion des valeurs de $\hat{\beta}^{OLS}$.
Classiquement, on mesure la précision d'un estimateur $\hat{\beta}$ de β par son erreur quadratique moyenne (MSE : Mean Squares Error) définie par (1.19) :

$$\begin{aligned} MSE(\hat{\beta}) &= \mathbb{E}\left[(\hat{\beta} - \beta)^T (\hat{\beta} - \beta)\right] \\ &= \underbrace{\left(\mathbb{E}[\hat{\beta}] - \beta\right)^T \left(\mathbb{E}[\hat{\beta}] - \beta\right)}_{\text{biais}^2 \text{ de } \hat{\beta}} + \underbrace{\mathbb{E}\left[(\hat{\beta} - \mathbb{E}[\hat{\beta}])^T (\hat{\beta} - \mathbb{E}[\hat{\beta}])\right]}_{\text{variance de } \hat{\beta}} \end{aligned} \tag{1.19}$$

Le MSE se décompose donc en deux termes. Le premier terme correspond au carré du biais de l'estimateur tandis que le second terme correspond au terme de variance de l'estimateur. D'après le théorème de Gauss-Markov, l'estimateur des moindres carrés, $\hat{\beta}^{OLS}$, est de tous les estimateurs sans biais celui de variance minimale. Il s'ensuit qu'on ne peut diminuer la variance de l'estimateur qu'au détriment d'estimateur biaisé. C'est l'objectif central des estimateurs régularisés.

L'erreur quadratique moyenne de $\hat{\beta}^{OLS}$ est définie par :

$$MSE(\hat{\beta}^{OLS})=\sum_{i=1}^{p}\mathbb{E}\left[\left(\hat{\beta}_i-\beta_i\right)^2\right]=\sum_{i=1}^{p}\text{var}\left(\hat{\beta}_i\right)=\sigma^2\text{trace}(X^TX)^{-1}=\sigma^2\sum_{i=1}^{p}1/\lambda_i$$

Si la matrice X^TX est mal conditionnée, au moins une des valeurs propres $\{\lambda_i\}_{i=1}^{p}$ sera petite impliquant alors un large MSE. X^TX est mal conditionnée soit parce que les variables explicatives sont fortement corrélées soit parce que le nombre de variables excède le nombre d'observations. Dans de telles situations, plutôt que de chercher l'estimateur de norme minimale solution de l'équation normale, on peut chercher un compromis entre minimisation de la fonction de coût et faible valeur de la norme $\|\beta\|$: cette approche est connue sous le nom de *Ridge Regression* [Hoerl and Kennard (1970)]. L'estimateur de la ridge regression est solution du problème d'optimisation (1.20)

$$\hat{\beta}^{RR}=\underset{\beta\in\mathbb{R}^p}{\text{argmin}}(y-X\beta)^T(y-X\beta)+\lambda\|\beta\|^2 \tag{1.20}$$

où λ est une constante positive à valeur réelle pondérant le compromis entre faible norme et minimisation de la fonction de coût.
En annulant la dérivée par rapport à β on obtient :

$$X^TX\hat{\beta}^{RR}+\lambda\hat{\beta}^{RR}=(X^TX+\lambda\mathbb{I}_p)\hat{\beta}^{RR}=X^Ty \tag{1.21}$$

Où $\mathbb{I}_p$ est la matrice identité de dimension p.
La solution est fournie par l'équation (1.22) :

$$\hat{\beta}^{RR}=(X^TX+\lambda\mathbb{I}_p)^{-1}X^Ty \tag{1.22}$$

Notons que $(X^TX+\lambda\mathbb{I}_p)$ est systématiquement inversible lorsque $\lambda>0$. Le modèle final est de la forme (1.23) :

$$\hat{f}_n(x)=\left\langle\hat{\beta}^{RR},x\right\rangle=y^TX(X^TX+\lambda\mathbb{I}_p)^{-1}x \tag{1.23}$$

Cette méthode est conceptuellement intéressante car elle permet de gérer les données corrélées. Cependant, la solution, $\hat{\beta}^{RR}$ est obtenue en inversant une matrice de dimension $p\times p$ rendant pour le moment sa mise en oeuvre difficile dans les espaces de grande dimension. Il est possible de contourner cette difficulté en remarquant que $\hat{\beta}^{RR}$ peut s'exprimer sous sa représentation dite duale, c'est-à-dire, non plus comme combinaison linéaires des variables, mais des observations [Saunders et al. (1998)].

$$\hat{\beta}^{RR}=\lambda^{-1}X^T(y-X\hat{\beta}^{RR})=X^T\alpha$$

où $\hat{\alpha}=\lambda^{-1}(y-X\hat{\beta}^{RR})$. Par conséquent, nous obtenons :

$$\begin{aligned}&\Rightarrow\lambda\hat{\alpha}=(y-XX^T\alpha)\\&\Rightarrow(XX^T+\lambda\mathbb{I}_n)\hat{\alpha}=y\\&\Rightarrow\hat{\alpha}=(XX^T+\lambda\mathbb{I}_n)^{-1}y\end{aligned}$$

Le modèle final est de la forme (1.24) :

$$\hat{f}_n(x) = \left\langle \hat{\beta}^{RR}, x \right\rangle = \left\langle \sum_{i=1}^{n} \hat{\alpha}_i x_i, x \right\rangle = \sum_{i=1}^{n} \hat{\alpha}_i \langle x_i, x \rangle \tag{1.24}$$

Le modèle (1.23) est évidemment équivalent au modèle (1.24).

Soulignons quelques propriétés fondamentales :

1. L'expression duale de $\hat{\beta}^{RR}$ ne s'exprime qu'au travers de produits scalaires entre observations. De la même manière, par le modèle (1.24), la prédiction d'un nouvel individu x ne nécessite que l'information fournie par le produit scalaire entre x et chacun des x_i. Ainsi, il est possible d'exprimer la solution de la ridge regression uniquement par les valeurs des produits scalaires entre individus. Cette propriété est d'une importance capitale.

2. La solution $\hat{\alpha}$ nécessite l'inversion de la matrice $(XX^T + \lambda \mathbb{I}_n)$ de dimension $n \times n$. Ceci est particulièrement intéressant lorsque $p >> n$.

Kernel Ridge Regression à travers l'astuce du noyau

Tel que présentées précédement, les relations entre la variable à expliquer y et les variables explicatives sont linéaires. Ces relations peuvent s'avérer insuffisantes lorsque l'on cherche à maximiser le pouvoir prédictif d'un modèle. On peut alors s'interroger sur la possibilité d'étendre la ridge regression au cadre non linéaire. Pour ce faire, nous allons nous placer dans le contexte des Espaces de Hilbert à Noyau Reproduisant. Puisque la solution (1.24) ne s'exprime que par le produit scalaire entre observations, on peut alors appliquer l'astuce du noyau au contexte de la ridge regression. Notons donc $K = (k(x_i, x_j))_{\substack{i=1,\dots,n \\ j=1,\dots,n}}$, la matrice de Gram associée à la fonction noyau k. La matrice K de dimension $n \times n$ fournit les valeurs des produits scalaires des n individus dans le RKHS $\mathcal{H}_K$ (que l'on ne connaît pas explicitement). Dans cet espace, $\hat{\alpha}$ est défini par l'équation (1.25).

$$\hat{\alpha} = (K + \lambda n I)^{-1} y \tag{1.25}$$

et la solution finale est fournie par l'équation (1.26)

$$\hat{f}_n(x) = \sum_{i=1}^{n} \hat{\alpha}_i k(x, x_i) \tag{1.26}$$

Kernel Ridge Regression à travers le théorème de Kimeldorf et Wahba

On peut également retrouver la solution de la Kernel Ridge Regression obtenue par l'équation (1.26) en utilisant le théorème de Kimeldorf et Wahba. Considérons le problème (1.27)

$$\hat{f}_n \quad = \quad \underset{f \in \mathcal{H}_K}{\operatorname{argmin}} \left[\tfrac{1}{n} \textstyle\sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda \|f\|^2_{\mathcal{H}_K} \right] \tag{1.27}$$

où $\mathcal{H} = \mathcal{H}_K$ défini, un RKHS associé à une fonction noyau k sur $\mathcal{X} \times \mathcal{X}$. Par le théorème du représentant, toute solution de (1.27) peut s'écrire sous la forme :

$$\hat{f}_n(x) \quad = \quad \textstyle\sum_{i=1}^{n} \alpha_i k(x_i, x) \tag{1.28}$$

Remarquons que le problème (1.27) peut s'exprimer matriciellement. En effet, soient $\alpha = (\alpha_1, ..., \alpha_n)^T \in \mathbb{R}^n$, K la matrice de Gram $(n \times n)$: $K_{ij} = k(x_i, x_j)$.

On a alors :

$$(\hat{f}_n(x_1), ..., \hat{f}_n(x_n))^T = K\alpha \tag{1.29}$$

et

$$\|\hat{f}_n\|^2_{\mathcal{H}_K} = \alpha^T K\alpha \tag{1.30}$$

Le problème (1.27) est donc équivalent à :

$$\hat{f}_n = \underset{\alpha\in\mathbb{R}^n}{argmin} \tfrac{1}{n}(K\alpha - y)^T(K\alpha - y) + \lambda\alpha^T K\alpha \tag{1.31}$$

C'est une fonction convexe et différentiable en α. Afin d'obtenir la solution de (1.31), il convient de calculer sa dérivée par rapport à α. Soit $F(\alpha) = \frac{1}{n}\sum_{i=1}^{n}(K\alpha - y)^T(K\alpha - y) + \lambda\alpha^T K\alpha$

$$\begin{aligned} \nabla F_\alpha &= \tfrac{2}{n}K(K\alpha - y) + 2\lambda K\alpha \\ &= \tfrac{2}{n}K^2\alpha - \tfrac{2}{n}Ky + 2\lambda K\alpha \end{aligned} \tag{1.32}$$

La matrice de Gram K étant symétrique définie positive, $\nabla F_\alpha = 0$ est une condition nécessaire et suffisante pour le calcul de α optimal. En multipliant par K^{-1}, la solution optimale est obtenue en résolvant le système linéaire (1.33)

$$(K + \lambda nI)\alpha = y \tag{1.33}$$

Et $\hat{\alpha}$ est défini par (1.34)

$$\hat{\alpha} = (K + \lambda nI)^{-1}y \tag{1.34}$$

Notons que $(K + \lambda nI)$ est systématiquement inversible quand $\lambda > 0$. On retrouve alors l'équivalence souhaitée en posant $K = XX^T$ (noyau de Mercer linéaire).

Dans cette section nous considérions le cas où $V = (y - f(x))^2$. Qu'en est-il en changeant la fonction de coût V ?

1.4.2 Les Support Vector Machines (SVM) : la Hinge loss

Cette section est dédiée à l'application des concepts de régularisation au contexte exclusif de la classification binaire $y \in \{-1, 1\}$. L'approche que propose Boser et al. [Boser et al. (1992)] et Vapnik [Vapnik (1995)] dans le contexte de la classification binaire repose sur la notion (intuitive et géométrique) d'hyperplan séparateur optimal. C'est le principe fondateur de la méthode bien connue des Support Vector Machines (SVM). Alors qu'à priori lointains de toute forme de régularisation de Tikhonov, les SVM se traduisent en fait comme un problème de la forme (1.15) au travers d'une fonction de coût V particulière : la Hinge loss. C'est ce que nous présentons dans cette section.
Commençons, néanmoins, par décrire les principes qui ont construit la notoriété des SVM. La motivation principale des SVM repose sur la généralité de la fonction de décision. Comment construire à partir d'une base d'apprentissage finie une fonction de décision capable de prédire la classe d'appartenance d'un nouvel individu (qui n'a pas servi à la construction du modèle). Initialement, la solution proposée par Vapnik pour éviter les problèmes liés au sur-apprentissage repose sur le concept de «marge» ou d' «hyperplan séparateur optimal». Il s'agit de construire un hyperplan tel que la distance minimale des observations à ce dernier soit maximale.

SVM linéaire et hyperplan séparateur optimal

Dans cette section, nous abordons les SVM de manière géométrique afin de fournir une approche pragmatique pour appréhender les principes de contrôle de complexité sous-jacents aux principes d'hyperplan séparateur optimal. Remarquons que la marge autour d'une séparatrice linéaire est associée à la richesse de l'espace d'hypothèses (donc à la VC-dimension h). En effet, plus la marge est large, plus il est difficile de faire passer une séparatrice linéaire avec cette marge entre les données des deux classes. Intuitivement, en cherchant une séparatrice de marge maximale, entre les données, nous nous obligeons à considérer un espace d'hypothèses plus limité et cela nous permet de maintenir un lien entre le risque empirique et le risque réel. Cela correspond à la justification avancée par Vapnik pour les SVM. Selon lui, cette méthode permettrait de régler directement le compromis entre adéquation aux données et limitation de la richesse de l'espace des hypothèses.

Soient n observations décrites par p variables réparties en deux classes. À chaque observation x_i est donc associée un label $y_i \in \{-1, 1\}$. Un hyperplan est défini par son orthogonalité à un vecteur unitaire β et sa distance à l'origine $-\beta_0$. La distance entre un individu x de la première classe et l'hyperplan $\{x : f(x) = x^T\beta + \beta_0 = 0\}$ où $\|\beta\| = 1$ est égale à

$$x^T\beta - (-\beta_0) = x^T\beta + \beta_0$$

De même, la distance entre un point x de la deuxième classe et l'hyperplan est égale à

$$-\beta_0 - x^T\beta$$

Supposons que les classes soient linéairement séparables. Alors il existe une infinité d'hyperplans tel que $y_i f(x_i) > 0$. L' «hyperplan séparateur optimal» (Figure 1.2) est défini par les vecteurs de poids β et β_0 vérifiant l'équation (1.35).

$$\arg\max_{\beta,\beta_0} \min\{\|x - x_i\|_2 : x \in \mathbb{R}^n, (x^T\beta + \beta_0) = 0, i = 1, ..., n\} \tag{1.35}$$

La figure 1.2 représente les individus de la première classe en bleu (carré) ($y = 1$) et ceux de la seconde en rouge (cercle) ($y = -1$).
Par conséquent $f(x) = x^T\beta + \beta_0$ représente la distance signée de x à l'hyperplan : positive pour les bleus et négative pour les rouges. Dans le cas séparable (figure 1.2) le produit $y_i f(x_i)$ est positif quelque soit l'observation. L'hyperplan séparateur optimal maximise donc les distances signées entre les points et l'hyperplan. On cherche donc à maximiser l'épaisseur C de la marge par rapport à β et β_0 sous les contraintes suivantes :

$$\|\beta\| \quad = \quad 1 \tag{1.36}$$

x_i à droite de H_1 pour les i correspondant à $y_i = 1$,

$$\begin{array}{lll} \Leftrightarrow & x_i^T\beta - (-\beta_0 + C) & > 0 \\ \Leftrightarrow & -(x_i^T\beta + \beta_0) & > C \ \forall i \text{ tel que } y_i = 1 \end{array} \tag{1.37}$$

x_i à gauche de H_2 pour les i correspondant à $y_i = -1$,

$$\begin{array}{lll} \Leftrightarrow & -x_i^T\beta - (\beta_0 + C) & > 0 \\ \Leftrightarrow & -(x_i^T\beta + \beta_0) & > C \ \forall i \text{ tel que } y_i = -1 \end{array} \tag{1.38}$$

Les contraintes (1.37) et (1.38) se résument en une seule :

$$y_i(x_i^T\beta + \beta_0) \quad > \quad C \tag{1.39}$$

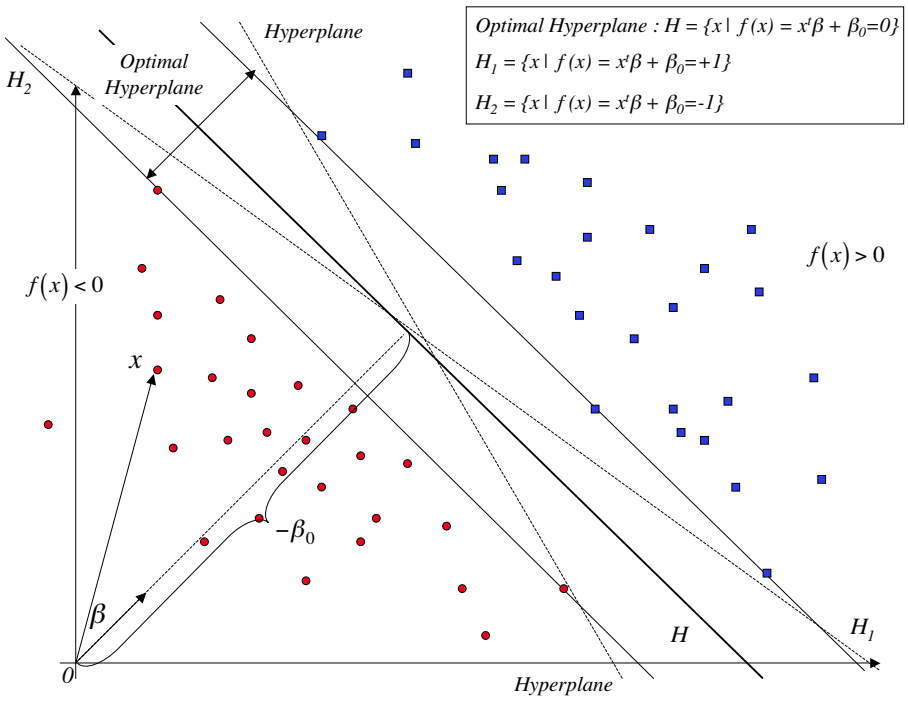

FIG. 1.2: Approche géométrique des Support Vecteur Machines. Représentation de l'Hyperplan Séparateur Optimal dans le cas de données linéairement séparables.

Si l'on souhaite s'affranchir de la contrainte de norme sur β, on peut remplacer β par $\beta/\|\beta\|$. Et (1.39) devient

$$\begin{aligned} \tfrac{1}{\|\beta\|} y_i(x_i^T\beta + \beta_0) &> C \\ \Leftrightarrow y_i(x_i^T\beta + \beta_0) &> \|\beta\| C \end{aligned} \tag{1.40}$$

Remarquons que β_0 est automatiquement redéfini : si le couple (β, β_0) vérifie l'inégalité (1.40) alors $\forall \alpha > 0$, le couple $(\alpha\beta, \alpha\beta_0)$ vérifie également (1.40). On peut donc choisir arbitrairement $\|\beta\| = 1/C$. Ainsi, maximiser C revient à minimiser $\|\beta\|$. Par conséquent, dans le cas linéairement séparable, la recherche de l'hyperplan optimal revient donc à résoudre le problème d'optimisation quadratique (1.41).

$$\min_{\beta,\beta_0} \tfrac{1}{2}\|\beta\|^2 \quad \text{sous la contrainte} \quad y_i(x_i^T\beta + \beta_0) > 1 \ \forall i = 1, ..., n \tag{1.41}$$

Si les données ne sont pas linéairement séparables, l'hyperplan séparateur optimal séparant les deux classes est celui qui sépare les données avec le minimum d'erreurs tout en maintenant une largeur de marge «raisonnable». L'hyperplan doit donc établir un compromis entre taux d'erreur et largeur de marge. Les contraintes doivent donc être relachées de façon douce en introduisant des «variables ressorts» ξ_i, $i = 1, ..., n$ permettant aux contraintes de marges d'être violées. La figure 1.3 illustre le cas de données non linéairement séparables. Dans ce cas, le problème d'optimisation (1.41) devient (1.42)

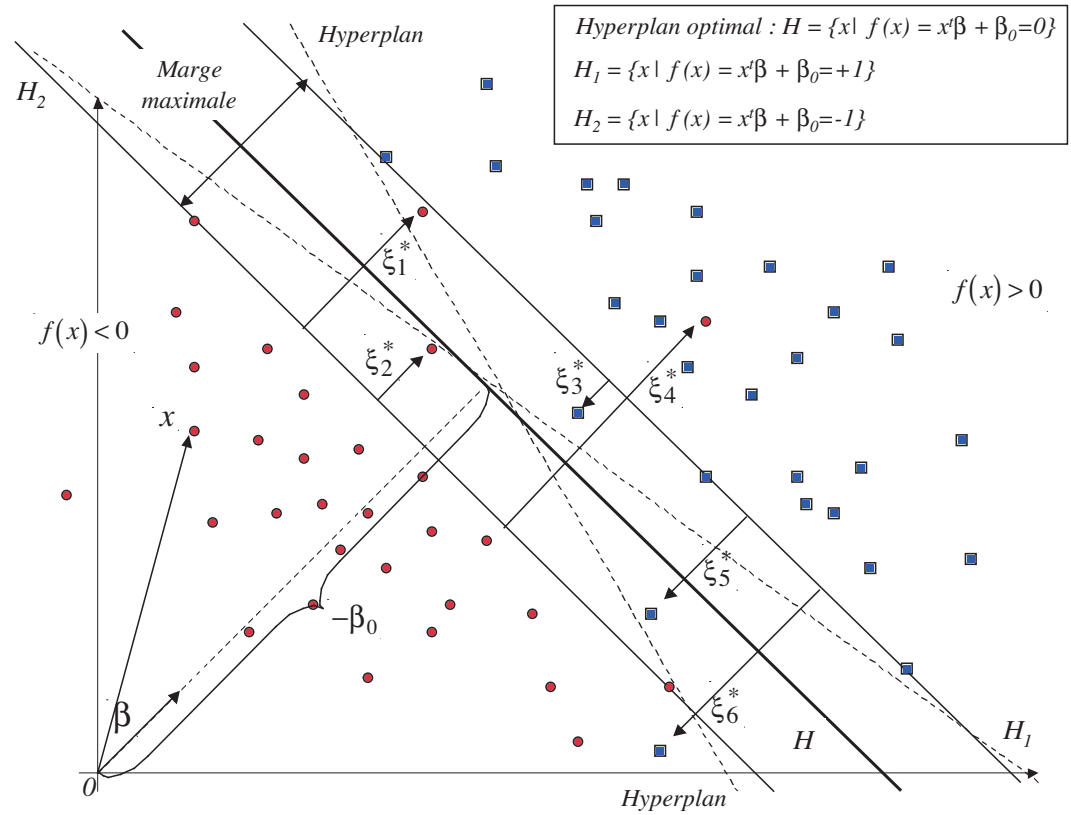

FIG. 1.3: Approche géométrique des Support Vecteur Machines. Représentation de l'Hyperplan Séparateur Optimal dans le cas de données non linéairement séparables.

$$\text{sous les contraintes} \quad \begin{array}{l} \min\limits_{\beta,\beta_0,\xi_i} \ \frac{1}{2}\|\beta\|^2 + \gamma \sum_{i=1}^{n} \xi_i \\ \left\{ \begin{array}{l} y_i(x_i^T\beta + \beta_0) \geq 1 - \xi_i, \forall i = 1, ..., n \\ \xi_i \geq 0 \ , \forall i = 1, ..., n \end{array} \right. \end{array} \tag{1.42}$$

où γ est le paramètre de régularisation (à fixer par l'utilisateur) permettant d'établir le compromis entre largeur de marge et taux d'erreur. Ce paramètre est généralement fixé par validation croisée et est à valeur dans $\mathbb{R}^+$.
La solution du problème (1.42) s'obtient en maximisant la forme duale du Lagrangien :

$$\left\{ \begin{array}{l} \max\limits_{\alpha} \{\sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j < x_i, x_j >\} \\ \gamma \geq \alpha_i \geq 0, \ i = 1, ..., n \\ \sum_{i=1}^{n} \alpha_i y_i = 0 \end{array} \right. \tag{1.43}$$

L'équation de l'hyperplan séparateur optimal s'écrit alors :

$$\hat{f}_n(x) \quad = \quad \sum_{i=1}^{n} \hat{\alpha}_i y_i < x, x_i > + \hat{\beta}_0 = 0 \tag{1.44}$$

où les coefficients $\hat{\alpha}_i$ sont obtenus par résolution du problème (1.43). Notons que seuls les x_i pour lesquelles les multiplicateurs de Lagrange $\hat{\alpha}_i$ sont non-nuls participent à la solution : ces

individus sont appelés les Support Vectors (d'où le nom de Support Vector Machines). Ajoutons enfin que $\hat{\beta}_0$ est obtenu indépendamment en résolvant l'équation :

$$\hat{\beta}_0 = -\frac{1}{2}\left[\sum_{i=1}^{n} \hat{\alpha_i} y_i < x_{sv_+}, x_i > + \sum_{i=1}^{n} \hat{\alpha_i} y_i < x_{sv_-}, x_i >\right]$$

où x_{sv_+} (respectivement x_{sv_-}) correspond à n'importe quel support vector étiqueté $+1$ (respectivement -1).

le signe de $\hat{f}_n(x)$ code la classe de l'individu x.

SVM non linéaire à travers l'astuce du noyau

Les SVM consistent à rechercher un hyperplan séparateur optimal. La fonction de décision est une séparation linéaire. Afin de considérer des frontières de classification non linéaires, l'astuce du noyau (explicité à travers le théorème de Mercer et la proposition 1) est exploitable puisque les individus n'interviennent que par leur produit scalaire (cf. (1.43)). On recherche désormais un hyperplan séparateur optimal, mais la recherche de cet hyperplan s'effectue dans le RKHS, $\mathcal{H}_K$ associé à la fonction noyau k. En utilisant comme précédemment l'astuce du noyau, le problème d'optimisation (1.43) est équivalent au problème (1.45) :

$$\begin{cases} \max\limits_{\alpha} \{\sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j k(x_i, x_j)\} \\ \gamma \geq \alpha_i \geq 0, \ i = 1, ..., n \\ \sum_{i=1}^{n} \alpha_i y_i = 0 \end{cases} \tag{1.45}$$

dont la solution est l'hyperplan séparateur d'équation :

$$\hat{f}_n(x) = \sum_{i=1}^{n} \hat{\alpha_i} y_i k(x, x_i) + \hat{\beta}_0 = 0 \tag{1.46}$$

où les coefficients $\hat{\alpha}_i$ et le $\hat{\beta}_0$ sont obtenus comme précédemment.

Remarque 4 *La formulation des SVM (1.45) est à la base des implémentations pratiques. Un des problèmes inhérents aux méthodes à noyau et que l'on retrouve également pour les SVM réside dans la gestion de la matrice de Gram K de dimension $n \times n$. Lorsque le nombre d'observations est trop important, on ne peut pas stocker K en mémoire et le problème devient intractable. Des implémentations ingénieuses des SVM permettent de contourner le problème. En partitionnant les individus par sous-blocs et en ne considérant ces sous-blocs que successivement, il est possible d'obtenir la solution des SVM. Citons les implémentations efiicaces des SVM fournies par Joachims [Joachims (1999)], Osuna [Osuna et al. (1997)], Platt [Platt (1999)] et Rifkin [Rifkin (2002)].* □

SVM à travers le théorème de Kimeldorf et Wahba

À l'instar de la Kernel Ridge Regression, il est possible de retrouver la solution (1.46) des SVM par l'utilisation du théorème de Kimeldorf et Wahba. Pour cela, considérons une fonction de coût particulière, la Hinge loss :

$$V(y, f(x)) = [1 - yf(x)]_+ = \max(0, 1 - yf(x)) \tag{1.47}$$

Nous pouvons fournir une interprétation géométrique de la Hinge loss. Nous avons vu que $y_i f(x_i)$ fournit la distance de l'individu x_i à l'hyperplan d'équation $f(x) = 0$. Pour que la fonction de coût $[1 - y_i f(x_i)]_+$ soit faible, il suffit de maximiser les valeurs de $y_i f(x_i)$ pour chaque x_i, et donc de construire un hyperplan tel que la distance minimale des individus soit maximale. Nous retrouvons ainsi le principe fondateur des SVM décrit en début de section.
Ainsi, pour obtenir la solution fournie par l'équation (1.46), il suffit d'estimer une fonction f qui minimise la fonctionnelle (1.48). L'équivalence entre la formulation (1.45) des SVM à variables ressorts et la formulation (1.48) a été intialement constaté par Wahba [Wahba (1999)] et Evgeniou et al. [Evgeniou et al. (2000)].

$$\min_{f \in \mathcal{H}_K} \frac{1}{n} \sum_{i=1}^{n} [1 - y_i f(x_i)]_+ + \lambda \|f\|^2_{\mathcal{H}_K} \tag{1.48}$$

où $\mathcal{H}_K$ est le RKHS généré par la fonction noyau $k(.,.)$ et λ définit le paramètre de régularisation permettant d'établir le compromis entre fidélité aux données et régularité de la fonction solution $\hat{f}_n$. La Hinge loss est représentée dans la figure 1.4.
Par le théorème du représentant, on sait que la solution $\hat{f}_n$ s'exprime sous la forme

$$\hat{f}_n(x) = \sum_{i=1}^{n} \alpha_i k(x_i, x) \tag{1.49}$$

Comme l'illustre la figure 1.4, la Hinge loss est convexe mais non différentiable. On ne peut donc pas résoudre le problème par dérivation comme dans le cadre de la Kernel Ridge Regression. Nous allons donc introduire des variables ressorts $\xi = \{\xi_1, \xi_2, ..., \xi_n\}$, afin de faciliter la résolution problème (1.48). Considérons donc le problème d'optimisation (1.50) :

$$\begin{aligned} &\min_{f \in \mathcal{H}_K, \xi \in \mathbb{R}^n} \tfrac{1}{n} \textstyle\sum_{i=1}^{n} \xi_i + \lambda \|f\|^2_{\mathcal{H}_K} \\ \text{sous les contraintes,} \quad &\\ &\xi_i \geq [1 - y_i f(x_i)]_+, \text{pour } i = 1, ..., n \end{aligned} \tag{1.50}$$

Ou de manière équivalente le problème d'optimisation (1.51)

$$\begin{aligned} &\min_{f \in \mathcal{H}_K, \xi \in \mathbb{R}^n} \tfrac{1}{n} \textstyle\sum_{i=1}^{n} \xi_i + \lambda \|f\|^2_{\mathcal{H}_K} \\ \text{sous les contraintes,} \quad &\\ &\begin{cases} y_i f(x_i) \geq 1 - \xi_i, \text{pour } i = 1, ..., n \\ \xi_i \geq 0, \text{pour } i = 1, ..., n \end{cases} \end{aligned} \tag{1.51}$$

Remarque 5 *Nous constatons alors d'étroites similarités entre les équations (1.42) et (1.51).* □

En utilisant la forme de $\hat{f}_n$ fournie par le théorème du représentant, on obtient le problème de minimisation en α (1.52)

$$\begin{aligned} &\min_{\alpha \in \mathbb{R}^n, \xi \in \mathbb{R}^n} \tfrac{1}{n} \textstyle\sum_{i=1}^{n} \xi_i + \lambda \alpha K \alpha \\ \text{sous les contraintes,} \quad &\\ &\begin{cases} y_i \sum_{j=1}^{n} \alpha_j k(x_i, x_j) + \xi_i - 1 \geq 0, \text{ pour } i = 1, ..., n \\ \xi_i \geq 0, \text{ pour } i = 1, ..., n \end{cases} \end{aligned} \tag{1.52}$$

C'est un problème d'optimisation quadratique sous contrainte linéaire que l'on peut résoudre à l'aide d'outil d'optimisation classique. En résolvant le problème d'optimisation (1.52), on

retrouve la solution (1.46).

Ainsi, à l'instar de la Kernel Ridge Regression, les SVM s'apparentent à un problème de régularisation de Tikhonov. Ils sont basés sur la minimisation du risque empirique régularisé (de fonction de perte linéaire par morceaux (la Hinge loss)) sur un espace fonctionnel de Hilbert (le RKHS). D'un point de vue plus pragmatique, la Hinge loss conduit à l'obtention d'un hyperplan séparateur optimal. Cette notion à l'interprétation claire et aux propriétés géométriques intéressantes joue un rôle central dans les derniers développements d'outils de classification car elle permet d'assurer un pouvoir de généralisation.

1.4.3 La Kernel Logistic Regression : la Logistic loss

Les SVM, dans leur forme initiale, estiment le signe de $p(x) - 1/2$ [Lin (2002)], où $p(x) = \mathbb{P}(y = 1/X = x)$. Ils ne fournissent donc pas la probabilité d'appartenance aux différentes classes de chacune des observations qui, dans de nombreux domaines, pourrait présenter un intérêt majeur (e.g. domaine médical). Une des approches les plus couramment utilisées pour prédire $p(x)$ est la régression logistique. C'est un modèle linéaire souffrant de problèmes de mauvais conditionnement lorsqu'il est confronté à des données corrélées et de grande dimension. Une configuration également mal gérée par la régression logistique est la séparation complète ou quasi-complète des deux classes d'individus Albert et Anderson [Albert and Anderson (1984)]. Nous allons, dans la prochaine section, expliciter les différents problèmes rencontrés par la régression logistique dans de telles situations et montrer que régulariser le problème fournit une alternative intéressante.

En régression logistique binaire, il est usuel de considérer les valeurs de la variable à expliquer y comme appartenant à l'ensemble $\{0, 1\}$. Posons $p(x_i) = \mathbb{P}(y = 1/x_i = x_{i1}, ..., x_{ip})$, le modèle de la régression logistique est de la forme (1.53) (transformation *logit*). Le lien *logit* a, apparamment, été suggéré initialement par Fisher [Fisher and Yates (1938)] pour établir des relations pour données binaires. Le modèle de la régression logistique s'écrit :

$$\begin{aligned} p(x_i) &= \frac{exp(\beta_0 + \beta_1 x_{i1} + ... + \beta_p x_{ip})}{1 + exp(\beta_0 + \beta_1 x_{i1} + ... + \beta_p x_{ip})} \\ 1 - p(x_i) &= \frac{1}{1 + exp(\beta_0 + \beta_1 x_{i1} + ... + \beta_p x_{ip})} \end{aligned}$$

où de manière équivalente :

$$log\left(\frac{p(x_i)}{1 - p(x_i)}\right) = \beta_0 + \beta_1 x_{i1} + ... + \beta_p x_{ip} \tag{1.53}$$

La probabilité d'observer la réponse y_i pour un individu x_i s'écrit de manière plus compacte :

$$\mathbb{P}(y_i/x_i) = p(x_i)^{y_i}(1 - p(x_i))^{1-y_i}$$

La vraisemblance des données représente la probabilité d'observer les n données (x_i, y_i). En supposant l'indépendance des individus, cette vraisemblance s'écrit :

$$l(\beta) = \prod_{i=1}^{n} p(x_i)^{y_i}(1 - p(x_i))^{1-y_i} \tag{1.54}$$

où $\beta = (\beta_0, \beta_1, ..., \beta_p)$. On recherche $\hat{\beta}$ maximisant la log-vraisemblance :

$$L(\beta) = log(l(\beta)) = \sum_{i=1}^{n} (y_i log[p(x_i)] + (1 - y_i) log[1 - p(x_i)]) \tag{1.55}$$

On obtient $\hat{\beta}$ en annulant les dérivées partielles :

$$\frac{\partial L}{\partial \beta_k} = \sum_{i=1}^{n} x_{ik}(y_i - p(x_i)) = 0,\ k = 0, ..., p$$

où $x_{i0} = 1$ pour tout $i = 1, ..., n$.
Soit V la matrice diagonale formée des $p(x_i)(1 - p(x_i))$. En notant que le Hessien H (matrice des dérivées secondes de la log-vraisemblance) défini par (1.56)

$$H(\beta) = \frac{\partial^2 L}{\partial \beta \partial \beta^T} = -X^T V X \tag{1.56}$$

est définie négative, on en conclut que la log-vraisemblance est une fonction concave et possède un maximum obtenu en annulant le vecteur score $U(\beta)$ formé des dérivées premières. Ce problème n'a pas de solution analytique et se résout en général par l'algorithme de Newton-Raphson.

La probabilité estimée pour l'observation $x_i = (x_{i1}, ..., x_{ip})$ est alors fournie par

$$\hat{p}(x_i) = \frac{exp(\hat{\beta}_0 + \hat{\beta}_1 x_{i1} + ... + \hat{\beta}_p x_{ip})}{1 + exp(\hat{\beta}_0 + \hat{\beta}_1 x_{i1} + ... + \hat{\beta}_p x_{ip})} \tag{1.57}$$

Et la matrice des covariances de $\hat{\beta}$ est estimée par :

$$\hat{cov}(\hat{\beta}) = [X^T \hat{V} X]^{-1}$$

où $\hat{V}$ est la matrice diagonale formée des $\hat{p}(x_i)(1 - \hat{p}(x_i))$.
Ainsi, la variance de l'estimateur du maximum de vraisemblance explose si les données sont corrélés ou que les éléments diagonaux de $\hat{V}$ sont tous proches de 0. Ce second phénomène apparaît dans le cas de complète ou quasi-complète séparation des deux classes d'individus (les probabilités estimées sont dans ce cas toutes très proches de 0 ou 1). Dans de telles situations, il peut être intéressant de régulariser le problème. En recodant la valeur de y par $\{-1, 1\}$, nous pouvons montrer que la log-vraisemblance se traduit par (1.58) :

$$-\sum_{i=1}^{n} \log(1 + e^{-y\beta^T x_i}) \tag{1.58}$$

où $x_i = [1\ x_{i1}\ ...\ x_{ip}]$, pour tout $i = 1, ..., n$.
Ainsi les paramètres de la régression logistique peuvent être obtenus en résolvant le problème d'optimisation (1.59)

$$\min_{\beta} \sum_{i=1}^{n} \log(1 + e^{-y_i \beta^T x_i}) \tag{1.59}$$

On peut alors pénaliser le problème en considérant (1.60)

$$\min_{\beta} \sum_{i=1}^{n} \log(1 + e^{-y_i x_i^T \beta}) + \lambda \|\beta\|^2 \tag{1.60}$$

Et enfin, considérer des solutions plus générales en étendant l'investigation de la solution $\hat{f}_n$ à un RKHS, $\mathcal{H}_K$, associé à une fonction noyau k. On déduit alors le sytème d'optimisation (1.61)

$$\underset{f\in\mathcal{H}_K}{min}\ \frac{1}{n}\sum_{i=1}^{n}\log(1+e^{-y_i f(x_i)})+\lambda\|f\|^2_{\mathcal{H}_K} \tag{1.61}$$

La fonction de coût $V(y, f(x)) = \log(1 + e^{-yf(x)})$ est appelée la *logistic loss*.

Le problème d'optimisation (1.61) est connu sous le nom de la Kernel Logistic Regression (KLR). La solution est fournie par le théorème de Kimeldorf et Wabha. Cette approche est donc particulièrement intéressante d'une part car elle donne une estimation naturelle de la probabilité d'appartenance aux différentes classes (alors que les SVM n'estiment que le signe de $p(x) - 1/2$) et, d'autre part, car de nombreux travaux proposent des implémentations ingénieuses du problème (1.61). Nous pouvons citer, par exemple, les travaux de [Zhu and Hastie (2005)] et [Keerthi et al. (2002)].

Les trois méthodes que nous avons présentées (KRR, SVM et KLR) dans cette partie s'apparentent à un problème de régularisation de Tikhonov et diffèrent uniquement au niveau du choix de la fonction de coût. La Hinge loss, relative aux SVM, présente l'intérêt de produire des modèles de marge maximale, ce qui pourrait assurer une meilleure capacité de généralisation. Ainsi, dans la prochaine section, nous allons nous intéresser au lien existant entre le choix de la fonction de coût et l'obtention d'un l'hyperplan séparateur optimal.

1.5 Hyperplan séparateur optimal et fonction de coût

La figure 1.4 représente l'allure des fonctions de coût associées à la régression logistique, à la ridge regression et aux Support Vector Machines. Nous constatons d'étroites similarités de comportement entre la *logistic loss* et la *hinge loss* laissant suggérer des performances équivalentes (le chapitre 5 de validation fournit des résultats comparatifs des SVM et de la KLR).

Nous pouvons donc supposer que les modèles générés au travers du problème général fournit par l'équation (1.14) et à partir d'une des deux fonctions coût ne diffère que peu, notamment sur la propriété d'hyperplan séparateur optimal.

Soit un ensemble de n observations décrites par p variables. Supposons les données linéairement séparables. Il existe alors un hyperplan séparateur optimal que l'on suppose unique. Intéressons nous au problème d'optimisation général (1.62)

$$\hat{\beta}(\lambda) = \arg\min_{\beta}\sum_{i=1}^{n}V(y_i, f(x_i))+\lambda\|\beta\|_2^2 \tag{1.62}$$

On retrouve, au travers de la Hinge loss la solution des SVM. Notons β^* cette solution. La question centrale de cette section est la suivante : pour quelle fonction de coût, $\hat{\beta}(\lambda)$ converge-t-il vers l'hyperplan séparateur optimal des SVM lorsque $\lambda \to 0$? Pour formaliser les choses, on cherche à déterminer les conditions sur l'allure de la fonction de coût pour que :

$$\underset{\lambda\to 0}{lim}\ \frac{\hat{\beta}(\lambda)}{\|\hat{\beta}(\lambda)\|} = \arg\max_{\|\beta\|=1}\min_{i}\ y_i x_i^T\beta = \beta^* \tag{1.63}$$

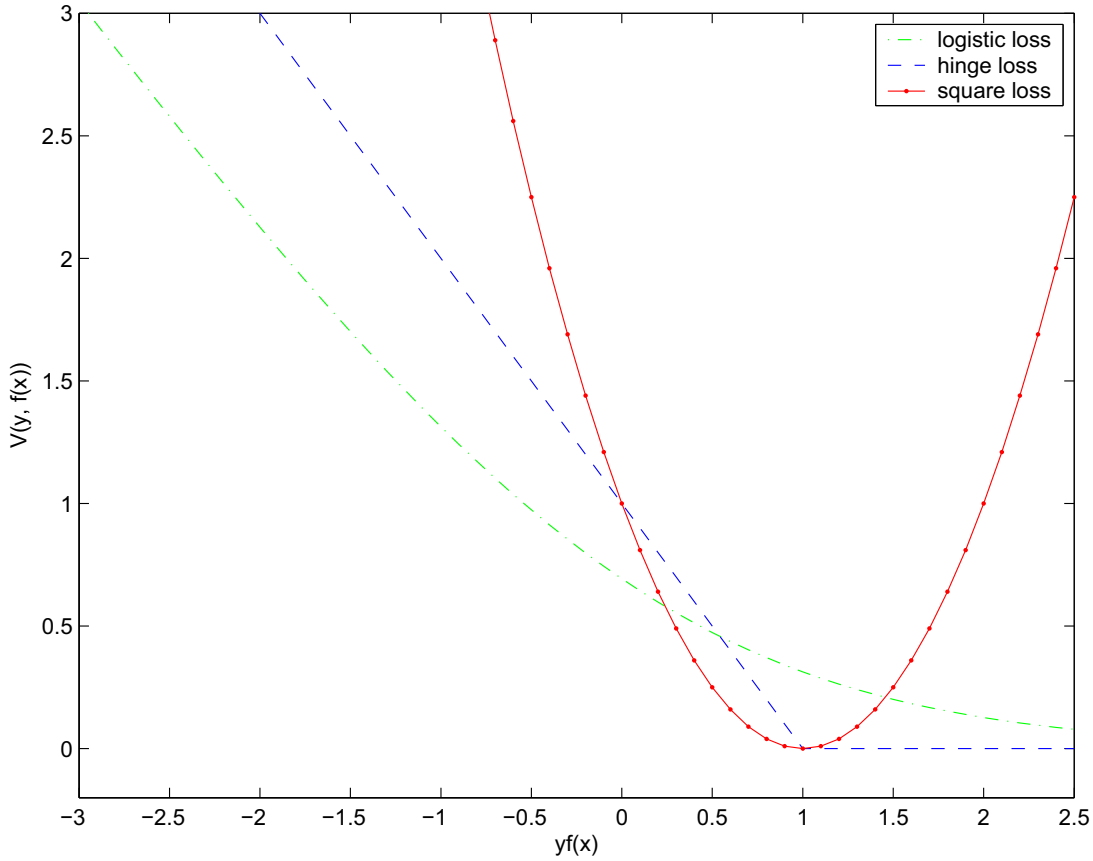

FIG. 1.4: Représentation des fonctions de coût associées aux Support Vector Machines (Hinge loss, $V(y, f(x)) = \max(0, 1-yf(x))$), à la Kernel Logistic Regression (Logistic loss, $V(y, f(x)) = \ln(1 + e^{-yf(x)})$) et à la Ridge Regression (Square loss, $V(y, f(x)) = (y - f(x))^2$).

L'équation (1.63) est intéressante puisqu'elle fournit une interprétation géométrique lorsque l'on relache les contraintes de régularisation et stipule que la fonction de coût cherche à séparer les observations en maximisant la distance des points les plus proches à l'hyperplan. Or les propriétés de marge sont, en un certain sens, une garantie de fiabilité et de généralité du modèle généré.

Rosset et al. [Rosset et al. (2004b)], par le théorème 3, établissent une condition suffisante sur le choix de la fonction de coût pour que la solution, $\hat{\beta}(\lambda)$ converge vers un hyperplan de marge maximale lorsque le terme de régularisation λ tend vers 0.

Théorème 3 *Supposons que les observations des deux classes soient linéairement séparables.* $\exists\beta$ *tel que* $y_i x_i^T \beta > 0$. *Soit* $V(y, f) = V(yf)$ *une fonction de coût non-croissante ne dépendant que de la valeur de la marge.*
S'il existe $t > 0$ *(éventuellement* $t = \infty$*) tel que*

$$\lim_{t \to T} \frac{V(t(1-\epsilon))}{V(t)} = \infty, \ \forall \epsilon > 0 \tag{1.64}$$

Alors V est une fonction de coût de marge maximale dans le sens où toute solution normalisée $\frac{\hat{\beta}(\lambda)}{\|\hat{\beta}(\lambda)\|}$ au problème de régularisation (1.62) converge vers l'hyperplan séparateur optimal quand $\lambda \to 0$. Ainsi, si l'hyperplan séparateur optimal est unique, $\frac{\hat{\beta}(\lambda)}{\|\hat{\beta}(\lambda)\|}$ converge vers cette solution.

$$\lim_{\lambda \to 0} \frac{\hat{\beta}(\lambda)}{\|\hat{\beta}(\lambda)\|} = arg \max_{\|\beta\|_p = 1} \min_i y_i x_i^T \beta \tag{1.65}$$

□

Pour une démonstration complète de ce théorème, nous renvoyons le lecteur à [Rosset et al. (2004b)].

Ce théorème stipule que si la fonction de coût décroit «suffisament rapidement», alors elle engendre des modèles de marge maximale.
Zhu et Hastie [Zhu and Hastie (2005)] ont montré que la fonction de coût associée à la Kernel Logistic Regression vérifie les conditions du théorème 3. Nous pouvons en conclure que l'estimateur de la KLR converge vers celui des SVM lorsque les contraintes de régularisation sont relachées. Cette propriété est d'autant plus intéressante que la séparation complète des classes d'individus n'est pas une configuration bien gérée par la régression logistique. Ainsi, dans une situation de complète séparation des classes d'individus, l'utilisation de la régression logistique régularisée fournit une alternative particulièrement intéressante à la régression logistique «standard».

1.6 Conclusion

La plupart des méthodes à noyaux (e.g. Kernel Ridge Regression et SVM) ont deux interprétations complémentaires : une interprétation géométrique dans le RKHS grâce à l'astuce du noyau et une interprétation fonctionnelle, grâce au théorème de Kimeldorf et Wahba. Afin d'éviter tout phénomène de sur-apprentissage l'idée générale, consiste à réduire la complexité des fonctions solution. Pour parvenir à cet objectif, l'ensemble des méthodes décrites dans ce chapitre s'appuie sur les principes de régularisation de Tikhonov. C'est sur ce principe que se fait le compromis entre complexité de la fonction solution et ajustement aux données.

Au delà des SVM et sous certaines conditions de comportement de la fonction coût, une panoplie d'algorithmes (e.g. KLR décrit dans chapitre, Boosting voir [Rosset et al. (2004a)]) peuvent être engloblées dans la notion d'hyperplan séparateur optimal. Il s'avère que les approches basées sur le calcul d'hyperplan séparateur optimal fournissent une garantie de fiabilité et de généralité des modèles générés. Comme le confirme la littérature et le chapitre de validation, ces algorithmes sont très performants et tout laisse penser qu'au delà de l'aspect régularisation sous-jacent, les propriétés de marge maximale jouent un rôle prépondérant dans l'efficacité de telles approches.

Le prochain chapitre présente d'autres techniques de régularisation basées sur des approches de type réduction de dimension. En effet, comme nous le verrons, la réduction de dimension lorsqu'elle précède une étape de construction de modèles agit sur la norme de l'estimateur.

Chapitre 2

Régularisation par réduction de dimension

Plaçons nous dans le cadre général où l'on cherche à estimer une fonction $\hat{f}_n : x_i \in \mathcal{X} \rightarrow y_i \in \mathcal{Y}$. Pour éviter le sur-apprentissage, il est fondamental de restreindre le choix des fonctions admissibles : c'est l'objectif central du SRM et plus généralement de toute méthode de sélection de modèles. Nous avons vu, dans le premier chapitre, que la mise en oeuvre du SRM pouvait s'apparenter à un problème de régularisation de Tikhonov (Cf. équation (1.14)).
Nous allons maintenant nous focaliser sur les méthodes de régularisation basées sur des approches type «réduction de dimension». En statistique, les principes de régularisation ont longtemps été associés quasi exclusivement à la ridge regression. Or, on sait maintenant que les méthodes de réduction de dimension, utilisées préalablement à toute étape de construction de modèles, fournissent également des propriétés de régularisation. Dans ce contexte, on peut dissocier les méthodes de réduction de dimension non-supervisées (e.g. l'analyse en composantes principales) des méthodes de réduction supervisées (e.g. la régression PLS). Nous étudions ici les propriétés de régularisation sous jacentes à ces approches. Nous verrons qu'il est possible d'interpréter ces approches de réduction de dimension (supervisée ou non) comme une mise en oeuvre du SRM.

Lorsque l'on cherche à relier une variable à expliquer à des variables explicatives et que ces dernières sont corrélées et/ou de grande dimension (mauvais conditionnement), les algorithmes d'apprentissage sont souvent précédés d'un pre-processing visant à synthétiser l'information pertinente. La dimension de la nouvelle représentation des individus étant généralement plus faible que la dimension de l'espace d'origine, on utilise fréquemment les termes de «réduction de dimension» pour désigner ce pre-processing. Un des pre-processing les plus utilisés est l'analyse en composantes principales (ACP). L'ACP a été introduite par Pearson [Pearson (1901)] et développé par Hotelling [Hotelling (1933)]. L'ACP considère exclusivement les données X (réduction de dimension non-supervisée) dans la construction de son espace de redescription. En effet, les axes de l'ACP (composantes principales) sont choisis selon des critères de variance maximale sur X et, de ce fait, indépendants de la variable à expliquer y. Ainsi, dans un contexte supervisé où l'on souhaite tenir compte de la variable explicative y pour construire la nouvelle représentation des individus, l'ACP n'est pas optimale. La régression PLS (PLS-R) offre alors une alternative Wold et al. [Wold et al. (1983)]. L'idée schématique de la PLS-R est la suivante : on cherche à extraire de X l'information pertinente

pour la prédiction de y. Les axes de la PLS-R (composantes PLS) sont ainsi choisis selon des critères de covariance maximale entre les variables explicatives X et la variable à expliquer y. On parlera alors de «réduction de dimension supervisée» dans la mesure où l'information extraite est généralement contenue dans un sous-espace et que la construction de ce sous-espace s'appuie sur la variable à expliquer. Dans le contexte de la régression PLS, nous pouvons distinguer deux situations :

- La régression PLS univariée (ou régression PLS1) correspond à la situation où l'on cherche à expliquer une réponse y à partir de p variables explicatives $X = [\mathbf{x_1} \ ... \ \mathbf{x_p}]$.

- La régression PLS multivariée (ou régression PLS2) correspond à la situation où l'on cherche à expliquer simultanément plusieurs réponses $Y = [\mathbf{y_1} \ ... \ \mathbf{y_q}]$, $q \geq 2$ à partir de p variables explicatives $X = [\mathbf{x_1} \ ... \ \mathbf{x_p}]$.

Dans la suite de ce document, on utilisera, une fois le contexte définie, le terme générique de PLS-R pour nommé à la fois la régression PLS1 et la régression PLS2.

Dans une première partie, nous présentons, la régression PLS qui offre l'avantage de pouvoir traiter de très grands ensemble de données. Dans une deuxième partie, nous situons la régression PLS par rapport à la Régression sur Composantes Principales puis fournissons des interprétations de la régression PLS en termes de régularisation et de contrôle de complexité. Dans une troisième partie, nous présentons des extensions non linéaires de la PLS-R pour se focaliser sur la Kernel PLS.

2.1 La Régression PLS (PLS-R)

La régression PLS2 est une technique visant à résumer deux blocs de variables X et Y par des variables latentes (composantes PLS). X est l'ensemble des p variables explicatives tandis que Y est l'ensemble des q variables à expliquer. X et Y sont observées sur n individus et ne joue pas de rôles symétriques. Les composantes PLS, $t_1, ..., t_m$, reliées à X sont contraintes d'être orthogonales tandis que les composantes PLS, $u_1, ..., u_m$, reliées à Y ne sont pas contraintes d'être orthogonales. La PLS-R décompose les versions centrées de X et Y à travers les formes matricielles (2.1)

$$\begin{aligned} X &= TP^T + E \\ Y &= UQ^T + F \end{aligned} \tag{2.1}$$

où $T = [t_1, ..., t_m]$ et $U = [u_1, ..., u_m]$ sont les matrices $n \times m$ composées des composantes PLS, P et Q représentent les matrices de poids de dimension $p \times m$ et $q \times m$, tandis que E et F représentent les matrices résiduelles de dimension $n \times p$ et $n \times q$.
N'oublions pas que la PLS-R cherche à évaluer et comprendre l'influence des variables explicatives X sur les variables à expliquer Y. Dans ce cadre, l'étape finale consiste à construire un modèle prédictif et explicatif de la forme (2.2) :

$$Y = XB + G \tag{2.2}$$

où B représente la matrice $p \times q$ de poids tandis que G représente la matrice résiduelle $n \times q$.

L'algorithme de la PLS-R a été décliné sous de nombreuses formes. Elles diffèrent au niveau de la construction des composantes PLS T et U. Les principes de l'algorithme NIPALS

Wold [Wold (1966)] (Nonlinear estimation by Iterative Partial Least Squares) sont à la base de l'algorithme PLS le plus couramment présenté Wold et al. [Wold et al. (1983)]. Cependant, nous avons choisi de décrire ici l'algorithme de la Kernel PLS linéaire [Lindgren et al. (1993) ; Lindgren et al. (1994) ; Rännar et al. (1994) ; Rännar et al. (1995)] puisque, comme nous le verrons, il facilite la présentation de la Kernel PLS non linéaire [Rosipal and Trejo (2001)].

2.1.1 Construction des composantes PLS

Construction des composantes PLS t_1^{PLS} ***et*** u_1^{PLS}

Höskuldsson [Höskuldsson (1988)] montre que les premières composantes PLS $t_1 = Xw_1^{PLS}$ et $u_1 = Yc_1^{PLS}$ sont obtenues par maximisation du critère de Tucker (2.3) :

$$cov^2\left(Xw_1^{PLS}, Yc_1^{PLS}\right) = var\left(Xw_1^{PLS}\right) corr^2\left(Xw_1^{PLS}, Yc_1^{PLS}\right) var\left(Yc_1^{PLS}\right) \tag{2.3}$$

sous les contraintes $\|w_1^{PLS}\| = 1$ *et* $\|c_1^{PLS}\| = 1$

[Höskuldsson (1988)] montre que w_1^{PLS} correspond au vecteur propre normalisé associé à la plus grande valeur propre de la matrice $X^T Y Y^T X$.

$$X^T Y Y^T X w_1^{PLS} = \lambda_1 w_1^{PLS} \tag{2.4}$$

et

$$u_1 = Y \underbrace{\frac{Y^T t_1}{t_1^T t_1}}_{c_1^{PLS}} \tag{2.5}$$

Construction des composantes PLS t_h^{PLS} ***et*** u_h^{PLS}

Les composantes PLS suivantes, $t_h = X_{h-1} w_h^{PLS}$ et $u_h = Y_{h-1} c_h^{PLS}$, sont obtenues par maximisation du critère de Tucker (2.6) :

$$cov^2\left(X_{h-1}w_h^{PLS}, Y_{h-1}c_h^{PLS}\right) = var\left(X_{h-1}w_h^{PLS}\right) corr^2\left(X_{h-1}w_h^{PLS}, Yc_h\right) var\left(Y_{h-1}c_h^{PLS}\right)$$

sous les contraintes $\|w_h^{PLS}\| = 1$ et $\|c_h^{PLS}\| = 1$ (2.6)

où $X_0 = X$, $Y_0 = Y$ et X_{h-1} (respectivement Y_{h-1}) est la matrice résiduelle de la régression de X (respectivement Y) sur $t_1, ..., t_{h-1}$.

Remarque 6 *X_{h-1} (respectivement Y_{h-1}) correspond également à la matrice résiduelle de la régression de X_{h-2} (respectivement Y_{h-2}) sur t_{h-1}.* □

Ainsi $X_h = X_{h-1} - t_h^{PLS} p_h^T$, où chaque p_{hj} correspond au coefficient de régression t_h^{PLS} dans la régression de $\mathbf{x_{h-1,j}}$ sur t_h^{PLS} ; et $Y_h = Y_{h-1} - t_h^{PLS} c_h^T$, où chaque c_{hj} correspond au coefficient de régression de t_h^{PLS} dans la régression de $\mathbf{y_{h-1,j}}$ sur t_h^{PLS}. Nous déduisons les formules de deflation (2.7) et (2.8) :

$$X_h \quad = \quad X_{h-1} - t_h^{PLS} \underbrace{\frac{t_h^{PLS^T} X_{h-1}}{t_h^{PLS^T} t_h^{PLS}}}_{p_h^T} = \underbrace{\left(I - \frac{t_h^{PLS} t_h^{PLS^T}}{t_h^{PLS^T} t_h^{PLS}}\right)}_{P_h^{\perp}} X_{h-1} \tag{2.7}$$

et

$$Y_h = Y_{h-1} - t_h^{PLS} \underbrace{\frac{t_h^{PLS^T} Y_{h-1}}{t_h^{PLS^T} t_h^{PLS}}}_{c_h^T} = \underbrace{\left(I - \frac{t_h^{PLS} t_h^{PLS^T}}{t_h^{PLS^T} t_h^{PLS}} \right)}_{P_h^{\perp}} Y_{h-1} \tag{2.8}$$

où $P_h^{\perp}$ définit l'opérateur de projection sur t_h^{PLS}.

Höskuldsson [Höskuldsson (1988)] montre que w_h^{PLS} correspond au vecteur propre normalisé associé à la plus grande valeur propre de la matrice $X_{h-1}^T Y_{h-1} Y_{h-1}^T X_{h-1}$.

$$X_{h-1}^T Y_{h-1} Y_{h-1}^T X_{h-1} w_h^{PLS} = \lambda_h w_h^{PLS} \tag{2.9}$$

et

$$u_h = Y_{h-1} \underbrace{\frac{Y_{h-1}^T t_h}{t_h^T t_h}}_{c_h} \tag{2.10}$$

Posons alors $W = [w_1, ..., w_m]$ et $C = [c_1, ..., c_m]$.

Une fois les matrices T et U obtenues, nous déduisons immédiatement les matrices P et Q de l'équation (2.1) : $P = [p_1, ..., p_m]$ et $Q = [q_1, ..., q_m]$ avec $q_h = u_h^T Y_{h-1} / u_h^T u_h$.

Pour le moment, les composantes PLS t_h sont définies à partir des résidus X_{h-1}. Manne [Manne (1987)] établit le lien (sous forme matricielle) entre les composantes PLS T et les variables originales X :

$$T = XW(P^T W)^{-1} = XW^* \tag{2.11}$$

Les composantes PLS peuvent facilement se calculer récursivement. En effet, il est possible d'exprimer les résidus X_h en fonction des variables d'origines X au travers de la relation (2.12).

$$X_h = X \prod_{j=1}^{h} (\mathbb{I} - w_j p_j^T) \quad \forall h \geq 1 \tag{2.12}$$

La démonstration de cette relation est fournie, par exemple, dans [Tenenhaus (1998), page 103]. Nous obtenons alors :

$$t_h = X_{h-1} w_h = X \underbrace{\prod_{j=1}^{h-1} (\mathbb{I} - w_j p_j^T) w_h}_{w_h^*} \tag{2.13}$$

Remarque 7 *Notons que le calcul récursif de T est préférable puisque dans ce cas, il n'est plus nécessaire d'inverser la matrice $P^T W$ de dimension $m \times m$.* □

Soient X_{new}, des observations n'ayant pas participées à la construction des composantes PLS T. Les coordonnées de ces individus sur l'espace engendré par les composantes PLS sont fournies par la relation (2.14).

$$T_{new} = X_{new} W^* \tag{2.14}$$

2.1.2 Écriture du modèle de régression PLS final

La dernière étape consiste à exprimer la régression de Y sur T en fonction des variables d'origine X.

$$Y = TC^T + F^* = X \underbrace{W(P^TW)^{-1}C^T}_{B} + F^* \tag{2.15}$$

Les valeurs estimées $\hat{Y}$ de X sont fournies par l'équation (2.16)

$$\hat{Y} = XB \tag{2.16}$$

Pour de nouveaux individus, on a bien sûr $\hat{Y}_{new} = X_{new}B$.

Quelques remarques sur l'algorithme

L'algorithme présenté ci-dessus nécessite le calcul des premiers vecteurs propres des matrices $X_h^T Y_h Y_h^T X_h$ de dimension $p \times p$. Cette approche n'est donc pas idéalement adaptée à la modélisation dans les espaces de grande dimension. L'algorithme NIPALS sur lequel s'appuie fréquemment les implémentations de la régression PLS ne nécessite cependant ni inversion ni diagonalisation de matrices. Une description détaillée de la PLS-R basée sur NIPALS est fournie dans Tenenhaus [Tenenhaus (1998)] et a été initialement proposée par Wold et al. [Wold et al. (1983)]. Nous verrons que l'extension non linéaire de la PLS-R basée sur l'astuce du noyau ne souffre plus de cet handicap.

2.2 Quelques interprétations de la régression PLS

Limitons nous au cas particulier où l'on cherche à prédire une variable $y \in \mathbb{R}^n$ à partir de p variables explicatives : c'est la régression PLS1. Nous nous situons donc dans une problématique analogue à celle du chapitre précédent.
Cette section se divise en trois partie. La première partie tente de situer la PLS-R par rapport à la régression sur composantes principales. La deuxième partie présente la PLS-R comme une mise en oeuvre de la minimisation du risque structurel. La troisième partie présente la PLS-R comme un outil de régularisation.

2.2.1 Régression sur Composantes Principales et régression PLS1

Par de légères modifications de l'algorithme de l'Analyse en Composantes Principales (ACP), Bennett et Embrechts [Bennett and Embrechts (2003)] montrent qu'il est possible de d'obtenir les composantes PLS.

Composante principale t_1^{ACP} vs. composante PLS t_1^{PLS}

L'objectif de l'ACP est de construire une projection linéaire des données qui préserve,«autant que possible», les caractéristiques du nuage de points. Ainsi la première composante principale t_1^{ACP} s'obtient par projection linéaire des données sur le vecteur normalisé w_1 (direction principale) choisi selon le critère de maximiser la variance du nuage projeté. Pour résoudre cette

problématique il suffit de résoudre le système (2.17).

$$\begin{array}{l} \underset{w_1}{max}\ var(Xw_1) \\ \text{sous la contrainte } \|w_1\| = 1 \end{array} \tag{2.17}$$

w_1 correspond au vecteur propre normalisé associé à la plus grande valeur propre de la matrice X^TX. La première composante principale $t_1^{ACP} = Xw_1$ est définie comme la projection de X sur le vecteur propre normalisé w_1.

Toutefois, on peut interpréter l'ACP de manière différente. La première composante principale peut être obtenue en recherchant le vecteur w_1, minimisant la distance des individus à leur projection linéaire sur ce vecteur. Pour calculer la projection qui minimise la déformation entre X et sa projection, il suffit de résoudre le problème (2.18)

$$\begin{array}{l} \underset{w_1}{min}\ \|X - Xw_1w_1^T\|^2 \\ \text{sous la contrainte } \|w_1\| = 1 \end{array} \tag{2.18}$$

Si on suppose que Xw_1 approxime y, que penser du problème de minimisation (2.19) ?

$$\begin{array}{l} \underset{w_1}{min}\ \|X - yw_1^T\|^2 \\ \textit{sous la contrainte } \|w_1\| = 1 \end{array} \tag{2.19}$$

Premièrement, il est intéressant de remarquer que $\|X - yw_1^T\|^2$ est un majorant de la fonction de coût usuelle de la régression :

$$\|Xw_1 - y\|^2 = \|(X - yw_1^T)w_1\|^2 \leq \|X - yw_1^T\|\|w\|^2 = \|X - yw_1^T\|^2.$$

Le problème (2.19) minimise un majorant de la fonction de coût de la régression mais le choix des w_1 est maintenant fortement restreint (dilemme biais-variance).
Deuxièment, il est intéressant de constater que la première composante $t_1 = Xw_1$ où w_1 est solution de du problème (2.19) maximise la covariance entre Xw_1 et y. En effet, en supposant X et y standardisées (centrées et réduites) et que $\|w_1\| = 1$

$$\begin{array}{rcl} \|X - yw_1^T\|^2 & = & (X - yw_1^T)(X - yw_1^T)^T \\ & = & XX^T - Xw_1y^T - yw_1^TX^T + yw_1^Tw_1y^T \\ & = & -2cov(Xw_1, y) + \text{constante} \end{array} \tag{2.20}$$

Le problème (2.19) peut ainsi être vu comme le problème de maximisation (2.21), plus connu sous le nom de critère de Tucker.

$$\begin{array}{l} \underset{w_1}{max}\ cov(Xw_1, y) \\ \text{sous la contrainte } w_1^Tw_1 = 1 \end{array} \tag{2.21}$$

La première composante $t_1^{PLS} = Xw_1$ satisfait le critère de Tucker et répond donc au problème de maximisation (2.19). Ainsi, contrairement à l'ACP, la PLS-R exploite l'information fournit par la variable explicative y dans la construction de la première composante PLS t_1^{PLS}.

Composante principale t_h^{ACP} vs. composante PLS t_h^{PLS}

Afin de construire la $h^{\text{ième}}$ composante principale, t_h^{ACP}, la matrice des données X est «déflatée» de manière à obtenir une matrice résiduelle, $X_{h-1} = [X_{h-2} - X_{h-2}w_{h-1}w_{h-1}^T]$,

contenant uniquement l'information de X inexpliquée par $w_1,\ w_2,\ ...,\ w_{h-1}$. Ainsi, à l'itération suivante, on cherche w_h tel que

$$\begin{array}{l} \underset{w_h}{min}\ \|X_{h-1} - X_{h-1}w_h w_h^T\|^2 \\ \text{sous la contrainte } \|w_h\| = 1 \end{array} \tag{2.22}$$

Précisons que w_h correspond au $h^{\text{ième}}$ plus grand vecteur propre normalisé de la matrice X^TX. La $h^{\text{ième}}$ composante principale t_h^{ACP} est définie comme la projection de X sur le vecteur propre normalisé w_h. En d'autres termes, $t_h = X_{h-1}w_h$ et $W_h = \text{vect}\{w_1,\ w_2,\ ...,\ w_h\}$ est le sous-espace vectoriel de dimension h qui, en projection linéaire, déforme le moins le nuage de points original.
De la même manière que précédemment (en supposant que y approxime $X_{h-1}w_h$), t_h^{PLS} est obtenu par optimisation du critère (2.23)

$$\begin{array}{l} \underset{w_h}{max}\ cov(X_{h-1}w_h, y) \\ \text{sous la contrainte } \|w_h\| = 1 \end{array} \tag{2.23}$$

où la matrice X_{h-1} est définie par la relation (2.7).

Remarque 8 *X_{h-1} correspond également à la matrice résiduelle de la régression multiple de X sur $t_1^{PLS}, t_2^{PLS}, ..., t_{h-1}^{PLS}$.* □

Ainsi, la PLS-R maximise la covariance entre $t_h^{PLS} = X_{h-1}w_h$ et y sous la contrainte de normalité des w_h. Par conséquent, contrairement à l'ACP, la PLS-R exploite l'information a priori fournie à la fois par X **et** y pour construire les variables latentes $t_1^{PLS}, ..., t_h^{PLS}$. En notant que

$$cov^2(X_{h-1}w_h, y) = \underbrace{\text{var}(X_{h-1}w_h)}_{\text{Critère de l'ACP}}\ \underbrace{\text{corr}^2(X_{h-1}w_h, y)}_{\text{Critère de la régression}}\ \text{var}(y)$$

$$\text{sous la contrainte } w_h^T w_h = 1$$

On en conclut que les composantes PLS établissent un compromis entre une corrélation maximum avec la variable à expliquer y (critère de la régression) et directions de variance maximale (critère de l'ACP).

Régression sur Composantes Principales vs. Régression PLS

La PLS-R consiste à réaliser une régression de y sur les composantes PLS, $t_1^{PLS}, t_2^{PLS}, ..., t_m^{PLS}$. La régression sur composantes principales (PCR : Principal Component Regression) consiste à réaliser une régression de y sur les composantes principales de l'ACP $t_1^{ACP}, t_2^{ACP}, ..., t_m^{ACP}$. Le nombre de composantes m est usuellement choisi par validation croisée, ce nombre est compris entre 1 et $r = \text{rang}(X)$.

On montre que le coefficient de corrélation de la régression PLS à m composantes est toujours plus grand que le coefficient de corrélation de la PCR à m composantes. La démonstration est fournie par De Jong [de Jong (1993a)].

Par conséquent, dans un contexte supervisé, on devrait privilégier la PLS-R à la régression sur composantes principales.

2.2.2 Régression PLS et Minimisation du Risque Structurel

L'équation (2.15) définit la PLS-R comme une méthode de régression linéaire appliquée non plus aux données d'origine mais à une nouvelle représentation des observations. Les t_h représentent les coordonnées des observations dans le nouveau repère tandis que c_h représente le coefficient de régression de t_h dans la régression de y sur $t_1, ..., t_m$. La nouvelle représentation des données est supervisée puisque la variable à expliquer, y, participe à la construction des t_h. Les observations sont donc représentés sur des variables non-observables directement (les variables latentes) et c'est la raison pour laquelle, l'acronyme de ***P**rojection to **L**atent **S**tructure* est fréquemment attribué à la PLS-R. Ce sont ces variables latentes qui participent à la construction du modèle de régression final prenant la forme fournie par l'équation (2.24).

$$\hat{f}_n(x) = c_1 t_1(x) + c_2 t_2(x) + ... + c_m t_m(x) = \sum_{h=1}^{m} c_h t_h(x) \tag{2.24}$$

où les $\{t_h(x)\}_{h=1}^{m}$ sont les projections de l'individu x sur les m premières composantes sélectionnées.

Soient $t_1^{PLS}, ..., t_m^{PLS}$, les m premières composantes PLS et $\mathcal{T}_k = \text{vect}\{t_1^{PLS}, ..., t_k^{PLS}\}$ l'espace engendré par ces composantes. Nous en déduisons immédiatement une suite d'espaces emboîtés :

$$\mathcal{T}_1 = \text{vect}\{t_1^{PLS}\} \subset \mathcal{T}_2 = \text{vect}\{t_1^{PLS}, t_2^{PLS}\} \subset ... \subset \mathcal{T}_m = \text{vect}\{t_1^{PLS}, ..., t_m^{PLS}\} \tag{2.25}$$

L'équation (2.25) atteste que la PLS-R, à l'instar des SVM, conduit à une structure d'espaces emboîtés. En effet, comme évoquée dans le chapitre précédent, la formulation des SVM comme une procédure de régularisation type Tikhonov est équivalente à

$$\begin{array}{l} \underset{f \in \mathcal{H}_K}{min} \ \frac{1}{n} \sum_{i=1}^{n} (1 - y_i f(x_i))_+ \\ \text{sous la contrainte : } \|f\|_{\mathcal{H}_K}^2 \leq a_j^2 \end{array}$$

et permet d'interpréter les SVM comme une procédure de sélection d'ellipsoïdes homothétiques [Zwald et al. (2004)]. En effet, si k est un noyau de Mercer, on a la relation suivante :

$$\{f; \|f\|_{\mathcal{H}_K} \leq a_m\} = \left\{ \sum_{i \leq 1} a_i \psi_i; \sum_{i=1}^{M} \frac{a_i^2}{\lambda_i} \leq a_m \right\} \tag{2.26}$$

A contrario, dans le cas de la PLS-R, l'élément m de la structure est définie par l'espace vectoriel de dimension m défini par $\mathcal{T}_m$. La différence fondamentale entre ces deux types de structure est que dans le contexte des SVM, la structure est construite indépendamment des données X et y alors que la structure basée sur la PLS-R s'appuie sur l'information fournit à la fois par X et y. Or, Selon Vapnik, «*According to the SRM principle the structure has to be defined a priori before the training data appear*»[Vapnik (1995), page161]. En effet, le SRM se décompose en deux étapes disjointes. Une première étape consiste à fixer la structure d'espace d'hypothèses. Une deuxième étape consiste à rechercher, parmi l'ensemble des hypothèses, la fonction solution $\hat{f}_n$ qui minimise le risque structurel défini au chapitre 1. Il est surprenant que les deux étapes ne soit pas davantage imbriquées. En effet, alors qu'intuitivement, on aimerait construire la structure en s'appuyant sur l'information a priori fournie par les données X et y, le contrôle de complexité qu'implique la minimisation du risque structurel «classique» proscrit ce cheminement. Néanmoins, la PLS-R s'inscrit dans une des approches génériques de mise en oeuvre du SRM que suggèrent Cherkassky et al. [Cherkassky et al. (1999)] et que nous décrivons brièvement dans la suite.

Mise en oeuvre du SRM selon Cherkassky

Plaçons nous dans le cadre de combinaison linéaire de fonctions de base $b(x, w_h) \in D$, où D représente un dictionnaire de fonctions préfixées. On définit la fonction suivante

$$f_m(x, c, W) = \sum_{i=1}^{m} c_h b(x, w_h) \tag{2.27}$$

w_h et c_h sont des paramètres d'ajustement estimés à partir des données. Une structure d'espaces de fonctions emboîtés est obtenue en faisant varier la valeur de m Cherkassky et al. [Cherkassky et al. (1999)], c'est-à-dire le nombre déléments intervenant dans la combinaison linéaire de l'équation (2.27). On a alors la structure $f_1 \subset f_2 \subset ... \subset f_k \subset ....$

Le modèle de la PLS-R, s'inscrit dans ce contexte selon l'équation (2.28)

$$\hat{f}_m^{PLS} = \sum_{i=1}^{m} c_i b(x, w_i) = \sum_{i=1}^{m} c_h t_h^{PLS} = \sum_{i=1}^{m} c_h \underbrace{\sum_{j=1}^{p} \mathbf{x_{h-1,j}} w_{hj}}_{t_h^{PLS} = X_{h-1} w_h} = \sum_{i=1}^{m} c_h \underbrace{\sum_{j=1}^{p} \mathbf{x_j} w_{hj}^*}_{t_h^{PLS} = X w_h^*} \tag{2.28}$$

où w_h est tel que $\text{cov}(X_{h-1}w_h, y)$ est maximale et c_h correspond au coefficient de régression de t_h dans la régression de y sur $t_1^{PLS}, ..., t_m^{PLS}$. Ainsi, la PLS-R peut être vue comme une implémentation de la minimisation du risque structurel dans laquelle participe à la fois les variables explicatives et la variable à expliquer.

Remarque 9 *Notons qu'un raisonnement analogue pourrait être appliqué à la régression sur composantes principales. Dans ce contexte, les fonctions de base sont les composantes principales. Le modèle de la PCR s'écrit alors :*

$$\hat{f}_m^{PCR} = \sum_{i=1}^{m} c_i b(x, w_i) = \sum_{i=1}^{m} c_h t_h^{ACP} = \sum_{i=1}^{m} c_h \underbrace{\sum_{j=1}^{p} \mathbf{x_{h-1,j}} w_{hj}}_{t_h^{ACP} = X_{h-1} w_h} = \sum_{i=1}^{m} c_h \underbrace{\sum_{j=1}^{p} \mathbf{x_j} w_{hj}^*}_{t_h^{ACP} = X w_h^*} \tag{2.29}$$

où w_h est tel que $var(X_{h-1}w_h)$ est maximale et c_h correspond au coefficient de régression de t_h dans la régression de y sur $t_1^{ACP}, ..., t_m^{ACP}$. Contrairement à la PLS-R, la structure est construite en s'appuyant exclusivement sur les variables explicatives. □

Nous pouvons généraliser le raisonnement : toute méthode de réduction de dimension lorsqu'elle précède une étape de construction de modèle peut être interprétée comme une mise en oeuvre du SRM et par là même constituer un outil de contrôle de complexité.
Nous avons vu, dans le chapitre précédent, que la régularisation de Tikhonov joue un rôle central dans l'analyse des données de grande dimension. C'est sur ce principe que repose la totalité des approches décrites au chapitre précédent. Le prochain paragraphe présente donc quelques propriétés de régularisation sous-jacentes à la PLS-R.

2.2.3 Régression PLS et Régularisation

Soit $\hat{\beta}_h^{PLS}$ l'estimateur de la PLS-R de y sur X à h composantes. D'après l'équation (2.15) on a :

$$\hat{y} = T_h c_h = X \underbrace{W_h (P_h^T W_h)^{-1} c_h}_{\hat{\beta}_h^{PLS}}$$

Martens [Martens (1985)] propose une autre formulation équivalente de $\hat{\beta}_h^{PLS}$ fournie par l'équation (2.30) :

$$\hat{\beta}_h^{PLS} = W_h(W_h^T C W_h)^{-1} W_h^T s \tag{2.30}$$

où $C = X^T X$ et $s = X^T y$, que l'on peut réexprimer sous la forme :

$$\begin{aligned} \hat{\beta}_h^{PLS} &= W_h(W_h^T C W_h)^{-1} W_h^T C C^{-1} s & (2.31) \\ &= W_h(W_h^T C W_h)^{-1} W_h^T C \hat{\beta}^{OLS} & (2.32) \end{aligned}$$

En s'appuyant sur l'équation (2.32), l'estimateur de la PLS-R de y sur X à h composantes correspond à la projection C-orthogonale de l'estimateur des moindres carrés de y sur X $\hat{\beta}^{OLS} = (X^T X)^{-1} X^T y$ sur l'espace engendré par les colonnes de $w_1, ..., w_m$. Or la norme de l'estimateur, $\hat{\beta}^{OLS}$, projeté sur l'espace engendré par $w_1, ..., w_m$ est, par construction, plus faible que la norme de l'estimateur, $\hat{\beta}^{OLS}$, projeté sur l'espace engendré par $w_1, ..., w_{m+1}$. Nous en déduisons immédiatement les propositions suivantes :

Proposition 2 *La norme de $\hat{\beta}^{OLS}$ de la régression de y sur X est supérieure où égale à la norme de l'estimateur de la PLS-R de y sur X à m composantes quand $m \leq r$ avec $r = rang(X)$.*

$$\forall h \leq r, \qquad \|\hat{\beta}_h^{PLS}\| \leq \|\hat{\beta}^{OLS}\|$$

et l'égalité est vérifiée lorsque que $h = r$ □

Et, de manière plus générale, on a :

Proposition 3 *L'application $m \rightarrow \|\hat{\beta}_m^{PLS}\|$ est croissante :*

$$\forall k \leq l \qquad \|\hat{\beta}_k^{PLS}\| \leq \|\hat{\beta}_l^{PLS}\|$$

□

Ainsi, la norme de l'estimateur de la PLS-R est une fonction croissante du nombre de composantes PLS-R selectionné. Les propriétés de régularisation de la PLS-R ont été initialement proposé par de Jong [de Jong (1995)] et Goutis [Goutis (1996)]. La démonstration de la proposition 3 a été initialement fournie par De Jong [de Jong (1995)]. Nous avons obtenu ici une démonstration beaucoup plus simple en considérant l'estimateur de la PLS-R de y sur X à m composantes comme une projection C-orthogonale de $\hat{\beta}$ sur l'espace engendré par $w_1, ..., w_m$.

Nous allons maintenant présenter la PLS-R comme un problème de la forme «minimisation d'une fonction de coût sous contrainte». Nous allons pour ce faire introduire la propriété 1. Manne [Manne (1987)] et Helland [Helland (1988)] ont montré que la suite de vecteurs $w_1, ..., w_h$ issue de la PLS-R pouvait être obtenue par orthogonalisation de Gram-Schmidt de l'espace engendré par les vecteurs $s, Cs, C^2 s, ..., C^{h-1} s$ où $s = X^T y$ et $C = X^T X$. Ce résulat a ensuite été étendu aux autres suites $\{w_h^*\}$, $\{t_h\}$ et $\{p_h\}$ issues de la PLS-R [de Jong (1993b)]. Les résultats de Manne, Helland et De Jong sont synthétisés dans la propriété 1.

Propriété 1 *Régression PLS et suite de Krylov*
Soient $s = X^T y$, $t = Xs$, $C = X^T X$ *et* $D = XX^T$.

(a) Il y a équivalence entre les espaces engendrés par les suites $\{w_h\}$, $\{w_h^*\}$, $\{t_h\}$ *et* $\{p_h\}$ *et ceux engendrés par les suites de Krylov définies ci-dessous :*

$$\begin{array}{ll} (1) & \{w_1, ..., w_h\} \approx \mathcal{W}_h = \{s, Cs, C^2s, ..., C^{h-1}s\} \\ (2) & \{w_1^*, ..., w_h^*\} \approx \mathcal{W}_h^* = \{s, Cs, C^2s, ..., C^{h-1}s\} \\ (3) & \{t_1, ..., t_h\} \approx \mathcal{T}_h = \{t, Dt, D^2t, ..., D^{h-1}t\} \\ (4) & \{p_1, ..., p_h\} \approx \mathcal{P}_h = \{Cs, C^2s, C^2s, ..., C^h s\} \end{array}$$

(b) La suite $\{w_1, ..., w_h\}$ *est obtenue en utilisant la procédure d'orthogonalisation de Gram-Schmidt sur la suite de Krylov* $\mathcal{W}_h = \{s, Cs, C^2s, ..., C^{h-1}s\}$.

(c) La suite de vecteurs $\{t_1, ..., t_h\}$ *normalisés est obtenue en utilisant la procédure d'orthogonalisation de Gram-Shmidt sur la suite de Krylov* $\mathcal{T}_h = \{t, Dt, D^2t, ..., D^{h-1}t\}$.
□

Une démonstation détaillée de chacune de ces propriétés est proposée dans Tenenhaus [Tenenhaus (1998), p. 110-112].

Höskuldsson [Höskuldsson (1988)] a montré que $\hat{\beta}_m^{PLS}$ pouvait être obtenu à partir de l'équation (2.33)

$$\hat{\beta}_m^{PLS} = R_m (R_m^T C R_m)^{-1} R_m^T s \tag{2.33}$$

où R_m est une matrice dont les colonnes forment une base orthogonale de la suite de Krylov $\mathcal{W}_m = \{s, Cs, ..., C^{m-1}\}$.

Remarque 10 *La propriété 1.b. stipule que les vecteurs colonnes de* $W_m = [w_1, ..., w_m]$ *forment une base orthogonale de la suite de Krylov* $\mathcal{W}_m$ *; puisque obtenus par orthogonalisation de Gram-Schmidt de* $\mathcal{W}_m$. *On en déduit immédiatement l'équation (2.30).* □

Une expression équivalente à (2.33) est que l'estimateur de la PLS-R de y sur X à m composantes, $\hat{\beta}_m^{PLS}$ est solution du système d'optimisation sous contraintes(2.34) :

$$\hat{\beta}_m^{PLS} = \underset{\beta \in \mathcal{W}_m}{\operatorname{argmin}} \|y - X\beta\|_2^2 \tag{2.34}$$

Ainsi, la PLS-R, à l'instar de la Ridge Regression ou des Support Vectors Machines, correspond à un problème de minimisation d'une fonction de coût pénalisée. Nous retrouvons la problématique centrale du chapitre précédent : la fonction solution $\hat{f}_n$ doit minimiser une fonction de coût tout en étant de norme faible. Le nombre de composantes retenues joue un rôle analogue au λ de la régularisation de Tikhonov au sens où, dans les deux cas, ces paramètres permettent d'établir un compromis en complexité (variance) et ajustement (biais). Ce compromis est, dans un cas, établi par le réglage d'un paramètre continu (à valeur dans $\mathbb{R}^+$) et dans l'autre cas par le nombre de composantes retenues (à valeur discrète inférieure au rang de la matrice des données).

Nous allons maintenant affiner l'analyse des propriétés de régularisation de l'estimateur de la PLS-R. Pour ce faire, commençons par rappeler quelques principes élémentaires de la régression par moindres carrés. Le modèle des moindres carrés est défini par l'équation (2.35) :

$$y = X\beta + \varepsilon \tag{2.35}$$

où β est le vecteur des coefficients de régression, $\text{cov}(y) = \sigma^2 \mathbb{I}_n$ et ε le vecteur résiduel.
La décomposition en valeur singulière de X est de la forme

$$X = V\Sigma U^T$$

où $V = [v_1, ..., v_p]$ et $U = [u_1, ..., u_p]$ sont des matrices orthonormales et Σ une matrice diagonale composée des valeurs propres de X.
Posons $\Lambda = \Sigma^2 = \text{diag}(\lambda_1, ..., \lambda_p)$, $C = X^T X = U\Lambda U^T$ et $s = X^T y$.
L'estimateur de moindres carrés $\hat{\beta}^{OLS}$ est donné par (2.36)

$$\min_{\beta} \|y - X\beta\|_2^2 \tag{2.36}$$

$\hat{\beta}^{OLS}$ est obtenu en résolvant l'équation normale $C\beta = s$. Il vient :

$$\hat{\beta}^{OLS} = C^{-1}s = \sum_{i=1}^{r} \frac{v_i^T y}{\sqrt{\lambda_i}} u_i = \sum_{i=1}^{r} \hat{\beta}_i^{OLS}$$

où

$$\hat{\beta}_i^{OLS} = \frac{v_i^T y}{\sqrt{\lambda_i}} u_i \tag{2.37}$$

est la composante de $\hat{\beta}^{OLS}$ le long de v_i et $r = \text{rang}(X)$.

L'estimateur des moindres carrés est sans biais et le terme de variance dépend des valeurs propres de $X^T X$:

$$var(\hat{\beta}^{OLS}) = \sigma^2 \sum_{i=1}^{r} \frac{1}{\lambda_i}$$

Ainsi, quelques valeurs propres faibles peuvent faire exploser la valeur du MSE. Ajoutons que de faibles valeurs propres de $X^T X$ correspondent à des directions principales v_i de X de faible étalement. Ainsi, une solution pour diminuer la valeur du MSE est de contrôler les directions de l'estimateur des moindres carrés responsables de l'explosion de la variance.
La forme générale d'un estimateur régularisé, noté $\hat{\beta}_{reg}$ est fournie par l'équation (2.38)

$$\hat{\beta}_{reg} = \sum_{i=1}^{r} f(\lambda_i) \hat{\beta}_i^{OLS} \tag{2.38}$$

où $f(.)$ est une fonction à valeur réelle. La valeur de $f(\lambda_i)$ est appelé le facteur de régularisation.

Nous allons maintenant étudier l'influence des facteurs de régularisation sur le comportement de l'erreur quadratique moyenne. Supposons, dans un premier temps, que l'estimateur soit de la forme $\hat{\beta} = Sy$ où la matrice S est indépendante de y.
Nous pouvons montrer (voir par exemple Krämer [Krämer (2006)]) que le MSE de l'estimateur régularisé est de la forme (2.39)

$$MSE\left(\hat{\beta}_{reg}\right) = \sum_{i=1}^{r} \left(f(\lambda_i) - 1\right)^2 \left(u_i^T \beta\right)^2 + \sigma^2 \sum_{i=1}^{r} \left(f(\lambda_i)\right)^2 / \lambda_i \tag{2.39}$$

Ainsi, dans le cas où le facteur de régularisation est indépendant de y (c'est-à-dire que $\hat{\beta}_{reg} = Sy$ est linéaire en y), nous avons les propriétés suivantes :

i. Si $f(\lambda_i) \neq 1$, le biais de l'estimateur augmente.

ii. Si $|f(\lambda_i)| < 1$, la variance de l'estimateur diminue.

iii. Si $|f(\lambda_i)| > 1$, la variance de l'estimateur augmente.

Par conséquent, il est clair qu'un $|f(\lambda_i)| > 1$ est toujours indésirable.

Remarque 11

1. Le facteur de régularisation associé à la Régression sur Composantes Principales à p composantes (indépendant de y) est fourni par l'équation (2.40) :

$$f(\lambda_i) = \begin{cases} 1 \ si \ i \leq p \\ 0 \ si \ i > p \end{cases} \tag{2.40}$$

Comme nous l'avons évoqué, les faibles valeurs propres de $X^T X$ correspondent à des directions principales de faible étalement. Or, l'équation (2.40) sélectionne précisément les directions d'étalement maximum. La régression sur composantes principales élimine donc, de fait, les composantes de $\hat{\beta}^{OLS}$ participant à l'explosion du terme de variance. Nous pouvons cependant ajouter que les composantes principales ne sont pas nécessairement les plus explicatives de y et qu'il peut être préférable de les sélectionner suivant des critères de corrélations avec y.

2. Le facteur de régularisation associé à la Ridge Regression (indépendant de y) est fourni par l'équation (2.41) :

$$f(\lambda_i) = \frac{\lambda_i}{\lambda_i + \gamma} \tag{2.41}$$

où γ est le paramère libre de la ridge regression.

Le paramètre libre de la ridge regression contraint, de fait, le facteur de régularisation à des valeurs inférieures à 1. □

Facteur de régularisation de la régression PLS

Nous allons maintenant décrire le facteur de régularisation de la régression PLS. Notons $L_m = W_m^T C W_m$. L'équation (2.32) fait intervenir l'inverse de la matrice L_m, dans l'expression de l'estimateur de la PLS-R de y sur X à m composantes. La matrice L_m est tridiagonale et ses p paires de vecteurs propres - valeurs propres $(r_i, \mu_i)_{i=1}^p$ sont appelés les pairs de Ritz. L'expression du facteur de régularisation de la PLS-R à m composantes est proposée par Lingjaerde et Christophersen [C.Lingjaerde and Christophersen (2000)], Butler et Denham [Butler and Denham (2000)] et Phatak et de Hoog [Phatak and de Hoog (2002)] et a la forme fournie par l'équation (2.42).

$$f(\lambda_i) = 1 - \prod_{j=1}^{m} \left(1 - \frac{\lambda_i}{\mu_j}\right) \tag{2.42}$$

Le facteur de régularisation de la PLS-R dispose de propriétés surprenantes, notamment pour certaines combinaisons de i et m, $f(\lambda_i) > 1$. Cette propriété est difficilement interprétable du

point de vue de l'équation (2.39) car le facteur de régularisation de la PLS-R dépend des valeurs propres de la matrice L et L, via W, dépend de y. Par conséquent, il est difficile d'évaluer dans quelle mesure les facteurs de régularisation de la PLS-R agissent sur la diminution de l'erreur quadratique moyenne. On retrouve une synthèse des propriétés de régularisation de la PLS-R dans [Krämer (2006)].

La régularisation est une méthode largement employée lorsqu'il s'agit d'analyser des données de grande dimension et les méthodes les plus performantes et reconnues (e.g. Ridge Regression, SVM) sont basées sur ce principe. La régression PLS s'appuie également sur ce procédé et le contrôle de la norme s'effectue par le choix du nombre de composantes sélectionnées. Par ailleurs, contrairement aux méthodes décrites au chapitre 2 (e.g. SVM, KLR), la construction des composantes PLS et du modèle PLS ne nécessite que la manipulation d'outils simples. La PLS-R est donc facile à mettre en oeuvre. De plus, l'algorithme de la PLS-R est largement utilisé en chimiométrie, domaine dans lequel l'analyse de données de grande dimension est courante (e.g. données de spectres). Ainsi de nombreux efforts ont été portés sur le développement d'algorithmes efficaces permettant la gestion de ce type de configuration ($n << p$). Citons, par exemple, les travaux de Rännar et al. [Rännar et al. (1994) ; Rännar et al. (1995)] et Dayal et MacGreggor [Dayal and MacGreggor (1997)]. Par conséquent, ce n'est pas sans raison que la communauté de l'apprentissage au travers des travaux récents de Rosipal et Trejo [Rosipal and Trejo (2001)], Rosipal et al.[Rosipal et al. (2003)], Bennett et Embrechts [Bennett and Embrechts (2003)] et Shawe-Taylor et Cristianini [Shawe-Taylor and Cristianini (2004)] s'est engouffrée dans la brêche ouverte par la PLS-R. En effet, l'intérêt de la PLS-R pour résoudre des problèmes d'apprentissage a été récemment relancé par l'obtention d'une version non linéaire de cet algorithme [Rosipal and Trejo (2001)] : La Kernel PLS non linéaire. En effet, telle que présentée précédemment, la PLS-R ne s'appuie que sur des dépendances linéaires entre variables et cela restreint son potentiel prédictif. Ainsi, la Kernel PLS permet d'exploiter les relations non linéaires entre variables. Nous décrivons donc, dans la prochaine section, cette approche.

2.3 La Kernel Partial Least Squares Regression non linéaire (KPLS)

Plaçons dans cette section dans le cadre de la régression PLS2. Un moyen simple d'introduire dans un modèle PLS de la non linéarité est d'appliquer des transformations aux variables originales (logarithme, polynômes d'ordre plus élevé, produits croisés). La PLS-R de Y sur cette nouvelle représentation des observations établit des liens non linéaires entre Y et les variables explicatives originales X. Cette voie simple a été proposée par Wold [Wold et al. (1989)] et approfondie par Berglund [Berglund and Wold (1997)]. Cependant, une telle transformation n'est intéressante que si la dimension initiale p de X est petite. Cette approche n'est donc pas adaptée lorsqu'il s'agit de traiter des données de grande dimension.

On peut également envisager de conserver l'étape de réduction de dimension initiale et de modifier uniquement les relations liant les composantes PLS à Y. Cette approche a été appliquée au cas où les relations sont polynômiales d'ordre 2 avec ou sans produit croisé [Baffi et al. (1999b) ; Wold et al. (1989)], étendue au cas où les relations sont approximées par des splines [Wold (1992)] ou encore des réseaux de neurones [Baffi et al. (1999a)]. Cette seconde approche présente un inconvénient majeur : les composantes PLS sont toujours obtenues par combinaison linéaire des variables originales et il n'est pas difficile de construire des exemples

inadaptés à ce schéma (par exemple, deux classes d'individus concentriques).

Nous nous intéressons, dans cette section, à la version RKHS de la PLS-R non linéaire. En effet, s'il est possible d'exprimer l'algorithme de la PLS-R uniquement au travers de produits scalaires entre observations, on pourrait alors appliquer l'astuce du noyau à la PLS-R. La Kernel PLS non linéaire est basée sur l'idée de plonger les données dans un RKHS $\mathcal{H}_K$ à travers une fonction noyau k. L'objectif est donc de construire une PLS-R linéaire sur $\mathcal{H}_K$ ou en d'autres termes, une PLS-R non linéaire sur $\mathcal{X}$. Cette approche, proposée par [Rosipal and Trejo (2001)] et explorée par [Bennett and Embrechts (2003), Shawe-Taylor and Cristianini (2004)] est présentée dans ce paragraphe.

2.3.1 Construction des composantes Kernel PLS

Dans le cadre de la construction des composantes PLS, l'équation (2.9) stipule que le vecteur de poids w_h^{PLS} correspond au vecteur propre normalisé associé à la plus grande valeur propre de la matrice $X_{h-1}^T Y_{h-1} Y_{h-1}^T X_{h-1}$. Cette équation ne faisant pas intervenir les observations à travers leur produit scalaire, nous sommes en dehors des conditions d'application de l'astuce du noyau. On souhaite donc s'absoudre du calcul des w_h^{PLS}. La Kernel PLS contourne ces calculs, voici comment procéder :

Construction de la première composante Kernel PLS

Nous cherchons à construire les premières composantes Kernel PLS t_1^{KPLS} et u_1^{KPLS}. Considérons donc le calcul matriciel de l'équation (2.4) :

$$X_0^T Y_0 Y_0^T X_0 w_1^{PLS} = \lambda_1 w_1^{PLS}$$

Or $t_1^{PLS} = X_0 w_1^{PLS}$ et en multipliant les deux membres de l'égalité par X_0, nous obtenons

$$X_0 X_0^T Y_0 Y_0^T \underbrace{X_0 w_1^{PLS}}_{t_1^{PLS}} = \lambda_1 \underbrace{X_0 w_1^{PLS}}_{t_1^{PLS}} \tag{2.43}$$

t_1^{PLS} correspond donc au vecteur propre normalisé associé à la plus grande valeur propre de la matrice $X_0 X_0^T Y_0 Y_0^T$ et $u_1^{PLS} = Y_0 Y_0^T t_1^{PLS}$. Ainsi, le calcul de t_1^{PLS} et u_1^{PLS} ne fait intervenir les observations qu'à travers leur produit scalaire. Il est alors possible, par application de l'astuce du noyau, de construire la première composante PLS, non plus dans l'espace d'origine $\mathcal{X}$, mais dans un RKHS $\mathcal{H}_K$ associé à une fonction noyau k (en pratique non linéaire). En effet, considérons la matrice de Gram K associé à la fonction noyau k et K_0 la version centrée de K.
La première composante Kernel PLS t_1^{KPLS} correspond au vecteur propre normalisé associé à la plus grande valeur propre de la matrice $K_0 Y_0 Y_0^T$.

$$K_0 Y_0 Y_0^T t_1^{KPLS} = \lambda_1 t_1^{KPLS} \tag{2.44}$$

et

$$u_1^{KPLS} = Y_0 Y_0^T t_1^{KPLS} \tag{2.45}$$

Remarque 12 *À l'instar de la PLS-R linéaire, la Kernel PLS doit être appliquée à une version centrée des données. Il s'agit donc de calculer la matrice de Gram K_0 associée à la version centrée Φ_0 de Φ, où Φ définit la matrice de coordonnées des n individus d'apprentissage dans $\mathcal{H}_K$. Wu et al. [Wu et al. (1997)] et Schölkopf et al. [Schölkopf et al. (1998)] propose :*

$$K_0 = (\mathbb{I} - \frac{1}{n} 1_n 1_n^T) K (\mathbb{I} - \frac{1}{n} 1_n 1_n^T) \tag{2.46}$$

où $\mathbb{I}$ est la matrice identité de dimension n et 1_n représente le vecteur de 1 de longueur n. Le calcul de matrice de Gram centrée K_0 est basé sur l'équation (2.47) fournissant la version centrée Φ_0 de Φ

$$\Phi_0(x) = \Phi(x) - \frac{1}{n}\sum_{i=1}^{n}\Phi(x_i) \tag{2.47}$$

□

Remarque 13 *L'algorithme de la PLS-R linéaire normalise les w_h et les c_h. Sa version via le RKHS normalise les t_h et les u_h.* □

Procédure de déflation

De la même manière que pour la PLS-R, la Kernel PLS est un processus itératif. Une fois la composante t_{h-1}^{KPLS} construite, l'algorithme est réitéré à partir des matrices résiduelles K_{h-1} et Y_{h-1} définies par :

$$h \geq 2,\ K_{h-1} = (I - t_{h-1}^{KPLS}\ t_{h-1}^{KPLS^T})K_{h-2}(I - t_{h-1}^{KPLS}\ t_{h-1}^{KPLS^T}) \tag{2.48}$$

$$h \geq 2,\ Y_{h-1} = Y_{h-2} - t_{h-1}^{KPLS}\ t_{h-1}^{KPLS^T}Y_{h-2} \tag{2.49}$$

Le calcul de la matrice résiduelle obtenue à travers l'équation (2.48) est basée sur le fait que Φ_{h-1} est déflatée selon l'équation (2.50)

$$h \geq 2,\ \Phi_{h-1} = (\Phi_{h-2} - t_{h-1}^{KPLS}\ t_{h-1}^{KPLS^T}\Phi_{h-2}) \tag{2.50}$$

où Φ_{h-1} définit la matrice résiduelle de la régression de Φ_{h-2} sur t_{h-1}^{KPLS}.

Construction de la $h^{ième}$ composante Kernel PLS

La $h^{\text{ième}}$ composante Kernel PLS, t_h^{KPLS} correspond au vecteur propre associé à la plus grande valeur propre de la matrice : $K_{h-1}Y_{h-1}Y_{h-1}^T$. En effet, considérons le calcul matriciel de l'équation (2.9) :

$$X_{h-1}^T Y_{h-1} Y_{h-1}^T X_{h-1} w_h^{PLS} = \lambda_h w_h^{PLS}$$

Or $t_h^{PLS} = X_{h-1}w_h^{PLS}$ et en multipliant les deux membres de l'égalité par X_{h-1} (respectivement Y_{h-1}), nous obtenons

$$X_{h-1}X_{h-1}^T Y_{h-1}Y_{h-1}^T \underbrace{X_{h-1}w_h^{PLS}}_{t_h^{PLS}} = \lambda_h \underbrace{X_{h-1}w_h}_{t_h^{PLS}} \tag{2.51}$$

et

$$u_h^{KPLS} = Y_{h-1}Y_{h-1}^T t_h^{KPLS} \tag{2.52}$$

Nous remarquons, comme précédemment, que le calcul de t_h^{PLS} et u_h^{PLS} ne fait intervenir les observations qu'à travers leur produit scalaire. Par simple application du kernel trick, t_h^{KPLS} est le premier vecteur propre normalisé associé à la plus grande valeur propre de la matrice $K_{h-1}Y_{h-1}Y_{h-1}^T$. Il devient alors possible de construire les composantes PLS dans un RKHS par application de l'astuce du noyau.

2.3.2 Écriture du modèle Kernel PLS final

La dernière étape consiste à exprimer la régression de Y sur T en fonction de K_0. Rappelons que le modèle final de la PLS-R linéaire peut s'écrire matriciellement sous la forme :

$$Y = XB + R$$

où $B = W(P^TW)^{-1}C^T$ avec $P = X^TT(T^TT)^{-1}$ et $C = Y^TT(T^TT)^{-1}$.

Il s'avère que la Kernel PLS s'absout du calcul des W. Cette formulation de B n'est donc pas adaptée au contexte non linéaire de la Kernel PLS. Or W représente la covariance entre les résidus de X et U, il s'ensuit que $W = X^TU$. De plus, la Kernel PLS construit des composantes t_h normalisées, il s'ensuit que $T^TT = 1$. Nous obtenons donc l'expression suivante de B.

$$B = \Phi_0^TU(T^TK_0U)^{-1}T^TY \tag{2.53}$$

et les valeurs estimées $\hat{Y}$ de Φ_0 sont fournies par l'équation (2.54)

$$\hat{Y} = \Phi_0B = K_0U(T^TK_0U)^{-1}T^TY = TT^TY \tag{2.54}$$

L'estimation d'individus de test est également basée sur l'équation (2.54) :

$$\hat{Y}_{test} = \Phi_0^{test}B = K_0^{test}U(T^TK_0U)^{-1}T^TY \tag{2.55}$$

K_0^{test} correspond à la version centrée Φ_0^{test} de Φ^{test}, où Φ^{test} définit la matrice de coordonnées des n_t individus de test dans $\mathcal{H}_K$. La version centrée K_0^{test} de K^{test} est obtenue par l'équation (2.56) [Wu et al. (1997) ; Schölkopf et al. (1998)] :

$$K_0^{test} = (K^{test} - \frac{1}{n}1_{n_t}1_n^TK)(\mathbb{I} - \frac{1}{n}1_n1_n^T) \tag{2.56}$$

Le calcul de matrice de Gram centrée, K_0^{test}, est basé sur l'équation (2.57) fournissant la version centrée Φ_0^{test} de Φ^{test}

$$\Phi_0^{test}(x) = \Phi^{test}(x) - \frac{1}{n}\sum_{i=1}^{n}\Phi(x_i) \tag{2.57}$$

Cette approche est intéressante puisque le nombre de variables p n'est plus une limite algorithmique. En effet, les contraintes sont davantage liées au nombre d'observations puisqu'il s'agit maintenant de manipuler les matrices $K_{h-1}Y_{h-1}Y_{h-1}^T$ de dimension $n \times n$. Ainsi, la limitation algorithmique principale de la Kernel PLS est reliée au nombre d'observations (inconvénients inhérents aux méthodes à noyau). La matrice de Gram K de dimension $n \times n$ doit être stockée en mémoire, notamment pour les étapes de centrage (Cf. équation (2.46)) et de déflation (Cf. équation (2.48)) de K. Dans ce contexte, il faut envisager utilisation d'autres extensions non linéaire de la PLS-R. Pour conclure, relevons la difficulté à interpréter les modèles Kernel PLS lorsqu'il s'agit d'évaluer l'influence des variables explicatives X dans la construction des variables à expliquer Y.

2.3.3 Quelques interprétations de la Kernel PLS

Plaçons nous dans le contexte de la régression Kernel PLS1, où l'on cherche à relier $y \in \mathbb{R}^n$ à p variables explicatives. Nous fournissons, à travers la proposition 4, une manière rapide de construire les composantes Kernel PLS.

Proposition 4 *Les composantes Kernel PLS $t_1^{KPLS}, ..., t_h^{KPLS}$ normalisés sont obtenues en utilisant la procédure d'orthogonalisation de Gram-Shmidt sur la suite de Krylov $\{Ky, K^2y, ..., K^hy\}$, où K définit la matrice de Gram associée à une fonction noyau k.* □

Preuve de la proposition 4
La démonstration est évidente puisqu'il ne s'agit que d'une simple application de l'astuce du noyau. Soient $t = Xs$ et $D = XX^T$. D'après la propriété 1, la suite des vecteurs $\{t_1, ..., t_h\}$ normalisés est obtenue en utilisant la procédure d'orthogonalisation de Gram-Shmidt de la suite de Krylov $\{t, Dt, D^2t, ..., D^{h-1}t\} = \{XX^Ty, (XX^T)^2y, (XX^T)^3y, ..., (XX^T)^hy\}$. L'astuce du noyau est ici directement appliquable puisque que les données n'interviennent qu'à travers leur produit scalaire. Nous en déduisons immédiatement la proposition 4.

L'équation (2.54), nous fournit deux interprétations possible de la KPLS :
• Notons $\hat{\alpha} = U(T^TKU)^{-1}T^Ty$. Nous pouvons alors exprimer la solution de la Kernel PLS sous la forme (2.58)

$$f(x) = \sum_{i=1}^{n} \hat{\alpha}_i k(x_i, x) \tag{2.58}$$

La forme de la solution (2.58) se réfère à la solution du théorème de Kimeldorf et Wabha présentée au chapitre 2. Ceci confirme l'expression de la régression PLS comme un problème de la forme fonction de coût pénalisée.

• La deuxième égalité de l'équation (2.54) définit la Kernel PLS comme une méthode de régression linéaire appliquée, non plus aux données d'origine, mais à une nouvelle représentation des observations.

$$f(x) = c_1t_1(x) + c_2t_2(x) + ... + c_mt_m(x) = c^Tt(x) = \sum_{h=1}^{m} c_ht_h \tag{2.59}$$

où $\{t_h(x)\}_{h=1}^m$ sont les projections de l'observation x sur les m premières composantes Kernel PLS et c est le vecteur de coefficients de régression de y sur $t_1, ..., t_m$. Précisons que les composantes t_h sont explicitement construites dans un espace de Hilbert à noyau reproduisant et capturent donc l'information non linéaire reliée à y (sous réserve de fonction de noyau non linéaire).
Puisque l'algorithme de la Kernel PLS est analogue à sa version linéaire (Il ne s'agit finalement que d'un changement d'espace), les propriétés décrites précédemment, notamment sur les principes de régularisation et de mise en oeuvre du SRM sont valables dans le cadre de la Kernel PLS.

Pour conclure ce chapitre sur la réduction de dimension, nous introduisons la Kernel ACP [Schölkopf et al. (1998)] sur laquelle s'appuie la Kernel Projection Machine (KPM) [Zwald et al. (2004)]. La KPM correspond à une nouvelle technique de classification fondée sur l'argument suivant : *le contrôle de complexité peut être obtenu par sélection d'un espace vectoriel via une méthode de réduction de dimension telle que la Kernel PCA*. Cet aphorisme est à la base des méthodes que nous développons dans les chapitres suivants avec la nuance que nous souhaitons privilégier des méthodes de réduction de dimension supervisée.

2.4 La Kernel PCA et la Kernel Projection Machine

Ce paragraphe est inspiré de la présentation de l'ACP fournie par Lebart [Lebart (1997)]. Comme nous l'avons vu en début de chapitre, l'obtention des composantes principales requiert la diagonalisation de la matrice $X^T X$ de dimension $(p \times p)$. Sous cette forme, il est clair que l'ACP n'est pas adapté à la gestion de données de grande dimension puisqu'il s'agit de calculer les vecteurs propres d'une matrice $p \times p$. L'objectif est donc de décrire une version de l'ACP basée sur la diagonalisation de la matrice $K = XX^T$ de dimension $n \times n$ fournissant ainsi une passerelle vers son extension non linéaire via l'astuce du noyau.

La Kernel PCA

Plaçons-nous donc dans l'espace des individus $\mathbb{R}^n$, où le tableau X est alors représenté par p observations dont les n coordonnées représentent les colonnes de X. La démarche pour ajuster le nuage des p «points-variables» est la même que pour le nuage des n «points-individus». Elle consiste à considérer le sous-espace à r dimensions dans $\mathbb{R}^n$ qui ajuste au mieux le nuage de points. Comme précédemment, nous sommes donc amenés à retenir les r vecteurs propres de XX^T correspondant aux r plus grandes valeurs propres. La matrice à diagonaliser sera alors XX^T de dimension $n \times n$. Notons v_j le vecteur propre associé à la j$^{\text{ième}}$ plus grande valeur propre. Il existe des relations de transitions entre les deux espaces vect$\{w_1, ..., w_r\}$ et vect$\{v_1, ..., v_r\}$. En effet,

$$\text{Dans } \mathbb{R}^p, \text{ nous avons } X^T X w_j = \lambda_j w_j \tag{2.60}$$

$$\text{et dans } \mathbb{R}^n, \text{ nous avons } XX^T v_j = \mu_j v_j \tag{2.61}$$

En prémultipliant les deux membres de (2.60) par X, nous obtenons :

$$(XX^T)Xw_j = \lambda_j (Xw_j)$$

Cette relation montre qu'à tout vecteur propre w_j de $X^T X$, relatif à une valeur propre λ_j non nulle, correspond un vecteur propre Xu_j de XX^T relatif à la même valeur propre λ_j. Nous pouvons montrer que toutes les valeurs propres non nulles des deux matrices $X^T X$ et XX^T sont égales. Remarquons que le vecteur Xw_j a pour norme λ_j et donc que le vecteur v_j unitaire correspondant à la même valeur propre λ_j est facilement calculable en fonction de w_j. On obtient ainsi, pour $\lambda_j \neq 0$ les formules de transitions entre les deux espaces $\mathbb{R}^p$ et $\mathbb{R}^n$:

$$\begin{aligned} v_j &= \tfrac{1}{\sqrt{\lambda_j}} X w_j \\ w_j &= \tfrac{1}{\sqrt{\lambda_j}} X^T v_j \end{aligned} \tag{2.62}$$

Dans $\mathbb{R}^p$, la j$^{\text{ième}}$ composante principale, notée $t_{j,\mathbb{R}^p}$ est définie par $t_{j,\mathbb{R}^p} = Xw_j$ et dans $\mathbb{R}^n$, la j$^{\text{ième}}$ composante principale, notée $t_{j,\mathbb{R}^n}$ est définie par $t_{j,\mathbb{R}^n} = X^T v_j$. Compte tenu de (2.62), les composantes principales peuvent se calculer par :

$$\text{et} \qquad \begin{aligned} t_{j,\mathbb{R}^p} &= v_j \sqrt{\lambda_j} \\ t_{j,\mathbb{R}^n} &= w_j \sqrt{\lambda_j} \end{aligned} \tag{2.63}$$

Sur le sous-espace de $\mathbb{R}^p$ engendré par les w_j, les coordonnées des points du nuage des individus sont les composantes Xw_j et sont aussi les composantes $v_j\sqrt{\lambda_j}$ où v_j est le jième vecteur propre de $K = XX^T$ de dimension $n \times n$. L'objectif est donc atteint : les composantes principales

sont obtenues par diagonalisation d'une matrice de dimensions $n \times n$ adapté à la configuration $n << p$. On peut évidemment accéder à une version non linéaire de l'ACP par application directe de l'asctuce du noyau. En pratique, l'algorithme de la Kernel ACP s'applique à une version centrée des données. L'équation (2.46) doit donc précéder l'algorithme.

La Kernel Projection Machine

Au lieu de minimiser le risque empirique associé à la Hinge loss sur des ellipsoïdes, comme c'est le cas pour les SVM, on peut minimiser le risque empirique sur des espaces vectoriels. Toute la difficulté réside alors dans la construction de l'espace vectoriel sur lequel on cherche à minimiser le risque empirique. Les travaux de Zwald et al. [Zwald et al. (2004)] se fondent sur cette idée : ils proposent de minimiser le risque empirique relatif à la Hinge loss sur l'espace vectoriel engendré par les D premières directions principales de la Kernel PCA. Notons S_D, cet espace. À chaque dimension, est alors associée la fonction (2.64)

$$\hat{f}_n = arg\min_{f \in S_D} \frac{1}{n} \sum_{i=1}^{n} (1 - y_i f(x_i))_+ \tag{2.64}$$

La KPM n'exploite donc pas l'information a priori fourni par la variable à expliquer y pour construire le sous-espace vectoriel S_D. Tout laisse ainsi penser que remplacer l'étape non supervisée de la Kernel PCA par une étape supervisée de type Kernel PLS fournit une alternative intéressante que nous n'explorerons pas ici.

2.5 Conclusion

Ce chapitre présente la régression PLS à travers le point de vue de la régularisation et du contrôle de complexité. En effet, nous avons vu d'une part que la régression PLS peut s'exprimer comme un problème de minimisation d'une fonction de coût sous contrainte, d'autre part que la PLS-R s'inscrit dans le schéma générique de mise en oeuvre du SRM fourni par Cherkassky et al. [Cherkassky et al. (1999)] et qu'il existe des implémentations efficaces de la PLS-R parmi lesquelles l'algorithme de la Kernel PLS permettant de gérer les données de grande dimension.Tous ces arguments justifient l'efficacité de la PLS-R dans les espaces de grande dimension.

Ajoutons qu'un des inconvénients majeurs des méthodes à noyau type SVM ou KLR réside dans l'incapacité à interpréter les modèles obtenus. Cet inconvénient est évidemment présent dans les méthodes de type Kernel PLS, mais est sans nul doute amoindri dans la mesure où la représentation des observations dans l'espace engendré par les premières composantes fournit un outil de visualisation et d'aide à l'interprétation de la structure des données. Ces algorithmes (SVM, KPLS, ...) ont une vocation pratique et nous pensons que l'analyse des résultats par inspection visuel représente un argument important.
C'est la raison pour laquelle les méthodes que nous proposons dans les prochains chapitres s'appuient sur des principes de réduction de dimension supervisée. Ces méthodes partent du constat suivant : la PLS-R a été historiquement développée pour prédire des variables continues et il est légitime de s'interroger sur la pertinence de cette approche lorsqu'il s'agit de prédire des variables catégorielles. Nous pouvons commencer par constater que dans le cadre de la PCR, les composantes principales sont obtenues indépendamment de la variable à expliquer y. Il suffit alors de modifier l'étape finale de la PCR : plutôt que de réaliser la régression multiple de y sur les composantes principales sélectionnées, on peut, par exemple, réaliser une régression

logistique. Les travaux de Zwald [Zwald et al. (2004)] s'inscrivent dans cette problématique en suggérant l'utilisation de la hinge loss. Il s'avère que dans le cadre de la PLS-R, la variable à expliquer participe à la construction des composantes PLS. On ne peut donc pas établir le même schéma de modification que celui opéré pour la PCR. Les modifications doivent être plus profondes et intervenir également dans la phase de construction des composantes PLS. Nous traitons précisément de ce sujet dans le prochain chapitre.

Chapitre 3

La Kernel Logistique PLS

La PLS-R a été initialement développée pour prédire des variables continues. Appliquer cette méthode directement à des données binaires ou catégorielles (classification) ne semble intuitivement pas approprié. Dans ce contexte, de nombreux travaux ont tenté d'adapter la régression PLS au contexte de la classification. On peut citer les travaux de Nguyen et Rock [Nguyen and Rocke (2002) ; Nguyen and Rocke (2004)], Barker et Rayens [Barker and Rayens (2003)], Fort et Lambert-Lacroix [Fort and Lambert-Lacroix (2005) et Bastien et al. [Bastien et al. (2005)]. Nous avons choisi de suivre l'approche suggérée par Bastien [Bastien et al. (2005)] conceptuellement simple et facile à mettre en oeuvre. Il s'avère qu'il est tout à fait possible de transposer les principes de la régression PLS au modèle linéaire généralisé (dont la régression logistique binaire est un cas particulier) et Bastien et al. [Bastien et al. (2005)] proposent la régression linéaire généralisé PLS (PLS-GLR). Nous présentons donc, dans une première partie, la PLS-GLR en se focalisant sur le cas particulier de la régression logistique PLS (PLS-LR). Dans une seconde partie, nous proposons une extension non linéaire de la PLS-LR : La Kernel Logistique PLS.

3.1 La Régression Logistique PLS (PLS-LR)

Lorsque la variable à expliquer y est limitée à une variable ($y \in \mathbb{R}^n$), le rang des matrices $X^T y y^T X$ et $y^T X X^T y$ égal 1. On obtient alors les simplifications suivantes pour le calcul des composantes PLS décrites dans le chapitre précédent (cf. équation (2.4)) :

$$w_1 = X^T y \Big/ \|X^T y\|$$
$$t_1 = X w_1 = X X^T y \Big/ \|X^T y\|$$

Garthwaite [Garthwaite (1994)] a remarqué que chaque coordonnée w_{1j} du vecteur de poids w_1 pouvait s'exprimer par l'équation (3.1)

$$w_{1j} = \mathbf{x_j}^T y \Big/ \|X^T y\| = \mathbf{x_j}^T \mathbf{x_j} \frac{\mathbf{x_j}^T y}{\mathbf{x_j}^T \mathbf{x_j} \|X^T y\|} = s_j^2 a_j \Big/ \sqrt{\sum_j (s_j^2 a_j)^2} \tag{3.1}$$

où s_j^2 défini la variance de $\mathbf{x_j}$ et a_j le coefficient de régression de $\mathbf{x_j}$ dans la régression de y sur $\mathbf{x_j}$. Ainsi, le vecteur de poids w_1 est obtenu par une succession de régressions simples de la variable à expliquer y sur chacune des variables explicatives $\mathbf{x_j}$. Ce résultat reste valide pour

les composantes suivantes : w_h est obtenu par une succession de régressions multiples de la variable y sur $t_1, ..., t_{h-1}$ et chaque colonne, $\mathbf{x_{h-1,j}}$, de la matrice X_{h-1}.

La régression linéaire généralisée (PLS-GLR) s'appuie sur l'architecture algorithmique de la régression PLS de Garthwaite [Garthwaite (1994)]. La principale différence entre l'approche de Garthwaite et la PLS-GLR repose sur l'utilisation du modèle linéaire généralisé en lieu et place de la régression standard. La PLS-GLR fournit donc un outil plus général que la PLS-R. Dans ce contexte, Tenenhaus [Tenenhaus (2002)], Bastien et al. [Bastien et al. (2005)] et Tenenhaus [Tenenhaus (2005)] ont étudié la régression logistique PLS et Bastien [Bastien (2004)] a appliqué la PLS-GLR au modèle de Cox.

3.1.1 Algorithme de la Régression PLS Généralisée

Nous présentons dans ce paragraphe, l'algorithme de la PLS-GLR et les descriptions plus spécifiques sont limitées au cadre logistique (PLS-LR) dans la prochaine section.

TAB. 3.1: Algorithme de la Régression PLS Généralisée

Algorithme de la Régression PLS Généralisée

A. Construction des composantes PLS-GLR

Construction de la première composante PLS-GLR $t_1^{PLS-GLR}$

1. GLR de y sur chaque $\mathbf{x_j}, j = 1, ..., p \Rightarrow a_{1j}$
2. Normalisation de $a_1 = (a_{1j})_{j=1,...,p} \Rightarrow w_1 = a_1/\|a_1\|$
3. La première composante PLS-GLR est définie par $t_1^{PLS-GLR} = Xw_1$

Construction de la $h^{ième}$ ***composante PLS-GLR*** $t_h^{PLS-GLR}$

1. Régression de chaque $\mathbf{x_j}, j = 1, ..., p$ sur $t_1^{PLS-GLR}, ..., t_{h-1}^{PLS-GLR}$
   $\Rightarrow X_{h-1} = [\mathbf{x_{h-1,1}}, ..., \mathbf{x_{h-1,p}}]$
2. GLR de y sur $t_1^{PLS-GLR}, ..., t_{h-1}^{PLS-GLR}$ et chaque $\mathbf{x_{h-1,j}}, j = 1, ..., p$
   $\Rightarrow a_{hj}, j = 1, ..., p$
3. Normalisation de $a_h = (a_{hj})_{j=1,...,p} \Rightarrow w_h = a_h/\|a_h\|$
4. La $h^{ième}$ composante PLS-GLR est définie par $t_h^{PLS-GLR} = X_{h-1}w_h$
5. Expression de la $h^{ième}$ composante PLS-GLR $t_h^{PLS-GLR}$ en fonction des X
   $\Rightarrow t_h^{PLS-GLR} = Xw_h^*$

B. GLR de y sur les m premières composantes PLS-GLR $t_1^{PLS-GLR}, ..., t_m^{PLS-GLR}$

L'algorithme de la PLS-GLR ne requiert qu'une succession de régressions dans lesquelles le nombre de variables explicatives n'excède pas le nombre de composantes retenues dans le modèle final. Ce nombre est typiquement faible et est, en pratique, choisi par validation croisée.

3.1.2 Construction des composantes PLS-LR

Dans ce paragraphe, nous nous focalisons sur le cas particulier de la Régression Logistique PLS (PLS-LR).

Construction de la première composantes PLS-LR t_1^{PLS-LR}

La première composante PLS-LR t_1^{PLS-LR} fournit le premier axe discriminant.

Étape 1 : Calcul des coefficients a_{1j} de $\mathbf{x_j}$ dans la régression logistique de y sur $\mathbf{x_j}, j = 1, ..., p$

Étape 2 : Normalisation du vecteur colonne $a_1 = [a_{11}, ..., a_{1p}] : w_1 = a_1/\|a_1\|$

Étape 3 : Calcul de la première composante PLS-LR : $t_1^{PLS-LR} = Xw_1$

Remarque 14 *Soulignons que la composante t_1^{PLS-LR} est construite par une succession de p régressions logistique simples et ne nécessite donc ni diagonalisation, ni inversion de matrice.* □

Procedure de déflation

Supposons construites, aux étapes précédentes, les $(h-1)$ premières composantes PLS-LR, $t_1^{PLS-LR}, ..., t_{h-1}^{PLS-LR}$. On cherche alors à construire une $h^{\text{ième}}$ composante PLS-LR, t_h^{PLS-LR} contenant l'information résiduelle (prédictive de y) non capturée par les $(h-1)$ précédentes. La $h^{\text{ième}}$ composante PLS-LR est donc construite à partir des résidus de la régression de chaque $\mathbf{x_j}$, $j = 1, ..., p$ sur $t_1^{PLS-LR}, ..., t_{h-1}^{PLS-LR}$.

La procédure de déflation se résume donc au calcul des résidus $\mathbf{x_{h-1,1}}, ..., \mathbf{x_{h-1,p}}$ de la régression de $\mathbf{x_j}$, $j = 1, ..., p$ sur $t_1^{PLS-LR}, ..., t_{h-1}^{PLS-LR}$. Notons $X_{h-1} = [\mathbf{x_{h-1,1}}, ..., \mathbf{x_{h-1,p}}]$ la matrice résiduelle.

Remarque 15

1. Il est préférable de remplacer la succession de régressions multiples par une succession de régressions simples en constatant que X_{h-1} est également la matrice résiduelle de X_{h-2} sur t_{h-1}^{PLS-LR}.

2. La construction de la matrice résiduelle ne nécessite que le calcul d'une succession de p régressions sur $h-1$ variables orthogonales ou, au regard de la remarque précédente, d'une succession de p régressions simples.

3. Toute combinaison linéaire des colonnes de X_{h-1} est orthogonale à $t_1^{PLS-LR}, ..., t_{h-1}^{PLS-LR}$. La $h^{ième}$ composante PLS-LR t_h^{PLS-LR} capture donc l'information discriminante résiduelle (i.e. absente des $h-1$ précédentes). □

Construction de la $h^{ième}$ composante PLS-LR t_h^{PLS-LR}

Étape 1 : Calcul des coefficients a_{hj} de $\mathbf{x_{h-1,j}}$ dans la régression logistique de y sur $t_1^{PLS-LR}, ..., t_{h-1}^{PLS-LR}$ et chaque $\mathbf{x_{h-1,j}}, j = 1, ..., p$

Étape 2 : Normalisation du vecteur colonne $a_h = [a_{h1}, ..., a_{hp}] : w_h = a_h/\|a_h\|$

Étape 3 : Calcul de la $h^{\text{ième}}$ composante PLS-LR : $t_h^{PLS-LR} = X_{h-1}w_h$

Étape 4 : Expression de la $h^{\text{ième}}$ composante PLS-LR t_h^{PLS-LR} en fonction de X : $t_h^{PLS-LR} = Xw_h^*$

Remarque 16 *La construction de t_h^{PLS-LR} nécessite l'inversion de p matrices de dimension h.* □

Expression des composantes PLS-LR en fonction des variables d'origine

L'expression des composantes PLS-LR en fonction des variables d'origine est une étape fondamentale dans l'analyse de nouvelles données.
Soit X_{new} les nouvelles observations. Le produit de matrice $T_{new} = X_{new} * W^*$ permet de calculer les valeurs des composantes PLS-LR pour de nouvelles observations.

Calcul de w_h^*

a. La première composante PLS-LR t_1^{PLS-LR} est déjà fonction des variables d'origine

$$t_1^{PLS-LR} = Xw_1 \text{ et } w_1^* = w_1 \tag{3.2}$$

b. La seconde composante PLS-LR t_2^{PLS-LR} s'exprime en fonction des résidus de la régression des variables d'origine sur t_1^{PLS-LR}. De $X = t_1^{PLS-LR}p_1^T + X_1$ et $t_2^{PLS-LR} = X_1w_2$, nous obtenons :

$$\begin{aligned} t_2^{PLS-LR} &= X_1w_2 = \left(X - t_1^{PLS-LR}p_1^T\right)w_2 = \left(X - Xw_1p_1^T\right)w_2 \\ &= \underbrace{X\left(I - w_1p_1^T\right)w_2}_{w_2^*} = Xw_2^* \end{aligned} \tag{3.3}$$

c. De manière similaire, il est facile de montrer que t_h^{PLS-LR} s'exprime en fonction des variables d'origine à travers l'équation (3.4).

$$\begin{aligned} t_h^{PLS-LR} &= X_{h-1}w_h = \left(X - \textstyle\sum_{i=1}^{h-1} t_i^{PLS-LR}p_i^T\right)w_h \\ &= \left(X - \textstyle\sum_{i=1}^{h-1} Xw_i^*p_i^T\right)w_h \\ &= \underbrace{X\left(I - \displaystyle\sum_{i=1}^{h-1} w_i^*p_i^T\right)w_h}_{w_h^*} \end{aligned} \tag{3.4}$$

3.1.3 Modèle final

L'étape finale consiste à réaliser la régression logistique de y sur les m premières composantes PLS-LR retenues, où la valeur de m est selectionnée par validation croisée.

$$\mathbb{P}\Big(y_i = 1/x_i = x_{i1}, ..., x_{ip}\Big) = \frac{e^{c_0+\sum_{h=1}^{m} c_h t_{hi}^{PLS-LR}}}{1+e^{c_0+\sum_{h=1}^{m} c_h t_{hi}^{PLS-LR}}} = \frac{e^{c_0+\sum_{h=1}^{m} c_h \sum_{j=1}^{p} w_{hj}^{*} x_{ij}}}{1+e^{c_0+\sum_{h=1}^{m} c_h \sum_{j=1}^{p} w_{hj}^{*} x_{ij}}} \quad (3.5)$$

où c_h correspond au coefficient de t_h^{PLS-LR} dans la régression logistique de y sur $t_1^{PLS-LR}...,t_m^{PLS-LR}$ et les w_h^* sont obtenus par application de l'équation (3.4).

Ainsi, la régression logistique de y sur les m premières composantes PLS-LR fournit une estimation naturelle de la probabilité conditionnelle d'appartenance des individus aux deux classes.

Pour conclure, soulignons trois aspects algorithmiques de la PLS-LR confirmant l'intérêt de cette approche :

1. La PLS-LR ne nécessite que l'inversion de matrice de faible dimension (au maximum, le nombre de composantes PLS-LR retenues dans le modèle final).

2. La PLS-LR permet la gestion de données où le nombre de variables est supérieur au nombre d'observations.

3. La PLS-LR n'est pas sensible aux fortes corrélations de variables.

Nous avons présenté au chapitre précédent, l'exemple générique de structure d'espace d'hypothèses emboîté fourni par Cherkassky [Cherkassky et al. (1999)]. Cette structure découle de l'équation (2.27). Dans ce contexte, nous avons établi que la PLS-R fournissait une mise en oeuvre du SRM. Or, il est tout à fait possible de transposer ces principes au contexte de la PLS-LR. En effet, l'équation (3.5), fournissant la forme du modèle final de la PLS-LR, se reformule de la manière suivante :

$$f_m(x_i, c, W^*) = \log\left(\frac{\mathbb{P}\Big(y = 1/x = x_{i1}, ..., x_{ip}\Big)}{1-\mathbb{P}\Big(y = 1/x = x_{i1}, ..., x_{ip}\Big)}\right) = c_0 + \sum_{h=1}^{m} c_h \sum_{j=1}^{p} w_{hj}^{*} x_{ij} \quad (3.6)$$

où les w_h^* et c_h sont des paramètres d'ajustement estimés à partir des données. On en conclut que le nombre d'éléments m intervenant dans la combinaison linéaire de l'équation (3.6) spécifie un élement de la structure $f_1 \subset f_2 \subset ... \subset f_k \subset ....$

La puissance discriminante de la régression logistique PLS reste néanmoins limitée puisque les variables explicatives $\mathbf{x_1}, ..., \mathbf{x_p}$ ne sont reliées à y que par des relations linéaires. De plus, le coût algorithmique de la régression logistique PLS est étroitement relié au nombre p de variables explicatives. Ainsi, nous proposons dans le prochain paragraphe une extension peu sensible au nombre de variables et capable de capturer les relations non linéaires : La Kernel Logistique PLS.

3.2 Approximation de rang faible de la matrice de Gram, Empirical Kernel Map et régression PLS

Les méthodes à noyau sont d'élégantes extensions d'approches linéaires au cas non linéaires. Elles sont néanmoins contraignantes dans la mesure où elles exigent des algorithmes linéaires où les observations n'interviennent que par leur produit scalaire. De plus, pour que les éléments de la matrice de Gram K fournissent les produits scalaires inter-individus dans un RKHS, K doit être symétrique et définie positive. On peut alors s'interroger sur la nécessité d'adopter la stratégie contraignante de l'astuce du noyau présentée au précédent chapitre plutôt que d'effectuer une régression PLS de y directement sur K. Cette stratégie, déjà étudiée par [Bennett and Embrechts (2003)] dans le cadre de la PLS-R, est présentée dans le paragraphe suivant puis comparée à la Kernel PLS.

3.2.1 Approximation de rang faible de la matrice de Gram via la régression PLS

Plusieurs publications traitent des approximations de rang faible de la matrice de Gram K. Citons les travaux de [Williams and Seeger (2000)] et [Smola and Schölkopf (2000)]. Ces deux approches recherchent une approximation de K indépendamment de la variable à expliquer. Récemment, une décomposition supervisée a été proposé par Bach et Jordan [Bach and Jordan (2005)]. Supposons que GG^T (où G est une matrice $n \times m$) fournisse une approximation de la matrice de Gram K. Ils suggèrent d'estimer α et β tels qu'ils vérifient l'égalité (3.7).

$$\min_{\alpha \in \mathbb{R}^{n \times d}} \frac{1}{2}\|Y - K\alpha\|_F^2 = \min_{\beta \in \mathbb{R}^{m \times d}} \frac{1}{2}\|Y - G\beta\|_F^2 \tag{3.7}$$

où $\|M\|_F = tr(MM^T)^{1/2}$.
Pour résoudre le problème d'optimisation (3.7), ils suggèrent de résoudre (3.8) :

$$J(G) = \lambda\|K - GG^T\|_1 + \mu \min_{\beta \in \mathbb{R}^{m \times d}} \|Y - G\beta\|_F^2 \tag{3.8}$$

où $\|M\|_1$ = somme des valeurs singulières de M, λ et μ sont choisies de sorte à établir un compromis entre approximation de K et prédiction de Y.

On peut tout naturellement penser à la régression PLS comme une approche résolvant ce type de problématique. L'idée est donc de réaliser directement la régression PLS de y sur K. Les m premières composantes PLS résultantes fournissent alors l'approximation de rang faible de K.

Comme l'atteste l'équation (2.20), on a l'égalité suivante :

$$\|X - yw_1^T\|_2^2 = -2cov(Xw_1, y) + \text{constante}$$

Nous concluons que le problème d'optimisation (3.9) fournit la premìere composante PLS, $t_1^{PLS} = Xw_1$.

$$\begin{array}{l} \min\limits_{w_1} \|X - yw_1^T\|_2^2 \\ \text{sous la contrainte } w_1^T w_1 = 1 \end{array} \tag{3.9}$$

Ainsi, la première composante PLS t_1^{PLS} peut-être obtenue en résolvant le système d'optimisation (3.10)

$$\min_{w_1} \|X - yw_1^T\|_2^2 + \lambda(\|w_1\|_2 - 1) \tag{3.10}$$

Et de la même manière, le problème d'optimisation (3.11) permet d'obtenir la $h^{\text{ième}}$ composante PLS $t_h^{PLS} = X_{h-1}w_h$, où X_{h-1} correspond à la matrice résiduelle définie par l'équation (2.7).

$$\min_{w_h} \|X_{h-1} - yw_h^T\|_2^2 + \lambda(\|w_h\|_2 - 1) \tag{3.11}$$

Ainsi, nous pouvons fournir un schéma analogue a celui proposé par [Bach and Jordan (2005)] : l'approximation de rang m de K est obtenue en résolvant m problèmes d'optimisation. Pour $h = 1, ..., m$,

$$\hat{w}_h = \underset{w_h}{\text{argmin}} \|K_{h-1} - yw_h^T\|_2^2 + \lambda(\|w_h\|_2 - 1) \tag{3.12}$$

où K_{h-1} correspond à la matrice résiduelle de la régression de K sur $t_1, ..., t_{h-1}$.
$T = [K_0\hat{w}_1, ..., K_{m-1}\hat{w}_m] = [t_1^{PLS}, ..., t_m^{PLS}]$ fournit alors l'approximation de rang m de K. Notons qu'en pratique, ces problèmes d'optimisation ne sont évidemment pas à résoudre puisqu'il suffit d'utiliser des implémentations efficaces de la régression PLS (i.e. Kernel PLS linéaire ou NIPALS). Une telle approche a donc l'intérêt d'être résolvable aisément et de maximiser des critères (covariance) intuitivement simples. Néanmoins, cette présentation permet de se situer dans un schéma proche de celui de [Bach and Jordan (2005)].

La régression PLS est ici vue comme une méthode de réduction de dimension supervisée et les composantes PLS issues de la régression PLS de y sur K fournissent alors l'approximation de rang faible de la matrice de Gram K.

3.2.2 Empirical Kernel Map et Régression PLS : DK-PLS

Nous pouvons également nous pencher sur le modèle explicite fourni par la régression PLS de y sur K : La Direct Kernel PLS (DK-PLS) proposé par Bennett et Embrechts [Bennett and Embrechts (2003)].

Définition 5 *Soit m observations $z_1, ..., z_m$ décrites par p variables et k une fonction noyau. L'empirical kernel map de $z_1, ..., z_m$ est définie par la transformation Φ :*

$$\begin{array}{rccl} \Phi : & \mathbb{R}^p & \rightarrow & \mathbb{R}^n \\ & x & \rightarrow & k(.,x)|_{(z_1,...,z_m)} = (k(z_1,x), ..., k(z_m,x)) \end{array} \tag{3.13}$$

□

Dans le cas où $(z_1, ..., z_m) = (x_1, ..., x_n)$, les observations sont décrites par n nouvelles variables : les colonnes de la matrice de Gram K associée à la fonction noyau k. Cette approche revient à considérer les colonnes de la matrice de Gram comme les nouvelles variables explicatives. Chaque cellule $K_{ij} = k(x_i, x_j)$ est une mesure de similarité entre les individus i et j. Chaque colonne de K, k_j , $j = 1, ..., n$ mesure la proximité de l'individu j à tous les autres individus. Les colonnes $k_1, ..., k_n$ représentent les nouvelles variables explicatives.

Comparaison de la Kernel PLS et de la Direct Kernel PLS

Proposition 5 *Soit X une matrice $n \times p$ représentant n individus définis par p variables. Soit $y \in \mathbb{R}^n$ une variable observée sur les n observations. Soit K la matrice de Gram associée à X*

et Φ la matrice de coordonnées des n individus dans $\mathcal{H}_K$.
La régression PLS de y sur $K^{1/2}$ est exactement la régression PLS de y sur Φ. □

preuve de la proposition 5
1. Pour toute matrice symétrique définie positive K, nous avons $K = U\Sigma U^T$ où U est une matrice orthogonale (i.e. $U^T = U^{-1}$) et Σ est une matrice diagonale composée des n valeurs propres positives. Par conséquent, nous pouvons définir $K^{1/2} = U\Sigma^{1/2}U^T$.

2. La démonstration se fait par récurrence. Soit K_h^{DKPLS} (respectivement K_h^{KPLS}), la $h^{\text{ième}}$ matrice déflatée de la Direct Kernel PLS (respectivement KPLS) : on a les deux égalités suivante :

$$K_h^{DKPLS} = (I - t_h^{DKPLS}t_h^{DKPLS^T})K_{h-1} \tag{3.14}$$

et

$$K_h^{KPLS} = (I - t_h^{KPLS}t_h^{KPLS^T})K_{h-1}(I - t_h^{KPLS}t_h^{KPLS^T}) \tag{3.15}$$

Soit $K_0^{KPLS} = K_0^{DKPLS} = K_0$. À partir de $K_0 = U\Sigma_0 U^T$, nous obtenons $K_0^{1/2} = U\Sigma_0^{1/2}U^T$.

La première composante DKPLS de y sur $K_0^{1/2}$ est définie par :

$$t_1^{DKPLS} = \left(K_0^{1/2}\right)^2 y = U\Sigma_0^{1/2}U^T U\Sigma_0^{1/2}U^T Y = U\Sigma_0 U^T y = K_0 y = t_1^{KPLS} \tag{3.16}$$

Ainsi, la premiére composante PLS de y sur Φ est la première composante PLS de y sur $K^{1/2}$

La deuxième composante DK-PLS est définie par :

$$\begin{aligned}
t_2^{DKPLS} &= \left(K_1^{DKPLS^{1/2}}\right)^2 y \\
&= \left[\left(I - t_1^{DKPLS}t_1^{DKPLS^T}\right)K_0^{1/2}\right]\left[\left(I - t_1^{DKPLS}t_1^{DKPLS^T}\right)K_0^{1/2}\right]^T y \\
&= \left[\left(I - t_1^{DKPLS}t_1^{DKPLS^T}\right)K_0\left(I - t_1^{DKPLS}t_1^{DKPLS^T}\right)\right]y \\
&= \left[\left(I - t_1^{KPLS}t_1^{KPLS^T}\right)K_0\left(I - t_1^{KPLS}t_1^{KPLS^T}\right)\right]y \\
&= K_1^{KPLS}y \\
&= t_2^{KPLS}
\end{aligned}$$

Supposons que la $h^{\text{ième}}$ composante KPLS de y sur Φ, t_h^{KPLS} soit la $h^{\text{ième}}$ composante DKPLS de y sur $K^{1/2}$, t_h^{DKPLS}. Nous devons montrer que la $(h+1)^{\text{ième}}$ composante KPLS de y sur Φ,

t_{h+1}^{KPLS}, est la $(h+1)^{\text{ième}}$ composante DKPLS de y sur $K^{1/2}$, t_{h+1}^{DKPLS}.

$$\begin{aligned}
t_{h+1}^{DKPLS} &= \left(K_h^{DKPLS^{1/2}}\right)^2 y \\
&= \left[\left(I - t_h^{DKPLS} t_h^{DKPLS^T}\right) K_{h-1}^{1/2}\right]\left[\left(I - t_1^{DKPLS} t_1^{DKPLS^T}\right) K_{h-1}^{1/2}\right]^T y \\
&= \left[\left(I - t_h^{DKPLS} t_h^{DKPLS^T}\right) K_{h-1}\left(I - t_h^{DKPLS} t_h^{DKPLS^T}\right)\right] y \\
&= \left[\left(I - t_h^{KPLS} t_h^{KPLS^T}\right) K_{h-1}\left(I - t_h^{KPLS} t_h^{KPLS^T}\right)\right] y \\
&= K_h^{KPLS} y \\
&= t_{h+1}^{KPLS}
\end{aligned}$$

Avantages de l'Empirical Kernel Map

Si l'on considère les colonnes de la matrice de Gram $K = [k_1, ..., k_n]$ comme les nouveaux descripteurs, on peut souligner qu'il n'est pas nécessaire de restreindre la transformation (3.13) au noyau de Mercer. Les valeurs de la matrice K doivent simplement fournir des indices de similarité et non les valeurs des produits scalaires dans un RKHS : les conditions de defini positivité sur K n'ont donc plus d'utilité et des mesures de similarité naturelles pour un problème donné peuvent alors être exploitées. Nous voyons ici un premier élément avantageux de l'Empirical Kernel Map.
Par ailleurs, un second avantage à se restreindre à des transformations du type Empirical Kernel Map réside dans les contraintes algorithmiques bien plus reliées au nombre d'observations (n) qu'au nombre de variables (p) (gestion de matrices $n \times n$). Cette transformation est donc idéale pour des configurations $p >> n$. À l'inverse, dans un contexte où le nombre d'observations est important, la gestion de matrices $n \times n$ devient intractable. On peut alors envisager de sélectionner certaines colonnes de la matrice K puisque il est possible de considérer des matrices rectangulaires. Ainsi, appliquer des méthodes d'échantillonnage aux colonnes de K rend possible la gestion de base de données où le nombre d'observations (n) est élevé.

On peut conclure qu'une transformation de type Empirical Kernel Map est bien plus flexible qu'une transformation d'envoi vers un Espace de Hilbert à Noyau Reproduisant par l'astuce du noyau. Précisons que les transformations de type Empirical Kernel Map, à l'instar des transformations vers un RKHS, engendrent des modèles non interprétables.

3.3 La Kernel Logistique PLS

En notant les liens entre la Kernel PLS et la Direct Kernel PLS et en considérant les avantages relatifs à l'Empirical Kernel Map, nous proposons la Kernel Logistique PLS [Tenenhaus et al. (2005)] qui n'est autre que la Régression Logistique PLS appliquée à la matrice de Gram K. L'objectif principal de la KL-PLS est de trouver une représentation des données (espace vectoriel) tel qu'un hyperplan sépare les deux classes. Cet espace vectoriel est engendré par les composantes KL-PLS $t_1^{KL-PLS}, ..., t_m^{KL-PLS}$. La construction de cet espace s'effectue de manière itérative en enrichissant l'espace d'investigation par l'ajout de composantes KL-PLS. Pour accéder aux informations non linéaires, on utilise des matrices de Gram associées à des fonctions noyaux non linéaires (par exemple, noyau gaussien ou polynomial).

En résumé, La KL-PLS construit une succession de composantes orthogonales dans l'espace induit par les colonnes de la matrice de Gram suivie d'une régression logistique dans l'espace engendré par les composantes KL-PLS. L'algorithme de la KL-PLS se décompose ainsi en trois étapes :

1. Construction de la matrice de Gram K

2. Construction des composantes KL-PLS

3. Régression Logistique de y sur les m composantes KL-PLS retenues

3.3.1 Algorithme de la Kernel Logistique PLS

Soient X le tableau de données représentant n observations définies par p variables explicatives et $y = \{0, 1\}$ une variable binaire observée sur les n observations. Soit K la matrice de Gram associée à X. Notons k_j le j$^{\text{ième}}$ vecteur colonne de K.
Le tableau 3.2 présente de manière algorithmique la KL-PLS.

TAB. 3.2: Algorithme de la Kernel Logistique PLS

Algorithme de la Kernel Logistique PLS

A. Construction des composantes KL-PLS

Construction de la première composante KL-PLS t_1^{KLPLS}

1. LR de y sur chaque $k_j, j = 1, ..., n \Rightarrow a_{1j}$
2. Normalisation de $a_1 = (a_{1j})_{j=1,...,n} \Rightarrow w_1 = a_1/\|a_1\|$
3. La première composante KL-PLS est définie par $t_1^{KLPLS} = Kw_1$

Construction de la $h^{ième}$ composante KL-PLS t_h^{KLPLS}

1. Régression de chaque $k_j, j = 1, ..., n$ sur $t_1^{KLPLS}, ..., t_{h-1}^{KLPLS}$
   $\Rightarrow K_{h-1} = [k_{h-1,1}, ..., k_{h-1,n}]$
2. LR de y sur $t_1^{KLPLS}, ..., t_{h-1}^{KLPLS}$ et chaque $k_{h-1,j}, j = 1, ..., n$
   $\Rightarrow a_{hj}, j = 1, ..., n$
3. Normalisation de $a_h = (a_{hj})_{j=1,...,n} \Rightarrow w_h = a_h/\|a_h\|$
4. La $h^{\text{ième}}$ composante KL-PLS est définie par $t_h^{KLPLS} = K_{h-1}w_h$
5. Expression de la $h^{\text{ième}}$ composante KL-PLS, t_h^{KLPLS}, en fonction de K
   $\Rightarrow t_h^{KLPLS} = Kw_h^*$

B. LR de y sur les m premières composantes KL-PLS $t_1^{KLPLS}, ..., t_m^{KLPLS}$

Mettons en lumière, au travers de cette présentation, le fait que cette implémentation ne requiert qu'une succession d'inversions de matrices hessiennes (sur chacune des régressions logistiques) de faible dimension (nombre de composantes KL-PLS).

Puisque l'algorithme de la KL-PLS est analogue à celui de la PLS-LR, nous ne présentons pas les détails de l'algorithme correspondant mais renvoyons le lecteur à la description de la Régression Logistique PLS : la seule modification réside dans la substitution de X par K. L'algorithme est donc simple à mettre en oeuvre puisqu'il ne requiert principalement qu'une

succession de régressions simples et de régressions logistiques.
Par ailleurs, il est intéressant de constater qu'il n'est pas nécessaire de stocker en mémoire la matrice de Gram puisque les colonnes de K sont utilisées individuellement. Il est alors possible de gérer des ensembles de données composées d'un grand nombre d'individus. Cependant, en pratique, cette implémentation est d'un intérêt limité car le temps de calcul s'en trouve augmenté (il faut réevaluer les colonnes de K à la construction de chaque nouvelle composante). La limite algorithmique de la KL-PLS est donc principalement reliée au nombre d'observations (gestion de matrices de dimension $n \times n$). Néanmoins, puisque basée sur des transformations de type Empirical Kernel Map, la KL-PLS ne se limite pas à l'étude complète de K et il est possible de traiter des matrices rectangulaires (contrairement à la Kernel PLS) sans modification de l'algorithme. Cette approche est donc bien plus souple car elle permet par une sélection de colonnes de K aléatoires ou ingénieuses par des techniques d'échantillonage de gérer des bases de données où le nombre d'observations est important.

Échantillonage et test de Wald

Nous suggérons une technique d'échantillonnage basée sur le test de Wald. On souhaite conserver les colonnes de K reliées à y de manière statistiquement significative. On effectue donc la régression logistique de y sur chacune des colonnes de $K = [k_1, ..., k_n]$.
Ainsi, pour $j = 1, ..., n$: on souhaite tester l'hypothèse nulle :

$$H_0 : \beta_j = 0 \text{ contre } H_1 : \beta_j \neq 0$$

où β_j est défini dans l'équation (3.17) :

$$\mathbb{P}(y = 1/k_j) = \frac{e^{\beta_{0j} + \beta_j k_j}}{1 + e^{\beta_{0j} + \beta_j k_j}} \tag{3.17}$$

Le test de Wald est utilisé dans le cadre de la régression logistique pour tester l'hypothèse nulle. On conserve les colonnes de K pour lesquelles le test de Wald est rejeté. Le risque α, à fixer par l'utilisateur, pourra être choisi par validation croisée.

3.3.2 Modèle final

L'étape finale consiste à réaliser la régression logistique de y sur les m premières composantes KL-PLS retenues, où m est selectionné par validation croisée.

$$\mathbb{P}\Big(y_i = 1/k_i = k_{i1}, ..., k_{in}\Big) = \frac{e^{c_0 + \sum_{h=1}^{m} c_h t_{hi}^{KLPLS}}}{1 + e^{c_0 + \sum_{h=1}^{m} c_h t_{hi}^{KLPLS}}} = \frac{e^{c_0 + \sum_{h=1}^{m} c_h \sum_{j=1}^{n} w_{hj}^{*} k_{ij}}}{1 + e^{c_0 + \sum_{h=1}^{m} c_h \sum_{j=1}^{n} w_{hj}^{*} k_{ij}}} \tag{3.18}$$

La régression logistique de y sur les m premières composantes KL-PLS fournit une estimation naturelle de la probabilité conditionnelle d'appartenance des individus aux différentes classes. Le nombre m de composantes retenues est un paramètre d'ajustement permettant d'établir un compromis entre adéquation aux données et complexité de la solution.

Remarque 17
1. L'estimateur du maximum de vraisemblance de la régression logistique dépend de la configuration spatiale des individus. Dans le seul cas d'un recouvrement des classes d'individus, l'estimateur du maximum de vraisemblance existe et est unique ([Allison (1999)] ; [Albert and Anderson (1984)]). Par conséquent, en appliquant la régression logistique sur les composantes

KL-PLS, il est fondamental de considérer la configuration spatiale des individus. Dans une situation de séparation complète ou quasi-complète des deux classes d'individus, l'analyse discriminante de Fisher de y sur les m composantes KL-PLS retenues pourrait être utilisée. On peut préférer l'utilisation systématique de l'approche de Heinze et Schemper [Heinze and Schemper (2002)] consistant à réduire le biais de l'estimateur du maximum de vraisemblance par application de l'approche de Firth [Firth (1993)]. On peut également envisager utiliser la régression logistique régularisée présentée dans le premier chapitre. Cette dernière approche présentant l'intérêt de converger vers des modèles de marge maximale lorsque les contraintes de régularisation sont relâchées.

2. À travers le terme de droite de l'équation (3.18), il est intéressant de constater, qu'une fois le modèle construit, il n'est pas nécessaire de connaître les coordonnées des individus dans l'espace engendré par les composantes KL-PLS pour estimer leur probabilité d'appartenance. D'un point de vue «implémentation-optimisation», cette propriété est utile lorsqu'il s'agit de prédire la classe d'appartenance d'un individu qui n'a pas participé à la construction du modèle.

3. Cette approche s'étend sans aucune difficulté au cadre multiclasse ordinale, c'est-à-dire qu'on peut également prédire des variables catégorielles ordonnées à plus de deux modalités. Il suffit de remplacer chaque régression logistique binaire par une régression logistique ordinale à rapport de chance proportionelle.

4. De la même manière que pour la KL-PLS, il est tout à fait possible d'étendre la PLS-GLR au cadre non linéaire par des transformations de type Empirical Kernel Map. Par conséquent, la portée de cette approche est bien plus étendue que celle présentée ici puisqu'elle s'applique à l'ensemble du modèle linéaire généralisé. □

3.4 Conclusion

Dans ce chapitre, nous avons présenté un nouvel outil de classification non linéaire, algorithmiquement peu sensible au nombre de variables. Cette approche est basée sur des principes de réduction de dimension supervisée et s'appuie sur des transformations de type Empirical Kernel Map. Ces transformations permettent d'accéder à des informations non linéaires et facilitent donc la recherche d'espace discriminants. De plus, ce chapitre met en avant la flexibilité de l'Empirical Kernel Map par rapport à l'astuce du noyau. En effet :

i. Il n'est plus necessaire de se restreindre à l'étude de matrice carrée puisque la KL-PLS s'applique sans modification de l'algorithme à des matrices rectangulaires.

ii. Les conditions de définie positivité sur K n'ont plus lieu d'être et il est tout à fait possible d'exploiter, lorsqu'elles existent, des mesures de similarités naturelles. En effet, dans différents domaines, de l'analyse textuelle à la génomique, de nombreux efforts ont été portés sur la construction de noyau adapté (e.g. kernel string). Ces noyaux doivent vérifier les conditions de Mercer qui, dans de nombreux cas, peuvent être difficile à respecter (et le sont au détriment de mesure de similarité intuitive). La KL-PLS est affranchie de ces contraintes.

De plus, puisque basée sur des techniques de réduction de dimension, la KL-PLS permet une inspection visuelle des données sur des espaces de faible dimension (1, 2 ou 3 dimensions). Ceci présente un double intérêt :

iii. La visualisation fournit des informations sur la structure des données : Les classes sont-elles linéairement séparables ? Les séparations sont-elles cohérentes ? Les individus mal classés sont ils proches de la frontière de séparation ?

iv. Cet outil de visualisation permet d'orienter (et donc de faciliter) le choix des paramètres d'ajustement du noyau intervenant dans la transformation (3.13).

Ainsi, les potentialités de visualisation fournissent un outil d'aide à l'interprétation très appréciable et qui constraste avec les méthodes types SVM, KLR ou Ridge Regression.

Nous montrons, dans le cinquième chapitre, que le comportement discriminant de la KL-PLS, concurrence les méthodes les plus performantes de la littérature telle que les SVM. Nous présentons également des résultats graphiques illustrant la KL-PLS comme un outil de visualisation et d'aide à l'interprétation dans des espaces de faible dimension.

Chapitre 4

La classification multiclasses

Les chapitres précédents sont concentrés exclusivement sur la classification binaire mais nous avons noté que la PLS-LR et son extension non linéaire, la Kernel Logistique PLS, s'étendent sans difficulté au cadre multiclasse ordinale.
Le présent chapitre traite de la classification multiclasse lorsque $y \in \{1, ..., G\}$ est une variable nomimale à plus de deux modalités. Il répond à une problématique souvent rencontrée en pratique : quelle approche adopter pour prédire une variable polytomique à partir d'un grand nombre de variables explicatives et corrélées ?

Nous allons donc, dans un premier temps, faire un bref état de l'art du SVM multiclasses puis, dans un second temps, présenter des alternatives multiclasses fondées sur les principes de la régression PLS. Dans ce contexte, nous proposons une extension de la PLS-LR au cas où la variable à prédire est catégorielle à plus de deux modalités : la Régression Logistique Multinomiale PLS (LM-PLS). Puis, inspirées des transformations de type Empirical Kernel Map sur lesquelles se fondent la Kernel Logistique PLS (cf. chapitre 3), nous proposons une extension de la LM-PLS peu sensible au nombre de variables et capable de produire des modèles prédictifs non linéaires : la Kernel Logistique Multinomiale PLS (KLM-PLS)

4.1 Les Support Vector Machines multiclasses

Soit un ensemble de n individus décrits par p variables. À chaque individu est associé une valeur $y \in \{1, ..., G\}$. Pour chaque individu i, y_i fournit sa classe d'appartenance. L'échantillon d'apprentissage D_n est alors défini par $D_n = \{(x_1, y_1), ..., (x_n, y_n)\}$. L'objectif est de trouver une fonction de décision $\hat{f}_n$ qui a tout x_i associe sa classe d'appartenance $y_i \in \{1, ..., G\}$.
Nous avons vu que la solution classique des SVM binaires était de la forme

$$f(x) = \sum_{i=1}^{n} c_i k(x, x_i) + \beta$$

Dans la plupart des situations, la forme multiclasses des SVM à G classes revient à estimer G fonctions de décision, $f_1, ..., f_G$ de la forme :

$$f_g(x) = \sum_{i=1}^{n} c_{ig} k(x, x_i) + \beta_g$$

On peut distinguer deux courants de pensées fondamentalement différents : les combinaisons de modèles binaires et les approches unifiées.

4.1.1 Combinaison de SVM binaire : One Versus All (OVA)

C'est la solution la plus simple et la première proposée (voir par exemple [Bottou et al. (1994)]). Cette approche consiste à construire G modèles. Le $g^{\text{ième}}$ modèle est construit en considérant les observations de la classe g comme appartenant à la classe positive $(+1)$ et tous les autres individus comme appartenant à la classe négative (-1). Chacun des G modèles doit donc résoudre le problème d'optimisation quadratique formulé au chapitre 1 au travers de l'équation (1.45). De cette procédure découlent donc G fonctions de décision définies par les équations (4.1) :

$$\begin{aligned} f_1(x) &= \textstyle\sum_{i=1}^{n} c_{i1} k(x, x_i) + \hat{\beta}_1 \\ &\vdots \\ f_G(x) &= \textstyle\sum_{i=1}^{n} c_{iG} k(x, x_i) + \hat{\beta}_G \end{aligned} \tag{4.1}$$

L'attribution d'une classe à l'individu x s'effectue au travers de la règle de décision (4.2)

$$\text{Classe de } x = \arg \max_{g=1,\ldots,G} \left[f_g(x) \right] \tag{4.2}$$

4.1.2 Combinaison de SVM binaire : One Versus One (OVO)

Une deuxième approche conceptuellement simple introduite pour les SVM par exemple par Friedman [Friedman (1996)] et Kreßel [Kreßel (1999)] consiste à construire $G(G-1)/2$ modèles. Chacun de ces modèles est construit en opposant toutes les combinaisons possibles de deux classes. De cette procédure découlent donc $G(G-1)/2$ fonctions. La fonction de décision opposant la classe k à la classe l est de la forme :

$$f_{kl}(x) = \sum_{i=1}^{n} c_{i,kl} k(x, x_i) + \hat{\beta}_{kl} \tag{4.3}$$

L'attribution d'une classe à l'individu x s'effectue au travers de la règle suivante : si le signe de $f_{kl}(x)$ attribue l'individu x à la classe k, alors on incrémente la classe k de 1 ; sinon, l'incrémentation est pour la classe l. On assigne alors à x la classe qui a reçu le plus grand nombre de votes. En cas d'égalité, on choisit le plus petit index de la classe.

Ces deux méthodes sont les plus communément utilisées car elles sont conceptuellement simples et fournissent, en pratique, de bons résultats.

4.1.3 Les approches unifiées

Plusieurs travaux récents abordent le problème SVM multiclasses en résolvant un unique problème d'optimisation quadratique afin d'en déduire G fonctions de décision. À l'inverse de l'approche OVA, les G fonctions de décision sont obtenues simultanément. On peut citer les travaux équivalents de Weston et Watkins [Weston and Watkins (1998)] et Vapnik [Vapnik (1998)] d'une part ainsi que ceux de Bredensteiner et Bennett [Bredensteiner and Bennett (1999)] d'autre part. L'équivalence de ces méthodes a été démontrée par Guermeur [Guermeur (2002)]. Citons également l'approche de Lee [Lee et al. (2001)].

Méthode de Weston et Watkins [Weston and Watkins (1998)]

L'idée fondamentale des SVM multiclasses de Weston et Watkins mais plus généralement de toutes les approches unifiées est de construire les G modèles, non pas successivement comme c'est le cas pour OVA, mais en les faisant interagir. Plus concrètement, soit une observation x appartenant à la classe i, OVA contraint le classifieur i si $f_i(x) < 1$, et pour toutes les autres classes j, si $f_j(x) > -1$. Dans le schéma proposé par Weston et Watkins, pour chaque paire $i \neq j$, des contraintes sur les modèles n'apparaisent que si $f_i(x) < f_j(x) + 2$. Par conséquent, si $f_i(x) < 1$, le modèle n'est contraint que si $f_j(x)$ est suffisament grand pour $i \neq j$; De manière analogue, si $f_j(x) > -1$, le modèle n'est contraint que si $f_i(x)$ est suffisament petit. Pour mettre en oeuvre ce principe, Weston et Watkins proposent d'utiliser $n(G-1)$ variables ressorts ξ_{ij} , où $i \in \{1,\ ...,\ n\}$ et $j \in \{1,\ ...,\ G\} \backslash y_i$ au travers du problème d'optimisation (4.4).

$$\begin{array}{l} \underset{f_1,...,f_G \in \mathcal{H}_K, \xi \in \mathbb{R}^{n(G-1)}}{min} \ \sum_{j=1}^{G} \|f_j\|^2_{\mathcal{H}_K} + C \sum_{i=1}^{n} \sum_{j \neq y_i} \xi_{ij} \\ \\ \text{sous les contraintes} \begin{cases} f_{y_i}(x_i) + \beta_{y_i} \geq f_j(x_i) + \beta_j + 2 - \xi_{ij} \\ \\ \xi_{ij} \geq 0,\ i = 1,\ ...,\ n,\ 1 \leq j \neq y_i \leq G \end{cases} \end{array} \tag{4.4}$$

Cette méthode s'inspire donc directement du problème d'optimisation (1.42) ou de manière équivalente du problème (1.51).

Méthode de Lee, Lin et Wahba [Lee et al. (2001)]

Nous avons vu que la solution $\hat{f}_n$ des SVM pouvait s'obtenir par résolution du problème (1.48) :

$$\hat{f}_n = \arg \underset{f \in \mathcal{H}_K}{min} \frac{1}{n} \sum_{i=1}^{n} [1 - y_i f(x_i)]_+ + \lambda \|f\|_{\mathcal{H}_K}$$

Lee [Lee et al. (2001)] propose d'étendre de manière particulièrement élégante cette formulation au contexte multiclasses. Voici comment procéder : considérons l'échantillon d'apprentissage $D_n = \{(x_1, y_1), ..., (x_n, y_n)\}$ avec $x_i \in \mathbb{R}^p$ et $y_i \in \{1, ..., G\}$. À chaque y_i est associé le vecteur $\mathbf{y_i} = (y_{i1}, ..., y_{iG})$ de dimension G défini par : si $y_i = j$ alors

$$\mathbf{y_i} = \left(-\frac{1}{G-1}, ..., 1, ..., -\frac{1}{G-1} \right)$$

où 1 est en j$^{\text{ème}}$ position.

Il s'agit d'estimer $\mathbf{f}(x) = (f_1(x), ..., f_G(x))$ sous la contrainte : $\forall x \in \mathbb{R}^p$, $\sum_{j=1}^{G} f_j(x) = 0$. Supposons que le label y_i de l'observation x_i soit égal à j et considérons la transformation L telle que $L(y_i)$, vecteur à G composantes, soit composé d'un 0 en j$^{\text{ème}}$ position et de 1 ailleurs. Lee propose alors de résoudre le problème d'optimisation (4.5).

$$\begin{array}{ll} \underset{f_1,...,f_G \in \mathcal{H}_K}{min} & \frac{1}{n} \sum_{i=1}^{n} L(\mathbf{y_i})[\mathbf{f}(x_i) - \mathbf{y_i}]_+ + \lambda \sum_{j=1}^{G} \|f_j\|^2_{\mathcal{H}_K} \\ \text{sous la contrainte} & \sum_{j=1}^{G} f_j(x) = 0,\ \forall x \end{array} \tag{4.5}$$

où $(\mathbf{f}(x_i) - \mathbf{y_i})_+$ signifie $[[f_1(x_i) - y_{i1}]_+,\ ...,\ [f_G(x_i) - y_{iG}]_+]$.

La règle de classification est fournie par :

$$\text{classe de x} = \arg \max_j f_j(x)$$

où chacune des G fonctions est de la forme :

$$f_j(x) = b_j + \sum_{i=1}^{n} c_{ij} k(x, x_i),\ j = 1,\ ...,\ G$$

Pour le détail de l'obtention de ces G fonctions, nous renvoyons à Lee [Lee et al. (2001)].

Les approches unifiées sont conceptuellement plus compliquées que les approches OVA et AVA sans amélioration significative des performances. En effet, en 2002, Hsu et Lin [Hsu and Lin (2002)] comparent l'ensemble des approches SVM multiclasses et concluent en la supériorité de l'approche AVA. En 2004, Rifkin et Klautau [Rifkin and Klautau (2004)] font une revue des approches SVM multiclasses les plus courantes. Ils concluent que l'approche OVA est au moins aussi compétitive que les existantes mais conceptuellement beaucoup plus intuitive. Dans ces deux papiers récents, les approches unifiées sont écartées.
Ainsi, dans le cinquième chapitre de validation, les deux approches SVM multiclasses, OVA et AVA qui ont largement prouvé leur efficacité face aux approches unifiées, serviront de base de comparaison de performances.

Dans la suite de ce chapitre, nous présentons des méthodes de classification multiclasses basées sur des approches de réduction de dimension supervisée type PLS. Nous avons évoqué dans le chapitre précédent le fait que la régression PLS n'est pas adaptée à la prédiction de variables catégorielles à plus de deux modalités. Nous étudions ce contexte dans la suite de ce chapitre.

4.2 De l'Analyse Discriminante PLS à la Régression PLS Discriminante

Soit un ensemble de n individus $x_1, ..., x_n$ décrits par p variables. À chaque observation $x_i,\ i = 1, ..., n$ est associée une valeur $y_i \in \{1, ..., G\}$. y_i fournit la classe d'appartenance de l'individu i. L'échantillon d'apprentissage D est alors défini par : $D = \{(x_1, y_1), ..., (x_n, y_n)\}$. Posons Y, le tableau disjonctif complet dérivé de y définie par :

$$Y = \begin{pmatrix} 1_{n_1} & 0_{n_1} & \dots & 0_{n_1} \\ 0_{n_2} & 1_{n_2} & \dots & 0_{n_2} \\ \vdots & \vdots & \ddots & 1_{n_g-1} \\ 0_{n_g} & 0_{n_g} & \dots & 0_{n_g} \end{pmatrix}$$

où les $\{n_i\}_{i=1}^{g}$ définissent le nombre d'individus de chacune des $g \in \{1, ...G\}$ classes.

4.2.1 L'Analyse Discriminante PLS (PLS-DA)

L'Analyse Discriminante PLS (PLS-DA) est définie comme la régression PLS2 de X sur le tableau disjonctif complet dérivé de y. Nous pouvons donc voir la PLS-DA comme une approche modélisant un ensemble de variables binaires à partir d'un ensemble de variables explicatives X. Rappelons que la régression PLS optimise le critère de Tucker :

$$cov^2(X_{h-1}w_h, Y_{h-1}c_h) = var(X_{h-1}w_h) corr^2(X_{h-1}w_h, Y c_h) var(Y_{h-1}c_h)$$

Or le terme de pénalité sur Y, $var(Y_{h-1}c_h)$, n'a pas de sens lorsque y est une variable catégorielle. La suite de ce chapitre présente des méthodes alternatives à la PLS-DA.

4.2.2 La régression PLS discriminante (PLS-D)

L'idée première est de retirer le terme de pénalité sur Y, $var(Yc_1)$, du scénario usuel de la PLS. Barker et Rayens [Barker and Rayens (2003)] suggèrent alors de maximiser le critère (4.6)

$$\begin{aligned} &var\left(Xw_1^{PLS-D}\right)corr^2\left(Xw_1^{PLS-D}, Yc_1^{PLS-D}\right) \\ &\text{sous les contraintes de normalité } \|w_1^{PLS-D}\| = \|c_1^{PLS-D}\| = 1 \end{aligned} \tag{4.6}$$

Construction de la première composante PLS-D

Barker et Rayens [Barker and Rayens (2003)] montrent que w_1^{PLS-D} correspond au vecteur propre normalisé associé à la plus grande valeur propre de la matrice $X^TY(Y^TY)^{-1}Y^TX$. $t_1^{PLS-D} = Xw_1^{PLS-D}$ correspond donc au vecteur propre associé à la plus grande valeur propre de la matrice $XX^TY(Y^TY)^{-1}Y^T$.

$$XX^TY(Y^TY)^{-1}Y^T \underbrace{Xw_1^{PLS-D}}_{t_1^{PLS-D}} = \lambda_1 \underbrace{Xw_1^{PLS-D}}_{t_1^{PLS-D}} \tag{4.7}$$

Construction de la $h^{ième}$ composante PLS-D

Les composantes PLS-D suivantes, $t_h = X_{h-1}w_h^{PLS-D}$ sont obtenues par maximisation du critère (4.8)

$$\begin{aligned} &var\left(X_{h-1}w_h^{PLS-D}\right)corr^2\left(X_{h-1}w_h^{PLS-D}, Y_{h-1}c_h^{PLS-D}\right) \\ &\text{sous les contraintes de normalité } \|w_h^{PLS-D}\| = \|c_h^{PLS-D}\| = 1 \end{aligned} \tag{4.8}$$

Barker et Rayens [Barker and Rayens (2003)] montrent que w_h^{PLS-D} correspond au vecteur propre associé à la plus grande valeur propre de la matrice $X_{h-1}^TY_{h-1}(Y_{h-1}^TY_{h-1})^{-1}Y_{h-1}^TX_{h-1}$. Ainsi, $t_h^{PLS-D} = X_{h-1}w_h^{PLS-D}$ correspond au vecteur propre associé à la plus grande valeur propre de la matrice $X_{h-1}X_{h-1}^TY_{h-1}(Y_{h-1}^TY_{h-1})^{-1}Y_{h-1}^T$.

$$X_{h-1}X_{h-1}^TY_{h-1}(Y_{h-1}^TY_{h-1})^{-1}Y_{h-1}^T \underbrace{X_{h-1}w_h^{PLS-D}}_{t_h^{PLS-D}} = \lambda_1 \underbrace{X_{h-1}w_h^{PLS-D}}_{t_h^{PLS-D}} \tag{4.9}$$

Notons que, comme dans le cas de la PLS-R, on souhaite exprimer les composantes PLS-D en fonction des variables d'origine X et non en fonction des matrices résiduelles X_{h-1}. La relation fournie par l'équation (2.11) est toujours valable ici.

En notant $T = \left[t_1^{PLS-D}, ..., t_r^{PLS-D}\right]$, $W = \left[w_1^{PLS-D}, ..., w_r^{PLS-D}\right]$ et $P = [p_1, ..., p_r]$ où p_{hj} correspond au coefficient de régression de $x_{h-1,j}$ sur t_h^{PLS-D}, alors

$$T = XW(P^TW)^{-1} = XW^*$$

Ajoutons, que comme pour la PLS-R, il est possible de calculer les composantes PLS-D récursivement (Cf. équation (2.13)).

Remarque 18 *Posons* $\widetilde{Y} = Y(Y^TY)^{-1/2}$. $\widetilde{Y}$ *représente le tableau disjonctif complet dérivé de y, normalisé par la racine carrée des effectifs. Dans ce contexte,* w_1^{PLS-D} *correspond donc au vecteur propre normalisé associé à la plus grande valeur propre de la matrice* $X^T\widetilde{Y}\widetilde{Y}^TX$ *et* $t_1^{PLS} = Xw_1^{PLS-D}$. *Par conséquent, la première composante PLS-D est équivalente à la première composante PLS de* $\widetilde{Y}$ *sur* X. *Ce résultat s'étend aux composantes suivantes : la* $h^{ième}$ *composante PLS-D correspond à la* $h^{ième}$ *composante PLS de* $\widetilde{Y}$ *sur* X. □

Modèle final de la PLS-D

Reste à construire un modèle reliant y aux composantes PLS-D. À l'issue de la PLS-D de y sur X, nous disposons de r composantes PLS-D où $r = rang(X)$. Le modèle final consiste en l'utilisation de méthodes supervisées type régression logistique ou analyse discriminante de y sur les m premières composantes PLS-D, où m est choisi par validation croisée.

4.2.3 Quelques interprétations de la PLS-D et de la PLS-DA

Dans cette section, nous présentons quelques liens entre la PLS-DA, la PLS-D et l'analyse discriminante.

La PLS-DA est définie comme la régression PLS d'un ensemble de variable binaire Y sur un ensemble de variables explicatives X. Les premières composantes PLS $t_1 = Xw_1$ et $u_1 = Yc_1$ sont obtenues par maximisation du critère (4.10)

$$cov^2(Xw_1, Yc_1) = var(Xw_1)corr^2(Xw_1, Yc_1)var(Yc_1) \tag{4.10}$$

sous les contraintes $\|w_1\| = \|c_1\| = 1$. w_1 correspond au vecteur propre normalisé de la matrice X^TYY^TX associé à la plus grande valeur propre.

Barker et Rayens [Barker and Rayens (2003)] ont noté que ce critère est en fait inapproprié pour des tâches de classification en mettant en avant le fait que le terme de var(Yc_1) n'a pas réellement de sens. Ils proposent alors de maximiser le critère (4.11)

$$\frac{cov^2(Xw_1, Yc_1)}{var(Yc_1)} = var(Xw_1)corr^2(Xw_1, Yc_1) \tag{4.11}$$

sous les contraintes $\|w_1\| = \|c_1\| = 1$. w_1 correspond au vecteur propre normalisé de la matrice $X^TY(Y^TY)^{-1}Y^TX$ associé à la plus grande valeur propre. Nous pouvons d'ores et déjà fournir une interprétation de la PLS-D intéressante :
La composante t_1^{PLS-D} est le résultat d'une analyse des redondances de X sur Y ou, de manière équivalente, correspond à la première composante principale d'une analyse en composante principale de X projeté sur l'espace engendré par les colonnes de la matrice Y.

Enfin, l'analyse discriminante est obtenue par maximisation du critère (4.12)

$$corr^2(Xw_1, Yc_1) \tag{4.12}$$

sous les contraintes $\|w_1\| = \|c_1\| = 1$. w_1 correspond au vecteur propre normalisé associé à la plus grande valeur propre de la matrice $(X^TX)^{-1}X^TY(Y^TY)^{-1}Y^TX$.

Les composantes suivantes sont obtenues en réitérant le procédé à partir des matrices résiduelles X_{h-1} de la régression de X sur les composantes précédentes $t_1, ..., t_{h-1}$ dans les critères (4.10), (4.11) et (4.12).

Remarque 19 *Notons que t_h^{PLS-D} est le résultat d'une analyse des redondances de X_{h-1} sur Y_{h-1} ou, de manière équivalente, correspond à la première composante principale de l'ACP de X_{h-1} projeté sur l'espace engendré par les colonnes de la matrice Y_{h-1}.* □

On peut alors remarquer que la PLS-D établit un pont entre la PLS-DA et l'analyse discriminante. En effet, considérons le premier vecteur propre associé à la plus grande valeur propre de la matrice

$$((1-\alpha)X^TX + \alpha\mathbb{I})^{-1}X^TY((1-\beta)Y^TY + \beta\mathbb{I})^{-1}Y^TX \tag{4.13}$$

Pour $\alpha = \beta = 1$, nous obtenons la PLS-DA.
Pour $\alpha = 1$ et $\beta = 0$, nous obtenons la PLS-D.
Pour $\alpha = \beta = 0$, nous obtenons l'analyse discriminante.

Extension de la PLS-D au cadre non linéaire : la Kernel PLS-D

Il s'avère que l'on peut étendre la PLS-D au cadre non linéaire via calcul des composantes dans un espace de Hilbert à noyau reproduisant. Nous ne décrivons pas dans les détails la KPLS-D, proposée par Rosipal et al [Rosipal et al. (2003)], mais relevons que la construction des composantes PLS-D par les équations (4.7) et (4.9) ne fait intervenir les observations qu'à travers leur produit scalaire.

La règle de décision finale de la Kernel PLS-D est obtenue de la même manière que précédemment, en réalisant par exemple, une régression logistique multinomiale de y sur $t_1^{KPLS-D}, ..., t_m^{KPLS-D}$.

4.3 La Régression Logistique Multinomiale PLS (LM-PLS)

Dans cette section, nous proposons une adaptation de la Régression Logistique PLS au cadre multiclasses : la Régression Logistique Multinomiale PLS (LM-PLS). Supposons que y soit une variable catégorielle à G modalités.
En s'appuyant sur la construction de variables latentes, l'idée principale de la LM-PLS est de rechercher un espace discriminant engendré par les composantes LM-PLS notées $t_1^{LM-PLS}, ..., t_m^{LM-PLS}$, où un simple modèle type régression logistique multinomiale suffit. Ainsi, l'objectif est de construire une représentation des données où $G-1$ hyperplans séparent les G classes. On cherche donc à fournir par cette nouvelle représentation, «l'ossature discriminante» des données originales. L'algorithme de la LM-PLS se décompose en deux étapes :

1. Construction des composantes LM-PLS, $t_1^{LM-PLS}, ..., t_m^{LM-PLS}$

2. Régression Logistique Multinomiale de y sur les m composantes LM-PLS retenues

Dans la suite de ce paragraphe, nous présentons la méthode en détail suivie d'un encart algorithmique récapitulatif.

4.3.1 Construction des composantes LM-PLS

Cette section décrit la construction itérative des composantes LM-PLS.

Construction de la première composantes LM-PLS t_1^{LM-PLS}

La première composante LM-PLS t_1^{LM-PLS} fournit le premier axe discriminant.

Étape 1 : Calcul des coefficients $a_{k1}^1, ..., a_{k,G-1}^1$ de $\mathbf{x_k}$ dans la régression logistique multinomiale de y sur $\mathbf{x_k}, k = 1, ..., p$:

$$\log\left(\frac{\mathbb{P}(y = g/x = x_k)}{\mathbb{P}(y = G/x = x_k)}\right) = a_{kg}^1 \mathbf{x_k} + b_{kg}^1,\ g = 1, ...G-1 \tag{4.14}$$

Étape 2 : Normalisation de chaque vecteur colonne de la matrice $A_1 = (a_{ij}^1)_{\substack{i=1,...,p \\ j=1,...,G-1}}$:

$$w_{1j} = \left(a_{1j}^1, a_{2j}^1, ..., a_{pj}^1\right)^T / \sqrt{\sum_{i=1}^{p} (a_{i,j}^1)^2} \tag{4.15}$$

et

$$W_1 = [w_{11}\ w_{12}\ ...\ w_{1,G-1}] \tag{4.16}$$

Étape 3 : Calcul de la première matrice LM-PLS :

$$T_1 = XW_1 \tag{4.17}$$

Étape 4 : Calcul de la première composante LM-PLS comme la première composante principale t_1^{LM-PLS} de la matrice T_1 :

$$t_1^{LM-PLS} = T_1 v_1 \tag{4.18}$$

Étape 5 : Expression de la première composante LM-PLS t_1^{LM-PLS} en fonction de X :

$$t_1^{LM-PLS} = XW_1 v_1 = Xw_1^* \tag{4.19}$$

Ainsi, la composante t_1^{LM-PLS} est construite par une succession de p MLR simples (pas d'inversion de matrice) et d'une analyse en composantes principales sur $G-1$ variables.

L'architecture algorithmique de la LM-PLS est donc très proche de celle de la PLS-LR. La différence principale réside dans l'obtention d'une matrice de coefficients W_1 de dimension $p \times G-1$ plutôt qu'un vecteur w_1 de dimension $p \times 1$. Considérons la j$^{\text{ème}}$ ligne de la matrice W_1. Elle est composée des $G-1$ poids associés à $\mathbf{x_j}$ estimés lors de la régression logistique multinomiale de y sur $\mathbf{x_j}$. La g$^{\text{ième}}$ valeur de la j$^{\text{ième}}$ ligne traduit l'importance de la variable $\mathbf{x_j}$ dans la prédiction de la g$^{\text{ième}}$ modalité de y. Cette matrice a donc la propriété intéressante de capturer les relations qui existent entre les variables explicatives et les différentes modalités de y. Il devient alors tout à fait naturel de construire la matrice LM-PLS comme le produit matriciel $T_1 = XW_1$. C'est un schéma tout à fait similaire à la régression logistique PLS. Pour le moment, T_1 est une matrice de dimension $n \times G-1$ et on souhaite condenser cette information pour ne produire, à la fin de cette étape, qu'une seule composante LM-PLS. On réalise donc une analyse en composantes principales de T_1. La perte d'information engendrée par cette ACP pourra être récupérée par la construction des composantes LM-PLS suivantes.

Construction de la $h^{ième}$ composante LM-PLS t_h^{LM-PLS}

Soit $X_{h-1} = \left[\mathbf{x_{h-1,1}}, ..., \mathbf{x_{h-1,p}}\right]$ la matrice résiduelle de la régression de X sur les $(h-1)$ composantes LM-PLS précédentes $t_1^{LM-PLS}, ..., t_{h-1}^{LM-PLS}$.

Remarque 20 *Notons que tout vecteur généré par les colonnes de X_{h-1} est orthogonal à $t_1^{LM-PLS}, ..., t_{h-1}^{LM-PLS}$. La $h^{ième}$ composante LM-PLS t_h^{LM-PLS} capture donc l'information discriminante résiduelle (i.e. absente des $h-1$ précédentes).* □

Étape 1 : Calcul des coefficients $a_{k1}^h, ..., a_{k,G-1}^h$ de $\mathbf{x_{h-1,k}}$ dans la régression logistique multinomiale de y sur $t_1^{LM-PLS}, ..., t_{h-1}^{LM-PLS}$ et $\mathbf{x_{h-1,k}}, k = 1, ..., p$: pour $g = 1, ...G-1$

Étape 2 : Normalisation de chaque vecteur colonne de la matrice $A_h = (a_{ij}^h)_{\substack{i=1,...,p \\ j=1,...,G-1}}$:

$$w_{hj} = \left(a_{1j}^h, a_{2j}^h, ..., a_{pj}^h\right)^T / \sqrt{\sum_{i=1}^{p} (a_{ij}^h)^2} \tag{4.20}$$

et

$$W_h = [w_{h1}\ w_{h2}\ ...\ w_{h,G-1}] \tag{4.21}$$

Étape 3 : Calcul de la $h^{\text{ième}}$ matrice LM-PLS :

$$T_h = X_{h-1} W_h \tag{4.22}$$

Étape 4 : Calcul de la $h^{\text{ième}}$ composante LM-PLS comme la première composante principale t_h^{LM-PLS} de la matrice T_h

$$t_h^{LM-PLS} = T_h v_h \tag{4.23}$$

Étape 5 : Expression de la $h^{\text{ième}}$ composante LM-PLS t_h^{LM-PLS} en fonction de X

$$t_h^{LM-PLS} = X w_h^* \tag{4.24}$$

En conséquence, la construction de t_h^{LM-PLS} nécessite l'inversion de p matrices de dimension $h \times h$ et d'une analyse en composantes principales sur $G-1$ variables.

4.3.2 Expression des composantes LM-PLS en fonction des variables d'origine

L'expression des composantes LM-PLS en fonction des variables d'origine est une étape fondamentale pour l'analyse de nouvelles observations (qui n'ont pas servi à la construction du modèle). Soit X_{new} de nouvelles observations. Le produit matriciel $T_{new} = X_{new} W^*$ permet de calculer les valeurs des composantes LM-PLS pour les nouvelles observations.

Calcul des W_h^*

a. La première composante LM-PLS t_1^{LM-PLS} est fonction des variables d'origine :

$$T_1 = XW_1 \tag{4.25}$$

et

$$t_1^{LM-PLS} = T_1 v_1 = X \underbrace{W_1^* v_1}_{w_{1*}} \tag{4.26}$$

où $W_1^* = W_1$

b. La seconde composante LM-PLS t_2^{LM-PLS} s'exprime comme combinaison des vecteurs colonnes de la matrice résiduelle de la régression des variables d'origine sur t_1^{LM-PLS}. À partir de $X = t_1^{LM-PLS} p_1^t + X_1$ et $T_2 = X_1 W_2$, nous obtenons :

$$\begin{aligned} T_2 &= X_1 W_2 = \left(X - t_1^{LM-PLS} p_1^T\right) W_2 \\ &= \left(X - T_1 v_1 p_1^T\right) W_2 = \left(X - X W_1^* v_1 p_1^T\right) W_2 \\ &= X \underbrace{\left(I - W_1^* v_1 p_1^T\right) W_2}_{W_2^*} \end{aligned} \tag{4.27}$$

et

$$t_2^{LM-PLS} = T_2 v_2 = X \underbrace{W_2^* v_2}_{w_2^*} \tag{4.28}$$

c. De manière équivalente, nous pouvons montrer que T_h et t_h^{LM-PLS} s'expriment en fonction des variables d'origine par les équations (4.29) et (4.30).

$$T_h = X \underbrace{\left(I - \sum_{i=1}^{h-1} W_i^* v_i p_i^t\right) W_h}_{W_h^*} \tag{4.29}$$

et

$$t_h^{LM-PLS} = X \underbrace{W_h^* v_h}_{w_h^*} \tag{4.30}$$

4.3.3 Modèle final

On réalise alors la régression logistique multinomiale de y sur les m premières composantes LM-PLS retenues par validation croisée. Pour $g = 1, ...G-1$, on a

$$\log\left(\frac{\mathbb{P}\left(y = g / t_1^{LM-PLS}, ..., t_m^{LM-PLS}\right)}{\mathbb{P}\left(y = G / t_1^{LM-PLS}, ..., t_m^{LM-PLS}\right)}\right) = \sum_{h=1}^{m} c_{hg} t_h^{LM-PLS} + b_g = h_g(X) \tag{4.31}$$

Ainsi, la régression logistique multinomiale de y sur les m premières composantes LM-PLS fournit une estimation naturelle des probabilités conditionnelles d'appartenance des n individus

à chacune des g classes $g = \{1, ..., G\}$ notée par $p_g(x) = P(y = g/X = x)$.
La règle de classification résultante est donnée par (4.32)

$$c(x) = \underset{g \in \{1,...,G\}}{\operatorname{argmax}} p_g(x). \tag{4.32}$$

où pour $g = 1, ..., G-1$, nous avons,

$$p_g(x) = \frac{exp[h_g(x)]}{1 + \sum_{g=1}^{G-1} exp[h_g(x)]} \tag{4.33}$$

et

$$p_G(x) = \frac{1}{1 + \sum_{g=1}^{G-1} exp[h_g(x)]} \tag{4.34}$$

Remarque 21 *L'estimateur du maximum de vraisemblance de la régression logistique dépend de la configuration spatiale des individus. Dans le seul cas d'un recouvrement des différentes classes d'individus, l'estimateur du maximum de vraisemblance existe et est unique ([Allison (1999)] ; [Albert and Anderson (1984)]). Par conséquent, en appliquant la régression logistique multinomiale sur les composantes LM-PLS, il est fondamental de considérer la configuration spatiale des individus. Dans une situation de séparation complète ou quasi-complète des individus d'au moins un des G groupes, l'analyse discriminante de Fisher de y sur les m composantes LM-PLS retenues pourrait être utilisée.*
On peut préférer l'utilisation systématique de l'approche de Bull [Bull et al. (2002)] consistant à réduire le biais de l'estimateur du maximum de vraisemblance par extension de l'approche de Firth [Firth (1993)] intialement prévue pour réduire le biais du maximum de vraisemblance des modèles de la famille exponentielle. Nous n'avons pas exploré cette voie.

En ce sens, la LM-PLS peut-être vue comme une méthode de réduction de dimension supervisée et servir de preprocessing à tout classifieur multiclasses. □

4.3.4 Algorithme de la LM-PLS

Pour résumer la section précédente nous présentons la LM-PLS sous forme algorithmique.

TAB. 4.1: Algorithme de la Régression Logistique Multinomiale PLS

Algorithme de la régression logistique multinomiale PLS

A. Calcul des composantes LM-PLS $t_1^{LM-PLS}, ..., t_m^{LM-PLS}$

Calcul de t_1^{LM-PLS}

1. MLR de y sur chaque $\mathbf{x_j}, j = 1, ..., p$
   $\Rightarrow$ coefficients de régression $a_{j1}^1, ..., a_{j,G-1}^1$ de $\mathbf{x_j}$
2. Normalisation des vecteurs colonnes de $A_1 = (a_{ij}^1)_{\substack{i=1,...,p \\ j=1,...,G-1}} \Rightarrow W_1$
3. La première matrice LM-PLS est $T_1 = XW_1$
4. Calcul de t_1 comme la première composante principale de T_1
   $\Rightarrow t_1^{LM-PLS} = T_1 v_1$
5. Expression de t_1 en fonction de X : $t_1^{LM-PLS} = XW_1 v_1 = Xw_1^*$

Calcul de t_h^{LM-PLS}

1. Régression de chaque $\mathbf{x_j}, j = 1, ..., p$ sur $t_1^{LM-PLS}, ..., t_{h-1}^{LM-PLS}$
   $\Rightarrow$ matrice résiduelle : $X_{h-1} = [\mathbf{x_{h-1,1}}, ..., \mathbf{x_{h-1,p}}]$
2. MLR de y sur $t_1^{LM-PLS}, ..., t_{h-1}^{LM-PLS}$ et chaque $\mathbf{x_{h-1,j}},\ j = 1, ..., p$
   $\Rightarrow$ coefficients de régression $a_{j1}^h, ..., a_{j,G-1}^h$ de $\mathbf{x_{h-1,j}}$
3. Normalisation des vecteurs colonnes de $A_h = (a_{jg}^h)_{\substack{j=1,...,p \\ g=1,...,G-1}} \Rightarrow W_h$
4. La $h^{\text{ième}}$ matrice LM-PLS est $T_h = X_{h-1} W_h$
5. Calcul de t_h comme la première composante principale de T_h
   $\Rightarrow t_h^{LM-PLS} = T_h v_h$
6. Expression de t_h en fonction de X : $t_h^{LM-PLS} = Xw_h^*$

B. Régression logistique multinomiale de y **sur** $t_1^{LM-PLS}, ..., t_m^{LM-PLS}$

En conclusion du paragraphe, nous mettons en lumière trois propriétés importantes de la LM-PLS :

1. Contrairement à la plupart des méthodes de classification, la LM-PLS nécessite uniquement l'inversion de matrice de faible dimension (nombre de composantes LM-PLS+1).
2. De par son algorithme, la LM-PLS autorise la manipulation de données où le nombre de variables est supérieur au nombre d'observations.
3. De par son algorithme, la LM-PLS gère les données hautement corrélées.

Nous allons, dans la suite de ce document étendre la LM-PLS au cadre non linéaire.

4.4 La Kernel Logistique Multinomiale PLS (KLM-PLS)

Inspiré de l'approche de la Kernel Logistique PLS, nous proposons une extension non linéaire de la régression ML-PLS basée sur des transformations de type Empirical Kernel

Map : la Kernel Logistique Multinomial PLS (KLM-PLS). L'algorithme de la KLM-PLS se décompose donc en trois étapes :

1. Calcul de la matrice de Gram K.
2. Construction des composantes KLM-PLS, c'est-à-dire les composantes ML-PLS dérivées de K.
3. Régression logistique multinomiale de y sur les m composantes KLM-PLS retenues.

4.4.1 Algorithme de la Kernel Logistique Multinomiale PLS

Le tableau 4.2 présente l'algorithme de la KLM-PLS.

TAB. 4.2: Algorithme de la Kernel Logistique Multinomiale PLS

Algorithme de la Kernel Logistique Multinomiale PLS

A. Construction de la matrice de Gram K

B. Construction des composantes KLM-PLS, $t_1^{KLM-PLS}, ..., t_m^{KLM-PLS}$

Construction de la première composante KLM-PLS $t_1^{KLM-PLS}$

1. MLR de y sur chaque $k_j, j = 1, ..., n$
   $\Rightarrow$ coefficients de régression $a_{j1}^1, ..., a_{j,G-1}^1$ de k_j
2. Normalisation des vecteurs colonnes de $A_1 = (a_{ij})_{\substack{i=1,...,n \\ j=1,...,G-1}} \Rightarrow W_1$
3. La première matrice KLM-PLS est définie par $T_1 = KW_1$
4. Calcul de $t_1^{KLM-PLS}$ comme la première composante principale de T_1
   $\Rightarrow t_1^{KLM-PLS} = T_1 v_1$
5. Expression de $t_1^{KLM-PLS}$ en fonction de K : $t_1^{KLM-PLS} = KW_1 v_1 = Kw_1^*$

Construction de la $h^{ième}$ composante KLM-PLS $t_h^{KLM-PLS}$

1. Régression de chaque $k_j, j = 1, ..., n$ sur $t_1^{KLM-PLS}, ..., t_{h-1}^{KLM-PLS}$
   $\Rightarrow$ matrice résiduelle $K_{h-1} = [k_{h-1,1}, ..., k_{h-1,n}]$
2. MLR de y sur $t_1^{KLM-PLS}, ..., t_{h-1}^{KLM-PLS}$ et chaque $k_{h-1,j}, j = 1, ..., n$
   $\Rightarrow$ coefficients de régression $a_{j1}^h, ..., a_{j,G-1}^h$de $k_{h-1,j}$
3. Normalisation des vecteurs colonnes de $A_h = (a_{jg}^h)_{\substack{j=1,...,n \\ g=1,...,G-1}} \Rightarrow W_h$
4. la $h^{\text{ième}}$ matrice KLM-PLS est définie par $T_h = K_{h-1} W_h$
5. Calcul de $t_h^{KLM-PLS}$ comme la première composante principale de T_h
   $\Rightarrow t_h^{KLM-PLS} = T_h v_h$
6. Expression de $t_h^{KLM-PLS}$ en fonction de K : $t_h^{KLM-PLS} = Kw_h^*$

C. MLR de y sur les m premières composantes KLM-PLS

Mettons en lumière au travers de cette présentation le fait que cette implémentation nécessite uniquement l'inversion de matrices de faible dimension (nombre de composantes KLM-PLS + 1).
Nous ne présentons pas en détail les étapes de l'algorithme de la KLM-PLS puisqu'il découle de manière naturelle de la LM-PLS.

Remarque 22
1. Les avantages algorithmiques de la KLM-PLS sont analogues à ceux de la Kernel Logistique

PLS (gestion de matrice $n \times n$, gestion de matrices rectangulaires, ...).
2. Il est tout à fait possible de considérer toute transformation non linéaire des variables originales et d'appliquer la LM-PLS à cette nouvelle représentation des individus mais les arguments valables pour la KL-PLS restent identiques ici. □

4.4.2 Modèle final

On réalise une régression logistique multinomiale de y sur les m premières composantes KLM-PLS retenues. Pour $g = 1, ...G-1$, on a,

$$\log\left(\frac{\mathbb{P}(y=g/t_1^{KLM-PLS},...,t_m^{KLM-PLS})}{\mathbb{P}(y=G/t_1^{KLM-PLS},...,t_m^{KLM-PLS})}\right) = \sum_{h=1}^{m} c_{hg} t_h^{KLM-PLS} + b_g = h_g(K) \tag{4.35}$$

Ainsi, la KLM-PLS fournit une estimation naturelle de la probabilité conditionnelle d'appartenance d'un individu à la classe $g = \{1, ..., G\}$ sachant $K = k$ notée par $p_g(k) = P(y = g/K = k)$. La règle de classification résultante est donnée par (4.36)

$$c(k) = \underset{g\in\{1,...,G\}}{\operatorname{argmax}}\, p_g(k). \tag{4.36}$$

où pour $g = 1, ..., G-1$, nous avons,

$$p_g(k) = \frac{exp[h_g(k)]}{1+\sum_{g=1}^{G-1} exp[h_g(k)]} \quad \text{et} \quad p_G(k) = \frac{1}{1+\sum_{g=1}^{G-1} exp[h_g(k)]}$$

4.5 Conclusion

L'approche la plus répandue des SVM multiclasses consiste à décomposer un problème à G (respectivement $G(G-1)/2$) classes en une collection de G (respectivement $G(G-1)/2$) sous-problèmes de la forme «one-versus-all » (respectivement «all-versus-all ») et fusionner les prédictions de sorte à attribuer à chacun des individus une classe unique. Cette approche est algorithmiquement coûteuse car elle nécessite, au minimum (pour OVA), la résolution de G problèmes d'optimisation quadratique.

Ainsi, dans ce chapitre, nous présentons des alternatives basées sur des approches de réduction de dimension supervisées. Dans ce contexte, nous proposons une extension de la régression logistique PLS au cas où la variable à prédire est polytomique à plus de deux modalités : la Régression Logistique Multinomiale PLS ainsi qu'une version non linéaire de cette approche, peu sensible au nombre de variables, basée sur des transformations de type Empircial Kernel Map : la Kernel Logistique Multinomiale PLS. Le prochain chapitre de validation évalue les performances de ces différentes approches.

Chapitre 5

Validation

Ce chapitre est dédié à l'évaluation des méthodes décrites dans les chapitres précédents. Il s'agit de quantifier et comparer le pouvoir discriminant des approches développées dans cette thèse selon leurs performances par rapport aux méthodes les plus performantes de la littérature (e.g. SVM, KLR, ...).

5.1 La validation croisée et la sélection de modèles

Le moyen le plus simple, le plus répandu et sans nul doute le plus fiable pour évaluer la qualité d'un classifieur est de mesurer ses performances de prédiction sur un ensemble d'observations, dit «échantillon de test», qui n'a pas participé à la construction du modèle. Ce processus d'évaluation a pris le nom de validation croisée (CV = **C**ross **V**alidation). Quelque soit la méthode, la capacité prédictive d'un modèle ne peut, en effet, se juger que sur des données indépendantes de celles qui ont participé à la construction du modèle. Lorsque l'on dispose d'un nombre suffisamment important d'observations, on partage les données en plusieurs sous-ensembles :

i. L'ensemble d'apprentissage sert à estimer chaque modèle en compétition.

ii. L'ensemble de validation sert à choisir le meilleur modèle, c'est-à-dire celui qui réalise les meilleures prédictions.

iii. L'ensemble de test sert uniquement à estimer la performance du modèle retenu.

On peut ainsi choisir le «bon» modèle quelque soit sa nature comme celui qui minimise le taux d'erreur sur l'ensemble de validation. La performance du modèle est alors mesurée sur l'échantillon de test. C'est cette mesure qui est soumise à comparaison de deux modèles. Par conséquent, pour évaluer effectivement les taux d'erreur que l'on peut espérer avec un modèle, il faut donc réaliser une double validation. Une première afin de régler les paramètres d'ajustement (paramètre du noyau, C, λ, nombre de composantes, ...) reliés à la complexité des fonctions admissibles et une deuxième pour mesurer la performance du modèle construit avec les paramètres de réglage ajustés. Cette double utilisation de la validation croisée est coûteuse mais est indispensable pour une évaluation fiable du modèle finalement fourni. Une procédure qui confondrait la phase de réglage des paramètres et l'évaluation des taux d'erreur

fournirait une estimation trop optimiste de la qualité du modèle.

Si les données sont en nombre insuffisant, on utilise la technique de validation croisée qui consiste à partager les données en K-sous ensembles disjoints de même taille et à calculer l'erreur de prédiction moyenne sur chacun de ces sous-ensembles, les $K-1$ autres blocs formant l'échantillon d'apprentissage. Pour $K = n$, nous retrouvons la méthode dite du *leave-one-out* introduite par Allen [Allen. (1974)]. Ce dernier choix n'est intéressant que lorsque n est relativement petit.
Afin de mesurer les performances de classifieurs, la validation croisée va être utilisée tout au long de ce chapitre. Notons que les performances d'un classifieur peuvent s'exprimer sous bien des formes : par une matrice de confusion, à partir de laquelle est calculé le pourcentage de bonne classification ; ou encore par la sensibilité, la spécificité et l'aire sous la courbe ROC (AUC : **A**era **U**nder the roc **C**urve) dans le cadre binaire (pour plus de détails sur la courbe ROC, voir Hanley [Hanley (1989)]).

5.2 La Kernel Logistique PLS

Cette section est dédiée à l'évaluation des performances de la Kernel Logistique PLS.

5.2.1 Benchmarks

L'efficacité de la Kernel Logistique PLS est mesurée sur un ensemble de 11 benchmarks. Ces jeux de données sont disponibles à l'adresse suivante : http ://ida.first.fraunhofer.de/projects/bench/benchmarks.htm. Le tableau 5.1 fournit des informations élémentaires sur ces 11 benchmarks de classification binaire : le nombre de variables (p), le nombre d'observations de la base d'apprentissage (n) et le nombre d'observations composant la base de test (n_{test}). Un descriptif plus détaillé est disponible dans [Rätsch et al. (2001)] et sur le site web. Chaque jeu de données est composé de 100 échantillons d'apprentissage auxquels sont associés 100 échantillons de test.

TAB. 5.1: Description des benchmarks binaires

Data set	**dimension (p)**	**Taille de la base d'apprentissage (n)**	**Taille de la base de test n_{test}**
Banana	2	400	4600
Breast Cancer	9	200	77
Diabetis	8	468	300
German	20	700	300
Heart	13	170	100
Ringnorm	20	400	7000
Flare Solar	9	666	400
Thyroid	5	140	75
Titanic	3	150	2051
Twonorm	20	400	7000
Twonorm	21	400	4600

Pour chacun de ces 100 jeux de données, le protocole d'évaluation d'une méthode est le suivant :

1. Pour chacune des 100 partitions d'apprentissage, construire un modèle. L'ajustement de paramètres libres (e.g. paramètre(s) du noyau, paramètre C des SVM, nombre de composantes PLS sélectionnées, ...) s'effectue à partir des 5 premières partitions et sont déterminés par validation croisée.

2. Tester chacun de ces 100 modèles sur l'échantillon de test associé.

3. Calculer le taux d'erreur de chacun des 100 échantillons de test.

4. Calculer la moyenne $\pm$ l'écart type de ces 100 taux d'erreurs de test.

Ce protocole a été appliqué à de nombreuses méthodes. On retrouve un catalogue des performances de certaines méthodes (et notamment des SVM) à l'adresse internet suivante : http ://ida.first.fraunhofer.de/projects/bench/benchmarks.htm. Nous avons reportés sur le tableau 5.2 les performances relatives aux méthodes suivantes :

• Les Support Vector Machines (SVM) proposés par Rätsch et al. [Rätsch et al. (2001)].

• La Kernel Logistic Regression (KLR) (basée sur 20 partitions) proposée par Zhu et Hastie [Zhu and Hastie (2005)].

• La Kernel PLS-SVC (Kernel PLS suivie des SVM sur les composantes retenues) proposée par Rosipal et al. [Rosipal et al. (2003)].

• La Kernel Projection Machine (KPM) proposée par Zwald et al. [Zwald et al. (2004)].

Les trois premières méthodes sont à notre connaissance, sur ces benchmarks, les plus performantes en termes de qualité de classification. C'est la raison pour laquelle nous avons décidé de nous y référer pour mesurer les performances de la Kernel Logistique PLS (KL-PLS).

TAB. 5.2: Taux d'erreur de classification mesuré sur les échantillons de test (moyenne ± écart-type) pour SVM [Rätsch et al. (2001)], Kernel PLS-SVC [Rosipal et al. (2003)], Kernel Logistic Regression (KLR), Kernel Projection Machine [Zwald et al. (2004)] et KL-PLS. La dernière colonne fournit le paramètre de la gaussienne ainsi que le nombre de composantes KL-PLS retenues. L'astérisque simple «$*$» indique lorsque l'hypothèse nulle du test de Student apparié est rejetée au risque $\alpha = 0.05$. Lorsque les statistiques individuelles des taux d'erreur moyens ne sont pas disponibles, on utilise le test de Student : l'astérisque simple «$\star$»indique lorsque l'hypothèse nulle du test de Student est rejetée au risque $\alpha = 0.05$.

Data set	**SVM**	**KLR**	**KPM**	**KPLS-SVC**	**KL-PLS**	**Paramètres de la KL-PLS**
Banana	$11.5 \pm 0.5^{*}$	10.34 ± 0.46	10.91 ± 0.57	10.5 ± 0.4	10.7 ± 0.5	$(0.9, 10)$
B. Cancer	26.0 ± 4.7	25.92 ± 4.79	$28.73 \pm 4.42^{\star}$	25.1 ± 4.5	25.8 ± 4.4	$(50, 7)$
Diabetis	$23.5 \pm 1.7^{*}$	?	$23.77 \pm 1.69^{\star}$	23.0 ± 1.7	23.0 ± 1.7	$(60, 4)$
German	23.6 ± 2.1	23.53 ± 2.48	$24.09 \pm 2.38^{\star}$	23.5 ± 1.6	23.2 ± 2.1	$(20, 2)$
Heart	16.0 ± 3.3	15.80 ± 3.49	$17.35 \pm 3.54^{\star}$	16.5 ± 3.6	16.0 ± 3.2	$(20, 3)$
Ringnorm	$1.66 \pm 0.12^{*}$	$1.97 \pm 0.29^{\star}$	?	1.43 ± 0.10	1.44 ± 0.09	$(200, 2)$
F. Solar	$32.4 \pm 1.8^{*}$	33.66 ± 1.64	32.52 ± 1.78	32.4 ± 1.8	32.7 ± 1.8	$(12, 1)$
Thyroid	4.80 ± 2.19	5.00 ± 3.02	?	4.39 ± 2.1	4.35 ± 1.99	$(15, 6)$
Titanic	22.4 ± 1.0	22.39 ± 1.03	?	22.4 ± 1.1	22.4 ± 0.04	$(300, 2)$
Twonorm	$2.96 \pm 0.23^{*}$	2.45 ± 0.15	?	2.34 ± 0.11	2.37 ± 0.10	$(40, 1)$
Waveform	$9.88 \pm 0.43^{*}$	$10.13 \pm 0.47^{\star}$	?	9.58 ± 0.36	9.74 ± 0.46	$(15, 4)$
Rang Moyen	3.3	2.9	4.2	1.8	2	

Commentaire du tableau de benchmarks binaires

Le noyau gaussien ($k(x,y) = exp(-\|x-y\|^2/\sigma)$) a été utilisé pour l'ensemble des méthodes. Les performances de la KL-PLS dépendent donc du paramètre de la gaussienne et du nombre de composantes retenues. À l'instar des autres méthodes, ces paramètres libres sont fixés tels qu'ils minimisent le taux d'erreur moyen observé par validation croisée sur les 5 premiers jeux de données.

Nous pouvons dans un premier temps relever que les méthodes les plus performantes de la littérature font partie de la famille des méthodes à noyau décrites aux chapitres 2 et 3. Le noyau gaussien, du fait de sa souplesse, est utilisé par toutes ces méthodes. Par ailleurs, il n'est pas surprenant que les méthodes basées sur des principes de marge maximale fournissent de bonnes propriétés de généralisation (KLR, SVM, KPLS-SVC, KPM).

Nous disposons des taux d'erreur individuels (mesurés sur les 100 échantillons de test) uniquement pour les SVM et la KPLS-SVC. Afin de comparer ces méthodes à la KL-PLS, nous utilisons un test de Student apparié. Ces tests conduisent aux résultats suivants :

KL-PLS vs. SVM : Pour 10 jeux de données sur 11, la KL-PLS fournit des taux d'erreur moyens inférieurs aux SVM. Dans 6 cas sur 10 (Banana, Diabetis, Ringnorm, Flare-Solar, Twonorm et Waveform), l'hypothèse nulle est rejetée. Inversement, le taux d'erreur moyen des SVM est plus faible que celui de la KL-PLS sur 1 jeu de données (Flare-Solar) et dans ce cas, l'hypothèse nulle est rejetée.

KL-PLS vs. KPLS-SVC : Pour 4 jeux de données, la KL-PLS fournit des taux d'erreur moyens inférieurs à la KPLS-SVC mais l'hypothèse nulle n'est rejetée dans aucun de ces cas. À l'inverse, pour 6 jeux de données, les taux d'erreur moyens de la KPLS-SVC sont inférieurs à ceux de la KL-PLS et l'hypothèse nulle est rejetée dans 5 cas sur 6 (Banana, Brest-Cancer, Flare-Solar, Twonorm et Waveform). Ces bonnes performances de la KPLS-SVC peuvent être dûes à l'utilisation du classifieur SVM linéaire, de marge maximale, assurant un pouvoir de généralisation. Soulignons qu'un classifieur SVM linéaire nécessite l'ajustement d'un paramètre supplémentaire (C) ce qui peut être contraignant en pratique. D'un autre côté, la KL-PLS fournit des probabilités d'appartenance et les règles de décision ont été obtenues par seuillage des probabilités à 0.5. Un gain en performance potentiel peut être relié au paramètre de seuillage. L'utilisation d'un classifieur de type SVM linéaire pourrait également être envisagée. Les différences entre les deux classifieurs étant néanmoins très faibles, nous n'avons pas exploré ces pistes.

Nous ne disposons pas des taux d'erreur individuels (mesurés sur les 100 échantillons de test) pour KLR et KPM. Afin de comparer ces méthodes à la KL-PLS, nous utilisons donc le test de Student (moins puissant que le test de Student apparié). Ces tests conduisent aux résultats suivants :

KL-PLS vs. KLR : Pour 8 jeux de données sur 10, la KL-PLS fournit des taux d'erreur moyens inférieurs à ceux obtenus par la KLR et l'hypothèse nulle est rejetée dans 2 cas (Ringnorm et Waveform). Inversement, dans 2 cas sur 10, la KLR surpasse la KL-PLS et l'hypothèse nulle est rejetée 1 fois. Il est intéressant de noter que la KLR et la KL-PLS s'appuient sur le même classifieur (régression logistique) et que seul le mode de contrôle de complexité diffère : régularisation de Tikhonov pour la KLR et réduction de dimension supervisée pour la KL-PLS.

KL-PLS vs. KPM : Pour 5 jeux de données sur 6, la KL-PLS fournit des taux d'erreur moyens inférieurs à ceux obtenus par la KPM et l'hypothèse nulle est rejetée dans 4 cas sur 5 (B. Cancer, Diabetis, German, Heart). Inversement, dans 1 cas sur 6, la KPM surpasse la KL-PLS et l'hypothèse nulle n'est pas rejetée. Ces résultats soulignent l'importance du mode supervisé de la réduction de dimension.

Comparaison globale : Nous souhaitons évaluer globalement les méthodes et non plus par jeu de données individuellement. Pour ce faire, nous procédons de la manière suivante : nous calculons, pour chaque jeu de données, le rang de performances de chacune des méthodes puis leur rang moyen. La dernière ligne du tableau 5.2 fournit la moyenne de ces rangs. Ces rangs permettent de distinguer trois types de méthodes : les méthodes basées sur un contrôle de complexité par réduction de dimension supervisée (KPLS-SVC et KL-PLS) fournissent le plus souvent les taux d'erreur moyens les plus faibles. Viennent ensuite les méthodes basées sur les principes de régularisation de Tikhonov (SVM et KLR) puis la KPM basée sur une approche de réduction de dimension non supervisée.

Ainsi, le contrôle de complexité obtenu par réduction de dimension supervisée fournit une alternative efficace à la régularisation de Tikhonov sur laquelle sont fondés à la fois les SVM et la KLR (cf. Chapitre 2).
Elle offre, de surcroît, l'avantage d'accéder à la représentation des observations dans l'espace engendré par les premières composantes et fournit de ce fait, un outil d'aide à l'interprétation.

5.2.2 Étude des données «Banana»

Banana est un jeu de données artificiel bi-dimensionnel permettant une inspection visuelle de la séparation des classes d'individus dans l'espace d'origine. Cette visualisation 2D atteste du côté hautement non linéaire de la séparation entre les deux classes. Nous allons, par conséquent, nous focaliser sur ce jeu de données.

Projection des données «Banana» dans l'espace engendré par les deux premières composantes KL-PLS

La figure 5.1 illustre la projection des individus de l'échantillon d'apprentissage (respectivement de l'échantillon de test) dans l'espace engendré par les deux premières composantes KL-PLS.
Nous constatons qu'une frontière linéaire suffit à séparer les deux classes d'individus dans l'espace engendré par les deux premières composantes KL-PLS et qu'ainsi, la régression logistique réalise une classification efficace.

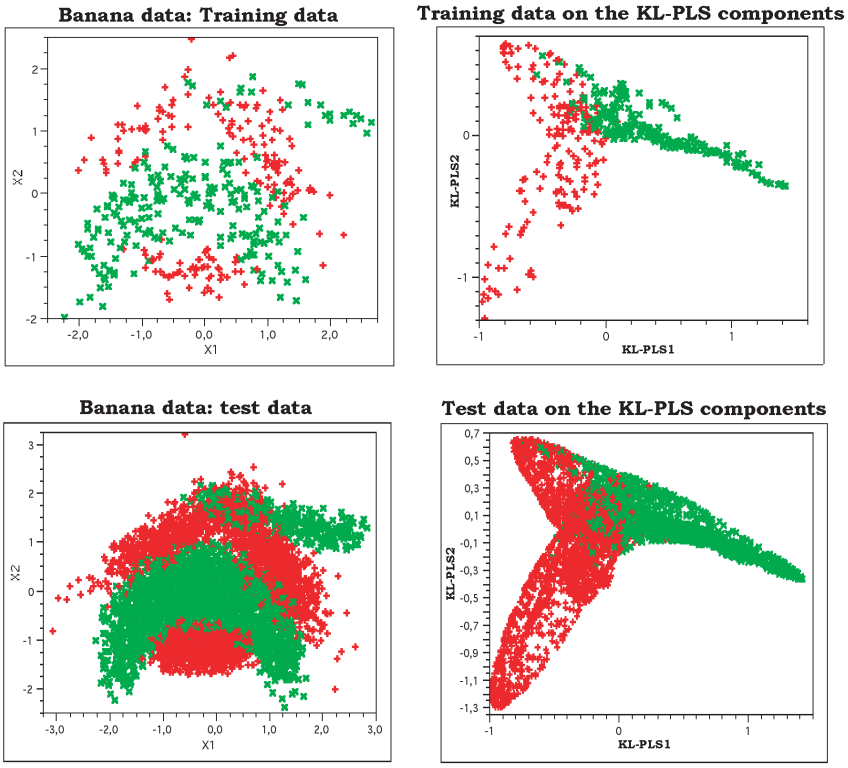

FIG. 5.1: Représentation des données «Banana» en apprentissage et en test sur les axes générés par la KL-PLS.

KL-PLS et probabilités

La KL-PLS fournit pour chacun des individus une probabilité d'appartenance aux différentes classes. La figure 5.2 représente les données «Banana» dans l'espace d'origine (partie gauche)

et leur projection sur l'espace engendré par les deux premières composantes KL-PLS (partie droite). La couleur symbolise la probabilité conditionnelle d'appartenance à la classe des croix vertes et les lignes représentent les contours d'isoprobabilité. Notons que la frontière de décision linéaire, dans l'espace induit par les composantes KL-PLS correspond à une frontière de décision non-linéaire complexe dans l'espace d'origine.

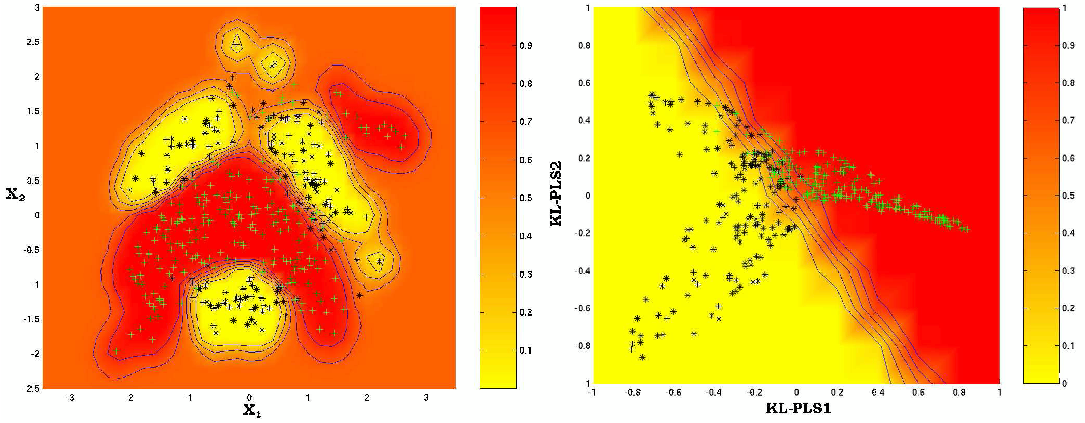

FIG. 5.2: Représentation des données «Banana» sur les deux premières composantes KL-PLS (partie droite) et sur l'espace d'origine (partie gauche). La palette de couleurs symbolise les probabilités d'appartenance à la classe des croix vertes tandis que les lignes, les contours d'isoprobabilité.

Taux d'erreur moyens vs. nombre de composantes KL-PLS

Dans ce paragraphe, nous allons évaluer (cf. figure 5.3) et visualiser (cf. figure 5.4) l'impact du choix du nombre de composantes sur la complexité de la frontière de décision.
La figure 5.3 montre les performances de la KL-PLS en fonction du nombre de composantes KL-PLS retenues. Sur cet exemple, notons que même en considérant un nombre relativement important de composantes KL-PLS, le phénomème de sur-apprentissage n'apparaît pas. Ajoutons que l'ajustement du nombre de composantes à retenir est simple, puisque'il s'agit d'un paramètre discret à l'inverse, par exemple, du paramètre continue (C) des SVM.

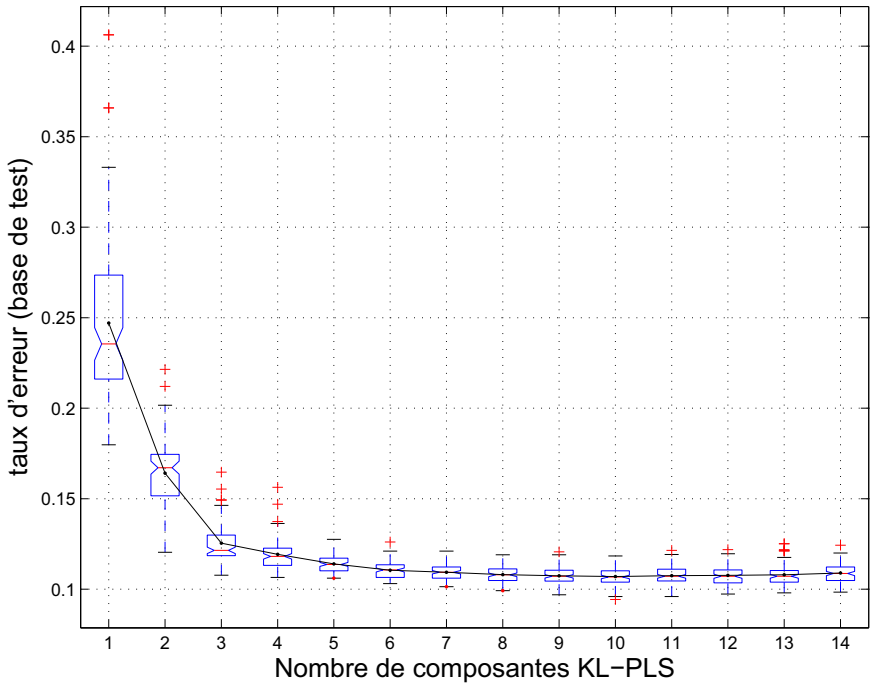

FIG. 5.3: Données «Banana» : Taux d'erreur moyen vs. nombre de composantes KL-PLS retenues. À chaque composante KL-PLS est associée une boîte à moustaches permettant la visualisation de manière compacte de la dispersion des 100 taux d'erreur pour les échantillons de test. La boîte centrale est construite à partir des quartiles inférieur et supérieur et partagée par la médiane. Les «moustaches» vont du premier quartile au minimum et du troisième quartile au maximum. Par convention, les moustaches ont une longueur qui ne doit pas dépasser une fois et demie la distance inter-quartiles. Si les points extrêmes sont trop loins des quartiles, ils apparaîtront comme isolés sur le graphique. Les entailles de la boîtes à moustaches sont centrées sur la médiane et ont pour largeur $(3.16 \times \text{distance inter-quartiles})/\sqrt{\text{effectif de l'échantillon}}$. Ces boîtes à moustache entaillées sont construites de manière à ce que deux boîtes ayant des entailles qui ne se chevauchent pas correspondent à des médianes significativement différentes au risque $\alpha = 0.05$.

La figure 5.4 illustre l'évolution de la complexité de la frontière en fonction du nombre de composantes KL-PLS sélectionnées.

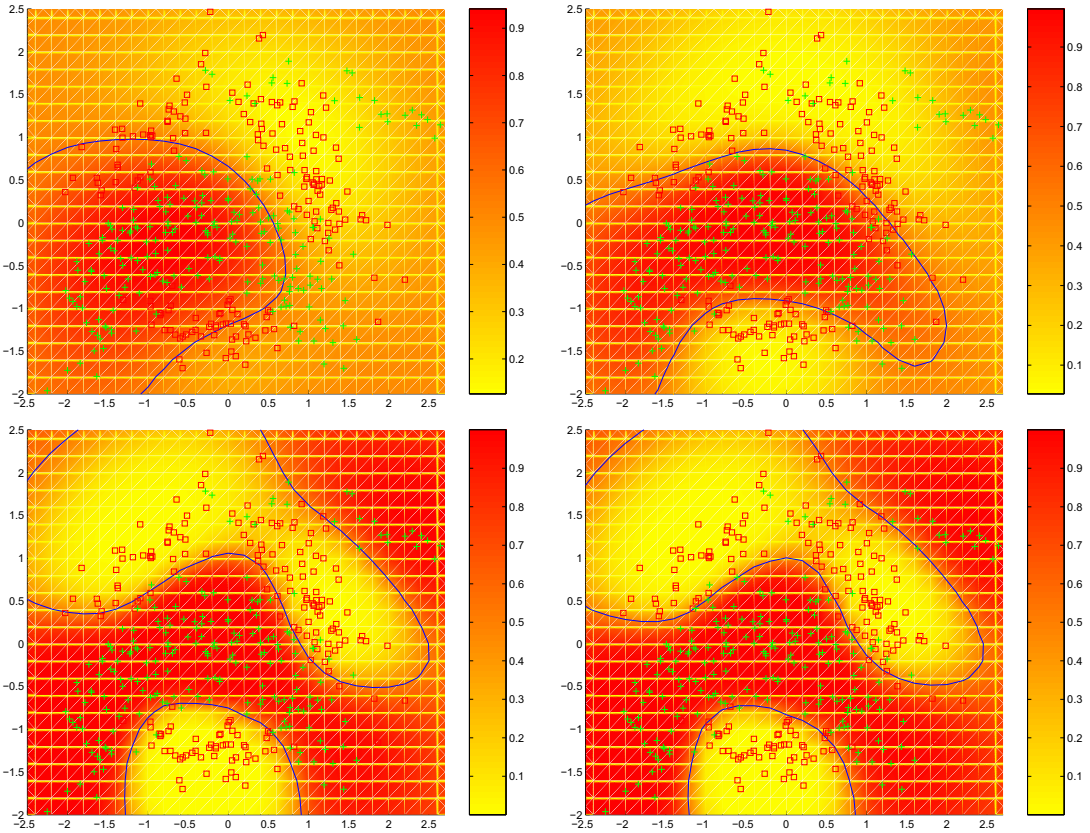

FIG. 5.4: Données «Banana» : évolution de la complexité de la frontière de décision en fonction du nombre de composantes KL-PLS retenues. En haut à gauche : une composante sélectionnée - En haut à droite : deux composantes sélectionnées - En bas à gauche : trois composantes sélectionnées - En bas à droite : quatre composantes sélectionnées.

Comme on pouvait s'y attendre à la lecture de la figure 5.3, la complexité de la fonction de décision augmente tant que $m \leq 3$ et ne varie quasiment plus à partir de la quatrième composante.

Kernel Logistique PLS et robustesse

La robustesse est ici définie comme la qualité du modèle à résister au bruit. Afin de mesurer la robustesse du modèle sur les données «Banana», le protocole suivant a été mis en place : la variable à expliquer y de la base d'apprentissage est «dégradée» par l'inversion d'étiquettes d'individus sélectionnés aléatoirement (tout en préservant l'équilibre initial des classes). L'erreur de classification sur la base de test est alors mesurée en fonction du pourcentage de bruit injecté.

La figure 5.5 montre, sur cette exemple, que la KL-PLS résiste au bruit de manière comparable aux SVM (SVM^{light} [Joachims (1999)]).

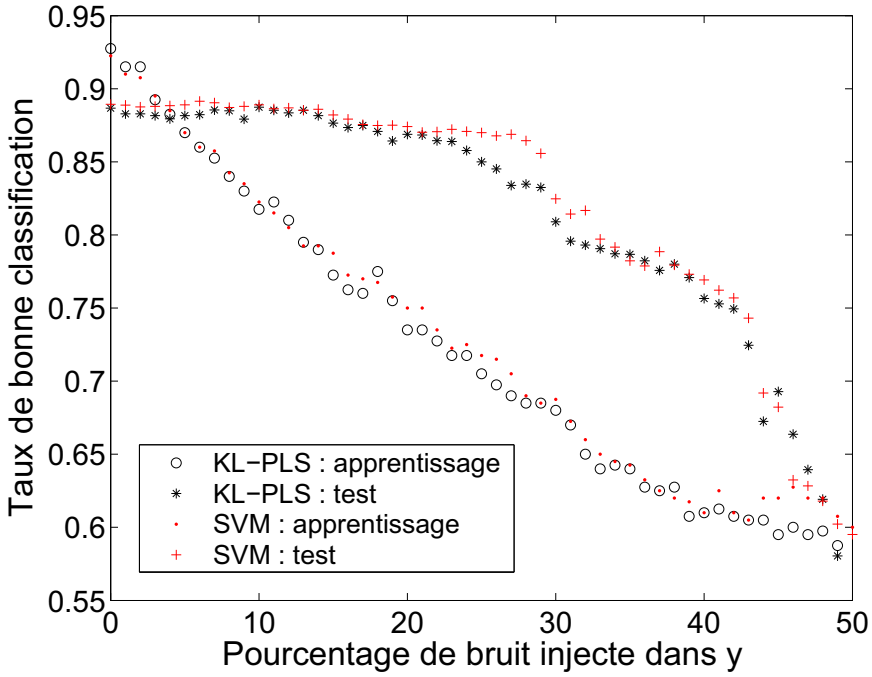

FIG. 5.5: Données «Banana» : évolution du taux de bonne classification mesuré sur la base de test en fonction du pourcentage de bruit injecté dans la variable à prédire (y) de la base d'apprentissage.

En effet, les courbes de taux d'erreur de test pour les SVM et la KL-PLS ont une allure très comparable. Il est remarquable de constater que jusqu'à 30% de dégradation, le taux d'erreur de test ne décline quasiment pas.

5.2.3 KL-PLS et données de grande dimension

Afin de valider la KL-PLS dans les espaces de grande dimension, 2 jeux de données issus de la problématique des puces à ADN ont été exploités : il s'agit de «Ovarian Cancer» et de «Lung Cancer». Ces jeux de données sont disponibles à l'adresse internet suivante : http ://sdmc.lit.org.sg/GEDatasets/. Ovarian Cancer est un jeu de données à deux classes en 15154 dimensions (150 observations d'apprentissage et 103 observations de test). Lung Cancer est un jeu de données à deux classes en 12533 dimensions (100 observations d'apprentissage et 81 observations de test). Les profils d'expression de gènes obtenus à partir des puces à ADN sont ici utilisés pour distinguer les deux types de tumeurs (cancéreuses ou non).

Le protocole d'évaluation d'une méthode est le suivant : le nombre d'observations composant les échantillons d'apprentissage et de test est fixé à 150 et 103 pour «Ovarian Cancer » et 100 et 81 pour «Lung Cancer». Trente partitions aléatoires sont alors générées. Le taux d'erreur moyen $\pm$ l'écart type, mesuré sur la base de test, permet d'évaluer la méthode. Dans les espaces de très grande dimension, un noyau linéaire est généralement suffisant pour

atteindre de bonnes performances de classification. Le noyau linéaire ne requiert l'ajustement d'aucun paramètre et l'unique paramètre libre est donc le nombre de composantes KL-PLS à retenir pour le modèle final. Sa valeur est fixée par validation croisée sur les 5 premiers jeux de données. Le nombre de composantes KL-PLS est fixé à 19 pour Ovarian et 6 pour Lung. Les résultats sont reportés dans le Tableau 5.3 et comparés à ceux obtenus par Shen avec un classifieur SVM linéaire [Shen and Tan (2005)].

TAB. 5.3: Taux d'erreur de classification mesuré sur les échantillons de test (moyenne ± écart-type) pour KL-PLS et SVM [Shen and Tan (2005)]. «∗» indique que l'hypothèse nulle associée au test de Student est rejetée au risque $\alpha = 0.05$.

Data set	**KL-PLS**	**SVM**
Ovarian	0.0±0.0	0.22* ±0.5
Lung	0.24±0.49	0.83* ±0.82

Afin de comparer les performances de la KL-PLS et des SVM sur «Ovarian Cancer» et «Lung Cancer», un test de student sur les moyennes des erreurs de test a été utilisé. Sur ces deux jeux de données, le test de Student rejette l'hypothèse d'égalité des moyennes. Nous pouvons donc conclure au bon comportement de la KL-PLS.

Soulignons que la configuration ($n << p$) est particulièrement favorable à la KL-PLS : en effet, la construction des composantes KL-PLS est algorithmiquement efficace puisqu'elles sont obtenues en gérant uniquement des matrices K de dimension $n \times n$. La figure 5.6 présente le taux d'erreur (base de test) en fonction du nombre de composantes KL-PLS retenues pour «Ovarian Cancer» et «Lung Cancer» (colonne de gauche) et temps CPU de la KL-PLS en fonction du nombre de composantes sélectionnées pour «Ovarian Cancer » et «Lung Cancer» (colonne de droite). Les deux lignes horizontales fournissent le temps CPU de SVM^{light} pour ces deux jeux de données.

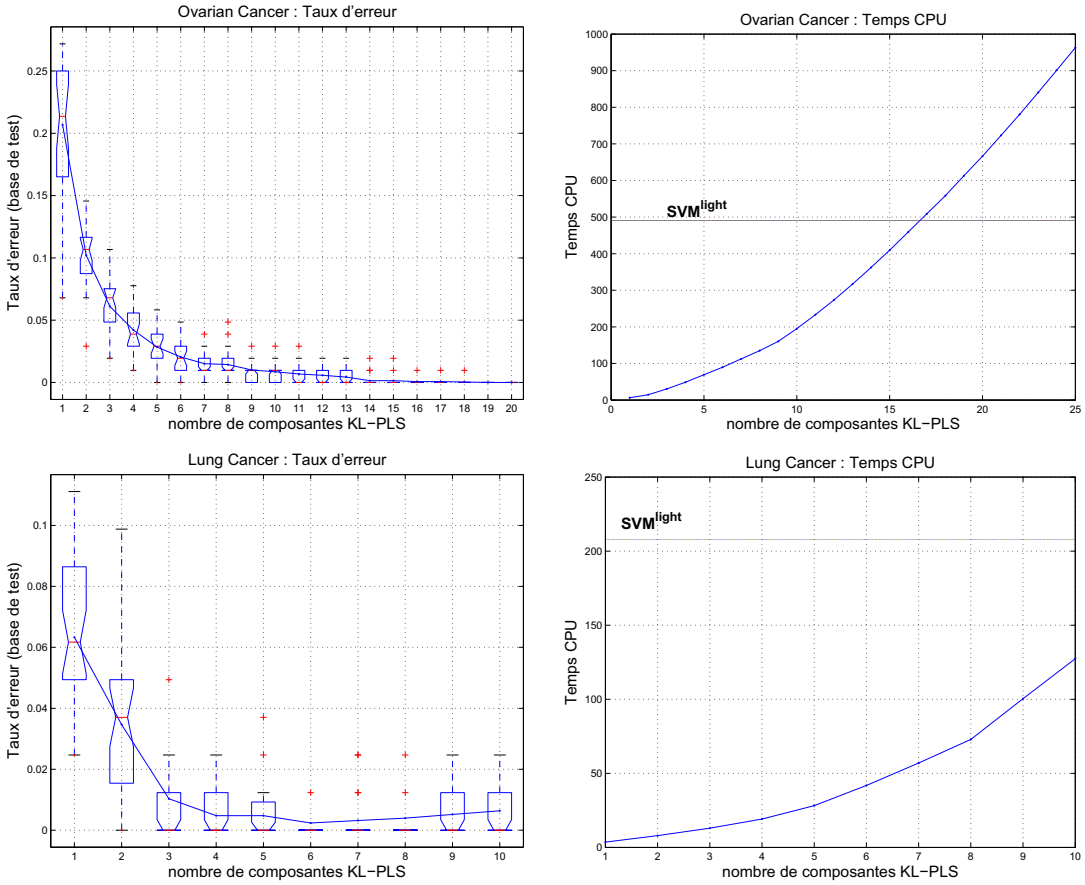

FIG. 5.6: Colonne de gauche : Taux d'erreur (base de test) de la KL-PLS en fonction du nombre de composantes KL-PLS retenues pour «Ovarian Cancer» (haut) et «Lung Cancer» (bas) ; Colonne de droite : Temps CPU de la KL-PLS calculé pour «Ovarian Cancer » (haut) et «Lung Cancer» (bas) en fonction du nombre de composantes sélectionnées. Comparaison avec le temps CPU des SVM^{light} (ligne horizontal).

Pour «Lung Cancer», le temps CPU de la KL-PLS à 6 composantes (implémenté sous Matlab version 6.5) est environ 5 fois inférieur au temps CPU des SVM^{light} (version C compilée). Pour «Ovarian Cancer» le temps CPU de la KL-PLS à 19 composantes surpasse le temps CPU de SVM^{light}. Néanmoins, à taux d'erreur égal (obtenu pour 14 composantes) le temps CPU de la KL-PLS est 30% inférieur à celui des SVM^{light}.

5.3 La Régression Logistique Multinomiale PLS

Nous allons reproduire pour la LM-PLS la même démarche de validation que celle effectuée pour la KL-PLS, à savoir validation sur benchmarks et représentation visuelle des classes d'individus.

5.3.1 Benchmarks

L'efficacité de la Régression Logistique Multinomiale PLS est mesurée sur un ensemble de 5 benchmarks utilisés dans [Duan and Keerthi (2003)]. Ces jeux de données sont disponibles à l'adresse suivante : http ://guppy.mpe.nus.edu.sg/∼mpessk/multiclass.shtml. Le Tableau 5.4 fournit quelques informations élémentaires sur ces jeux de données : nombre d'observations (n), nombre de variables (p), nombre de classes (G), le nombre d'observations composant l'échantillon d'apprentissage (n_{app}) et le nombre d'observations composant l'échantillon de test (n_{test}).

TAB. 5.4: Description des benchmarks multiclasses

Data	$\mathbf{n} \times \mathbf{p}$	G	n_{app}	n_{test}
ABE	2323×16	3	280	2083
DNA	3186×180	3	300	2886
SAT	6435×36	6	1000	5435
SEG	2310×18	7	250	2060
WAV	5000×21	3	150	4850

Chaque jeu de données est composé de 20 échantillons d'apprentissage auxquels sont associés 20 échantillons de test. Pour un jeu de données, le protocole d'évaluation est identique pour toutes les méthodes :

1. Pour chacune des 20 partitions d'apprentissage, construire un modèle. L'ajustement éventuel de paramètres libres (e.g. nombre de composantes sélectionnées, ...) s'effectue à partir des 5 premières partitions et sont fixés par validation croisée.

2. Une fois les paramètres d'ajustement déterminés, tester chacun de ces 20 modèles sur l'échantillon de test associé.

3. Le taux d'erreur moyen ± l'écart type de ces 20 taux d'erreur de test permet d'évaluer la méthode.

Le protocole a été appliqué à la régression logistique multinomiale, l'analyse discriminante, la PLS-D et la LM-PLS. Les performances sont reportées dans le tableau 5.5.

TAB. 5.5: Taux d'erreur de classification mesuré sur les échantillons de test (moyenne ± écart-type) pour la régression logistique multinomiale (MLR), l'analyse discriminante de Fisher (FDA), la PLS discriminante (PLS-D) et la régression logistique multinomiale PLS (LM-PLS). Le nombre de composantes retenues est fourni pour PLS-D et LM-PLS. Le symbole «$\star$ » signifie que le modèle final est une analyse discriminante de y sur les composantes retenues. L'astérisque simple «$*$» indique que l'hypothèse nulle du test de Sudent apparié est rejetée au risque $\alpha = 0.05$ et l'astérisque double «$**$» que l'hypothèse nulle est rejetée au risque $\alpha = 0.0001$.

Data	MLR	FDA	PLS-D	LM-PLS
ABE	4.47 ± 1.1	$6.08 \pm 1.27^{*}$	4.2 ± 1.16 (9)	4.28 ± 1.09 (11)
DNA	$25.45 \pm 2.05^{**}$	$20.77 \pm 1.75^{**}$	$11.11 \pm 1.08^{*}$ (3)	9.75 ± 1.26 (3)
SAT	$28.35 \pm 20.31^{*}$	$16.88 \pm 0.44^{*}$	15.28 ± 0.41 (8)	15.28 ± 0.64(9)
SEG	85.33 ± 5.76	9.94 ± 0.88	17.46 ± 18.01 (6)	36.38 ± 28.06 (3)
SEG*			9.83 ± 0.81 (14)*	9.78 ± 0.85 (13)*
WAV	$26.87 \pm 10.54^{**}$	$24.35 \pm 2.0^{**}$	14.42 ± 0.46 (2)	14.39 ± 0.57 (2)

Commentaire du tableau de benchmarks multiclasses

Le test de Student apparié a été utilisé pour comparer les résultats de MLR, FDA et PLS-D à ceux obtenus par LM-PLS. Ces test conduisent aux résultats suivants :

LM-PLS vs. MLR : Dans la totalité des cas, la LM-PLS fournit des taux d'erreur moyens inférieurs à MLR. Ces différences sont significatives dans 4 cas sur 5 (différence non-significative pour ABE). Les performances de MLR peuvent s'expliquer par de fortes colinéarités entre variables et/ou du fait de la complète ou quasi-complète séparation entre classes d'individus.
Cette forte colinéarité est décelée de façon classique par l' «indice de conditionnement» : notons $\lambda_1, ..., \lambda_p$, les valeurs propres de la matrice des corrélations, R, rangées par ordre décroissant. Le déterminant de R est égal au produit des corrélations et des problèmes d'instabilité ou de variance excessive apparaissent dès lors que les dernières valeurs propres sont relativement trop petites. L'indice de conditionnement, noté κ, est défini par le rapport de la plus grande valeur propre sur la plus petite valeur propre : $\kappa = \lambda_1/\lambda_p$. Cet indice de conditionnement donne un aperçu global des problèmes de colinéarité. En pratique, si $\kappa < 100$ on considère qu'il n'y a pas de problème de colinéarité. Celui-ci devient sévère pour $\kappa > 1000$. Le tableau 5.6 fournit les indices de conditionnement des 5 jeux de données. Du tableau 5.6, nous déduisons pour SAT et SEG de fortes colinéarités entre variables. La MLR n'est donc pas adaptée à ces deux jeux de données.

TAB. 5.6: Mesure de la multicolinéarité des variables sur les benchmarks multiclasses : indice de conditionnement.

Data	ABE	DNA	SAT	SEG	WAV
κ	76	226	2002	5×10^{15}	49

Par ailleurs, nous avons vu que la régression logistique multinomiale ne fournit pas de modèles satisfaisants dans le cas de complète ou quasi-complète séparation d'au moins une classe d'individus. Ce phénomème survient dans le cas de SEG. En effet, il existe une complète séparation d'au moins une classe d'individus. Nous pouvons visualiser la complète séparation de deux classes d'individus par la représentation des observations sur l'espace engendré par les deux premières composantes LM-PLS sur la figure 5.7.

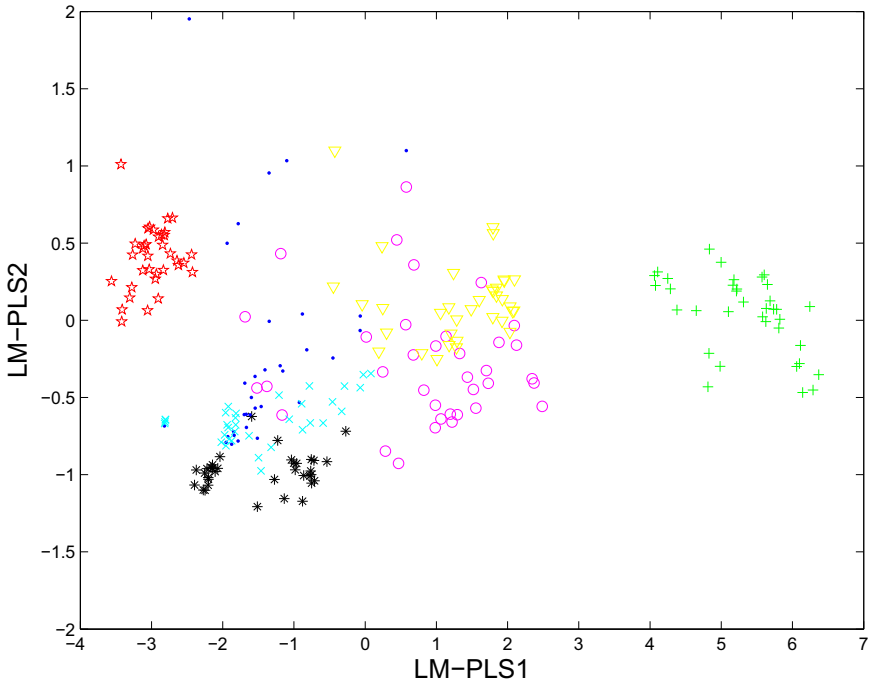

FIG. 5.7: Séparation complète d'au moins une classe d'individus : représentation de SEG sur l'espace engendré par les deux premières composantes LM-PLS.

Les classes d'individus caratérisées par «+» et par «⋆» sont linéairement séparés des autres individus. Dans de telles situations, il est alors préférable d'effectuer une analyse discriminante de y sur les composantes sélectionnées. Les résultats de l'analyse discriminante de y sur les composantes LM-PLS (respectivement PLS-D) sélectionnées sont reportés sur tableau 5.5.

LM-PLS vs. FDA : Dans la totalité des cas, la LM-PLS fournit des taux d'erreur moyens inférieurs à FDA. Ces différences sont significatives dans 4 cas sur 5 (différence non-significative pour SEG). FDA requiert l'inversion des matrices de covariances internes à chaque classe. Ces matrices calculées sur un effectif n_g inférieur à n peuvent être mal conditionnées. C'est le cas notamment de DNA. Pour contourner ce problème, Friedman propose une version régularisée de l'analyse discriminante [Friedman (1989)].

LM-PLS vs. PLS-D : Dans 3 cas sur 5, la LM-PLS fournit des taux d'erreur moyens inférieurs à PLS-D. Ces différences sont significatives dans 1 cas sur 3 (DNA). Inversement la PLS-D fournit des taux d'erreur moyens inférieurs à la LM-PLS dans 2 cas et ces différences

ne sont pas significatives.

À partir de ce tableau de benchmarks, nous concluons au bon comportement de la LM-PLS.

Taux d'erreur moyen vs. nombre de composantes LM-PLS

La figure 5.8 montre les performances de la LM-PLS en fonction du nombre de composantes retenues.

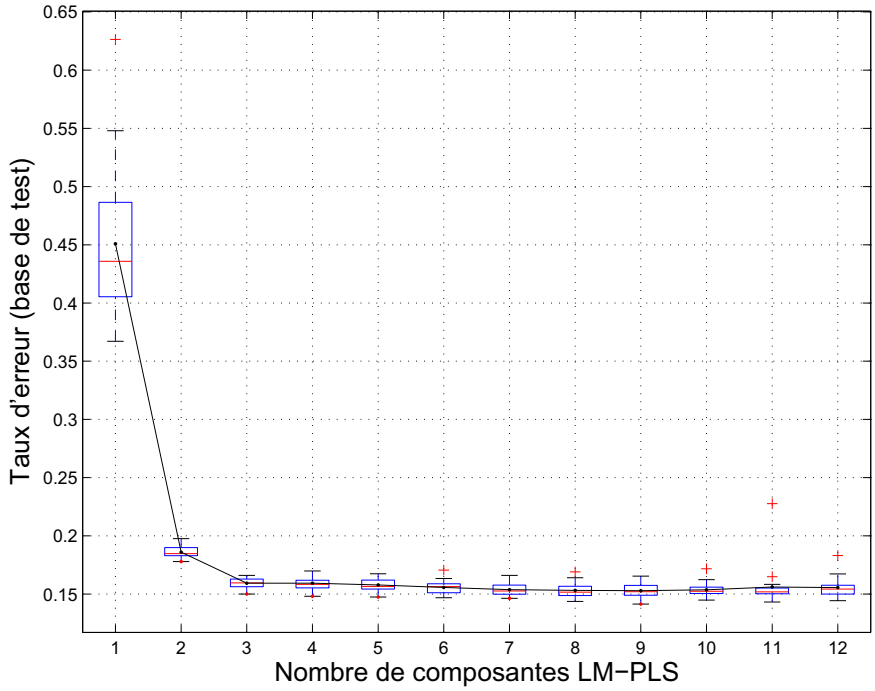

FIG. 5.8: données «SAT» : taux d'erreur moyen vs. nombre de composantes LM-PLS retenues.

Sur ces données, notons qu'à partir de 3 composantes, les performances prédictives s'améliorent peu. Par ailleurs, même en considérant un nombre relativement important de composantes LM-PLS, le phénomème de sur-apprentissage n'apparaît pas. Le choix du nombre de composantes est donc relativement simple à effectuer.

5.3.2 Visualisation

Exactement de la même manière que dans le cadre de la KL-PLS, il est possible de visualiser la représentation des individus dans l'espace engendré par les composantes LM-PLS. Nous allons nous focaliser sur les données «SAT».

La LM-PLS fournit pour chacun des individus une probabilité d'appartenance aux différentes classes. La figure 5.9 représente «SAT» dans l'espace engendré par les deux premières compo-

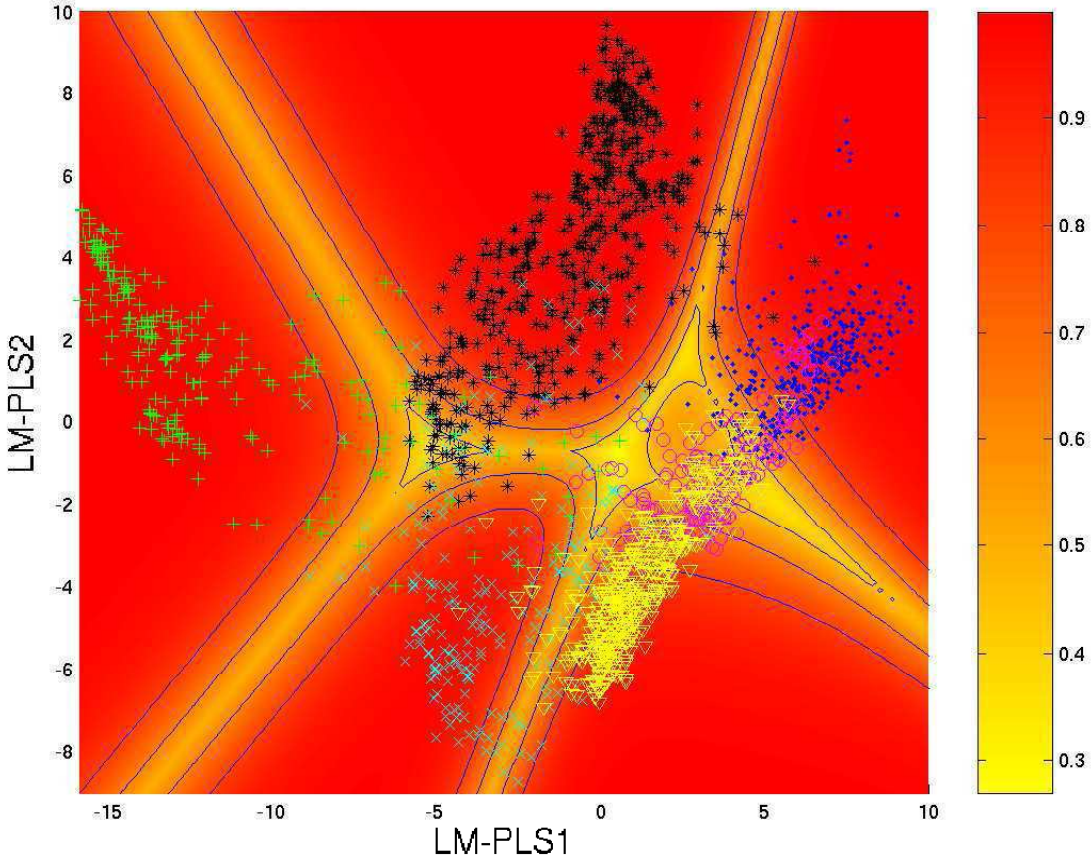

FIG. 5.9: Représentation de SAT sur les deux premières composantes LM-PLS. Chaque point représente un individu tandis que sa forme/couleur représente sa classe d'appartenance. La palette de couleur symbolise la probabilité maximale (i.e le maximum des $p_g(x)$, $g = 1, ..., G$) et les lignes les contours d'isoprobabilité.

santes LM-PLS. La couleur symbolise la probabilité maximale $\max_{g \in \{1,...,G\}} p_g(x)$ et les lignes, les contours d'isoprobabilité. Cette visualisation est intéressante pour plusieurs raisons : elle permet par exemple d'évaluer le degré de liaison d'une classe d'individus par rapport aux autres ; mais elle permet surtout d'évaluer la fiabilité de prédiction d'un nouvel individu. En effet, si le nouvel individu se place au milieu des individus de sa classe, on aura tendance à accorder de la confiance à sa prédiction. Si à l'inverse, l'individu est à l'écart de son groupe (bien que sa probabilité d'appartenance soit élevée) la confiance accordée à sa prédiction est sans aucun doute à nuancer car le classifieur n'est plus dans son domaine de compétence.

5.4 La Kernel Logistique Multinomiale PLS

5.4.1 Benchmarks

Nous restreignons les résultats relatifs à la KLM-PLS à la validation par benchmarks. Le protocole d'évaluation de la KLM-PLS est le même que pour la LM-PLS. On retrouve les performances (même protocole de validation) de OVA et AVA dans Duan [Duan and Keerthi (2003)].

TAB. 5.7: Taux d'erreur de classification mesuré sur les échantillons de test (moyenne ± écart-type) pour OVA, AVA et KLM-PLS. La dernière colonne fournit, en plus du pourcentage d'erreur, le paramètre γ du noyau gaussien ($k(x,y) = exp(-\gamma\|x-y\|^2)$) et le nombre de composantes KLM-PLS retenues. L'astérisque simple «$*$» indique que l'hypothèse nulle du test de Student apparié est rejetée au risque $\alpha = 0.05$ et l'astérisque double «$**$» que l'hypothèse nulle est rejetée au risque $\alpha = 0.0001$.

Data set	OVA	AVA	KLM-PLS
ABE	$1.92 \pm 0.65^{*}$	$1.96 \pm 0.65^{*}$	1.47 ± 0.41 $(0.08, 40)$
DNA	10.15 ± 1.26	9.87 ± 0.90	9.42 ± 1.08 $(0.0001, 3)$
SAT	$11.07 \pm 0.58^{*}$	$11.03 \pm 0.73^{*}$	$11.56 \pm 0.0.53$ $(0.08, 60)$
SEG	$9.43 \pm 0.54^{**}$	7.97 ± 1.23	7.84 ± 0.89 $(0.001, 40)$
WAV	$17.21 \pm 1.37^{**}$	$17.75 \pm 1.39^{**}$	14.22 ± 0.60 $(0.001, 2)$

Sur le site http ://guppy.mpe.nus.edu.sg/∼mpessk/multiclass.shtml sont disponibles les taux d'erreur individuels (pour chaque partition et chacune des deux méthodes). Il est donc possible d'utiliser le test de Student apparié pour comparer les différents modèles.

KL-PLS vs. OVA : Pour 4 jeux de données sur 5, la KLM-PLS fournit des taux d'erreur moyens inférieurs à OVA. Dans 3 cas sur 4 (ABE, SEG et WAV), l'hypothèse nulle est rejetée. Inversement le taux d'erreur moyen de OVA est plus faible que celui de la KLM-PLS sur 1 jeu de données et dans ce cas, l'hypothèse nulle est rejetée.

KLM-PLS vs. AVA : Pour 4 jeux de données sur 5, la KLM-PLS fournit des taux d'erreur moyens inférieurs à AVA et l'hypothèse nulle est rejetée dans 2 cas sur 4 (ABE et WAV). À l'inverse, le taux d'erreur moyen de AVA est inférieur à celui de la KLM-PLS pour SAT et l'hypothèse nulle est rejetée dans ce cas.

Aux vues des ces résultats, nous pouvons conclure au bon comportement discriminant de la KLM-PLS face à des méthodes de classification reconnues comme très performantes [Rifkin and Klautau (2004) ; Hsu and Lin (2002)].
Notons que la KLM-PLS bénéficie des avantages relatifs à l'inspection des résultats par examen visuel et des prédictions en termes de probabilités.

Chapitre 6

Caractérisation des tumeurs noires de la peau

Ce chapitre est le fruit d'une étroite collaboration : il s'inscrit dans la continuité des travaux de doctorat de Camille Serruys [Serruys (2003)] et exploite les résultats de stage de DEA de Jean-Francois Horn [Horn (2004)] et d'Alex Nkengne [Nkengne (2004)] avec qui j'ai eu le plaisir de travailler.

6.1 Contexte médical

Le mélanome est une tumeur maligne qui se développe principalement dans la zone cutanée, à partir des mélanocytes, cellules responsables de la pigmentation. C'est le cancer dont l'incidence augmente actuellement le plus rapidement, de 3 à 7 % par an, pour les 10 dernières années [Grob and Richard (2004)]. Le taux de mortalité associé augmente de façon plus modeste. Cette tendance de la mortalité par rapport à l'incidence peut s'expliquer par une détection plus précoce mais également par une meilleure prévention. Cette affection touche principalement les pays occidentaux. L'Australie présente le taux le plus élevé (40-60 cas pour 100 000) ; le mélanome y est le cinquième cancer le plus fréquent [Grob and Richard (2004)]. Les pays européens se situent dans une moyenne de 10-12 cas pour 100 000 (Allemagne), le taux le plus important revenant à la Scandinavie (15 cas pour 100 000) et le plus faible aux pays méditérranéens (5-7 cas pour 100 000) [Grob and Richard (2004)]. Le tableau 6.1 illustre l'évolution du mélanome en France entre 1980 et 2000.

Tab. 6.1: Nombre annuel de nouveaux cas de mélanomes en France (nombre annuel de décès). Chiffres extraits du rapport INVS-INSERM-FRANCIM-Hôpitaux de Lyon : évolution de l'incidence et de la mortalité par cancer en France entre 1978-2000.

	1980	1985	1990	1995	2000
Hommes	777 (318)	1092 (389)	1543 (480)	2199 (588)	3066 (704)
Femmes	1476 (348)	1859 (407)	2415 (484)	3184 (571)	4165 (660)

7 231 nouveaux cas de mélanomes ont été diagnostiqués en 2 000, dont 58% chez les femmes. Le mélanome représente 2.6% des cancers incidents. Il se situe au treizième rang des cancers

pour l'homme, au septième rang pour la femme. En 2000, 1364 décès lui sont imputables (704 hommes et 660 femmes), ce qui représente 0.9% des décès par cancer. Pour l'année 2 000, l'âge moyen de survenue du mélanome est de 58 ans chez l'homme et 56 ans chez la femme. Le mélanome est l'un des cancers dont l'incidence a augmenté le plus au cours des deux dernières décennies : elle a doublé en dix ans. Ainsi, en France, le taux annuel moyen d'évolution de l'incidence entre 1978 et 2000 est de +5.9% chez l'homme et + 4.33% chez la femme. La mortalité augmente régulièrement (mais à un moindre degré) durant cette période, et pour les deux sexes.

L'évolution spontanée d'un mélanome est très variable. Certaines tumeurs progressent rapidement alors que d'autres évoluent sur plusieurs années avec des régressions mais très rarement des guérisons spontanées. L'évolution habituelle est marquée par un envahissement local avec extension possible à la peau adjacente puis atteinte des ganglions régionaux et apparition de métastases, habituellement multiples. Tissus mous, poumons, foie, cerveau sont dans cet ordre les localisations métastatiques les plus fréquentes. L'atteinte osseuse est plus tardive ; tous les organes sont susceptibles d'être atteints secondairement. Malheureusement, le mélanome est une tumeur peu radiosensible et peu chimiosensible. Par conséquent, l'exérèse avec reprise chirurgicale éventuelle demeure le traitement principal et permet la guérison de la plupart des mélanomes non métastasés. Ainsi, toute lésion suspecte de mélanome doit être excisée en vue d'un examen histopathologique. La biopsie-exérèse doit être complète, emportant la tumeur dans son entier. Ajoutons que des traitements adjuvants par chimiothérapie, radiothérapie ou immunothérapie, sont parfois utilisés, une fois l'exérèse effectuée. Il est cependant encore difficile aujourd'hui de démontrer l'efficacité de ces traitements, malgré les multiples essais cliniques mis en place. Ainsi, le seul moyen de réduire efficacement la mortalité du mélanome est le dépistage précoce. L'exérèse de la tumeur permet d'enrayer son évolution et d'augmenter considérablement les chances de survie.
Les médecins, généralistes et dermatologues, doivent donc être les fers de lance de la prévention des cancers cutanés, certes en relayant l'information sur les dangers de l'exposition solaire ou du bronzage artificiel, mais également en assurant un dépistage précoce des lésions. «Reconnaître un mélanome de moins de 1 millimètre d'épaisseur histologique - une lésion accessible à la vue - c'est presque toujours le guérir ; à l'opposé un mélanome négligé et épais a peu de chance d'être curable »[Delaunay (2004)]. Sachant que l'observation à l'oeil nu suffit à repérer les lésions suspectes, cette phrase doit faire partie des aphorismes clés de la médecine préventive. En complément de ce travail quotidien (l'examen systématique de la peau et le recours à la biopsie, si besoin), le rôle des médecins et en particulier des dermatologues, est important, notamment au travers des journées de dépistage et de sensibilisation. Le retentissement médiatique de ces campagnes contribue à l'autosurveillance préconisée par de nombreux pays et peut ainsi raccourcir le délai de consultation (les tumeurs découvertes sont alors moins épaisses et donc de meilleur pronostic). La création récente (mai 2005) de l'Institut National du Cancer, conformément au plan de sensibilisation nationale contre le cancer impulsé par le président de la République, permettra un renforcement des actions de prévention, en regard de ce cancer évitable.

Actuellement, un dermatologue expérimenté peut identifier un mélanome avec 75% d'efficacité. Pour cela, il se fie en grande partie à l'observation visuelle de la tumeur en s'intéressant à des caractéristiques comme la coloration, la forme ou la texture (nous reviendrons plus loin sur ces critères). La difficulté étant de pouvoir différencier les mélanomes, illustrés sur la figure 6.1 de certaines lésions mélanocytaires ressemblantes comme les naevus dysplasiques (ou atypiques) (figure 6.2).

Ces dernières années, des systèmes autonomes combinant des caméras d'acquisition d'images

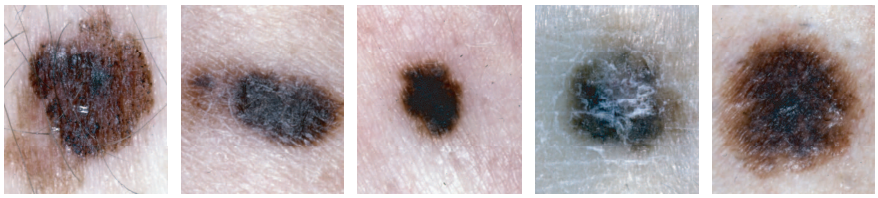

FIG. 6.1: Exemples de mélanomes malins.

FIG. 6.2: Exemples de naevus atypiques.

numériques et des possibilités d'archivage et de gestion de dossiers patients sont apparus sur le marché. Ces systèmes permettent, entre autres, de conserver des photos cliniques des naevi nécessitant une surveillance régulière. Partant du constat que de nombreux dermatologues ont l'habitude de prendre des clichés des lésions observées et de les associer au dossier médical, ils sont déjà sensibilisés aux informations que peuvent révéler ces images. De plus, le développement actuel de l'informatisation, encouragé par les pouvoirs publics, devrait conduire à doter chaque cabinet médical d'une station de travail informatique.

Dans ce contexte, de nombreux travaux consacrés à l'exploitation des images numériques de tumeurs noires apparaissent dans la littérature. Clairement, l'objectif affiché par les équipes travaillant dans ce domaine d'application est la caractérisation automatique et le diagnostic des mélanomes [Binder et al. (1998) ; Binder et al. (2000) ; Ganster et al. (2001) ; Sboner et al. (2003)]. La validité des systèmes actuellement proposés a été critiquée par Rosado [Rosado et al. (2003)]. Ces objections s'appuient notamment sur la validation des classifieurs (validation sur des images indépendantes), la représentativité des tumeurs constituant la base, les conditions d'acquision des images, la reproductibilité de l'acquisition, la confrontation des performances à celles de dermatologues.

C'est dans ce contexte extrêmement favorable et actif que s'inscrit le projet de recherches appliquée du laboratoire d'imagerie médicale de l'unité INSERM 678. Ce projet à pour objectif d'améliorer le dépistage précoce des mélanomes. Il s'agit de développer un outil d'aide au diagnostic capable d'alerter le praticien en proposant les éléments d'un diagnostic argumenté. Cet outil devra être conçu pour traiter des images directement obtenues à partir d'un appareil photo numérique standard, éventuellement muni d'un objectif dermatoscopique (sorte de loupe munie d'un système d'éclairage homogène), et fonctionner, pour un investissement modéré, en routine clinique. Compte tenue de la situation actuelle en imagerie et en analyse de données, il apparaît possible de réaliser cet outil en jumelant traitement d'images et technique d'analyse statistique. L'objectif de ce chapitre est donc de présenter l'assise méthodologique du système de caractérisation proposé.

La règle ABCD

Compte tenu des difficultés de mise au point d'un traitement efficace contre le mélanome, les développements actuels sont orientés vers la prévention et le dépistage précoce. Néanmoins, le diagnostic reste difficile pour des tumeurs malignes de stade précoce. Nous avons vu qu'un dermatologue expérimenté peut identifier un mélanome avec environ 75 % d'efficacité. Les performances varient de 60%, pour des dermatologues avec 3 à 5 ans d'expérience en dermatologie, à 80%, pour des médecins qui ont plus de 10 ans d'expérience. Du fait de nombreux traits communs avec d'autres tumeurs pigmentées, moins dangereuses voire bénignes, il est difficile de porter un diagnostic de façon catégorique. Le dermatologue a donc recours à plusieurs types d'informations : d'une part des informations qui concernent le patient et ses antécédents (dossier clinique), d'autre part des informations recueillies à partir de l'examen de la lésion, principalement des caractéristiques de contour, de forme, de texture et de couleur. Ainsi, lors de l'examen d'une lésion cutanée pigmentée, plusieurs critères cliniques peuvent faire suspecter la malignité. L'association de ces critères permet de définir des règles de décision. La règle la plus courante (et sur laquelle s'appuie le système que nous proposons) se base sur les critères ABCD :

A = **A**symétrie
B = **B**ords irréguliers
C = **C**ouleur hétérogène
D = **D**iamètre supérieur à 6-7 millimètres

Un mélanome se présente habituellement sous la forme d'une lésion asymétrique (A), à bords (B) irréguliers. Les bords sont souvent encochés ou polycycliques ou se prolongent en coulées d'encre accentuant l'asymétrie de la lésion. La couleur (C) est inhomogène avec des nuances variables dans les teintes du brun au noir, mais éventuellement des zones décolorées blanches, inflammatoires rouges ou cicatricielles bleutées. La diamètre (D) d'un mélanome est généralement supérieur à 6 millimètres. L'évolution (E), documentée par l'interrogatoire tient compte du changement non seulement de taille, mais aussi de forme de couleur et de relief. Ce critère E peut quelquefois être documenté par des photographies comparatives. Il existe cependant une nécessité à s'accorder sur la définition précise de ces critères et, bien que des conférences de consensus soient organisées, il y a encore matière à progresser.

Les outils d'aide à la décision sont ainsi apparus dans le but de soutenir la démarche diagnostique. Il devient possible d'établir une correspondance entre la significativité des paramètres extraits et l'information dont se sert le médecin pour fournir son diagnostic. Ce schéma d'analyse s'apparente clairement a des procédés courant en analyse d'images et en classification. On dispose d'une base d'images à partir desquelles on cherche à extraire des paramètres que l'on pense être caractéristiques de signes d'intérêt qui sont alors donnés à un classifieur fournissant ainsi une affectation aux différentes classes. Ce processus comporte plusieurs étapes dont la finalité est de prédire la nature de la lésion (naevus bénin vs naevus malin) :

1. Acquisition de l'image
2. Segmentation de la lésion
3. Extraction de paramètres liés à la couleur, la forme, la taille, ...
4. Classification

6.2 Description de la base de données

La base de données de tumeurs noires de la peau sur laquelle se fonde notre étude est composée de 227 images. Cette base a été fournie par les départements de Dermatologie de l'hôpital Louis Mourier et du British Hertfort Institut. Cinq dermatologues ont expertisé chacune des 227 lésions. Nous allons décrire, dans un premier temps, la base d'images et, dans un second temps, l'expertise fournie par les dermatologues.

6.2.1 Base d'images

Afin de respecter les critères nécessaires à un bon apprentissage, la base d'images doit représenter la grande variabilité des tumeurs. Le tableau 6.2 illustre la diversité de notre base d'images.

Tab. 6.2: Composition de la base d'images de tumeurs noires de la peau

Diagnostic	Nombre de cas	(%)	Exérèse
Mélanome	27	(11.9%)	excisé
Mélanome nodulaire	1	(0.45%)	excisé
Mélanome de Dubreuilh	4	(1.8%)	excisé
Naevus Dysplasique	118	(51.2%)	excisé
Naevus bénin	62	(27.3%)	non excisé
Naevus bleu bénin	2	(0.9%)	excisé
Naevus congenital bénin	5	(2.2%)	excisé
Naevus bénin junctionel et dermique	7	(3.1%)	excisé
Naevus bénin palmo-plantaire	1	(0.45%)	excisé
Total	227	100%	72.7%

La base d'images est composée de 77 naevus bénins (Classe 1), 118 naevus dysplasiques (classe 2) et 32 mélanomes (Classe 3). La majorité des lésions de la classe 1 n'ont pas été exérésées (pour ne causer aucun désagrément inutile aux patients), les lésions de la classe 2 et 3 ont toutes étaient exérésées et hystologiquement analysées. Dans un premier temps, nous considérons 2 catégories de lésions : les mélanomes (Classe 3) contre les lésions bénignes et dysplasiques (Classe 1 + Classe 2). Cette partition des tumeurs basée sur l'analyse histologique fournit le «gold standard ».

6.2.2 Expertise des dermatologues

À chacune des images de tumeurs noires est associée l'expertise de 5 dermatologues. L'expertise des dermatologues comporte, au-delà de l'information sur l'avis «diagnostic» (mélanome ou non), des informations sur la décision thérapeutique (exérèse ou non) et sur la présence-absence (réponse dichotomique) des signes de la règle ABCD. La Figure 6.3 illustre l'interface présentée aux cinq dermatologues afin de collecter leurs diagnostics.

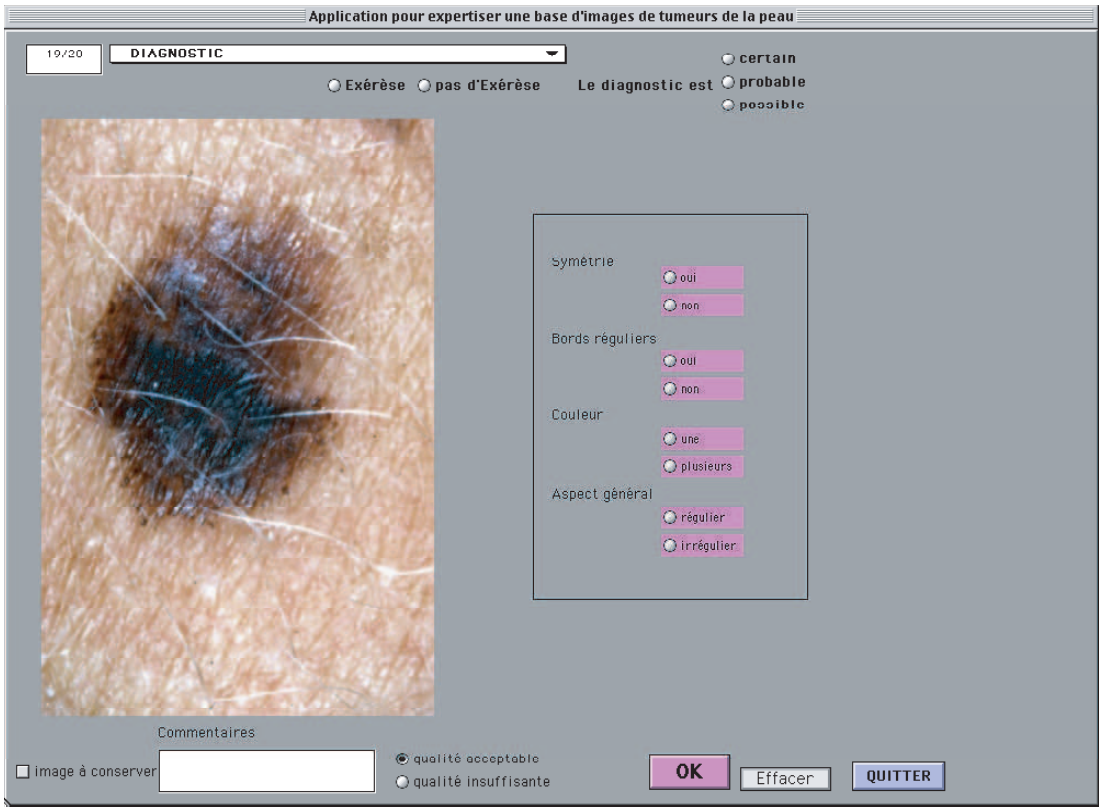

FIG. 6.3: Interface présenté aux cinq dermatologues afin de recueillir leurs diagnostics.

Il est intéressant d'évaluer le degré de concordance entre dermatologues et en particulier de définir un concensus pour chaque signe. Ce consensus, sous forme d'un processus de vote, va servir de référence pour indiquer la présence du signe dans la base d'images. Le résultat est compris entre 0 - les dermatologues sont unanimes et constate l'absence du signe et 5 - les dermatologues sont unanimes et constatent la présence du signe. Un vote égal à 2 signifie que 2 dermatologues repèrent le signe tandis que 3 ne l'ont pas observé. Le vote majoritaire est alors définie par une variable binaire égale à 0 si, au plus, 2 dermatologues ont détecté le signe d'intérêt et à 1 si au moins 3 dermatologues ont repéré le signe d'intérêt.
Dans ce paragraphe, nous allons nous intéresser au concensus des dermatologues pour chacun des signes de la règle ABCD, pour la décision thérapeutique et pour le diagnostic.

Appuyons nous sur le signe d'asymétrie (première ligne) pour expliciter la figure 6.4. L'histogramme représente la distribution des votes des dermatologues concernant le signe d'asymétrie. Ainsi, environ 75 des 227 lésions ont unanimement été diagnostiquées (par les cinq dermatologues) comme asymétriques. Parmi ces 75 lésions, la zone sombre (verte foncé) fournit le nombre de mélanomes (un peu moins de 25). Le camembert nous indique que 54% des tumeurs ont été unanimement diagnostiquées comme asymétriques ou symétriques (0-5). À l'inverse, nous pouvons lire sur le camembert que 20% des lésions disposent d'un concensus faible (2-3). La dernière colonne de la première ligne fournit la sensibilité et la spécificité du vote majoritaire relatif au signe d'asymétrie face au gold standard histologique.

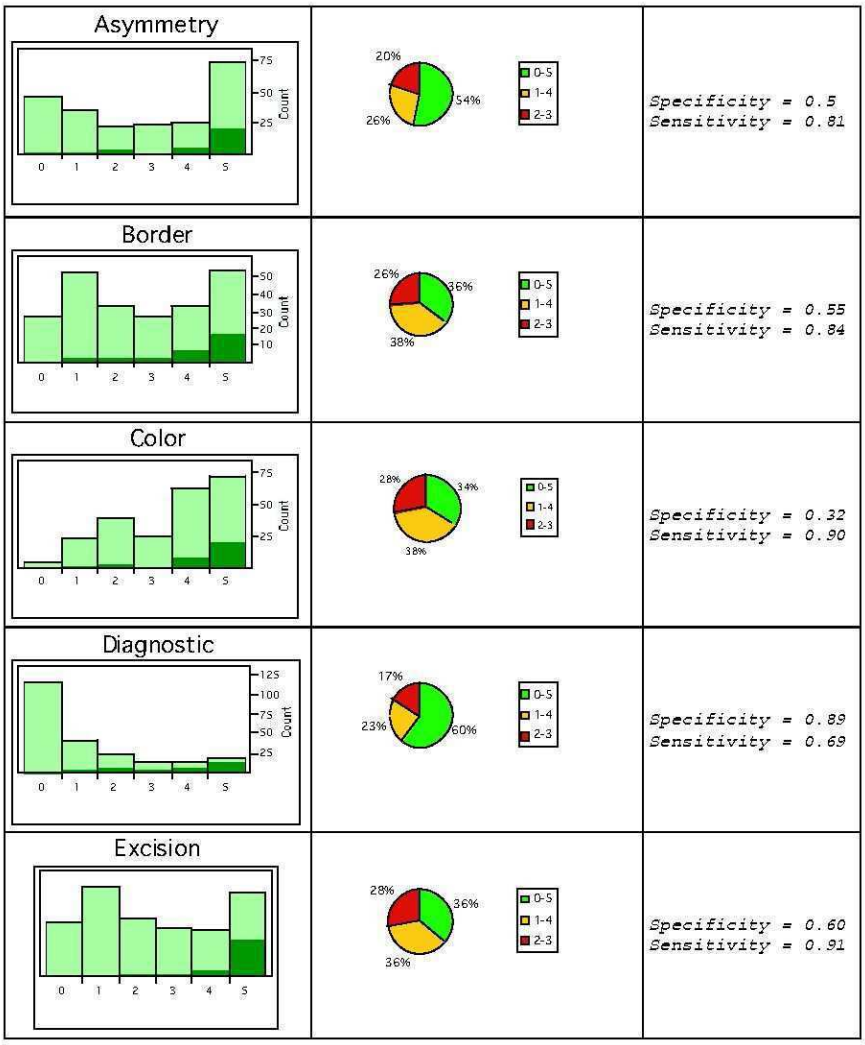

FIG. 6.4: Analyse de l'expertise des cinq dermatologues.

Le «faible» taux de concordance entre dermatologues peut s'expliquer par plusieurs raisons : l'expertise a été faite uniquement à partir de l'image. Bien que les dermatologues soient maintenant sensibilisés à l'image comme outil de suivi des patients, il peut-être difficile d'évaluer la nature d'une lésion sans tenir compte du «dossier clinique». Par exemple, l'exposition intense aux rayonnements ultraviolets (UV) (naturels et/ou artificiels), notamment dans la petite enfance et chez les personnes à phototypes clairs, constitue le principal facteur de risque du mélanome. Les risques sont également plus importants chez les personnes porteuses de nombreux naevus (plus de 50), atypiques et congénitaux géants. Dans une proportion bien moindre, certains facteurs de prédisposition génétique peuvent également être en cause (agrégation familiale dans 5 à 10% des cas). La localisation du mélanome et son évolution récente sont également des informations utiles à l'établissement du diagnostic. Il est donc important de prendre en compte ces informations qui interviennent en complément de l'examen visuel et l'image n'est pas le

meilleur des relais.
La figure 6.4 illustre la difficulté de détection des signes mais également la difficulté de garantir la certitude du diagnostic. En effet, il est surprenant de constater, que sur les 32 mélanomes composant la base d'images, seuls 40% d'entre eux sont unanimement diagnostiqués comme tels par les cinq dermatologues. Ce chiffre illustre clairement la difficulté de détection des mélanomes de notre base d'images.

6.3 Description et analyse des variables d'intérêt extraites des images de tumeurs noires de la peau

Pour réaliser le classifieur de tumeurs noires, nous allons reproduire la démarche suivie par un dermatologue lorsqu'il est confronté à l'analyse d'une lésion. Une fois la lésion localisée, il évalue chacun des critères de la règle ABCD. La fusion de toutes les informations receuillies fournit au dermatologue les éléments nécessaires à un diagnostic argumenté. Il est possible de reproduire ce schéma d'analyse en s'appuyant sur des techniques d'analyse d'images et d'apprentissage statistique. En effet, s'il est possible d'extraire de chacune des images de tumeurs (les n individus), l'information codant chacun des signes de la règle ABCD (les p variables), il est alors possible, en s'appuyant sur le gold standard histologique (y) de construire un classifieur fournissant pour chacune des images une probabilité d'appartenance à la classe des mélanomes. Nous allons décrire, dans cette section, le processus général de construction d'un tel modèle.

Considérons donc une image de tumeur noire de la peau. La première étape de l'analyse consiste à localiser la lésion dans l'image (étape de segmentation). La qualité de la localisation est capitale car elle permet d'évaluer directement la taille de la lésion (D), et fournit les informations sur lesquels s'appuie l'asymétrie (A) et de la régularité de bord (B). Par ailleurs, la segmentation délimite l'espace d'analyse de la couleur (C). Il s'agit donc de développer une approche peu sensible aux artefacts (e.g. poils) et capable de gérer la grande variabilité des tumeurs noires de la peau, en terme de couleur, de forme, de taille, ... La figure 6.5 illustre la grande variabilité des tumeurs noires rencontrées.

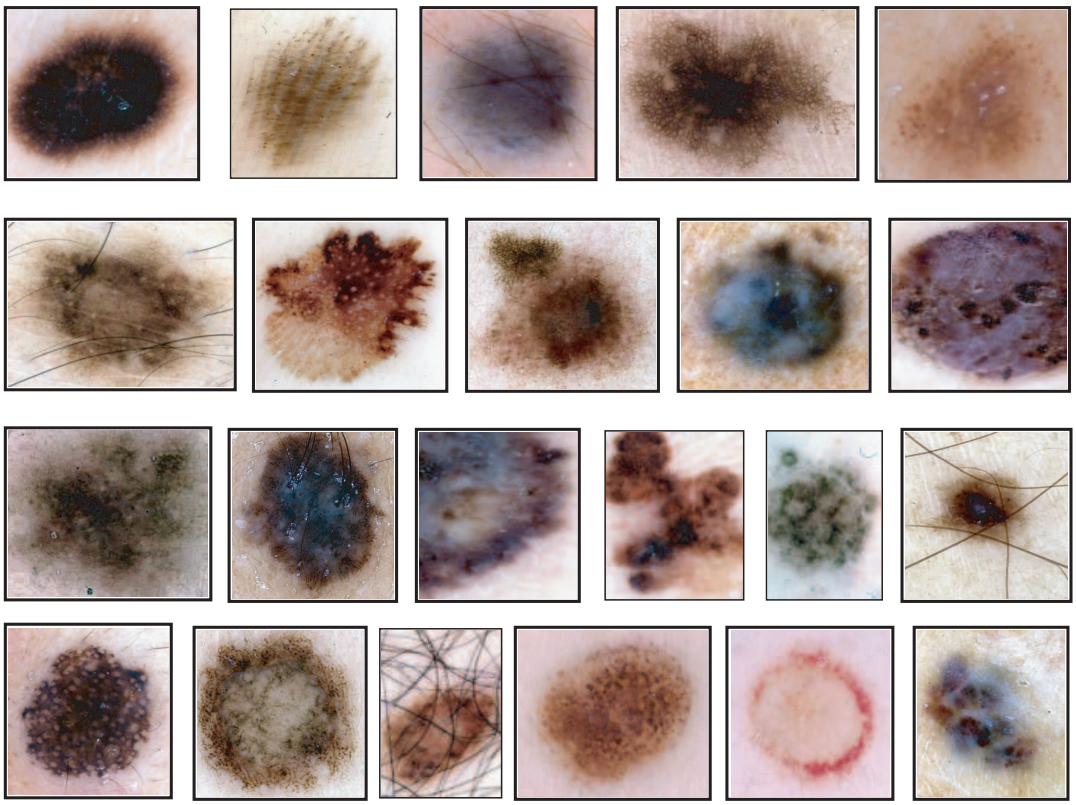

FIG. 6.5: Variabilité des tumeurs noires de la peau.

La segmentation achevée, reste à extraire les paramètres codant chacun des signes de la règle ABCD. Nous présentons dans les prochains paragraphes les approches d'extraction proposées. Afin de mesurer la pertinence des paramètres extraits, nous avons adopté la stratégie générique consistant à confronter les paramètres à l'expertise des dermatologues. Dans de nombreux problèmes médicaux, le taux de bonnes classification n'est pas l'indice central de la qualité d'un classifieur. La sensibilité et la spécificité fournissent souvent des indices plus intéressants car elles tiennent compte des coûts respectifs des différentes erreurs de classification. La sensibilité et la spécificité se calculent à partir d'une matrice de confusion définit par le tableau 6.3.

TAB. 6.3: Matrice de confusion

	Signe observé	**Signe non-observé**
Signe observé par le classifieur	Vrai Positifs (VP)	Faux Positifs (FP)
Signe non-observé par le classifieur	Faux Négatifs (FN)	Vrai Négatifs (VN)

La sensibilité est alors définie $\frac{VP}{VP+FN}$ et la spécificité par $\frac{VN}{FP+VN}$.

Afin de mesurer la qualité d'un modèle en s'appuyant à la fois sur la sensibilité et la

spécificité, Sboner [Sboner et al. (2003)] propose la mesure fournie par (6.1).

$$d = \sqrt{(1 - \text{sensibilité})^2 + (1 - \text{spécificité})^2} \tag{6.1}$$

Remarque 23 *Puisque la sensibilité et la spécificité d'un classifieur idéal égalent* 1*, on en déduit que pour ce classifieur,* $d = 0$*. Ainsi, un modèle est d'autant plus performant que la valeur du d associée est petite. Notons que cette mesure de qualité est intéressante lorsque les classes sont déséquilibrées et que l'on cherche un modèle établissant un compromis entre sensibilité et spécificité. En revanche, si l'on souhaite privilégier soit la sensibilité soit la spécificité, on préférera la courbe ROC.* □

6.3.1 Segmentation : taille et forme

Segmenter une lésion pour un dermatologue est une tâche peu «coûteuse». Pourtant, mettre en place une méthode de segmentation automatique robuste des tumeurs noires est extrêmement difficile. Le «bruit» lié par exemple à l'acquisition d'images dans des conditions plus ou moins bien contrôlée, la perte d'information spatiale et de profondeur, font que la segmentation n'est possible que si la tumeur est caractérisée par des couleurs et/ou des textures qui la différencient du fond. Cette étape est cependant essentielle si nous voulons caractériser les informations de forme et de couleur de la lésion. De nombreuses méthodes ont été mises au point afin de fournir un outil de segmentation robuste. Les difficultés rencontrées lors de cette opération sont en effet nombreuses et variées ; elles concernent à la fois, les variations de luminosité rencontrées dans l'image, la présence d'artefacts (e.g. poils), et la variabilité de couleur, de texture et de taille. Notre processus de segmentation du mélanome intialement proposé par Serruys [Serruys (2003)] et amélioré par Nkengne [Nkengne (2004)] repose sur deux idées principales. La première idée s'inspire des méthodes de segmentation de type région. L'objectif est de regrouper les pixels présentant des similitudes. On se ramène alors à une problématique de classification statistique, en considérant chaque pixel comme un individu. La classification (régression logistique) s'effectue à l'aide d'informations portées par le pixel, son voisinage immédiat et l'image entière. Elle est supervisée par le contour des lésions fourni par les dermatologues. La deuxième idée est de segmenter la tumeur à plusieurs niveaux de résolution. Nous commençons par réduire l'image d'un facteur de 1/16 afin de supprimer tous les détails et d'obtenir un contour grossier de la tumeur. À cette échelle, certains artefacts, comme les poils fins disparaissent. Le contour ainsi obtenu est ensuite affiné en résolution 1/2 et 1. Les classifieurs de niveau 1/2 et 1 ne travaillent que sur une bande, suivant les contours définis au niveau précédent. Ils bénéficient de la meilleure connaissance que nous avons de la couleur de la tumeur et de l'extérieur. La processus segmentation se décompose donc en trois étapes illustrée par la figure 6.6.

Validation des contours de segmentation

Notre objectif est d'évaluer les performances de l'algorithme de segmentation. Du fait de la variabilité intra et inter-observateurs, une réflexion est à mener sur les contours fournis par les dermatologues. Pour ce faire, dix images ont été segmentées par 4 dermatologues. La comparaison de ces contours fournit alors un seuil de bonne segmentation. Les critères de comparaison retenus sont :

i. La distance entre deux contours : Soient $A = \{a_1, a_2, ..., a_{N_a}\}$ et $B = \{b_1, b_2, ..., b_{N_b}\}$ deux

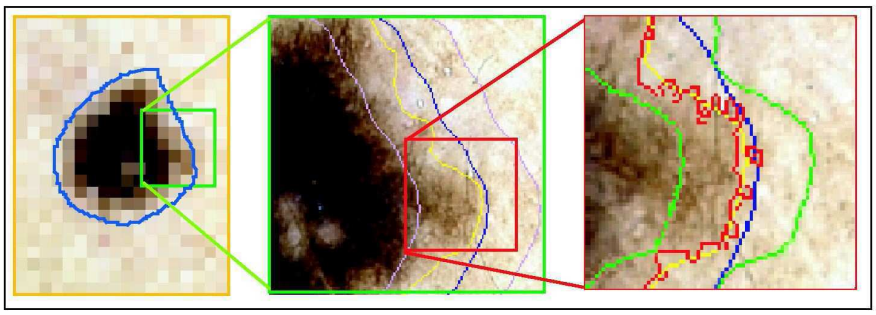

FIG. 6.6: Processus de segmentation des tumeurs noires de la peau.

contours à comparer. On définit alors la distance entre deux contours par :

$$D(A,B) = \frac{1}{N_a}\sum_{i=1}^{N_a} d(a_i, B) + \frac{1}{N_b}\sum_{i=1}^{N_b} d(b_i, A) \tag{6.2}$$

où $d(a_i, B)$ représente la plus petite distance euclidienne entre le point a_i et l'ensemble des points du contour B.

ii. La sensibilité définie par le pourcentage de pixels appartenant à la fois au contour A et au contour B

iii. La spécificité définie par le pourcentage de pixels en dehors du contour de segmentation et classé comme tel par le contour de référence.

Les 10 lésions segmentées par les quatre dermatologues permettent de définir des seuils de référence. Le système de segmentation proposé fournit, pour 86% des lésions, des contours respectant ces seuils.
La Figure 6.7 illustre les résultats de segmentation des images de lésions présentées en Figure 6.5 choisies pour illustrer la grande variabilité des lésions rencontrées. Les erreurs de segmentation apparaissent le plus fréquemment lorsque les lésions atteignent les limites de l'image ou que les bords de la lésion ne contrastent que très peu avec le reste de l'image.

6.3.2 Couleur des tumeurs noires de la peau

La multiplicité de couleurs est caractérisée par la présence de plusieurs plages de couleurs différentes au sein d'une tumeur. Malgré une définition simple, il s'agit d'un signe difficile à détecter : comment discerner une plage de couleurs homogène ? À partir de combien de couleurs la multiplicité de couleurs est-elle anonciatrice de la présence de mélanomes ? Il s'agit d'un signe fondamental du mélanome car très sensible (cf. Figure 6.4). En pratique, un dermatologue localise les plages de couleurs homogènes et détermine ainsi le «nombre de couleurs» en tenant compte de la taille de chacune des zones. Afin de simuler la démarche des dermatologues et quantifier la variété des couleurs d'une image, deux approches non-supervisées ont été utilisées : les cartes de Kohonen et les K-means. Nous présentons, dans le prochain paragraphe, les travaux de Horn [Horn (2004)] dédiés à l'analyse colorimétrique des tumeurs noires.

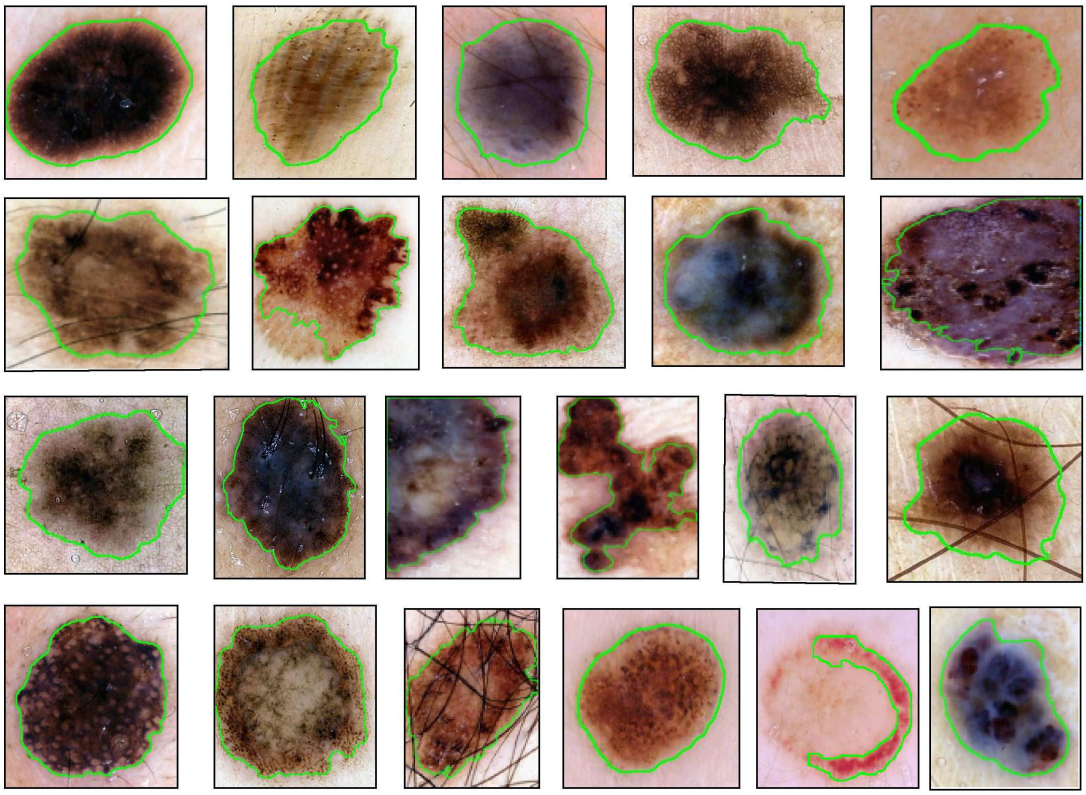

FIG. 6.7: Exemple de segmentation des tumeurs noires de la peau.

Carte de Kohonen

Les cartes de Kohonen sont utilisées afin de capturer les spécificités de couleurs d'une lésion par rapport aux autres. Les cartes de Kohonen fournissent le spectre des couleurs rencontrées sur l'ensemble des tumeurs noires de notre base d'images. Cette approche est donc globale et permet d'évaluer la singularité de chacune des lésions face aux autres.
5 pixels extraits aléatoirement de chacune des 227 lésions servent à la construction de la carte de Kohonen 5×5. La carte de Kohonen traduit donc la diversité de couleur présente sur la totalité des lésions. Considérons alors pour une lésion donnée, la projection sur la carte de Kohonen de l'ensemble des pixels qui la compose. La proportion des pixels de la tumeur projetée sur chacun des 25 neurones définit la signature de la lésion. En effet, les pixels d'une image mono-couleur sont projetés sur des régions voisines tandis que la projection des pixels de lésions multi-couleur s'étalera sur une grande partie de la carte. La figure 6.8 représente les cartes de Kohonen de deux types de lésions : une lésion pour laquelle les dermatologues sont unanimes dans l'évaluation mono-couleur (respectivement multi-couleur). La proportion de pixels projetés sur chacun des 25 neurones de la carte fournit alors un vecteur de caractéristiques de 25 valeurs caractérisant la diversité de couleurs d'une lésion.

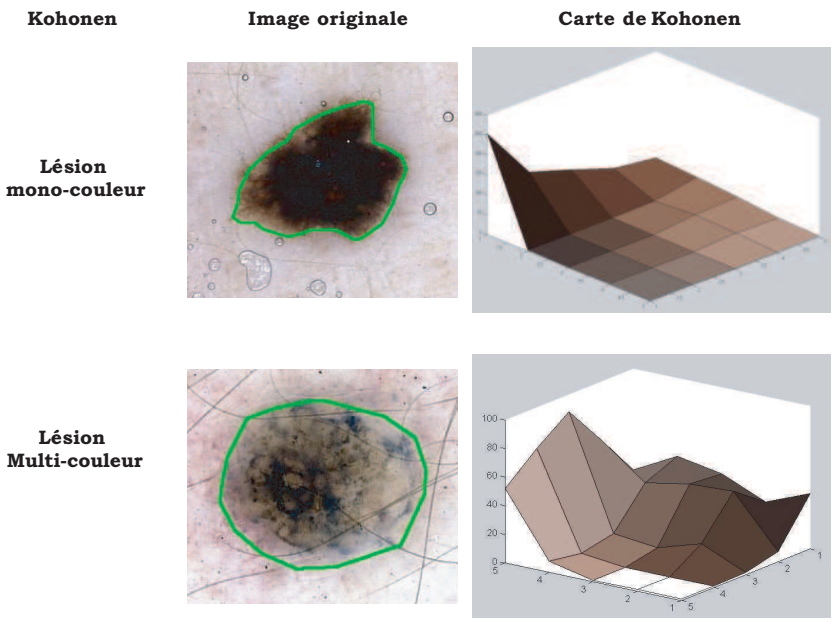

FIG. 6.8: Analyse couleur des tumeurs noires de la peau basée sur les cartes de Kohonen.

Kmeans clustering

Les K-means vont nous permettre de caractériser les couleurs composant les tumeurs noires de manière moins globale que l'approche fondée sur Kohonen. C'est une approche «image spécifique» tenant compte des conditions d'acquisition des images (e.g. surexposition, faible contraste, ...). L'algorithme des K-Means consiste à former K groupes dans une population de n individus. Chaque groupe est caractérisé par un centroïde représentant l'isobarycentre du groupe. Ainsi, les individus sont rattachés à un groupe en fonction de leur distance au centroïde. Dans notre contexte, chacun des n individus (les n pixels composant la lésion) est caractérisé par 3 dimensions (les valeurs R = rouge, V =Vert, B = Bleu). En considérant qu'un dermatologue n'observe que rarement plus de 4 couleurs différentes dans une lésion. Nous avons alors décidé de construire un K-means où $K = 4$. La figure 6.9 illustre l'approche proposée sur deux types d'images.

Les clusters possèdent des propriétés intéressantes :

i. La taille de chaque cluster : les tumeurs mono-couleur présentent 1 ou 2 clusters qui regroupent la majorité des pixels, alors que les pixels des tumeurs multi-couleurs sont répartis équitablement entre les quatre clusters. Cette information est donc codée par 4 variables (nombre de pixels composant chacun des 4 clusters).

ii. La couleur des clusters : les 4 centroïdes de tumeurs mono-couleur sont de couleurs proches, à l'inverse des tumeurs multi-couleurs dont les 4 centroïdes sont de couleurs éloignées. Cette information est codée par 12 variables (la couleur RVB de chacun des centroïdes des 4 clusters).

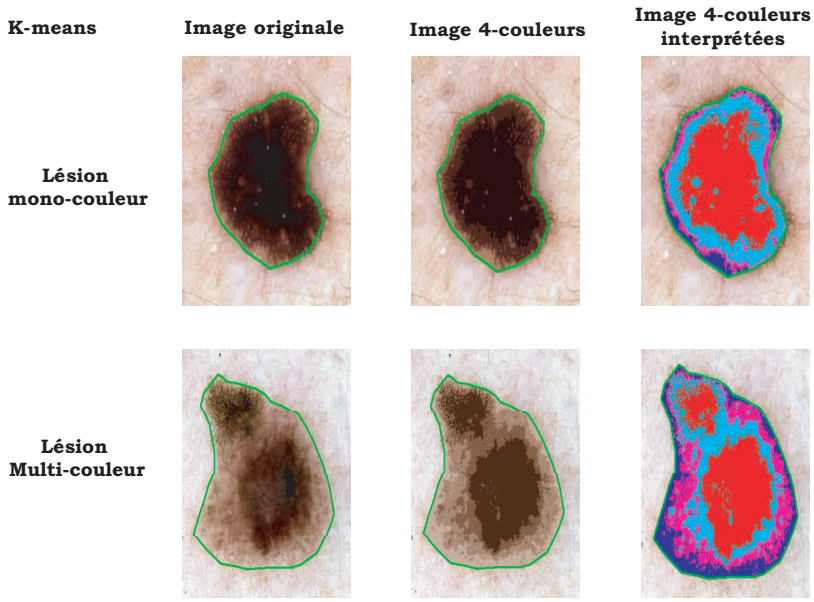

FIG. 6.9: Analyse couleur des tumeurs noires de la peau basée sur l'algorithme de segmentation des K-means.

Ainsi, les K-means engendrent un vecteur de caratéristiques de 16 variables et la carte de Kohonen, un vecteur de caractéristiques de 25 variables. L'information de couleur est donc codée par 41 variables.

Validation des paramètres de couleur

Il est possible d'évaluer la pertinence des paramètres de couleur en construisant un classifieur de type KL-PLS (nombre de composantes retenues = 10 et paramètre de la gaussienne ($k(x,y) = exp(\|x-y\|^2/\sigma)$ où $\sigma = 1000$) reliant ces 41 paramètres à l'expertise des cinq dermatologues relative à la couleur résumée par le vote majoritaire (1 si au moins 3 dermatologues ont détecté la multiplicité de couleur et 0 sinon). La validation du modèle couleur s'appuie sur la technique du leave-one-out.

La Figure 6.4 montre que les dermatologues sont unanimes sur le signe de multiplicité de couleur dans seulement 34% dans ce cas. Il s'ensuit que les indices de sensibilité, de spécificité et de qualité (d) varient grandement d'un dermatologue à l'autre. Le classifieur fournit des performances équivalentes à celle des dermatologues, laissant suggérer que les paramètres extraits contiennent l'information souhaitée.

6.3.3 Asymétrie des tumeurs noires de la peau

Pour déterminer l'asymétrie d'une lésion, les dermatologues s'intéressent à la fois à l'asymétrie de forme et de texture. Notre principe proposé par Nkengne [Nkengne (2004)] consiste à associer à chacune des tumeurs deux axes de symétrie de forme, et deux axes de

TAB. 6.4: Évaluation des paramètres de couleur

Couleur	Sensibilité / Spécificité	d
Dermatologue 2	0.84/0.95	0.17
Dermatologue 4	0.91/0.77	0.24
Classifieur	0.78/0.67	0.39
Dermatologue 5	0.59/0.94	0.44
Dermatologue 3	0.97/0.54	0.46
Dermatologue 1	0.97/0.26	0.74

symétrie de texture. À chacun des axes, est associé un indice de symétrie. Les axes de symétrie de texture et de forme sont déterminés indépendamment car rien ne laisse présager qu'ils sont confondus.

Axes de symétrie de forme

L'indice de symétrie de forme caractérise le recouvrement entre la surface initiale de la tumeur et sa surface après une rotation de 180 degrés autour de son axe de symétrie de forme. Cet indice est uniquement fonction du contour de la lésion. Nous avons choisi de calculer l'axe de symétrie de forme dans l'espace de Hough. Il s'agit d'un espace dans lequel une droite affine d'équation $y = ax + b$ est représentée par un point de coordonnées (R, θ) où R représente la distance de la droite à l'origine et θ l'angle qu'elle fait avec l'axe des abscisses. Supposons que la lésion possède un périmètre de longueur N (en pixels) et admette un axe de symétrie Δ. Le contour de la lésion est porteur de toute l'information sur sa forme. On considère donc tous les couples de points appartenant au bord, et on calcule leurs médiatrices que l'on représente dans l'espace de Hough. Dans le cas idéal où il existe un vrai axe de symétrie, il sera observé $N/2$ fois, car $N/2$ points sont symétriques par rapport à cet axe, et l'admettent donc comme médiatrice. De manière générale, le meilleur axe de symétrie sera donc représenté par le point d'accumulation maximal dans l'espace de Hough. Un deuxième axe de symétrie sera calculé en choisissant le prochain point d'accumulation maximale suivant, séparé du premier d'une distance angulaire d'au moins 30 degrés.

Axes de symétrie de texture

L'indice de symétrie de texture exprime l'erreur quadratique moyenne entre l'intensité des pixels de la lésion en position initiale et l'intensité des pixels de la tumeur retournée de 180 degrés autour de l'axe de symétrie de texture. Schmid-Saugeon [Schmid-Saugeon (2000)] montre que les axes obtenus par l'analyse en composantes principales des pixels de la lésion pondérés par leur intensité fournissent de bons axes de symétrie de texture.

Validation des paramètres d'asymétrie

Ainsi, l'information d'asymétrie est codée par 4 variables : les deux index de symétrie de forme, calculés à partir des axes de Hough et les deux index de symétrie de texture calculés à partir des axes de l'ACP pondérée. Il est alors possible d'évaluer la pertinence de ces paramètres d'asymétrie en construisant un classifieur de type KL-PLS (nombre de composantes retenues = 1 et $k(x, y) = exp(\|x - y\|^2/\sigma)$ où $\sigma = 100$) reliant ces 4 paramètres à l'expertise des cinq dermatologues relative à l'asymétrie résumée par le vote majoritaire (1 si au moins 3

dermatologues ont détecté l'asymétrie et 0 sinon). La validation du modèle d'asymétrie s'appuie sur la technique du leave-one-out. La Figure 6.4 montre que les dermatologues sont unanimes

Tab. 6.5: Évaluation des paramètres d'asymétrie

Asymétrie	Sensibilité / Spécificité	d
Dermatologue 2	0.94/0.87	0.14
Dermatologue 5	0.84/0.88	0.2
Dermatologue 1	0.84/0.88	0.2
Dermatologue 4	0.78/0.98	0.22
Classifieur	0.73/0.72	0.38
Dermatologue 3	0.98/0.61	0.39

sur le signe d'asymétrie à 54% et d'accord à 80% lorsque l'on inclut le 4/1 et 1/4. Il s'ensuit que les indices de sensibilité, de spécificité et de qualité (d) sont élevés pour la majorité des dermatologues. En comparaison, les performances du classifieur sont constrastées laissant suggérer que les paramètres extraits ne capturent pas la totalité de l'information qu'exploitent les dermatologues pour diagnostiquer l'asymétrie d'une lésion. Bien que les différences soient non négligeables, la sensibilité et la spécificité du classifieur restent néanmoins acceptables et les 4 paramètres d'asymétrie seront considérés pour la construction du modèle final.

6.4 Caractérisation des tumeurs noires de la peau par la Kernel Logistique PLS

Afin d'élaborer le modèle final, nous disposons, pour chacune des 227 lésions, de 4 variables d'asymétrie, 41 variables de couleurs et une variable de taille ; soit 46 variables. Nous cherchons à construire un modèle reliant ces variables à l'histologie. L'objectif de la classification est de fournir un modèle prédictif fournissant pour une lésion donnée sa probabilité d'appartenir à la classe des mélanomes. Le classifieur doit gérer la configuration où les données présentent un nombre important de variables, éventuellement corrélées et disposer d'un pouvoir de discrimination performant. La KL-PLS est donc une bonne candidate. Les variables explicatives sont donc les 46 paramètres extraits de l'image et la variable à expliquer est le résultat de l'histologie. Si le problème est uniquement de prédire (comme c'est le cas ici), une méthode doit être jugée par son efficacité et sa robustesse et les techniques de validation croisée apportent alors une solution. En effet, il est fondamental que le classifieur soit pertinent sur la base d'apprentissage, mais également sur des individus qui n'ont pas servi à la construction du modèle. Compte tenu du nombre limité d'images, nous avons opté pour la validation croisée par leave-one-out. Ainsi à chacune des lésions testées est associée une probabilité d'appartenance sur laquelle se base l'évaluation du modèle.
Le tableau 6.6 fournit la sensibilité et la spécificité du diagnostic des cinq dermatologues par rapport à l'histologie. Étonnamment, on constate une grande variabilité de la sensibilité et de la spécificité du diagnostic. Ceci peut s'expliquer par le contexte libre de la tâche (aucune implication réelle) mais également par le fait que le diagnostic n'est établi qu'à partir de l'image. Notons que les performances les plus élevées sont à attribuer aux dermatologues ayant une activité hospitalière, et donc plus couramment confrontés à la vision de mélanomes.
Le tableau 6.7 fournit la sensibilité et la spécificité de la décison thérapeutique des cinq dermatologues par rapport à l'histologie.

TAB. 6.6: Évaluation du diagnostic des cinq dermatologues

Diagnostic	Sensibilité / Spécificité
Dermatologue 1	0.62/0.90
Dermatologue 2	0.78/0.85
Dermatologue 3	0.59/0.71
Dermatologue 4	0.81/0.90
Dermatologue 5	0.71/0.80

TAB. 6.7: Évaluation de la décision thérapeutique des cinq dermatologues

Exérèse	Sensibilité / Spécificité
Dermatologue 1	0.84/0.63
Dermatologue 2	0.93/0.63
Dermatologue 3	0.84/0.39
Dermatologue 4	0.90/0.55
Dermatologue 5	0.87/0.63

Comme on pouvait s'y attendre, nous constatons une dégradation de la spécificité au profit de la sensibilité : les dermatologues ne prennent plus le moindre risque. Il est néanmoins surprenant de constater qu'aucun des dermatologues n'atteint une sensibilité de 1. Nous pouvons avancer les mêmes arguments que précédemment, à savoir le contexte de l'expérience et l'image comme unique support diagnostic. De la même manière que précédemment les médecins ayant une activité hospitalière prennent les décisions les meilleures.
Nous souhaitons comparer la sensibilité et la spécificité de la KL-PLS à celles des cinq dermatologues. Dans ce contexte, un outil plus adapté que l'indice d est la courbe ROC (Abréviation de l'anglais **R**eceiver **O**perating **C**haracteristic curve). Cette courbe résume les performances de toutes les règles de classement (en termes de sensibilité et de spécificité) que l'on peut obtenir en faisant varier le seuil de décision. La courbe ROC est fréquemment utilisée dans le domaine médical. En effet, les répercussions d'une erreur de diagnostic ne sont pas équilibrées : prédire un mélanome comme bénin n'a pas le même impact que prédire un naevus bénin comme mélanome. Les résultats de la KL-PLS ($\sigma = 1e^{-3}$ et nombre de composantes = 3) et de la régression logistique (fournie comme référence) sous forme de courbe ROC sont représentés sur la figure 6.10. L'aire sous la courbe ROC (AUC : Area Under The ROC Curve) fournit un indice de qualité du modèle générée. L'AUC de la KL-PLS (0.84) exprime une amélioration significative par rapport à l'AUC de la régression logistique (0.73). Nous pouvons constater que la KL-PLS atteint des performances équivalentes à celles des dermatologues en termes de diagnostic et de décision thérapeutique et souligner l'utilité de la prédiction en terme de probabilité. En effet, elle permet de reproduire le comportement des dermatologues : le diagnostic et la décision thérapeutique correspondent simplement à une modulation du seuil de décision. Cette souplesse dans la prise de décision est fondamentale dans le contexte du mélanome car le coût des erreurs n'est pas symétrique : la sensibilité est bien plus importante que la spécificité puisque la priorité «absolue» est de détecter les mélanomes (quitte à diagnostiquer des naevus bénins comme malins).

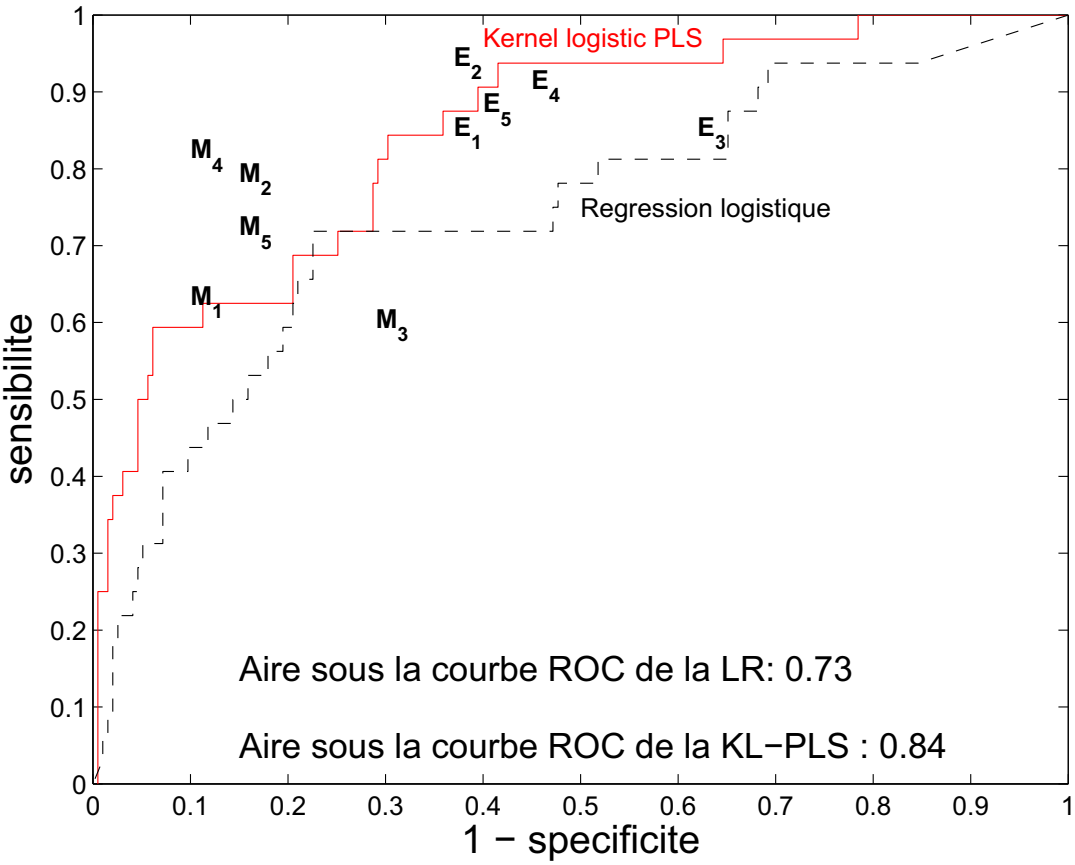

FIG. 6.10: Applications tumeurs noires de la peau : courbes ROC de la Kernel Logistique PLS et de la régression logistique.

6.5 Conclusion

La complexité du modèle fourni par la KL-PLS implique qu'il est difficilement interprétable. Dans le contexte de la caractérisation du mélanome, il est évident que la détection des mélanomes est une priorité mais il est également indéniable que les justifications qui nous ont conduit à prendre cette décision sont essentielles. Le modèle final ne fournit pas ces justifications. En revanche, les modèles intermédiaires (couleur, asymétrie) permettent, outre de mesurer la pertinence des paramètres extraits, d'argumenter et justifier le diagnostic. Ces modèles intermédiaires sont donc particulièrement intéressants. Les résultats obtenus (en terme de sensibilité et spécificité) par notre système de caratérisation sont comparables aux performances des cinq dermatologues ayant participé au projet. Ces résultats, encourageants nous ont conduit à entamer une collaboration avec un industriel afin de développer un prototype du système de caractérisation. La figure 6.11 présente le prototype. La validation à grande échelle du système de caractérisation devra être entreprise et le projet DANAOS (Diagnostic and Neural Analysis of Skin Cancer), étude multi-centrique regroupant 14 pays européens devrait faciliter l'évaluation du système proposé si les données sont mises à la disposition des chercheurs. En effet, ce projet a permis d'évaluer plus de 25 000 images sur lesquelles nous pourrions valider le prototype. Pour

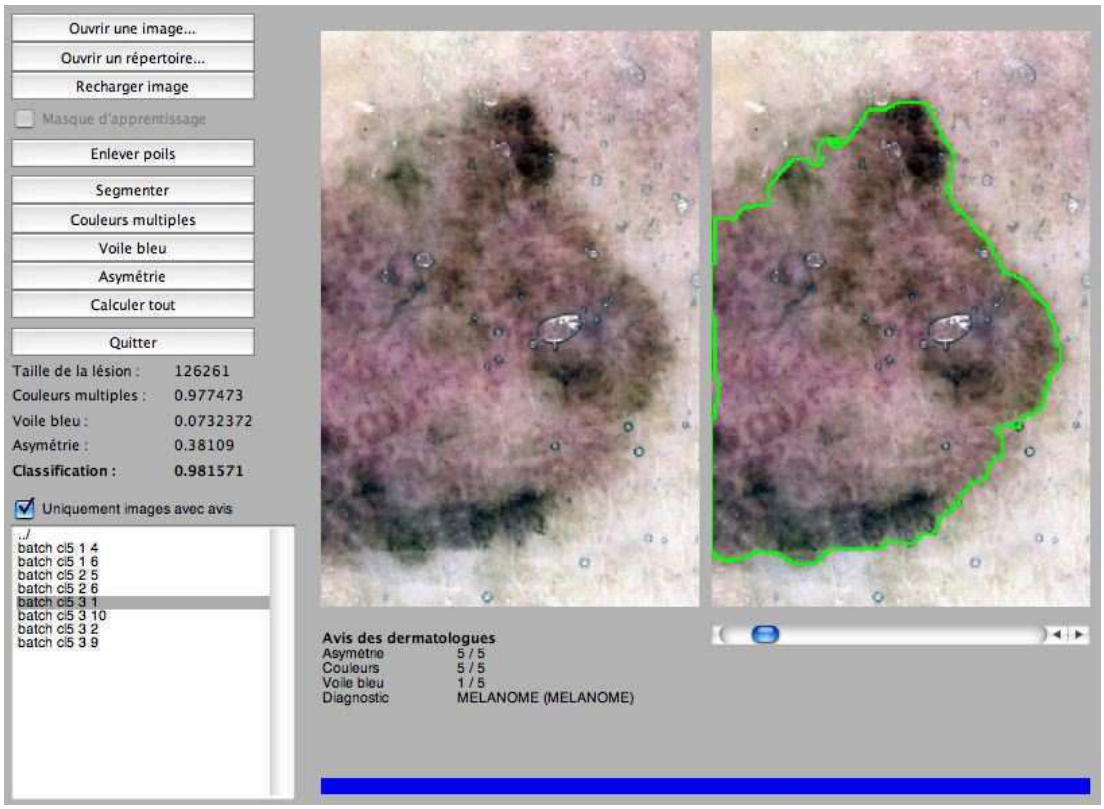

FIG. 6.11: Prototype du système de caractérisation des tumeurs noires de la peau.

conclure, divers paramètres non inclus dans l'image mais cependant informatifs et d'importance parfois non négligeable dans l'établissement d'un diagnostic (antécédents, âge du patient, etc.) devront être pris en compte pour affiner le diagnostic.

Conclusion

Ce travail nous a permis d'explorer la problématique des données de grande dimension et de proposer de nouvelles approches compétitives. Nous avons vu que les méthodes à noyau fournissent un cadre algorithmique propice à la gestion de ce type de données et que, jumelées à des principes de contrôle de complexité telles que la régularisation, elles offrent de très bonnes performances. Ces performances se gagnent au prix d'une perte de lisibilité des modèles générés. Prédire ou comprendre ? La compexité des méthodes à noyau font souvent d'elles des boîtes noires que l'on ne peut, en général, pas interpréter. Ces modèles n'ont pour vocation que de prédire et doivent donc être jugés du point de vue leur efficacité et de leur robustesse : comment donc choisir une méthode ? Les techniques de choix de modèle, par exemple par validation croisée, fournissent l'outil de prédilection. Le chapitre de validation nous conduit à conclure à l'équivalence en terme de pouvoir prédictif de l'ensemble des méthodes décrites.

Le choix d'une méthode doit donc s'appuyer sur d'autres arguments : des critères de simplicité de mise en oeuvre, de facilité d'ajustement, de rapidité d'exécution, de visualisation de la structure des données, ... Nous pensons avoir montré que les méthodes basées sur les principes de réduction de dimension offre une alternative intéressante aux SVM et autres KLR et que les approches que nous proposons dans ce document, répondent à ces critères de qualité. En premier lieu, le chapitre de validation atteste de leur pouvoir prédictif compétitif (cf. KL-PLS, KLM-PLS ...). Sur benchmarks, les performances des méthodes proposées sont au moins équivalentes à celles des méthodes références de la littérature. De plus, le réglage des paramètres des méthodes de réduction de dimension semble plus naturel que celui du paramètre λ de la régularisation de Tikhonov : le nombre de composantes est un paramètre discret inférieur au rang de la matrice des données alors que le paramètre λ est à valeur dans $\mathbb{R}^+$. Enfin, l'exploration des résultats par examen visuel permet, outre l'évaluation des degrés de liaison entre classes d'individus, de caractériser la nature des erreurs : les erreurs de classification sont-elles frontières ou structurelles ? Ajoutons que les prédictions en terme de probabilités peuvent être particulièrement utiles dans de nombreux contextes et notamment dans le domaine médical : l'application de nos méthodes à la problématique des tumeurs noires de la peau en atteste.

Ajoutons que, dans le cadre de nos activités de laboratoire de l'INSERM 678, nos méthodes ont été appliquées à la détection d'anomalies de contraction du ventricule gauche en échocardiographie, Ruiz Domingez et al. [Ruiz Dominguez et al. (2005)] et également à l'analyse d'images scintigraphiques 3D afin d'établir un diagnostic différentiel de la maladie de Parkinson (maladies neuro-dégénératives), Billard [Billard (2006)].

En termes de perpectives, trois idées se profilent :

1. La visualisation permet de circonscrire le domaine de compétence du modèle. En effet, lorsque l'on cherche à prédire la classe d'un nouvel individu, sa position sur la carte peut fournir des informations essentielles : on accordera une grande confiance à la prédiction d'un individu fondu dans la masse de ses partenaires et donc caratéristique de la classe prédite. À l'opposé, une probabilité d'appartenance proche de 1 pour un individu éloigné de ses partenaires est sans aucun doute à nuancer. Les problèmatiques liées au domaine de compétence d'un classifieur ne sont pas abordés dans ce travail mais paraissent être des perpectives de recherches prometteuses.

2. Automatiser le choix du nombre de composantes apparaît comme un développement futur important à mettre en place. La validation croisée sur laquelle nous nous sommes appuyés pour sélectionner les paramètres d'ajustement, coûteuse en temps de calcul, devrait pouvoir servir de référence dans ce but. On peut penser à des choix de modèles basées sur la minimisation d'une borne à la SRM. En effet, la borne de l'inégalité (1.3) est la somme du risque empirique et d'un terme dépendant du rapport h/n. Ainsi, à n fixé, la minimisation de la borne fournit un critère de choix de modèles qui ne fait appel à aucune procédure de validation croisée. En pratique la difficulté d'une telle approche réside dans le calcul de la VC dimension. Or, dans le contexte des modèles de la forme (2.27), dans laquelle s'inscrivent par exemple la KL-PLS, Cherkassky et al. [Cherkassky et al. (1999)] proposent d'estimer la VC dimension par le nombre d'éléments intervenant dans la combinaison linéaire + 1 (c'est à dire le nombre de paramètres libres). De plus, partant du constat que le niveau de confiance de la borne croît avec la valeur de n, Cherkassky et al. proposent de fixer la probabilité η à $1/\sqrt{n}$. Reste alors à choisir le nombre de composantes fournissant la borne minimum.

3. Enfin, la limite algorithmique des approches proposées est reliée au nombre d'observations bien plus qu'au nombre de variables (à l'instar des méthodes à noyau). Ces approches s'appuient sur des transformations de type Empirical Kernel Map et ne se restreignent nullement à la gestion de matrice $n \times n$: les matrices rectangulaires sans modification des algorithmes peuvent alors être analysées. Les méthodes de sélection de variables (basées, par exemple, sur le test de Wald), couplées à la KL-PLS ou à la KLM-PLS sont également des perpectives de recherches intéressantes. La KL-PLS et la KLM-PLS devraient alors pouvoir s'appliquer à des problématiques où à la fois la taille de l'échantillon et le nombre de variables sont importants.

Bibliographie

Albert, A. and Anderson, J. A. (1984). On the existence of maximum likelihood estimates in logistic regression models. *Biometrika*, 71 :1–10.

Allen., D. M. (1974). The relationship between variable selection and data augmentation and a method for prediction. *Technometrics*, 16(1) :125–127.

Allison, P. D. (1999). *Logistic regression using the SAS system*. SAS intitute.

Aronszajn, N. (1950). Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68 :337–404.

Bach, F. R. and Jordan, M. I. (2005). Predictive low-rank decomposition for kernel methods. In *Proceedings of the twenty-second International Conference on Machine Learning*.

Baffi, G., Martin, E. B., and Morris, A. J. (1999a). Non-Linear Projection to Latent Structures Revisited : (The neural network PLS Algorithm. *Computer and Chemical engineering*, 23 :1293–1307.

Baffi, G., Martin, E. B., and Morris, A. J. (1999b). Non-Linear Projection to Latent Structures Revisited : The Quadratic PLS Algorithm. *Computer and Chemical engineering*, 23 :395–411.

Barker, M. and Rayens, W. S. (2003). Partial Least square for discrimination. *Journal of Chemometrics*, 17 :166–173.

Bastien, P. (2004). PLS-Cox model. In *Proceedings in Computational Statistics*, pages 655–662. Physica-Verlag, Springer.

Bastien, P., Vinzi, V. E., and Tenenhaus, M. (2005). PLS Generalized Linear Regression. *Computational Statistics and Data Analysis*, 48 :17–46.

Bennett, K. P. and Embrechts, M. J. (2003). An Optimization Perspective on Kernel Partial Least Squares Regression. *Advances in learning Theory : Methods, Models and Applications, NATO Sciences Series III : Computer & Systems Sciences*, 190 :227–250.

Berglund, A. and Wold, S. (1997). INLR, Implicit Non-Linear Latent Variable Regression. *Journal of Chemometrics*, 11 :141–156.

Billard, C. (2006). Développement d'un outil de diagnostic différentiel des maladies à symptômes parkinsoniens. Mémoire de fin d'étude, Telecom INT.

Binder, M., Kittler, H., Dreiseitl, S., Ganster, H., Wolff, K., and Pehamberger, H. (2000). Computer-aided epiluminescence microscopy of pigmented skin lesions : the value of clinical data for the classification process. *Melanoma Research*, 10(6) :556–561.

Binder, M., Kittler, H., Seeber, A., Steiner, A., Pehamberger, H., and Wolff, K. (1998). Epiluminescence microscopy-based classification of pigmented skin lesions using computerized image analysis and an artificial neural network. *Melanoma Research*, 8(3) :261–266.

Boser, B., Guyon, I., and Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *COLT'92*, pages 144–152.

Bottou, L., Cortes, C., Denker, J., Drucker, H., Guyon, I., L. Jackel, Y. Lecun, U. M., Sackinger, E., Simard, P., and Vapnik, V. N. (1994). Comparison of classifier method : A case study in handwritting digit recognition. In *Proceedings of the International Conference of Pattern Recognition*, pages 77–87.

Bredensteiner, E. and Bennett, K. P. (1999). Multicategory classification by Support Vector Machines. *Computational Optimization and Applications*, 12.

Bull, S. B., Mak, C., and Greenwood, C. M. T. (2002). A modified score function estimator for multinomial logistic regression in small samples. *Computational Statistics and Data Analysis*, 39.

Butler, N. and Denham, M. (2000). The peculiar shrinkage properties of Partial Least Squares regression. *Journal of the Royal Statistical Society*, 62 :585–594.

Cherkassky, V., Shao, X., Mulier, F. P., and Vapnik, V. N. (1999). Model Complexity Control for Regression Using VC Generalization Bounds. *IEEE Transactions on neural networks*, 10 :1075–1989.

C.Lingjaerde, O. and Christophersen, N. (2000). Shrinkage structure of Partial Least Squares. *Scandinavian Journal of Statistics*, 27 :459–473.

Dayal, B. S. and MacGreggor, J. F. (1997). Improved the PLS Algorithms. *Journal of Chemomerics*, 11 :73–85.

de Jong, S. (1993a). PLS fits closer than PCR. *Journal of Chemometrics*, 9 :551–557.

de Jong, S. (1993b). SIMPLS : An alternative approach to Partial Least Square Regression. *Chemometrics and Intelligent Laboratory Systems*, 18 :251–263.

de Jong, S. (1995). PLS shrinks. *Journal of Chemometrics*, 9 :323–326.

Delaunay, M. (2004). Mélanome cutané : le diagnostic précoce, un devoir d'efficacité. In *La revue du praticien*.

Duan, K. and Keerthi, S. (2003). Which is the best multiclass SVM method ? An empirical study. Technical report, Department of Mechanical Engineering, National University of Singapore.

Evgeniou, T. and Pontil, M. (1999). On the V gamma dimension for regression in Reproducing Kernel Hilbert Space. Technical report, Technical Report A.I. Memo No. 1656, Artificial Intelligence Lab, MIT.

Evgeniou, T., Pontil, M., and Poggio, T. (2000). Regularization networks and Support Vector Machines. *Advances in Computational Mathematics*, 13 :1–50.

Firth, D. (1993). Bias reduction of maximum likelihood estimates. *Biometrika*, 80 :27–38.

Fisher, R. A. and Yates, F. (1938). Statistical tables for biological, agricultural and medical research. *Edinburgh : Oliver and Boyd*.

Fort, G. and Lambert-Lacroix, S. (2005). Classification using partial least squares with penalized logistic regression. *Bioinformatics*, 21 :1104–1111.

Friedman, J. H. (1989). Regularized discriminant analysis. *Journal of the American Statistical Association*, 84 :165–175.

Friedman, J. H. (1996). Another Approach to Polychotomous Classification. Technical report, Department of Statistics, Stanford University.

Ganster, H., Pinz, A., and Rohrer, R. (2001). Automated melanoma recognition. *IEEE Transactions on Medical Imaging*, 20(3) :233–239.

Garthwaite, P. H. (1994). An Interpretation of Partial Least Squares. *Journal of the American Statistical Association*, 89 :122–127.

Goutis, C. (1996). Partial Least Squares algorithm yields shrinkage estimators. *The Annals of Statistics*, 24 :816–824.

Grob, J. J. and Richard, M. A. (2004). épidémiologie et prévention du mélanome. In *La revue du praticien*.

Guermeur, Y. (2002). Combining discriminant models with multiclass SVMs. *Pattern Analysis and Applications*, 5(2) :168–1799.

Hanley, J. A. (1989). Receiver Operating Characteristic methodology : the state of the art. *Critical Reviews in Diagnostic Imaging*, 29 :307–335.

Heinze, G. and Schemper, M. (2002). A solution to the problem of separation in logistic regression. *Statistics in medecine*, 21 :2409 – 2419.

Helland, I. S. (1988). On the structure of Partial Least Squares regression. *Communications in Statistics Simulation and Computation*, 17 :581–607.

Hoerl, A. E. and Kennard, R. W. (1970). Ridge regression : biased estimation for nonorthogonal problems. *Technometrics*, 12 :55–77.

Horn, J. (2004). étude colorimétrique et détection des mélanomes. Mémoire de dea d'informatique médicale et technologies de la communication, université Paris VI.

Höskuldsson, A. (1988). PLS regression methods. *Journal of Chemometrics*, 2 :211–228.

Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of Educational PsychologyChemometrics*, 24 :417–441 and 498–520.

Hsu, C. W. and Lin, C. J. (2002). A comparison of methods for multiclass Support Vector Machines. *IEEE on neural networks*, 13 :415–425.

Ivanov, V. V. (1962). On linear problems which are not well-posed. *Soviet Math. Docl.*, 3 :981–983.

Joachims, T. (1999). Making Large-Scale SVM Learning Pratical. *Advances in Kernel Methods - Support Vector Learning*.

Keerthi, S., Duan, K., Shevade, S., and Poo, A. (2002). A Fast Dual Algorithm for Kernel Logistic Regression. In *Proceeding of the nineteenth international conference on machine learning*.

Kimeldorf, G. and Wahba, G. (1971). Some results on Tchebycheffian spline functions. *Journal of Mathematical Analysis and Applications*, 33 :82–95.

Krämer, N. (2006). An Overview of the Shrinkage Properties of Partial Least Squares Regression. *to appear in the Special Issue on PLS in Computational Statistics*.

Kreßel, U. (1999). Pairwise classification and support vectors machines. *Advances in Kernel Methods - Support Vector Learning*, pages 255–268.

Lebart, L. (1997). Méthodes factorielles. In Thiria, S., Lechevallier, Y., Gascuel, O., and Canu, S., editors, *Modèles statistiques pour données qualitatives*. Dunod.

Lee, Y., Lin, Y., and Wahba, G. (2001). Multicategory Support Vector Machines. Technical report 1043, Department of Statistics, University of Wisconsin.

Lin, Y. (2002). Support Vector Machines and the Bayes Rule in Classfication. *Data Mining and Knowledge Discovery*, 6 :259–275.

Lindgren, F., Geladi, P., and Wold, S. (1993). The Kernel algorithm for PLS. *Journal of Chemometrics*, 7 :45–59.

Lindgren, F., Geladi, P., and Wold, S. (1994). Kernel-based PLS regression : cross-validation and applications to spectral data. *Journal of Chemometrics*, 8 :377–389.

Manne, R. (1987). Analysis of Two Partial Least Squares Algorithms for Multivariate Calibration. *Chemometrics and Intelligent Laboratory Systems*, 2 :187–197.

Martens, H. (1985). Multivariate Calibration. Thesis, Technical university of Norway, Trondheim.

Mercer, J. (1909). Functions of positive and negative type and their connection with the theory of integral equations. *Philosophical Transactions Royal Society London*, A209 :415–446.

Nguyen, D. and Rocke, D. (2002). Tumor classification by Partial Least Squares using microarray gene expression. *Bioinformatics*, 18 :39–50.

Nguyen, D. and Rocke, D. (2004). On Partial Least Squares dimension reduction for microarray-based classification. *Computational Statistics and Data Analysis*, 46 :407–425.

Nkengne, A. (2004). Segmentation du mélanome par apprentissage et étude de l'asymétrie. Mémoire de dea d'informatique médicale et technologies de la communication, université Paris VI.

Osuna, E., Freund, R., and Girosi, F. (1997). An improved training algoithm for Support Vector Machines. *Neural Networks for Signal Processing*, pages 276–285.

Pearson, K. (1901). On lines and planes of closest fit to points in space. *Philosophical Magazine*, 2 :559–572.

Phatak, A. and de Hoog, F. (2002). Exploiting the Connection between PLS, Lanczos, and Conjugate Gradient : Alternative proofs of some Properties of PLS. *Journal of Chemometrics*, 16 :361–367.

Platt, J. C. (1999). Fast Training of Support Vector Machines using Sequential Minimal Optimization. *Advances in Kernel Methods - Support Vector Learning*, pages 169–185.

Rännar, S., Geladi, P., Lindgren, F., and Wold, S. (1995). A PLS Kernel algorithm for data sets with many variables and few objects : Part II : Cross-validation, missing data and examples. *Journal of Chemometrics*, 9 :459–470.

Rännar, S., Lindgren, F., Geladi, P., and Wold, S. (1994). A PLS Kernel algorithm for data sets with many variables and fewer objects. Part I : Theory and algorithm. *Journal of Chemometrics*, 8 :111–125.

Rätsch, G., Onoda, T., and Muller, K. R. (2001). Soft margin for adaboost. *Machine Learning*, 42 :287–320.

Rifkin, R. (2002). Everything Old is New Again : A Fresh Look at Historical Approaches in Machine Learning. Thesis, Massachusetts Institute of Technology.

Rifkin, R. and Klautau, A. (2004). In defense of one-vs-all classification. *Journal of Machine Learning Research*, 5 :101–141.

Rosado, B., Menzies, S., Harbauer, A., Pehamberger, H., Wolff, K., Binder, M., and Kittler, H. (2003). Accuracy of computer diagnosis of melanoma : a quantitative meta-analysis. *Archives of Dermatology*, 139(3) :361–367.

Rosipal, R. and Trejo, L. J. (2001). Kernel Partial Least Squares Regression in Reproducing Kernel Hilbert Space. *Journal of Machine Learning Research*, 2 :97–123.

Rosipal, R., Trejo, L. J., and Matthews, B. (2003). Kernel PLS-SVC for linear and nonlinear classification. In *Proceeding of the twentieth international conference on machine learning (ICML-2003)*.

Rosset, S., Zhu, J., and Hastie, T. (2004a). Boosting as a regularized path to a maximum margin classifier. *Journal of Machine Learning Research*, 5 :941–973.

Rosset, S., Zhu, J., and Hastie, T. (2004b). Margin maximizing loss functions. In *Advances in Neural Information Processing Systems (NIPS) 16*.

Ruiz Dominguez, C., Kachenoura, N., Mulé, S., Tenenhaus, A., Delouche, A., Nardi, O., Gérard, O., Diebold, B., Herment, A., and Frouin, F. (2005). Classification of segmental wall motion in echocardiography using quantified parametric images. In *Functional Imaging and Modeling of the Heart*. Springer Berlin / Heidelberg.

Saunders, C., Gammerman, A., and Vovk, V. (1998). Ridge regression learning algorithm in dual variables. In *Proceeding of the fifteenth international conference on machine learning*, pages 515–521.

Sboner, A., Eccher, C., Blanzieri, E., Bauer, P., Cristofolini, M., Zumiani, G., and Forti, S. (2003). A multiple classifier system for early melanoma diagnosis,. *Artificial Intelligence in Medecine*, 27 :29–44.

Schmid-Saugeon, P. (2000). Symmetry axis computation for almost-symmetrical and asymmetrical objects : Application to pigmented skin lesions. *Medical Image Analysis*, 4 (3) :269–282.

Schölkopf, B., Smola, A. J., and Müller, K. R. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10 :1299–1319.

Serruys, C. (2003). Classification automatique des tumeurs noires de la peau par des techniques numériques d'analyse d'images fondées sur des méthodes d'apprentissage par l'exemple : aide au dépistage des mélanomes. Thèse d'informatique médicale, université Paris V.

Shawe-Taylor, J. and Cristianini, N. (2004). *Kernel methods for pattern analysis*. Cambridge University Press.

Shen, L. and Tan, E. C. (2005). Dimension reduction-based penalized logistic regression for cancer classification using microarray data. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2(2) :166–175.

Smola, A. J. and Schölkopf, B. (2000). Sparse greedy matrix approximation for machine learning. In *Proceeding of the seventeenth international conference on machine learning*.

Tenenhaus, A. (2002). La Régression Logistique PLS validée par bootstrap. In *Mémoire de DEA de Statistique, Université Pierre et Marie Curie*.

Tenenhaus, A., Giron, A., Saporta, G., and Fertil, B. (2005). Kernel Logistic PLS : a new tool for complex classification. In *11th International Symposium on Applied Stochastic Models and Data Analysis*, Brest.

Tenenhaus, M. (1998). *La Régression PLS*. Éditions Technip.

Tenenhaus, M. (2005). La régression logistique PLS. In Droebeke, J., Lejeune, M., and Saporta, G., editors, *Modèles statistiques pour données qualitatives*. Technip.

Tikhonov, A. N. and Arsenin, V. Y. (1977). *Solutions of Ill-posed problems*. Wiley.

Vapnik, V. N. (1982). *Estimation of Dependences Based on Empirical Data*. Springer-Verlag.

Vapnik, V. N. (1995). *The Nature of Statistical Learning Theory*. Springer-Verlag.

Vapnik, V. N. (1998). *Statistical Learning Theory*. Wiley.

Wahba, G. (1999). Support Vector Machines, Reproducing Kernel Hilbert Spaces and randomized GACV. *Advances in Kernel Methods - Support Vector Learning*, pages 69–88.

Weston, J. and Watkins, C. (1998). Multiclass Support Vector Machines. Technical Report CSD-TR-98-04, University of London, Department of Computer Science.

Williams, C. K. I. and Seeger, M. (2000). Effect of the input density distribution on kernel-based classifiers. In *Proceeding of the seventeenth international conference on machine learning*.

Wold, H. (1966). Estimation of principal components and related models by iterative least squares.

Wold, S. (1992). Non-Linear Partial Least Squares Modelling II : Spline inner function. *Chemometrics and Intelligent Laboratory Systems*, 14 :71–84.

Wold, S., Kettaneh-Wold, N., and Skegerberg, B. (1989). Non-Linear PLS Modelling. *Chemometrics and Intelligent Laboratory Systems*, 7 :53–65.

Wold, S., Martens, L., and Wold, H. (1983). The multivariate calibration problem in chemistry solved by the PLS method. In *Proceedings Conf. Matrix Pencils, Ruhe A. & Kåstrøm B, Lecture Notes in Mathematics*, pages 286–293. Springer Verlag.

Wu, W., Massart, D. L., and de Jong, S. (1997). The Kernel PCA algorithms for wide data – part II : Fast cross validation and application in classification of NIR data. *Chemometrics and Intelligent Laboratory Systems*, 37 :271–280.

Zhu, J. and Hastie, T. (2005). Kernel Logistic Regression and the Import Vector Machine. *Computational and Graphical Statistics*, 14(1) :185–205.

Zwald, L., Vert, R., Blanchard, G., and Massart, P. (2004). Kernel Projection Machine : a New Tool for Pattern Recognition. In *Advances in Neural Information Processing Systems 17*, pages 1649–1656.