Université Toulouse II

Thèse

présentée par **M^me^ Nathalie VILLA - VIALANEIX**
pour obtenir le grade de docteur de
l'Université Toulouse II, Le Mirail.

Spécialité : Mathématiques Appliquées.

Éléments d'apprentissage en statistique fonctionnelle

Classification et régression fonctionnelles par réseaux de neurones et Support Vector Machine

Soutenue le 21 octobre 2005 devant la commission d'examen composée de :

Louis Ferré	Université Toulouse II	Directeur de thèse
Philippe Besse	Université Toulouse III	Examinateur
Marie Cottrell	Université Paris I	Rapporteur
Fabrice Rossi	INRIA, Rocquencourt	Examinateur
Gilbert Saporta	CNAM, Paris	Rapporteur
Pascal Sarda	Université Toulouse II	Examinateur

Thèse préparée au sein de l'**Équipe GRIMM**, Université Toulouse II, Le Mirail

Remerciements

Tout d'abord, je tiens à remercier Louis Ferré pour son aide et sa disponibilité tout au long de cette thèse malgré les moments particulièrement difficiles qu'il a vécus. Je lui suis reconnaissante de m'avoir offert la possibilité de travailler sur des thématiques ouvertes et novatrices.

Je tiens ensuite à exprimer ma vive gratitude à Fabrice Rossi avec qui j'ai collaboré pendant cette dernière année : cette coopération fut pour moi enrichissante et motivante mais aussi très agréable, grâce à son grand sens de l'humour. Je le remercie également d'avoir examiné, avec attention et compétence, mon travail.

J'adresse toute ma reconnaissance à Marie Cottrell pour avoir accepté d'être rapporteuse[1] de cette thèse et pour m'avoir fait le très grand plaisir de trouver un peu de temps pour assister à ce jury ; je tiens à lui faire connaître mon admiration pour son travail et son engagement. Je suis également très reconnaissante à Gilbert Saporta de m'avoir fait l'honneur d'être le second rapporteur ; je le remercie vivement pour les remarques fructueuses qu'il a pu me faire afin d'améliorer ce mémoire.

Je remercie aussi particulièrement Philippe Besse et Pascal Sarda de participer à ce jury. Les échanges que j'ai eus avec eux ont été très enrichissants pour moi et je suis donc particulièrement touchée qu'ils aient accepté d'évaluer cette thèse.

Je remercie chaleureusement ceux de mes collègues de l'Université Toulouse Le Mirail qui rendent le travail plus joyeux et réussissent à supporter mon mauvais caractère. Je rends particulièrement hommage à Bertrand pour son enthousiasme communicatif et pour son investissement permanent : il nous a, à tous, souvent rendu la vie plus facile. Je tiens également à exprimer toute mon amitié à Martin Paegelow avec qui les collaborations sont toujours un plaisir. Je n'oublie pas non plus tous ceux qui ont œuvré pour me permettre de réaliser cette thèse dans de bonnes conditions ; ainsi, je voudrais exprimer ma gratitude à Jean-Marie Cellier pour sa grande ouverture d'esprit.

Enfin, j'ai une pensée particulière pour mes amis et ma famille ainsi que pour mon petit coin de verdure, au fin fond de la Corrèze, où je puise calme et réconfort. J'adresse surtout toute mon affection à Jean, qui m'a soutenue et supportée (dans les deux sens du terme) tout au long de cette thèse, et au petit Marius qui est le moteur et le soleil de ma vie.

[1] qu'elle m'excuse si le féminin de "rapporteur" n'est pas très heureux mais mes convictions féministes me poussent à essayer de rappeler que tous les mots français devraient avoir deux genres

Table des matières

Introduction

Résumé :

Le développement des capacités de calcul des ordinateurs a permis l'émergence de nouvelles méthodes de traitement des données dont les réseaux de neurones et les Support Vector Machines font partie. Ainsi, un nombre croissant de travaux scientifiques s'intéressent à ces deux outils.

Dans la Partie I, nous présentons les résultats d'un travail interdisciplinaire dans lequel nous avons utilisé les qualités d'adaptation des perceptrons multi-couches pour la prédiction de cartes géographiques d'occupation du sol. Dans la suite de la thèse (Partie II), nous nous focalisons sur la généralisation de l'utilisation des réseaux de neurones et des SVM au traitement de données fonctionnelles. Le but est de disposer d'outils non linéaires pour l'étude de ce type de données. Une partie de nos travaux (Chapitres 4, 5 et 7) est basée sur une approche semi-paramétrique utilisant une généralisation de la méthode de régression inverse au cadre fonctionnel. Enfin, dans le Chapitre 6, nous explorons une approche différente par la construction de noyaux pour SVM qui prennent en compte la nature spécifique des données.

Dans tous ces travaux, la théorie de l'apprentissage statistique ([Vapnik, 1995]) joue un rôle important et nous nous attachons, autant que possible, à expliciter des résultats de convergence des méthodes décrites.
Mots clés : Analyse des données fonctionnelles, réseau de neurones, perceptron multi-couches, SVM, régression inverse fonctionnelle, discrimination de courbes, régression fonctionnelle, apprentissage statistique, projection sur des espaces de Hilbert.

Abstract:

The increase of computational power has allowed the development of new statistical methods, among which neural networks and Support Vector Machines and a growing number of scientific works are devoted to them.

In Part I, we present the results of an interdisciplinary project in which we use the approximation abilities of multilayer perceptrons in order to predict land cover maps. Subsequently, we focus on the extension of the neural networks and of the SVM for functional data analysis (Part II). Our purpose is to build non linear tools for functional data. A part of our work (Chapters 4, 5 et 7) is based on a semi-parametric approach which uses a functional inverse regression method. Then, in Chapter 6, we present another approach which allows us to build kernels for SVM in order to take into account the functional nature of the data.

In this work, the statistical learning theory (see [Vapnik, 1995]) plays a central role and we apply ourselves to give consistency results for our methods, as much as possible.
Key words: Functional data analysis, neural networks, multilayer perceptron, SVM, functional inverse regression, curves classification, functional regression, statistical learning theory, projection onto Hilbert spaces.

Chapitre 1

Présentation générale

Les dernières années ont connu l'explosion de la masse d'information disponible, par le développement des moyens de communication et de nombreuses machines permettant des relevés divers et complexes (spectromètres en biologie, analyse du génome, satellites en météorologie ou en géographie environnementale, ...). La quantité et la nature des données à exploiter, ont permis l'émergence de nouveaux outils statistiques qui ont également tiré profit des capacités de calcul croissantes des ordinateurs. Les méthodes neuronales ainsi que les machines à vecteurs supports (SVM) sont au nombre de celles-ci, bien que leurs développements respectifs aient connu des voies très différentes : si les premiers sont apparus dans le domaine des neurosciences avant de connaître une grande popularité en statistique où leurs propriétés théoriques ont alors été étudiées, les seconds sont issus de la théorie de l'apprentissage statistique dont ils découlent naturellement par leurs bonnes "capacités de généralisation", capacités à généraliser correctement ce que l'on a appris à partir d'exemples.

Un des exemples de la difficulté à tirer parti de cette nouvelle et volumineuse source d'information est l'analyse des données fonctionnelles : ce domaine a fait l'objet de nombreux travaux dans la communauté statistique. En effet, de manière naturelle, le relevé de nombreuses grandeurs physiques possède une structure sous-jacente particulière qui permet de les représenter de manière naturelle par une ou plusieurs fonctions : c'est le cas des relevés de spectromètres, des courbes de croissance d'individus, des courbes de températures, des enregistrements de voix, ... Ces données sont particulièrement volumineuses et fortement corrélées et les techniques traditionnelles de traitement conduisent à des problèmes mal posés qui ne permettent pas de tirer profit de ce type de données. Des approches fonctionnelles tenant compte de leur structure sous-jacente particulière, ont alors été développées avec succès.

Dans cette thèse, nous nous intéressons à l'utilisation du perceptron multi-couches et des SVM dans le domaine de l'analyse de données fonctionnelles. Peu de travaux ont exploré ce champ qui offre de nombreuses perspectives de recherche. Nous développons des approches fonctionnelles nouvelles pour ces deux outils, qui conduisent à des modèles fonctionnels non linéaires. Nous montrons enfin des résultats théoriques de consistance de l'estimation des paramètres ou bien de l'erreur commise.

Cette introduction s'organise de la manière suivante : dans le paragraphe 1.1, nous posons de manière formelle notre problème et faisons une rapide revue de l'état de l'art en analyse des données fonctionnelles. Dans le paragraphe 1.2, nous présentons le perceptron multi-couches auquel nous nous sommes intéressés en montrant comment il peut être utilisé pour le traitement de données fonctionnelles. Dans le paragraphe 1.3, nous exposons des éléments de la théorie de l'apprentissage qui nous conduisent, dans le paragraphe 1.4, à la présentation

des SVM et à leur utilisation dans le cadre fonctionnel.

1.1 Notations et introduction

Soit (X,Y) un couple de variables aléatoires dans lequel X est à valeurs dans un espace de Hilbert, $(\mathcal{H}, \langle ., . \rangle)$, et Y est, soit un élément de $\{-1; 1\}$ (le problème est alors un problème de discrimination à 2 classes), soit un élément de $\mathbb{R}$ (on parle alors de régression). Le problème de l'apprentissage statistique consiste à prévoir les valeurs de Y connaissant celles de X.

De manière concrète, on suppose connues N observations indépendantes du couple (X, Y), $(x_1, y_1), \ldots, (x_N, y_N)$ à partir desquelles on construit une fonction

$$\hat{\phi}_N : \mathcal{H} \to \left\{ \begin{array}{l} \{-1; 1\} \\ \mathbb{R} \end{array} \right. ,$$

qui doit jouer le rôle de prédicteur. Ce simple problème engendre une série de questions ; nous en isolons deux : connaissant les observations $\{(x_i, y_i)\}_i$,

- *comment construire un « bon » $\hat{\phi}_N$?*
- *comment évaluer l'erreur commise par ce $\hat{\phi}_N$?*

Dans [Vapnik, 1995] et [Vapnik, 1998], V. Vapnik expose des éléments théoriques permettant de mieux cerner ces deux questions ; il met particulièrement l'accent sur deux types d'outils permettant de répondre aux problèmes de régression et de classification : d'un côté les *réseaux neuronaux* dont il explique les limites tout en soulignant leurs bons comportements pratiques et d'un autre côté les *Support Vector Machines* (SVM) dont il expose les bonnes capacités de généralisation. De nombreux auteurs se sont également intéressés à la question de la capacité de généralisation des modèles statistiques en améliorant les résultats de Vapnik ou en soulignant leurs limites. Ainsi, des approches variées permettent d'atteindre la notion de consistance : dans [Devroye *et al.*, 1996], on trouvera un exposé de celles-ci et de nombreux résultats théoriques concernant un grand nombre de modèles.

1.1.1 L'analyse des données fonctionnelles

Dans ce travail, nous nous sommes intéressés à ces deux outils statistiques mais sous un angle particulier. En effet, dans la plupart des problèmes généralement présentées dans le domaine de l'apprentissage statistique, X est une variable aléatoire à valeurs dans l'espace $\mathbb{R}^d$. Ici, nous avons étudié le cas où X vit dans un espace de Hilbert mais de dimension quelconque (éventuellement infinie). L'intérêt de ce type de données est qu'elles apparaissent fréquemment dans les problèmes concrets : puisque les appareils d'enregistrement modernes collectent des données qui se présentent sous la forme de courbes qui peuvent être considérées comme des fonctions discrétisées en certains points. Ces données sont appelées *données fonctionnelles*. De manière plus formelle, on considère généralement que X est à valeurs dans un espace de Hilbert séparable $\mathcal{H}$ qui est, par exemple, l'espace L^2_τ, ensemble des fonctions de l'intervalle compact τ de $\mathbb{R}$ dans $\mathbb{R}$, de carré intégrable, que l'on munit du produit scalaire :

$$\forall\ f, g \in L^2_\tau, \qquad \langle f, g \rangle = \int_\tau f(t)g(t)\, dt.$$

L'analyse statistique des données fonctionnelles conduit à des problèmes nouveaux qui sont liés au fait que la dimension de l'espace $\mathcal{H}$ est infinie. Cela implique en particulier que certains problèmes qui sont résolus simplement lorsque $\mathcal{H} = \mathbb{R}^d$ deviennent mal posés dans un espace de dimension infinie et donnent lieu, d'un point de vue pratique, à des solutions

inappropriées si le caractère fonctionnel des données n'est pas pris en compte dans leur traitement. Une des raisons de cet état de fait est la difficulté d'inverser les opérateurs définis dans des espaces de dimension infinie. Rappelons d'abord que pour tout espace de Hilbert $\mathcal{H}$ (et ceci est donc valable pour son dual $\mathcal{H}'$), on définit l'espérance d'une variable aléatoire à valeurs dans $\mathcal{H}$, Z, comme étant l'unique $\mathbb{E}(Z) \in \mathcal{H}$ tel que $\forall\, v \in \mathcal{H}$, $\langle \mathbb{E}(Z), v\rangle = \mathbb{E}(\langle Z, v\rangle)$ et que le produit tensoriel dans $\mathcal{H}$ est donné par $\forall\, u \in \mathcal{H}$, $u \otimes u : v \in \mathcal{H} \to \langle u, v\rangle u$. Prenons alors l'exemple de l'opérateur de variance de X : $\Gamma_X = \mathbb{E}(X \otimes X) - \mathbb{E}(X) \otimes \mathbb{E}(X)$; cet opérateur est un opérateur de Hilbert-Schmidt : il n'est donc pas bijectif dans L^2_τ. De plus, restreint à son image, l'opérateur Γ_X peut être bijectif mais n'est jamais inversible (dans l'ensemble des opérateurs linéaires continus de L^2_τ) car son inverse n'est pas borné.

La plupart des modèles statistiques classiques ont pourtant été étendus au cas fonctionnel, moyennant cependant quelques adaptations : tout d'abord, [Deville, 1974], [Dauxois and Pousse, 1976], [Besse and Ramsay, 1986] et [Besse, 1991] proposent diverses approches pour l'extension des analyses factorielles au cadre fonctionnel. Parallèlement, [Saporta, 1981] présente une étude synthétique des méthodes exploratoires d'analyse des processus. Plus tard, [Ramsay and Silverman, 1997] donnent une vision globale du traitement des données fonctionnelles avec des méthodes de régression, de discrimination et également des analyses factorielles. [Aguilera *et al.*, 1997] ont également développé un modèle linéaire de prédiction de séries chronologiques basé sur l'ACP d'un processus et qui utilise une approximation spline des facteurs principaux. Par ailleurs, [Cardot *et al.*, 1999] ont généralisé la régression linéaire au cadre fonctionnel alors que [Ferraty and Vieu, 2002] ont proposé une approche non paramétrique du problème de la régression fonctionnelle. Une alternative à ces approches est proposée par [Preda and Saporta, 2002] qui développent un modèle fonctionnel de régression PLS (Partial Least Squares) : il s'agit d'une méthode itérative de régression linéaire qui se révèle particulièrement efficace dans le cadre fonctionnel. Dans [Preda and Saporta, 2005a], les auteurs proposent une variante de ce modèle dans le cas où l'on considère une partition de l'espace des prédicteurs en K groupes distincts. Enfin, dans [Preda and Saporta, 2005b], les auteurs proposent une application de la régression PLS dans le cadre de l'analyse discriminante pour une réponse binaire. [Dauxois *et al.*, 2001], [Ferré and Yao, 2003] et [Ferré and Yao, 2005] développent un modèle semi-paramétrique pour variable aléatoire hilbertienne qui est une version fonctionnelle de la SIR ([Li, 1991]) ; voir, à ce sujet, le paragraphe 1.2.4 ou les chapitres 4 et 5. Ce dernier modèle peut être utilisé aussi bien à des fins de régression que de discrimination (cf [Dauxois *et al.*, 2001] et [Ferré and Villa, 2005a], Chapitre 4). Par ailleurs, [Rossi *et al.*, 2004], [Rossi and Conan-Guez, 2005a] et [Rossi *et al.*, 2005] ont développé des méthodes de traitement de données fonctionnelles par réseaux de neurones.

Dans le domaine de la discrimination aussi les dernières années sont riches en travaux : un certain nombre d'entre eux présentent des approches par *pénalisation* de l'opérateur de covariance. Cette méthode, qui permet une régularisation du problème initial, a été développée, à l'origine par [Ivanov, 1962], [Tihonov, 1963a] et [Tihonov, 1963b]. Elle a été appliquée avec succès à l'Analyse en Composantes Principales ([Pezzulli and Silverman, 1993] et [Silverman, 1996]), à l'Analyse Discriminante ([Hastie *et al.*, 1994] et [Hastie *et al.*, 1995]) et à l'Analyse Canonique ([Leurgans *et al.*, 1993]). Une approche alternative aux méthodes de régularisation est le *filtrage* qui consiste à projeter les données sur une base de fonctions préalablement fixées : c'est ce que font, par exemple, [Biau *et al.*, 2005] qui développent une méthode consistante de discrimination, dans des espaces de Hilbert, par plus proches voisins. Enfin, [James and Hastie, 2001] montrent, dans le cadre de l'Analyse Discriminante, la limite de ces deux approches, particulièrement lorsque les données sont échantillonnées de manière non uniforme : ils proposent alors un modèle dans lequel la fonction sous-jacente, et non les observations, est exprimée par filtrage sur une base Spline.

1.2 Réseaux de neurones

Dans cette section, nous nous concentrons sur une méthode classique qui a fait ses preuves en classification et régression. Les perceptrons ont été introduits par Rosenblatt à la fin des années 1950 et ont donné lieu à de multiples généralisations regroupées sous le terme générique de *réseaux de neurones*. Leur richesse en terme d'approximation de fonctions ainsi que leur simplicité en ont fait des méthodes très populaires dans le domaine du traitement de données statistiques. Les travaux de [Ripley, 1994] et [Bishop, 1995] proposent une présentation générale et complète de ces outils.

1.2.1 Approximation universelle

Nous nous concentrerons ici sur l'étude de *réseaux de neurones à 1 couche* (feed-foward neural network with one hidden layer) ; ceux-ci conduisent à la construction d'une fonction de décision de la forme

$$\phi : x \in \mathbb{R}^d \to g_2 \left[\sum_{i=1}^{q} w_i^{(2)} g_1 \left(\langle x, w_i^{(1)} \rangle + w_i^{(0)} \right) + w_0^{(2)} \right] \tag{1.1}$$

où

- $w = \left(\{w_i^{(0)}\}_{i=1,\dots,q}, \{w_i^{(1)}\}_{i=1,\dots,q}, w_0^{(2)}, \{w_i^{(2)}\}_{i=1,\dots,q} \right)$ sont appelés *poids* du réseau et sont des paramètres à déterminer dans $(\mathbb{R})^q \times (\mathbb{R}^d)^q \times \mathbb{R} \times (\mathbb{R})^q$;
- g_j $(j = 1, 2)$ sont les *fonctions d'activation* du réseau de neurones qui sont, par exemple, $g : x \to \frac{1-e^{-x}}{1+e^{-x}}$ (tangente hyperbolique), $g : x \to \mathbb{1}_{\{x>0\}} - \mathbb{1}_{\{x<0\}}$ (fonction à seuil) ou $g : x \to x$ (fonction linéaire, souvent utilisée pour g_2).

De nombreux travaux ont traité de la richesse d'approximation d'une telle famille de fonctions : leur principal intérêt, démontré par de nombreux auteurs sous des formes différentes (voir [Pinkus, 1999] pour une revue exhaustive des travaux dans ce domaine), est leur propriété d'approximateur universel :

Théorème 1.1. *Soit g_1 une fonction continue de $\mathbb{R}$ dans $\mathbb{R}$, non polynomiale. Alors, l'ensemble des fonctions de la forme*

$$\phi : x \in \mathbb{R}^d \to \sum_{i=1}^{q} w_i^{(2)} g_1 \left(\langle x, w_i^{(1)} \rangle + w_i^{(0)} \right)$$

(où $q \in \mathbb{N}$, $\forall\, i = 1, \dots, q$, $w_i^{(2)} \in \mathbb{R}$, $w_i^{(1)} \in \mathbb{R}^d$ et $w_i^{(0)} \in \mathbb{R}$) est dense dans l'ensemble des fonctions continues de $\mathbb{R}^d$ pour la topologie de convergence uniforme sur des compacts de $\mathbb{R}^d$.

[Hornik, 1991] donne également un résultat de densité des réseaux de neurones sur les compacts de l'ensemble des fonctions continues de $\mathbb{R}^d$ dans $\mathbb{R}$ alors que [Hornik, 1993] généralise ce résultat aux fonctions de $L^p(\mu)$ où μ est une mesure finie de $\mathbb{R}^d$. Par ailleurs, [Stinchcombe, 1999] donne une version de ces résultats pour des espaces vectoriels arbitraires. Enfin, [Rossi *et al.*, 2002] et [Rossi and Conan-Guez, 2005a] proposent une version des résultats de [Stinchcombe, 1999] qui sont directement applicables dans le cas de perceptrons multi-couches lorsque les entrées sont des fonctions de $L^P(\mu)$ (où μ est une mesure borélienne finie de $\mathbb{R}^d$ et $p \in \mathbb{N}^*$).

Les poids du réseau de neurones sont déterminés de manière à minimiser l'erreur empirique commise ; classiquement, on choisit

$$\hat{w}_{opt} = \arg\min \sum_{n=1}^{N} \| y_n - \phi(x_n) \|^2, \tag{1.2}$$

où ϕ est définie comme dans (1.1). [White, 1989] donne un résultat qui montre que l'estimation des paramètres du réseau de neurones est consistante : $\hat{w}_{opt}$ converge presque sûrement vers les paramètres optimaux théoriques du réseau, c'est-à-dire, vers l'ensemble des minimima de :

$$\mathbb{E}\left(\| Y - \phi(X) \|^2 \right).$$

Là encore, [Rossi and Conan-Guez, 2005a] et [Rossi and Conan-Guez, 2005d] généralisent ce résultat aux perceptrons multi-couches à entrées fonctionnelles.

1.2.2 Application à un problème de discrimination issu des sciences humaines

Dans un premier travail élaboré avec une équipe de géographes [1], nous avons cherché à tirer parti de cette grande capacité d'adaptation (voir [Villa *et al.*, 2005], Chapitre 2.4 et [Paegelow *et al.*, 2004b], Chapitre 3).

Dans ce travail interdisciplinaire, notre but est d'analyser les dynamiques de l'évolution de l'occupation du sol afin d'estimer son évolution future. D'un point de vue géographique, cette question revêt une importance majeure pour pouvoir aider les décideurs locaux à organiser le développement des zones de montagnes isolées. L'originalité de ce problème est la nature très diverse des variables explicatives :

- *un processus temporel* : les occupations du sol à trois dates différentes (1980, 1990 et 2000) de pixels de 20 mètres de large (variable qualitative à 8 modalités) ;
- *un processus spatial* : les occupations du sol des voisins de chaque pixel (fréquences de chaque type d'occupation du sol) ;
- *des variables explicatives quantitatives* qui sont des variables environnementales, telle l'altitude, la distance aux plus proches infrastructures humaines, ...

Dans ce problème de discrimination, alliant processus spatio-temporel et variables explicatives quantitatives, les réseaux de neurones sont de bons outils qui permettent d'obtenir des solutions non linéaires ; leurs performances ont été confrontées, de manière positive, avec un modèle linéaire généralisé et un outil de modélisation issu de la recherche en géographie (SIG) qui permet d'introduire des connaissances expertes dans le modèle.

Ce problème se place dans un cadre multi-dimensionnel mais avec un espace de départ de dimension relativement importante (18) pour lequel les variables explicatives sont fortement corrélées. Les problèmes pratiques rencontrées (temps de calcul important, nécessité de relancer plusieurs fois l'apprentissage pour échapper aux minima locaux du problème de minimisation (1.2)...) sont donc de nature proche des problèmes rencontrés lors du traitement de données fonctionnelles par perceptron multi-couches. Nous proposons, dans la prochaine section, un modèle fonctionnel pour le traitement de données par perceptron multi-couches ; cette approche permet de réduire le temps de calcul et les principaux problèmes rencontrés grâce à un pré-traitement pertinent des données.

[1] Equipe GEODE, UMR 5602 CNRS, Université Toulouse Le Mirail

1.2.3 Perceptrons multi-couches à entrées fonctionnelles : approche directe et approche par projection sur une base déterministe

Dans ce chapitre, nous nous concentrons sur la généralisation des perceptrons multi-couches au traitement de données fonctionnelles. Il s'agit donc ici d'étudier l'ensemble des fonctions de décisions de la forme

$$\phi : x \in \mathcal{H} \rightarrow \sum_{i=1}^{q} w_i^{(2)} g\left(\langle x, w_i^{(1)} \rangle + w_i^{(0)}\right) \tag{1.3}$$

où $\forall\, i = 1, \ldots, q$, $w_i^{(2)} \in \mathbb{R}$, $w_i^{(1)} \in \mathcal{H}$ et $w_i^{(0)} \in \mathbb{R}$. La différence avec le modèle (1.1) réside dans le fait que les entrées du perceptron et donc également les poids $(w_i^{(1)})_i$ sont des fonctions ou, plus généralement, des éléments d'un espace de Hilbert de dimension infinie, $\mathcal{H}$.

L'utilisation des réseaux de neurones pour données fonctionnelles est confrontée à plusieurs problèmes :

- d'un point de vue théorique, la minimisation (1.2) est un problème non linéaire, difficile à résoudre, a priori, dans un espace de dimension infinie ;
- les fonctions $x_1, \ldots, x_N$ ne sont pas complètement connues mais seulement discrétisées aux points $t_1, \ldots, t_K$ qui, au pire, peuvent différer d'une fonction à l'autre ; comment, dès lors, approcher les opérations usuelles dans $\mathcal{H}$ et quelle représentation choisir pour les poids fonctionnels du réseau ?

Dans [Rossi and Conan-Guez, 2005a], [Rossi and Conan-Guez, 2005d] et dans [Rossi *et al.*, 2005], les auteurs explorent deux voies différentes :

- Une approche *directe* qui consiste à utiliser directement la discrétisation en entrée du réseau de neurones ; les poids du réseau sont alors estimés par une technique quelconque d'approximation de fonctions et les produits scalaires dans $\mathcal{H}$ sont approchés par le produit scalaire multi-dimensionnel. Cette approche est particulièrement coûteuse en temps de calcul puisque l'approximation de chaque produit scalaire présent dans (1.3) conduit à effectuer un nombre d'opérations égal au nombre de points de discrétisation de la fonction. Dans les problèmes concrets, celui-ci est souvent élevé et l'apprentissage du perceptron est alors particulièrement long.
- Une approche *par projection* sur une base fonctionnelle classique (B-Spline par exemple) : les fonctions d'entrées sont alors approchées par leur projection sur une base finie (à d fonctions) et les poids du réseau sont exprimés sur cette base $(\Psi_j)_{j=1,\ldots,d}$:

$$x^d = \sum_{j=1}^{d} x_j \Psi_j(x) \qquad \text{et} \qquad w = \sum_{j=1}^{d} w_j \Psi_j(x).$$

  Ceci conduit à estimer le produit scalaire dans $\mathcal{H}$ par $\langle x, w \rangle \simeq \sum_{i,j=1}^{d} x_i w_j \langle \Psi_i, \Psi_j \rangle$, ce qui revient à travailler dans $\mathbb{R}^d$ muni du produit scalaire induit par les $\{\Psi_j\}_j$. Le temps de calcul est donc ainsi considérablement réduit par rapport à la première approche. Cependant, la principale difficulté reste le choix d'une base de projection adaptée. [Rossi and Conan-Guez, 2005c] et [Rossi *et al.*, 2005] suggèrent de procéder par comparaison de modèles non linéaires mais là encore, le temps de calcul est alors très important.

Dans notre approche, nous suggérons une méthode automatique de détermination de la base de projection qui permet, via un modèle de régression inverse fonctionnelle, de trouver rapidement un espace de projection « exhaustif » de faible dimension.

1.2.4 Perceptrons à entrées fonctionnelles : une approche par régression inverse

La régression inverse multi-dimensionnelle

Le modèle de régression inverse fonctionnelle est une version généralisée aux espaces Hilbertiens de la SIR de [Li, 1991] ; celle-ci propose un modèle dans lequel Y dépend de X uniquement au travers de sa projection sur un sous-espace de faible dimension appelé espace EDR (Effective Dimension Reduction). Nous présentons donc d'abord, dans le cadre multi-dimensionnel, ce modèle.

Soit donc Y et X des variables aléatoires à valeurs, respectivement, dans $\mathbb{R}$ et $\mathbb{R}^d$. Pour faire face au fléau de la dimension dans la régression non paramétrique de Y sur X, [Li, 1991] introduit le modèle suivant :

$$Y = f(a_1' X, a_2' X, \dots, a_q' X, \epsilon)$$

où ϵ est une variable aléatoire centrée, indépendante de X, f est une fonction inconnue et $(a_j)_{j=1,\dots,q}$ sont des vecteurs linéairement indépendants. L'espace engendré par les $(a_j)_{j=1,\dots,q}$ est appelé espace EDR (Effective Dimension Reduction). Le modèle traite donc de l'estimation de cet espace EDR par le biais des vecteurs propres de la matrice $Var(X)^{-1}Var(\mathbb{E}(X|Y))$. Pour une description complète de cette méthode, nous renvoyons à [Li, 1991] ou à [Aragon and Saracco, 1997] qui proposent diverses approches pour l'estimation de $Var(\mathbb{E}(X|Y))$. Les avantages de cette approche sont multiples : l'espace EDR est obtenu de manière très simple par une analyse spectrale classique et permet de projeter la variable explicative sur un espace qui tient compte à la fois des observations de cette variable mais aussi de la cible. D'autre part, l'originalité de la méthode est l'introduction, dans le modèle, d'une composante non linéaire, f qui peut, une fois déterminé l'espace EDR, facilement être estimée par diverses méthodes non paramétriques. Son extension au cadre fonctionnel est donc justifiée par la possibilité d'obtenir une alternative en partie non linéaire et non paramétrique aux méthodes fonctionnelles déjà existantes (régression linéaire, PLS, etc) et par la présence, dans le modèle, d'une méthode de réduction des données qui permet de traiter efficacement les données fonctionnelles dont la dimension intrinsèque est infinie.

La régression inverse fonctionnelle

[Ferré and Yao, 2003] proposent une version fonctionnelle de la SIR qui s'écrit :

$$Y = f(\langle X, a_1 \rangle, \dots, \langle X, a_q \rangle, \epsilon) \tag{1.4}$$

où $a_1, \dots, a_q$ sont des vecteurs inconnus, linéairement indépendants, ϵ est une variable aléatoire centrée, indépendante de X et f est une fonction inconnue. Dans ce modèle, l'estimation de l'espace engendré par les $\{a_i\}_i$, l'espace EDR, est centrale et la fonction f est ensuite estimée facilement par diverses techniques d'estimation non paramétriques. La clé de la méthode vient du résultat suivant :

Théorème 1.2. *Soit* $A = (\langle X, a_1 \rangle, \dots, \langle X, a_q \rangle)^T$. *Si,*

(H1) $\quad \forall\, u \in L^2_\tau$, *il existe* $v \in \mathbb{R}^q$ *tel que* $\mathbb{E}(\langle u, X \rangle / A) = v^T A$,

alors $\mathbb{E}(X/Y)$ *est à valeurs dans le sous-espace vectoriel engendré par* $\Gamma_X a_1, \dots, \Gamma_X a_q$.

Ce résultat a pour conséquence que l'espace EDR contient le sous-espace vectoriel engendré par les vecteurs propres Γ_X-orthonormés associés aux valeurs propres non

nulles de l'opérateur $\Gamma_X^{-1}\Gamma_{\mathbb{E}(X/Y)}$. Ainsi, l'estimation de l'espace EDR repose sur la décomposition spectrale de l'opérateur $\Gamma_X^{-1}\Gamma_{\mathbb{E}(X/Y)}$, ce qui est l'analogue du résultat connu dans le cadre multi-dimensionnel. Malheureusement, l'opérateur Γ_X^{-1} n'est pas défini (Γ_X, opérateur défini positif, n'est pas inversible comme opérateur de $\mathcal{L}_\tau^2$ dans $\mathcal{L}_\tau^2$). Si nous notons $(\delta_i)_{i\geq 1}$ et $(u_i)_{i\geq 1}$ respectivement ses valeurs propres et vecteurs propres (issus de la décomposition spectrale des opérateurs compacts), R_Γ son espace image et $R_\Gamma^{-1} = \left\{h \in \mathcal{L}_\tau^2 \ : \ \exists\, , f \in R_\Gamma,\ h = \sum_i \frac{1}{\delta_i}(u_i \otimes u_i)(f)\right\}$ Γ_X est une application bijective de R_Γ dans R_Γ^{-1} dont l'inverse, que nous noterons donc Γ_X^{-1} est défini par $\Gamma_X^{-1} = \sum_i \frac{1}{\delta_i} u_i \otimes u_i$. [Ferré and Yao, 2005] démontrent alors le résultat suivant : si $X = \sum_i \zeta_i u_i$ est la décomposition de Karhunen Loève de la variable aléatoire X et si $\sum_{i,j} \frac{1}{\delta_i \delta_j}\mathbb{E}(\mathbb{E}(\zeta_i/Y)\mathbb{E}(\zeta_j/Y)) < +\infty$ alors une base de l'espace EDR est donnée par les vecteurs propres de $\Gamma_X^{-1}\Gamma_{\mathbb{E}(X/Y)}$ qui sont alors des éléments de $\mathcal{L}_\tau^2$. On notera désormais $(a_j)_{j=1,\ldots,q}$ les vecteurs propres Γ_X-orthonormés de l'opérateur $\Gamma_X^{-1}\Gamma_{\mathbb{E}(X|Y)}$.

Cependant, pour les raisons invoquées dans le paragraphe 1.1.1, l'estimateur empirique $\Gamma_X^N = \frac{1}{N}\sum_{n=1}^N x_n \otimes x_n - \overline{X}^N \otimes \overline{X}^N$ (avec $\overline{X}^N = \frac{1}{N}\sum_{n=1}^N x_n$) est mal conditionné et ne permet pas une estimation convergente des $\{a_j\}_j$. Plusieurs solutions sont proposées pour contourner cette difficulté : [Ferré and Yao, 2003] proposent une approche consistant à projeter les données sur une base de vecteurs propres de Γ_X alors que [Ferré and Yao, 2005] procèdent en utilisant un inverse généralisé de l'opérateur de rang fini $\Gamma_{\mathbb{E}(X/Y)}^N$. Ces deux méthodes donnent lieu à des estimations convergentes de l'espace EDR.

Dans [Ferré and Villa, 2005a] et [Ferré and Villa, 2005b], nous proposons une approche basée sur une régularisation : la matrice Γ_X^N est pénalisée par une forme quadratique $[.,.]$ qui, dans nos applications, est définie pour des fonctions deux fois différentiables par : $[u,v] = \int_\tau D^2u(t)D^2v(t)\,dt$. Cette approche est similaire à celle de [Leurgans *et al.*, 1993] pour l'analyse canonique. Nous démontrons un résultat de consistance de la méthode (voir [Ferré and Villa, 2005b], Chapitre 5 et Annexe A.1) :

Notons $\mathcal{S}$ le sous-espace de L_τ^2 des fonctions de τ dans $\mathbb{R}$, deux fois différentiables, dont la différentielle d'ordre 2 est un élément de L_τ^2 et

$$\forall\, f,g \in L_\tau^2, \qquad Q_\alpha(f,g) = \langle \Gamma_X f, g\rangle + \alpha[f,g]$$

où $[f,g]$ est la forme quadratique $\int_\tau D^2f(t)D^2g(t)dt$.
Supposons alors que

(H2) $\mathbb{E}(\| X \|^4) < +\infty$;

(H3) pour tout $\alpha > 0$,

$$\inf_{\|a\|=1,\ a\in\mathcal{S}} Q_\alpha(a,a) = \rho_\alpha > 0\ ;$$

(H4) $\Gamma_{\mathbb{E}(X/Y)}^N$ est un opérateur continu qui converge en probabilité vers $\Gamma_{\mathbb{E}(X/Y)}$ à la vitesse de $\sqrt{N}$;

(H5) $\lim_{N\to+\infty}\alpha = 0$, $\lim_{N\to+\infty}\sqrt{N}\alpha = +\infty$;

(H6) $\{a_j\}_{j=1,\ldots,q}$ appartiennent à $\mathcal{S}$ et vérifient, pour tout u tel que $\langle\Gamma_X u, a_1\rangle = 0$ et que $\langle\Gamma_X u, u\rangle = 1$,

$$\langle\Gamma_{\mathbb{E}(X/Y)}u,u\rangle \ \leq\ \langle\Gamma_{\mathbb{E}(X/Y)}a_2,a_2\rangle = \lambda_2 < \lambda_1\ ;$$

Théorème 1.3. *Sous les hypothèses **(H1)-(H6)**, la fonction* $\gamma^N : a \to \frac{\langle \Gamma_{\mathbb{E}(X/Y)} a, a\rangle}{\langle \Gamma_X^N a, a\rangle + \alpha[a,a]}$ *atteint son maximum sur* $\mathcal{S}$ *avec une probabilité qui tend vers 1 lorsque* N *tend vers* $+\infty$.

Soit alors a_1^N *un vecteur de* $\mathcal{S}$ *pour lequel* γ^N *est maximale et tel que* $\langle \Gamma_X a_1^N, a_1\rangle = 1$, *on a*

$$\langle \Gamma_X (a_1^N - a_1), a_1^N - a_1\rangle \to_{\mathbb{P}} 0.$$

On voit que cette approche conduit à maximiser un critère de Rayleigh pénalisé,

$$\frac{\langle \Gamma_{\mathbb{E}(X/Y)} a, a\rangle}{\langle \Gamma_X^N a, a\rangle + \alpha[a,a]}.$$

D'autres choix de $\Gamma^N_{\mathbb{E}(X/Y)}$ permettent également de traiter le cas où la variable réponse est fonctionnelle : ainsi, [Dauxois *et al.*, 2001] ont montré que la régression inverse fonctionnelle se généralise au cas où la variable réponse et la variable explicative sont des éléments d'un espace de Hilbert quelconque et [Setodji and Cook, 2004] appliquent ce modèle à un problème dans lequel seule la variable cible est une fonction en utilisant une estimation de l'opérateur $\Gamma^N_{\mathbb{E}(X/Y)}$ par la méthode « k-means » .

Elle peut s'appliquer aussi bien aux problèmes de régression qu'aux problèmes de classification suivant le choix de l'estimateur $\Gamma^N_{\mathbb{E}(X/Y)}$.

Régression inverse et classification

De manière plus précise, si $\mathcal{C}_1, \ldots, \mathcal{C}_H$ sont H groupes et que l'on cherche à prédire, connaissant une variable aléatoire fonctionnelle X, le groupe d'appartenance de l'individu observé, le modèle (1.4) peut encore être utilisé. Posons, en effet, $Y = (\mathbb{1}_{\{\mathcal{C}_1\}}, \ldots, \mathbb{1}_{\{\mathcal{C}_H\}})$ où $\mathbb{1}_{\{\mathcal{C}_h\}}$ désigne la fonction indicatrice d'appartenance au groupe $\mathcal{C}_h$; si on cherche à estimer le vecteurs des probabilités $P = E(Y/X)$, on obtient le modèle

$$P = f(\langle a_1, X\rangle, \ldots, \langle a_q, X\rangle).$$

Estimons alors $\Gamma_{\mathbb{E}(X/Y)}$ par

$$\Gamma^N_{\mathbb{E}(X/Y)} = \frac{1}{N}\sum_{h=1}^{H} N_h \widehat{\mathbb{E}(X/Y=h)} \otimes \widehat{\mathbb{E}(X/Y=h)} - \overline{X}\otimes\overline{X}$$

avec $N_h = \sum_{n=1}^N \mathbb{1}_{\{\mathcal{C}_h\}}$ et $\widehat{\mathbb{E}(X/Y=h)} = \frac{1}{N_h}\sum_{n=1}^N x_n \mathbb{1}_{\{\mathcal{C}_h\}}$; l'estimation de l'espace EDR donne alors les mêmes résultats qu'une analyse discriminante (pénalisée dans le cas fonctionnel). En conséquence, puisque l'opérateur $\Gamma^N_{\mathbb{E}(X/Y)}$ est de rang au plus $H-1$, la dimension, q, de l'espace EDR est, à l'instar de la dimension d'une AFD, inférieure ou égale à $H-1$.

La différence avec une AFD classique réside ici aussi dans le fait que le modèle (1.4) admet une part non linéaire qui apparaît au travers de la fonction f. L'estimation de celle-ci conduit à une règle de classification naturelle puisque

$$f(x) = \mathbb{E}(Y/X=x) = (\mathbb{P}(\mathcal{C}_1/X=x), \ldots, \mathbb{P}(\mathcal{C}_H/X=x)).$$

L'estimation de f coïncide donc avec l'estimation des probabilités d'appartenance aux groupes connaissant X ce qui induit la règle de classification suivante :

$$\hat{h}(x) = \arg\max_{h=1,\ldots,H} \left\{ \widehat{\mathbb{P}(\mathcal{C}_h/X=x)} \right\}.$$

Dans le Chapitre 4 ([Ferré and Villa, 2005a]), nous comparons, sur des problèmes de discrimination réels et simulés, diverses approches de régression inverse dont l'approche régularisée qui se révèle particulièrement efficace dès que le nombre de points de discrétisation est suffisamment élevé et fait apparaître plus clairement le caractère fonctionnel des données.

Perceptrons multi-couches fonctionnels

L'originalité de notre approche de perceptrons fonctionnels repose donc sur un prétraitement préalable des données qui consiste à déterminer l'espace EDR et à effectuer une projection des observations de la variable explicatives, $x_1, \ldots, x_N$, sur celui-ci. La projection ainsi obtenue présente plusieurs avantages : l'espace EDR est déterminé automatiquement à partir des données et tient compte de la variable cible Y ; c'est ainsi un espace « optimal » , au sens de la régression. Notre méthode combine donc une procédure de réduction de la dimension particulièrement efficace avec la richesse d'approximation d'un réseau de neurones.

Nous démontrons dans [Ferré and Villa, 2005b] un résultat de consistance de l'estimation des paramètres du réseau de neurones ainsi construit, sous des hypothèses très générales pour le choix des fonctions d'activation et de fonctions d'erreur à minimiser. Le résultat est voisin de celui de [White, 1989] et [Rossi and Conan-Guez, 2005d] : la différence principale provient du fait que les entrées du perceptron ne sont pas iid, du fait de la projection des observations sur une estimation de l'espace EDR, Vect $\left\{a_1^N, \ldots, a_N^N\right\}$, qui est une variable aléatoire dépendant de l'ensemble des données $\{(x_n, y_n)\}_n$. Ceci nécessite donc une modification de la preuve du résultat de consistance qui s'appuie dans ce cas sur le résultat du Théorème 1.3.

Ce résultat ne démontre pas la consistance de l'estimateur obtenu vers la fonction de régression $\mathbb{E}(Y/X)$ mais seulement la consistance de la procédure d'estimation des paramètres. Dans [White, 1990], l'auteur démontre la consistance en probabilité, dans L^2, du perceptron obtenu par minimisation de l'erreur empirique vers la fonction de régression, $\mathbb{E}(Y/X)$ et dans [Barron, 1994], A. Barron donne une vitesse de convergence de l'estimation d'une fonction par minimisation de l'erreur quadratique d'un réseau de neurones en fonction du nombre d'observations, N, du nombre de neurones, de la dimension de l'espace d'entrée et de moments de la série de Fourier de la fonction cible.

Du point de vue de la discrimination à deux classes, la théorie de l'apprentissage, introduite par Vapnik, apporte des réponses à la question de la consistance du classifieur construit par réseaux de neurones.

1.3 Eléments de la théorie de l'apprentissage

Le but de la théorie de l'apprentissage statistique est l'analyse des facteurs de la capacité de généralisation des modèles statistiques afin de sélectionner les méthodes permettant d'accéder à un meilleur contrôle en terme de généralisation (voir [Vapnik, 1998] et [Devroye *et al.*, 1996]).

1.3.1 Le risque

De manière formelle, un modèle statistique est un ensemble de fonctions admissibles $\mathcal{F} = \{\phi\} \subset \{\mathcal{H} \to \mathbb{R}\}$. L'erreur commise par une fonction $\phi \in \mathcal{F}$ est définie grâce à une

fonction de *risque*, $R : \mathbb{R} \times \mathbb{R} \to [0; +\infty[$:

$$\forall\, \phi \in \mathcal{F},\ Err(\phi) \equiv L\phi = \mathbb{E}(R(\phi(X), Y)).$$

Par exemple, dans un problème de discrimination ($Y \in \{-1; 1\}$ et $\phi : \mathcal{H} \to \{-1; 1\}$), la fonction de perte la plus commune est le taux de mauvais classements :

$$R(\phi(x), y) = \left\{ \begin{array}{ll} 0 & \text{si } \phi(x) = y \\ 1 & \text{sinon} \end{array} \right. = \frac{1}{2}|\phi(x) - y| = \mathbb{1}_{\{\phi(x) \neq y\}}.$$

Dans un problème de régression, plusieurs possibilités sont offertes :

$$R(\phi(x), y) = (y - \phi(x))^2$$

(Fonction quadratique de perte), ou

$$R(\phi(x), y)_\epsilon = \left\{ \begin{array}{ll} 0 & \text{si } |y - \phi(x)| \leq \epsilon \\ |y - \phi(x)| - \epsilon & \text{sinon,} \end{array} \right.$$

(Fonction de perte linéaire ϵ-insensible). Cette dernière est à l'origine de la généralisation des SVM pour l'estimation des fonctions réelles.

L'idée principale de la théorie de l'apprentissage est de quantifier la différence entre l'erreur réellement commise en choisissant ϕ comme « décideur » et l'idée que l'on peut se faire de cette erreur au travers de l'échantillon d'apprentissage par le biais du *risque empirique* :

$$\forall\, \phi \in \mathcal{F},\ Err_{emp}(\phi) \equiv \widehat{L}_N\phi = \frac{1}{N}\sum_{n=1}^{N} R(\phi(x_n), y_n).$$

Dans le cas de la discrimination à deux classes, on connaît une borne inférieure de $L\phi = \mathbb{P}(\phi(X) \neq Y)$ sur l'ensemble $\mathcal{G} = \{\mathcal{H} \to \mathbb{R}\}$; c'est l'*erreur de Bayes* : $L^* = L\phi^*$ où

$$\phi^*(x) = \left\{ \begin{array}{ll} 1 & \text{si } \mathbb{E}(Y/X = x) > 1/2 \\ -1 & \text{sinon} \end{array} \right. .$$

Ainsi, une règle de classification $\hat{\phi}_N$, construite à partir de l'échantillon d'apprentissage $(x_1, y_1), \ldots, (x_N, y_N)$ est dite *universellement consistante* si, pour toute distribution du couple (X, Y),

$$L\hat{\phi}_N = \mathbb{P}(\hat{\phi}_N(X) \neq Y) \xrightarrow{N \to +\infty} L^*.$$

Dans [Devroye *et al.*, 1996], on trouvera une vision complète des multiples résultats connus de l'apprentissage statistique dans le cadre de la discrimination à deux classes.

1.3.2 Principal résultat

Les premiers résultats en théorie de l'apprentissage se placent dans le cadre de la discrimination à deux classes. Nous citons ici le Théorème 12.6 de [Devroye *et al.*, 1996] (page 199) mais de nombreux autres résultats similaires ont été démontrés ; celui-ci a été établi dans le cas où $\mathcal{H} = \mathbb{R}^d$ mais nous montrons (dans [Rossi and Villa, 2005b], voir paragraphe 1.4.4 ou Chapitre 6 et Annexe A.2) comment il peut être généralisé dans le cas d'espaces de dimension infinie (voir aussi [Biau *et al.*, 2005]).

Théorème 1.4. *Soit $\mathcal{F}$ un ensemble de classifieurs de la forme $\phi : \mathbb{R}^d \to \{-1; 1\}$. Alors,*

$$\mathbb{P}\left(\sup_{\phi\in\mathcal{F}} |\widehat{L}_N\phi - L\phi| > \epsilon\right) \leq 8\mathcal{S}(\mathcal{F}, N)e^{-N\epsilon^2/32},$$

où $\widehat{L}_N\phi = \frac{1}{N}\sum_{n=1}^{N} \mathbb{1}_{\{\phi(x_n)\neq y_n\}}$ et $L\phi = \mathbb{P}(\phi(X) \neq Y)$.

Ici, la notion centrale de ce résultat est la quantité $\mathcal{S}(\mathcal{F}, N)$ appelée *coefficient de pulvérisation* (shatter coefficient) dont la définition est la suivante :

Définition 1.5. *On appelle N-*coefficient de pulvérisation *de l'ensemble de fonctions $\mathcal{F}$, $\mathcal{S}(\mathcal{F}, N)$, le nombre maximum de sous-ensembles différents de N points $x_1, \ldots, x_N$ qui peuvent être séparés par les classifieurs de la classe $\mathcal{F}$.*

Ainsi, les coefficients de pulvérisation mesurent le pouvoir de classification de la classe $\mathcal{F}$. Dans tous les cas, $\mathcal{S}(\mathcal{F}, N) \leq 2^N$ et, dans le cas où il existe un $N_0 \in \mathbb{N}$ tel que $\mathcal{S}(\mathcal{F}, N_0) < 2^{N_0}$, le nombre maximum $\mathcal{V}$ tel que $\mathcal{S}(\mathcal{F}, \mathcal{V}) = 2^{\mathcal{V}}$ est appelé *VC-dimension* de la classe $\mathcal{F}$; la VC-dimension d'une classe $\mathcal{F}$ est donc le nombre maximum de points qui peuvent être arrangés de telle sorte que $\mathcal{F}$ les sépare de toutes les façons possibles. Ainsi, lorsqu'elle existe, on démontre que la VC-dimension $\mathcal{V}$ vérifie : $\mathcal{S}(\mathcal{F}, N) \leq (N+1)^{\mathcal{V}}$. C'est cette notion de *VC-dimension* (qui apparaît comme une borne supérieure de la capacité de généralisation d'une classe de fonctions) qui est au centre de la théorie de Vapnik.

Le théorème 1.4 est relié à la notion de consistance par le résultat suivant (voir [Devroye *et al.*, 1996]) :

Proposition 1.6.

$$L\hat{\phi}_N - \inf_{\phi\in\mathcal{F}} L\phi \leq 2\sup_{\phi\in\mathcal{F}} |\widehat{L}_N\phi - L\phi|$$

Ainsi, dans le cas de la discrimination à deux classes, la consistance d'un modèle est controlée :

- par sa *capacité de généralisation*, $\sup_{\phi\in\mathcal{F}} |\widehat{L}_N\phi - L\phi|$, qui dépend elle-même de la VC-dimension de $\mathcal{F}$;
- par la *richesse* de $\mathcal{F}$, $\inf_{\phi\in\mathcal{F}} L\phi - L^*$.

Un « bon candidat » classifieur devra donc faire un bon compromis entre une classe de fonctions suffisamment riche et une classe de fonctions ayant une faible VC-dimension.

Moyennant des hypothèses plus importantes sur la fonction de risque et la classe $\mathcal{F}$, des résultats similaires existent dans le cadre de la régression ; nous renvoyons à [Vapnik, 1998] pour plus de détails.

1.3.3 Application aux réseaux de neurones

Si on considère le cas des réseaux de neurones, un classifieur naturel est défini par la forme (1.1) où g_2 est la fonction à seuil :

$$\phi(x) = \text{sign}\left[\sum_{i=1}^{q} w_i^{(2)} g_1\left(\langle x, w_i^{(1)}\rangle + w_i^{(0)}\right) + w_0^{(2)}\right] \tag{1.5}$$

où $\text{sign}(x) = \mathbb{1}_{\{x>0\}} - \mathbb{1}_{\{x<0\}}$ et g_1 est une sigmoïde arbitraire (comme dans le Théorème1.1). Le Théorème 1.1 assure que, si $\mathcal{C}^{(q)}$ désigne l'ensemble des classifieurs définis par un réseaux de neurones à q neurones sur la couche cachée, définis comme dans (1.5), alors

$$\lim_{q\to+\infty} \inf_{\phi\in\mathcal{C}^{(q)}} L\phi - L^* = 0.$$

Quant au problème de la VC-dimension des réseaux de neurones, il a d'abord été résolu pour une fonction d'activation g_1 à seuil ($g_1(x) = \text{sign}(x)$) : [Baum and Haussler, 1989] donnent des bornes inférieures et supérieures pour la VC-dimension de tels classifieurs dont [Farago and Lugosi, 1993] déduisent la consistance universelle de ce type de réseau de neurones lorsque l'on minimise l'erreur empirique de classification. Malheureusement, la plupart des réseaux de neurones pratiquement utilisés ont une fonction d'activation g_1 plus générale et souvent continue. [Macintyre and Sontag, 1993] prouvent finalement que la VC-dimension de réseaux de neurones est finie pour une large famille de sigmoïdes dont la sigmoïde standard.

Malheureusement, ces résultats sont de peu d'intérêts en pratique puisqu'ils ne donnent pas d'algorithme permettant de déterminer le nombre de neurones q à utiliser. [Lugosi and Zeger, 1990] proposent alors un cadre général pour la consistance de l'erreur L^p ($p \geq 1$) des perceptrons en reliant le nombre de neurones nécessaires à l'approximation avec la taille de l'échantillon et la borne admissible supérieure des poids du réseau ; de manière plus précise, ils démontrent la convergence de l'erreur L^p commise par le perceptron optimal vers l'erreur L^p optimale $\mathcal{C}^* = \min_{\phi:\mathcal{H}\to\mathbb{R}} \mathbb{E}\left[(\phi(X) - Y)^p\right]^{1/p}$. Dans [Rossi and Conan-Guez, 2005b], les auteurs généralisent ce résultat pour une approche par projection de perceptron fonctionnel.

Cependant, les algorithmes itératifs habituels de minimisation sont encore confrontés à des problèmes de minima locaux (nombreux) et la qualité de la solution trouvée dépend donc de nombreux paramètres, dont, notamment, l'initialisation des poids w du réseaux. Ce problème conduit souvent à préférer aux perceptrons multi-couches des outils dont la construction est proche (basés sur des combinaisons linéaires de la variable explicative) mais qui ne connaissent pas les mêmes problèmes dans l'optimisation de leurs paramètres : ce sont les Support Vector Machines (SVM).

1.4 SVM

Dans cette partie, nous nous intéresserons aux SVM sous l'angle de la discrimination à deux classes de variables aléatoires fonctionnelles. Nous dirons également un mot sur la généralisation au problème de la discrimination multi-classes, sans entrer dans les détails théoriques. Les SVM ont également été généralisés pour l'estimation de fonctions (problèmes de régression) ; pour cet aspect, qui sera l'objet d'une ouverture future de notre travail, nous renvoyons à [Vapnik, 1995] ou [Vapnik, 1998] qui explique le principe de l'utilisation des SVM pour le traitement de problèmes de régression lorsque les variables aléatoires sont à valeurs dans des espaces de dimension finie.

Le principe des SVM est basé sur une discrimination linéaire avec hyperplan à marge maximale. Introduits par [Boser *et al.*, 1992], ils ont été l'objet de nombreux travaux ; [Cristianini and Shawe-Taylor, 2000] proposent une introduction aux SVM et à ses liens avec la théorie de l'apprentissage statistique. Dans [Shawe-Taylor and Cristianini, 2004], de nombreux noyaux sont proposés pour des données de natures multiples, parfois inhabituelles comme des textes ou des arbres : ils permettent d'effectuer, pour ce type de données, des traitements statistiques divers (discrimination, régression, analyses factorielles non linéaires par noyau, etc). Dans le domaine de la bio-statistique, [Schölkopf *et al.*, 2004] montrent l'efficacité des approches par noyaux (analyse canonique par noyau, analyse en composantes principales par noyau...) pour la résolution de problèmes réels dans lesquels les données peuvent, par exemple, être représentées par des chaînes de caractères, des arbres ou des graphes.

Dans le travail dont nous proposons un résumé en Section 1.4.4, nous introduisons des

noyaux qui ont également été construits pour un usage spécifique : le traitement de données fonctionnelles.

1.4.1 Rappels sur le principe des SVM

Soit $(x_1, y_1), \ldots, (x_N, y_N)$, N réalisations du couple aléatoire (X, Y) à valeurs dans $\mathbb{R}^d \times \{-1; 1\}$. Le principe des SVM, pour la réalisation de ce problème de discrimination multi-dimensionnel, est basé sur la recherche d'une fonction discriminante affine avec une marge maximale. Formellement, ceci se traduit par la résolution du problème suivant : trouver $(w, b) \in \mathbb{R}^d \times \mathbb{R}$ qui minimisent $\| w \|^2$ sous la contrainte :

$$y_n(\langle w, x_n \rangle + b) \geq 1 \tag{1.6}$$

pour tout $1 \leq n \leq N$. Dans le cas où $\min_{n=1,\ldots,N} |\langle w, x_n \rangle + b| = 1$, la marge, c'est-à-dire, la distance du point le plus près à l'hyperplan de séparation, est exactement égale à $\frac{1}{\|w\|}$, ce qui explique pourquoi le problème d'optimisation décrit ci-dessus conduit à la recherche d'un hyperplan de marge maximale. Finalement, le classifieur se résume à

$$\phi(x) = \text{sign}(\langle x, w \rangle + b)$$

où (w, b) sont les solutions du problème d'optimisation ci-dessus.

En pratique, la contrainte (1.6) ne peut être satisfaite exactement et on est souvent amené à considérer le problème d'optimisation à contraintes relachées (marge molle) :

$$(P_C) \quad \left\{ \begin{array}{ll} \min_{w,b,\xi} \langle w, w \rangle + C \sum_{n=1}^N \xi_n & \\ \text{sous les contraintes} & y_n(\langle w, w \rangle + b) \geq 1 - \xi_n, \ 1 \leq n \leq N, \\ & \xi_n \geq 0, \ 1 \leq n \leq N, \end{array} \right.$$

où C est un nombre réel positif. Par l'utilisation du Théorème de Kuhn-Tucker, le problème (P_C) peut être remplacé par le problème dual :

$$(D_C) \quad \left\{ \begin{array}{ll} \max_{\alpha \in \mathbb{R}^N} \sum_{n=1}^N \alpha_n - \sum_{n,m=1}^N \alpha_n \alpha_m y_n y_m \langle x_n, x_m \rangle, & \\ \text{sous les contraintes} & \sum_{n=1}^N \alpha_n y_n = 0, \\ & 0 \leq \alpha_n \leq C, \ 1 \leq n \leq N. \end{array} \right.$$

D'un point de vue pratique, ce problème de minimisation sous contrainte ne connaît pas les mêmes problèmes de minima locaux que celui menant à la détermination des poids d'un perceptron multi-couches. Les solutions peuvent donc être déterminées de manière exacte (problème d'optimisation quadratique) ; on trouve alors,

$$w = \sum_{n=1}^N y_n \alpha_n^* x_n.$$

La formulation (D_C) a un second avantage : les vecteur $(x_n)_n$ n'y apparaissent qu'au travers de leur produit scalaire $\langle x_n, x_m \rangle$; ceci permet, par l'utilisation d'un noyau, de finalement procéder à une discrimination non linéaire. Le théorème suivant (voir, par exemple, [Berlinet and Thomas-Agnan, 2004]) en donne la clé théorique :

Théorème 1.7 ([Aronszajn, 1950]). *Soit* $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ *un* noyau positif *i.e. tel que*

$$\forall\, N \geq 1,\ \forall\, (\alpha_1, \ldots, \alpha_N) \in \mathbb{R}^N,\ \forall\, (x_1, \ldots, x_N) \in (\mathbb{R}^d)^N,$$
$$\sum_{n,m=1}^{N} \alpha_n \alpha_m K(x_n, x_m) \in \mathbb{R}^+.$$

Alors, il existe un espace de Hilbert $\mathcal{F}$*, muni du produit scalaire,* $\langle ., . \rangle_{\mathcal{F}}$ *et une application* $\Phi : \mathbb{R}^d \to \mathcal{F}$ *tels que*

$$\forall\, x, y \in \mathbb{R}^d, \qquad K(x, y) = \langle \Phi(x), \Phi(y) \rangle_{\mathcal{F}}.$$

Ainsi, K agit comme un produit scalaire dans un certain espace, $\mathcal{F}$, appelé espace image. Le produit scalaire usuel de (D_C) peut donc être remplacé par K et la discrimination linéaire est alors effectuée dans l'espace $\mathcal{F}$; l'application Φ n'a pas besoin d'être connue : le problème ne la fait intervenir qu'au travers du produit scalaire et seul le noyau est donc fixé. De même, l'espace image $\mathcal{F}$ n'est pas explicité mais sa structure est celle d'un RKHS (Reproducing Kernel Hilbert Spaces ; voir [Berlinet and Thomas-Agnan, 2004] pour plus de détails).

Les noyaux usuellement utilisés sont

- le noyau linéaire : $K(x, y) = \langle x, y \rangle$ qui revient à choisir pour Φ la fonction identité et effectue donc directement la discrimination linéaire dans l'espace $\mathbb{R}^d$ usuel ;
- le noyau gaussien : $K(x, y) = e^{-\|x-y\|^2/2\sigma^2}$;
- le noyau polynomial : $K(x, y) = (\langle x, y \rangle + 1)^D$;
- ...

1.4.2 Capacité de généralisation des SVM

Historiquement, [Vapnik, 1998] met en valeur la bonne capacité de généralisation des SVM en explicitant la VC-dimension de SVM à marges dans des espaces de dimension quelconque : cette VC-dimension est bornée par le ratio entre le carré du rayon de la plus petite boule contenant toutes les observations et le carré de la marge du classifieur. En pratique, ce résultat est inutilisable car, si les SVM déterminent le classifieur linéaire à marge optimale, il est néanmoins impossible de connaître à l'avance une borne inférieure de cette marge. Pour pallier cet inconvénient, les travaux de [Guo *et al.*, 2002] proposent donc une borne praticable qui utilise la notion de *nombre de ϵ-couverture* (que nous définirons précisément un peu plus loin). Ce résultat donne une borne pratique des nombres de ϵ-couverture qui est directement reliée à la structure spectrale du noyau et particulièrement à la vitesse de convergence de ses valeurs propres vers 0. Ce résultat est exploité, à titre d'exemple, pour les noyaux gaussiens, en dimension 1, pour lesquels la vitesse de convergence des nombres de ϵ-couverture est logarithmique en ϵ et polynomiale en σ. Le résultat sur les nombres de ϵ-couverture permet de retrouver la capacité de généralisation des SVM par les résultats de [Pollard, 1984]. On trouvera également de nombreux détails sur ce thème dans [Williamson *et al.*, 1998].

Enfin, utilisant les résultats sur les nombres de ϵ-couverture, [Steinwart, 2002] démontre la propriété d'universelle consistance des SVM sur les compacts de $\mathbb{R}^d$, pour une suite de paramètres de régularisation (voir (D_C)) dont la forme dépend de la nature du noyau utilisé et particulièrement, là encore, du nombre de ϵ-couverture de l'espace image qu'il induit. De plus, le noyau considéré doit être *universel* : si Φ est la fonction image associée au noyau, l'ensemble des applications de la forme $x \to \langle w, \Phi(x) \rangle$, $w \in \mathcal{F}$, doit être dense dans l'ensemble des fonctions continues sur les compacts de $\mathbb{R}^d$. Le noyau gaussien, par exemple, est universel et induit un SVM consistant pour une suite de régularisation de la forme $C_N = N^{\beta-1}$ avec $0 < \beta < 1/d$ (cf [Steinwart, 2002]). Ce résultat théorique a deux limitations : d'abord, la variable aléatoire X doit prendre ses valeurs dans un compact et ensuite, la suite de régularisation n'est pas arbitraire. Nous généralisons ce résultat aux

espaces de Hilbert de dimension quelconque (voir paragraphe 1.4.4 ou Chapitre 6, Annexe A.2 et [Rossi and Villa, 2005b]) et, en utilisant un partage de l'échantillon d'apprentissage pour effectuer une procédure de validation, nous nous libérons de la contrainte de vitesse de C_N.

1.4.3 SVM à entrées hilbertiennes

Dans [Villa and Rossi, 2005], [Rossi and Villa, 2005a] et [Rossi and Villa, 2005b], nous proposons une approche permettant d'envisager le traitement des problèmes de discrimination à entrées fonctionnelles.

Dans [Rossi and Villa, 2005b] (voir Chapitre 6), nous mettons en avant les caractéristiques théoriques de SVM fonctionnels : de manière plus formelle, nous supposons que la variable aléatoire explicative X n'est plus à valeurs dans $\mathbb{R}^d$ mais dans un espace de Hilbert séparable de dimension quelconque, $(\mathcal{H}, \langle ., . \rangle)$. Classiquement, $\mathcal{H}$ peut être n'importe quel sous-espace de $L^2(\mu)$ où μ est une mesure de Borel finie sur $\mathbb{R}$.

Dans un espace de dimension infinie, il est toujours possible de trouver une solution au problème d'optimisation à marges dures dès lors que les observations $(x_n)_n$ ne sont pas linéairement dépendantes. Cependant, en pratique, un tel classifieur, comme c'est habituellement le cas en dimension infinie, ne fournira pas une solution satisfaisante au problème de discrimination : [Hastie *et al.*, 2004] montrent que les classifications linéaires par SVM en grande dimension sont également des problèmes mal posés et soulignent l'importance du paramètre de régularisation C dans ce cas. On préfèrera donc résoudre systématiquement le problème d'optimisation (P_C) en choisissant de manière appropriée le paramètre C : [Hastie *et al.*, 2004] décrivent une procédure permettant de choisir C sur n'importe quel intervalle de la forme $[0; \mathcal{C}]$ avec un temps de calcul très compétitif.

Enfin, [Lin, 2001] montre que les résultats concernant la résolution du problème d'optimisation (P_C) s'appliquent à un espace vectoriel de dimension quelconque (y compris infinie) ; dans ce contexte, il est donc aussi possible d'utiliser la formulation duale (D_C) et de remplacer le produit scalaire usuel de $\mathcal{H}$ par un noyau positif. Cette formulation présente alors deux avantages : il est plus facile de résoudre un problème d'optimisation dans $\mathbb{R}^N$ que dans $\mathcal{H}$, quitte à approcher les produits scalaires et normes de $\mathcal{H}$ par des méthodes habituelles (comme [Rossi and Conan-Guez, 2005a] le font pour les réseaux de neurones) ; d'autre part, l'utilisation de noyaux permet, outre le fait d'obtenir des solutions non linéaires, de travailler dans des RKHS : à chaque type de noyau correspond des types de régularisation différents. [Girosi, 1997] montre les liens existant entre régularisation et support vector machine.

1.4.4 Implémentation pratique

Dans [Villa and Rossi, 2005], [Rossi and Villa, 2005a] et [Rossi and Villa, 2005b], nous montrons l'intérêt de la construction de noyaux spécialement construits pour le traitement de données fonctionnelles. En effet, comme nous l'avons déjà souligné pour les réseaux de neurones, les fonctions $x_1, \ldots, x_N$ ne sont connues qu'au travers de points de discrétisation, éventuellement échantillonnés de manière non uniforme. Au travers de diverses applications sur données réelles, nous montrons que le noyau linéaire qui peut toujours, dans des espaces de grande dimension, parvenir à la construction d'un classifieur linéaire exact, n'obtient pas de bonnes performances ; des noyaux tenant compte de la nature fonctionnelle des données permettent alors d'améliorer considérablement les résultats. Nous proposons plusieurs approches :

Une approche consistante

Dans [Biau *et al.*, 2005], les auteurs proposent une démarche générale permettant la construction de classifieurs consistants pour données fonctionnelles. Si leur article se concentre sur la méthode des k plus proches voisins, la démarche est suffisamment générale pour être adaptée à tout modèle consistant dans $\mathbb{R}^d$. Dans [Rossi and Villa, 2005b] (voir Chapitre 6), nous adaptons cette méthodologie aux SVM à entrées fonctionnelles. Plus précisément, nous recommandons l'application de l'algorithme suivant :

1. choisir une base orthogonale de l'espace de Hilbert séparable $(\Psi_j)_{j\geq 1}$ et projeter les observations sur cette base :
$$\forall\, n = 1, \ldots, N, \qquad x_n = \sum_{j\geq 1} x_{nj}\Phi_j.$$
La base peut, par exemple, être une base trigonométrique ou une base d'ondelettes (voir les applications du Chapitre 6) ;
2. pour tout $d \in \mathbb{N}^*$ et toute valeur des paramètres du SVM (σ pour le noyau gaussien et le paramètre de régularisation C), résoudre le problème d'optimisation à partir d'une partie des données (ensemble d'apprentissage) $(x_n^{(d)})_{n=1,\ldots,l}$, projection des données initiales sur la base tronquée $(\Psi_j)_{j=1,\ldots,d}$: $x_n^{(d)} = (x_{n1}, \ldots, x_{nd})$. Le SVM ainsi déterminé est donc le SVM multi-dimensionnel classique sur les données projetées ;
3. choisir, de manière optimale, les valeurs de d et des paramètres du SVM par une procédure de validation croisée sur un ensemble de validation : $(x_n)_{n=l+1,\ldots,N}$.

Cette procédure est équivalente à la construction d'un noyau fonctionnel qui peut être écrit de la manière suivante :

$$\forall\, x, x' \in \mathcal{H}, \qquad \mathcal{K}_d(x, x') = K(\mathcal{P}_d(x), \mathcal{P}_d(x')) \tag{1.7}$$

où $\mathcal{P}_d$ désigne la projection sur le sous-espace engendré par $(\Psi_j)_{j=1,\ldots,d}$ et K, n'importe quel noyau standard. Le SVM fonctionnel ainsi construit, conjugue un pré-traitement fonctionnel des données avec une procédure de recherche de classifieur linéaire à marge optimale classique.

L'utilisation de ce type de procédure conduit à la détermination de trois types de paramètres :

- les paramètres dus au pré-traitement fonctionnel : il s'agit ici uniquement de la dimension d de la projection ;
- le paramètre du noyau ou même, le noyau lui-même : on peut en effet fixer à l'avance un ensemble fini de noyaux, avec un nombre fini de paramètres pour chacun d'eux, et appliquer la procédure de validation croisée à l'ensemble des noyaux, $\mathcal{J}_d$, qui est un ensemble fini. Le résultat de consistance sera alors assuré dès lors que l'un au moins de ces noyaux induit un SVM consistant dans $\mathbb{R}^d$ (dès lors, par exemple, qu'il y a un noyau gaussien dans $\mathcal{J}_d$) ;
- le paramètre de régularisation du SVM, C ; C peut être déterminé dans une grille de recherche contenant au moins un paramètre C qui permette la consistance du SVM d-dimensionnel (voir [Steinwart, 2002] pour le lien entre C et le noyau choisi). Cependant, les récents travaux de [Hastie *et al.*, 2004] proposent une procédure de recherche optimale de C dans des intervalles de la forme $\mathcal{I}_d = [0; \mathcal{C}_d]$.

Pour chaque valeur fixée des paramètres, $a = (d, C, K) \in \cup_{d\geq 1}\{d\} \times \mathcal{I}_d \times \mathcal{J}_d$, un ensemble d'apprentissage $(x_1, y_1), \ldots, (x_l, y_l)$ est utilisé pour déterminer la règle de classification $\phi_a^l = \text{sign}\left(\sum_{n=1}^l \alpha_n^* y_n \mathcal{K}_d(., x_n) + b^*\right)$ où $(\{\alpha_n^*\}_n, b^*)$ sont les solutions du problème

d'optimisation (D_C) dans lequel le produit scalaire usuel a été remplacé par le noyau fonctionnel décrit par (1.7). Ensuite, un ensemble de validation est choisi pour sélectionner le méta-paramètre a de manière optimale :

$$a^* = \arg\min_{a \in \cup_{d\geq 1}\{d\}\times\mathcal{I}_d\times\mathcal{J}_d} \left\{ \widehat{L}_{N-l}\hat{\phi}_a^l + \frac{\lambda_d}{\sqrt{N-l}} \right\},$$

où la fonction de risque choisie est l'habituel taux de mal classés $\widehat{L}_{N-l}\hat{\phi}_a^l = \frac{1}{N-l}\sum_{n=l+1}^{N} \mathbb{1}_{\{\phi_a^l(x_n)\neq y_n\}}$ et $\frac{\lambda_d}{\sqrt{N-l}}$ est un terme de pénalisation. Nous démontrons que les SVM fonctionnels ainsi construits sont consistants ; pour cela, définissons, tout d'abord, pour tout $\epsilon > 0$, $\mathcal{N}(\mathcal{F}, \epsilon)$, le nombre de ϵ-couverture d'un espace de Hilbert $\mathcal{F}$, comme le nombre minimum de boules de rayon ϵ nécessaires pour recouvrir $\mathcal{F}$ (cf Chapitre 28 de [Devroye *et al.*, 1996]). Dans le cas des SVM, on considère $\mathcal{F}$ l'espace image induit par un noyau K et on notera donc $\mathcal{N}(K, \epsilon) = \mathcal{N}(\mathcal{F}, \epsilon)$. Par exemple, les noyaux gaussiens induisent des espaces images dont les nombres de ϵ-couverture sont de la forme $\mathcal{O}(\epsilon^{-d})$ où d est la dimension de l'espace de départ (cf [Steinwart, 2002]). On a alors le résultat suivant :

Théorème 1.8. *Soit $(\mathcal{H}, \langle ., . \rangle)$ un espace de Hilbert et X une variable aléatoire à valeurs dans un sous-espace borné de $\mathcal{H}$. On suppose que*

$$\begin{array}{lll} \forall\, d \geq 1, & \mathcal{J}_d \text{ est fini}, & \\ & \exists\, K_d \in \mathcal{J}_d \text{ tel que} & K_d \text{ est universel}, \\ & & \exists\, \nu_d > 0 \text{ tel que } \mathcal{N}(K_d, \epsilon) = \mathcal{O}(\epsilon^{-\nu_d}) \\ & \mathcal{C}_d > 1 & \end{array}$$

et que

$$\sum_{d\geq 1} |\mathcal{J}_d| e^{-2\lambda_d^2} <= \infty$$

et finalement que

$$\lim_{N\to+\infty} l = +\infty, \quad \lim_{N\to+\infty} N - l = +\infty, \quad \lim_{N\to+\infty} \frac{l \log(N-l)}{N-l} = 0,$$

alors, les SVM fonctionnels décrits dans la procédure ci-dessus sont universellement consistants, c'est-à-dire

$$L\phi_{a^*}^l \xrightarrow{N\to+\infty} L^*.$$

La démonstration de ce théorème est donnée en Annexe A.2. Elle est proche de celle de [Biau *et al.*, 2005] ; cependant, cette dernière est basée sur une inégalité « oracle » déduite d'une règle de classification intégrant une recherche de paramètres sur un ensemble fini. L'ensemble $\mathcal{I}_d$ étant éventuellement infini et non dénombrable, une modification de la démonstration est nécessaire ici.

Approche utilisant une base de B-Splines

Une autre manière de choisir un espace de projection (et donc un pré-traitement fonctionnel des données) consiste à utiliser une base de B-Splines ; dans ce contexte, les résultats de consistance exposés dans le paragraphe précédent ne s'appliquent pas mais, néanmoins, leur utilisation pratique est montrée (voir Chapitre 6, Section 6.6.3). Une propriété intéressante des B-Splines est qu'elles permettent d'obtenir facilement des traitements fonctionnels supplémentaires puisque la recherche de la dérivée d'ordre q de n'importe quelle fonction

est alors calculée très facilement. Ainsi, n'importe quel noyau peut ensuite être utilisé sur les dérivées, ce qui permet de se focaliser sur des aspects particuliers des fonctions, telles leur courbure (qui est déterminée à partir de la dérivée d'ordre 2). Dans certains domaines, tel le traitement de données issues de spectromètres, ces informations sont particulièrement pertinentes.

Approche par régression inverse

Enfin, dans le chapitre 7, nous présentons une approche encore à l'étude : l'approche par régression inverse. Celle-ci est illustrée par une application sur données simulées. L'idée de cette approche est similaire à ce que nous avons décrit dans la Section 1.2.4 pour les réseaux de neurones : là encore, il s'agit d'un pré-traitement fonctionnel qui consiste à projeter l'ensemble des données sur une estimation de l'espace EDR obtenue par les méthodes usuelles de régression inverse fonctionnelle. Cette approche consiste donc en la construction d'un noyau fonctionnel :

$$\forall\, x, x' \in \mathcal{H}, \qquad \mathcal{K}_q(x, x') = K(\mathcal{P}_q(x), \mathcal{P}_q(x')),$$

où $\mathcal{P}_q$ désigne la projection des éléments de $\mathcal{H}$ sur l'espace EDR. La différence avec le noyau de (1.7) réside dans le fait que la projection est ici une variable aléatoire qui dépend des observations puisque l'espace de projection est estimé. A contrario, elle permet également d'obtenir un espace de projection optimal en un certain sens et adapté aux données : ici, le choix de la base est effectué automatiquement. Comme dans le cas des réseaux de neurones, une extension de notre travail consiste à obtenir un résultat de consistance de cette approche sous l'hypothèse du modèle de régression inverse fonctionnelle (1.4).

Liste de travaux

Cette thèse regroupe les travaux suivants :

1. en **Partie I** :

   Chapitre 2 « Various approaches for predicting land cover in Mediterranean moutain areas » par Villa N., Paegelow M., Cornez L., Ferraty F., Ferré L. et Sarda P. (2005) *Soumis à publication* ;

   Chapitre 3 « Modélisations prospectives de données géoréférencées par approches croisées SIG et statistiques. Application à l'occupation du sol en milieu montagnard méditerranéen » par Paegelow M., Villa N., Cornez L., Ferraty F., Ferré L. et Sarda P. (2004) *Cybergéo*, **295** (6 décembre 2004), pages 1-19 ;

2. en **Partie II** :

   Chapitre 4 « Discrimination de courbes par régression inverse fonctionnelle » par Ferré L. et Villa N. (2005) *Revue de Statistique Appliquée*, **LIII** (1), pages 39-57 (Chapitre 4) ;

   Chapitre 5 « Multi-Layer Neural Network with functional inputs : an inverse regression approach » par Ferré L. et Villa N. (2005) *Scandinavian Journal of Statistics, accepté pour publication* (Chapitre 5) ;

   Chapitre 6 « Support Vector Machine for Functional Data Classification » par Villa N. et Rossi F. (2005) *Neurocomputing, accepté pour publication* ; cet article est une version longue et complétée de deux publications issues d'actes de congrès :
   - « Support Vector Machine For Functional Data Classification » par Villa N. et Rossi F. (2005) In *ESANN'2005 Proceedings*, Bruges, Belgique, pages 467-472 ;
   - « Classification in Hilbert Spaces with Support Vector Machines » par Rossi F. et Villa N. (2005) In *ASMDA 2005 Proceedings*, Brest, France, pages 635-642 ;

   Chapitre 7 est une brève présentation du travail en cours.

Dans les Annexes,

- l'**Annexe A** développe les preuves des principaux résultats de la thèse : la Section A.1 est dévolue aux démonstrations de deux théorèmes du Chapitre 5 et la Section A.2 se consacre à la preuve d'un théorème présenté dans le Chapitre 6 ;
- l'**Annexe B** présente quelques aspects de programmation utilisées dans les simulations : la Section B.1 présente quelques programmes utilisés pour les simulations des Chapitres 4 et 5 et la Section B.2 ceux du Chapitre 6.

Une **bibliographie** complète des travaux regroupés dans la thèse est présente page 153.

Première partie

Application des réseaux de neurones à un problème issu des sciences humaines

Résumé :

Ce chapitre est consacré à la présentation de l'utilisation de réseaux de neurones dans le cadre d'un problème de discrimination réel issu de la recherche en géographie. Ce projet, mené en partenariat avec le laboratoire de géographes GEODE, UMR 5602 CNRS, Université Toulouse Le Mirail, est un véritable travail interdisciplinaire répondant à l'attente des géographes de voir une partie de leur tâches automatisées par des outils mathématiques performants. Les diverses approches ont été étudiées par

- Martin Paegelow [2] qui a développé les aspects géographiques du problème et proposé une approche "spécialiste" par l'utilisation de techniques de SIG (Système d'Information Géographique, voir [Paegelow, 2004]) ;
- Pascal Sarda, Louis Ferré [3], Frédéric Ferraty [4] et Laurence Cornez [5] qui ont tous contribué aux approches statistiques.

Nous présentons ici deux exposés de ces travaux, l'un (Chapitre 2) sous l'angle de la méthodologie statistique, l'autre (Chapitre 3) sous la vision géographique.

[2] Maître de conférence, habilité à diriger des recherches, Laboratoire GEODE, Université Toulouse II.

[3] Professeurs, Département de Mathématiques et Informatique, Université Toulouse II.

[4] Maître de conférence, habilité à diriger des recherches, Département de Mathématiques et Informatique, Université Toulouse II.

[5] Doctorante à l'ONERA, Toulouse.

Chapitre 2

Various approaches for predicting land cover in Mediterranean mountain areas

Nathalie Villa
GRIMM, Equipe d'Accueil 3686, Université Toulouse Le Mirail, France
Martin Paegelow
GEODE, UMR 5602 CNRS, Université Toulouse Le Mirail, France
Laurence Cornez
ONERA, Toulouse, France
Frédéric Ferraty
GRIMM, Equipe d'Accueil 3686, Université Toulouse Le Mirail, France
Louis Ferré
GRIMM, Equipe d'Accueil 3686, Université Toulouse Le Mirail, France
Pascal Sarda
GRIMM, Equipe d'Accueil 3686, Université Toulouse Le Mirail, France

Référence : Various approaches for predicting land cover in Mediterranean moutain areas (2005), *Soumis*.

Abstract:

Using former maps, geographers intend to study the evolution of the land cover in order to have a prospective approach on the future landscape; these simulations are usually done through the GIS (Geographic Information System). Here we propose to confront this classical geographical approach with statistical approaches, a generalized linear model (polychotomous regression modelling) and a non linear one (neural network). These various methodologies have been tested on a real area whose land cover is known on various dates; this allows us to compare the different models in order to underline their respective advantages.
Key words: *polychotomous regression modelling, neural network, Geographic Information System, classification, prediction, comparison*

2.1 Predicting land cover

From the sketch maps made by geographers or from the analysis of satellite images or aerial photographs, we can build land cover maps for a given country which can be rather precise: the studied area is then cut into several squared pixels whose sides are about 20 meters long and whose land cover is known on various dates. The type of land cover can be chosen from a pre-determined list: coniferous forests, deciduous forests, scrubs, ...

Here, we are not interested in making such maps (for satellite data analysis, see [Cardot *et al.*, 1993]). Our purpose is to analyse the land cover dynamics in order to estimate its future evolution; on a geographical point of view, prospective simulations have a great interest to help the local administrations to develop these mountain areas. The idea is then to compare different approaches in order to confront their ability to be generalized to various mountain areas.

For a given pixel, determined by its spatial coordinates, latitude (i) and longitude (j), the value of the land cover on date t, $c_{i,j}(t)$, is a categorical random variable depending on several variables:

- the land cover of this pixel on previous dates: $c_{i,j}(t-1), \ldots, c_{i,j}(t-T)$ (*time serie of length* T);
- the land covers of the neighbouring pixels on previous dates: $V_{i,j}(t-1), \ldots, V_{i,j}(t-T)$, where $V_{i,j}(t-\tau)$ is a set of values of land cover on date $t-\tau$ for the pixels in a neighbourhood of the pixel (i,j) (*vectorial time serie*);
- some environmental variables: for example, the elevation, the aspect, the proximity of roads and villages, ...: $Y^1_{i,j}, \ldots, Y^p_{i,j}$.

Finally, we face a problem of classification in which the predictors are both qualitative and quantitative and are also highly dependant (spatial time process). To solve this question, we propose and compare three approaches. Two of them are statistical approaches: the first one is a generalized linear model in which we estimate the parameters of the model by maximizing a log-likelihood type criterion. The second one uses a supervised multilayer neural network. The last one comes from the GIS (Geographic Information System) approach and allows the use of the knowledge of an expert (a geographer guides the model by integrating their knowledge, *i.e.* by introducing some constraints and suitability factors). By confronting these various approaches, we expect to give ideas in order to improve the GIS approach by underlining its advantages and its limits.

A comparison of these three approaches has been made on a little area of the "Pyrénées Orientales" (south west of France) where land cover maps have been made. We confront the various scenarios constructed with the real maps.

In the following, we describe the data more precisely and present the three approaches. Then we apply these methodologies on this data set and finally, we compare the results obtained by analyzing the advantages and the limits of the three models.

2.2 Description of the data set

The area under study is named Garrotxes; it is a basin of 5 villages and 8 570 hectares located in the "Massif des Pyrénées" (south west of France). As in many moutain areas, the demography reached its maximum at the begining of the 19^{th} century but, since then, a big drift from the land has led to the desertion of the land under cultivation and the recovery

of the fields by scrubs and forests. Human action on the land is then very low and the evolution of the landscape is fast enough on a geographical point of view.

This area was divided into about 241 000 pixels. For each pixel, we know:

- a categorical variable which is the land cover at 3 different dates: 1980, 1990 and 2000. This variable was taken from a list of 8 choices ("Coniferous forests", "Deciduous forests", "Scrubs", "Broom lands", "Grass pastures", "Grassland", "Agriculture" and "Urban") and has been used to make 3 maps of the studied area (*cf.* Figure 2.1);

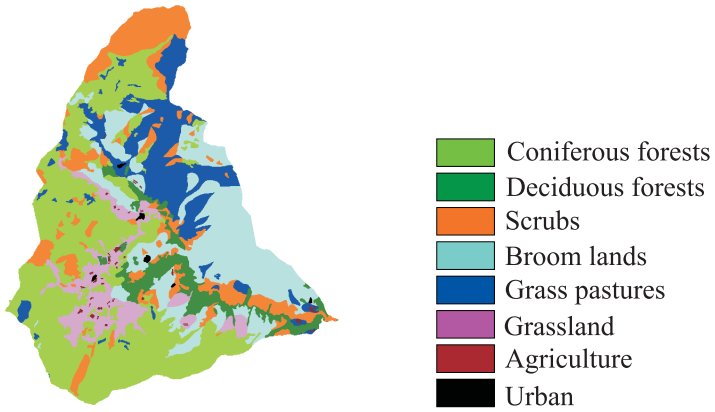

Figure 2.1: Land cover for the Garrotxes area on date 1980

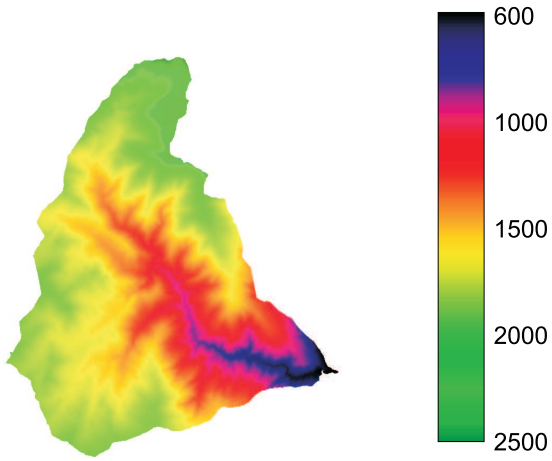

Figure 2.2: Elevation (meters) for the Garrotxes area

- several environmental variables; some of them are of numeric type (the elevation, the slope, the aspect, the distance of roads and villages) and others are of categorical type (Type of forest management, administrative or not ? Type of area, pastoral or not ? - a pastoral area is another type of administration subdivision). None of these variables has changed during the studied period. (*cf.* Figure 2.2).

2.3 Presentation of the three approaches

Three approaches have been developed to estimate the evolution of the land cover: first, we present the two statistical approaches; we develop the polychotomous regression modelling and then the neural network model. Finally, we focus on a classical approach for the geographers: the GIS.

We will see that these three methods have common points: if the statistical models give, by their constructions, a smooth solution to any given problem (here, to estimate a spatial time process), the GIS first takes into account the temporal aspect of the problem and, at the end, makes a smoothing of the results by using the spatial aspect of the data set. The idea is then similar in both cases: we have time series smoothed by the use of the geographical aspect of the problem.

2.3.1 Statistical models

We confront two statistical approaches for this particular problem: the first one, polychotomous regression modelling is a generalized linear approach based on the maximum log-likelihood method. The second one, neural network is a popular method which has recently proved its great efficiency to solve various types of problems. These two approaches take into account the particular spatial aspect of the problem. The models include the spatial dependance of the land cover whereas the GIS approach proceed in several steps, one of them consisting in a king of smoothing of the map (see GIS description for details).

Let us now describe the statistical setting more formally. We note $X_{i,j}(t)$ the vector of variables that explain the value of the land cover for a given pixel (i,j) on date t. We suppose that the time dependance is of order 1; then, $X_{i,j}(t)$ contains:

- *for the time series:* the value of the land cover for the pixel (i,j) at the previous time $t-1$;
- *for the spatial aspect:* the frequency of each type of land cover in the neighbourhood of pixel (i,j) on the previous date. Then we have to choose the shape and the size of the neighbourhood. For the shape, we have many choices: the simpler one is a square neighbourhood or a star-shaped neighbourhood around the pixel (i,j); the most sophisticated can use the slope to better take into account the morphological influences of the land. For the size of the neighbourhood, we have to find at which distance a pixel can influence the land use of pixel (i,j). Moreover, for the neural network approach, in order to respect the spatial aspect of the problem, we weight the influence of a pixel by a decreasing function of its distance to the pixel (i,j) (*cf.* Figure 2.3).
- environmental variables (slope, elevation, ...).

Let us repeat that $c_{i,j}(t)$ is the land cover for a given pixel on date t. We note $\mathcal{C}_1, \ldots, \mathcal{C}_K$ the different types of land cover. Then, for every $k = 1, \ldots, K$, we try to estimate the probability $P(c_{i,j}(t) = \mathcal{C}_k | X_{i,j}(t))$ that the pixel (i,j) has a land cover equal to $\mathcal{C}_k$ given the vector $X_{i,j}(t)$; thus, the model is of the following form :

$$P(c_{i,j}(t) = \mathcal{C}_k | X_{i,j}(t)) = f_k(X_{i,j}(t)). \tag{2.1}$$

Once this probability is estimated, the main idea consists in predicting the type of land cover, $c_{i,j}(t)$, by the quantity:

$$\arg \max_{k=1,\ldots,K} P(c_{i,j}(t) = \mathcal{C}_k | X_{i,j}(t)).$$

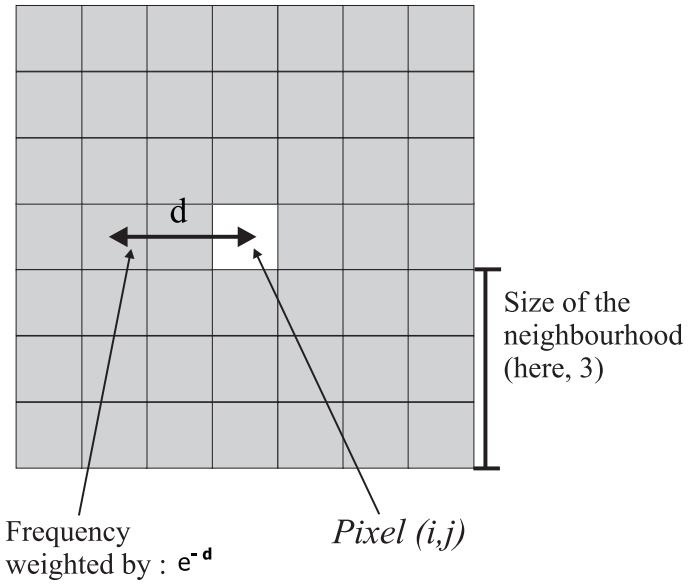

Figure 2.3: An example of neighbourhood

In both approaches, we estimate f_k thanks to a training sample which is used to estimate parameters for each model. To that end, we have collected the values of the predictors and of the land cover for many pixels on various dates (see next section for more details); the observations are denoted by $(X^{(1)}, c^{(1)}), \ldots, (X^{(N)}, c^{(N)})$.

Finally, the difference between the two approaches is in the way they try to estimate the functions f_k. We will now clarify it.

Polychotomous Regression Modelling

When we wish to predict a categorical response given a random vector, a useful model is the *multiple logistic regression* (or *polychotomous regression*) model ([Hosmer and S., 1989]). A smooth version of this kind of method can be found in [Kooperberg *et al.*, 1997]. These statistical techniques have been applied to several situations. Their good behaviour both on theoretical and practical grounds have been emphasized. In our case, where the predictors are both categorical and scalar, we may think of generalizing this method. We describ the derived model below.

Let us note, for $k = 1, \ldots, K$

$$\theta\left(\mathcal{C}_k | X_{i,j}(t)\right) = \log \frac{P\left(c_{i,j}(t) = \mathcal{C}_k | X_{i,j}(t)\right)}{P\left(c_{i,j}(t) = \mathcal{C}_K | X_{i,j}(t)\right)}.$$

Then, we get the following expression

$$P\left(c_{i,j}(t) = \mathcal{C}_k | X_{i,j}(t)\right) = \frac{\exp \theta\left(\mathcal{C}_k | X_{i,j}(t)\right)}{\sum_{k'=1}^{K} \exp \theta\left(\mathcal{C}_{k'} | X_{i,j}(t)\right)}. \tag{2.2}$$

Now, to estimate these conditional probabilities, we use the parametric approach to the

polychotomous regression problem, that is the linear model

$$\theta\left(\mathcal{C}_k|X_{i,j}(t)\right) = \alpha_k + \sum_{c\in V_{i,j}(t-1)} \sum_{l=1}^{K} \beta_{kl} \mathbb{1}_{[c=\mathcal{C}_l]} + \sum_{r=1}^{p} \gamma_{kr} Y_{i,j}^r, \tag{2.3}$$

where we recall that $V_{i,j}(t-1)$ are the values of the land cover in the neighbourhood of the pixel (i,j) on the previous date $t-1$ and $(Y_{i,j}^r)_r$ are the values of the environment variables. Let us call $\delta = (\alpha_1, \ldots, \alpha_{K-1}, \beta_{1,1}, \ldots, \beta_{1,K}, \beta_{2,1}, \ldots, \beta_{2,K}, \ldots, \beta_{K-1,1}, \ldots, \beta_{K-1,K}, \gamma_{1,1}, \ldots, \gamma_{1,K}, \ldots, \gamma_{K-1,1}, \ldots, \gamma_{K-1,p})$ the parameters of the model to be estimated. We have to notice that since $\theta\left(\mathcal{C}_K|X_{i,j}(t)\right) = 0$, we have $\alpha_K = 0$, $\beta_{Kl} = 0$ for all $l = 1, \ldots, K$, and $\gamma_{Kr} = 0$ for all $r = 1, \ldots, p$. We now have to estimate the vector of parameters δ. For that end, we use a penalized likelihood estimator which is performed on the training sample. Let us write the penalized log-likelihood function for model (2.3). It is given by

$$l_\varepsilon(\delta) \;=\; l(\delta) - \varepsilon \sum_{n=1}^{N} \sum_{k=1}^{K} u_{nk}^2, \tag{2.4}$$

where the log-likelihood function is

$$l(\delta) = \log\left(\prod_{n=1}^{N} P_\delta\left(c^{(n)}|X^{(n)}\right)\right). \tag{2.5}$$

In this expression, $P_\delta(c^{(n)}|X^{(n)})$ is the value of the probability given by (2.2) and (2.3) for the observations $(X^{(n)}, c^{(n)})$ and the value δ of the parameter.

In expression (2.5), ε is a penalization parameter and, for $k = 1, \ldots, K$, $u_{nk} = \theta_\delta(\mathcal{C}_k|X^{(n)}) - \frac{1}{K}\sum_{k'=1}^{K} \theta_\delta(\mathcal{C}_{k'}|X^{(n)})$. Our penalized likelihood estimator $\widehat{\delta}_\varepsilon$ satisfies:

$$\widehat{\delta}_\varepsilon \;=\; \arg\max_{\delta\in\ \mathbb{R}^M} l_\varepsilon(\delta),$$

where $M = K^2 + (K-1) * p - 1$ denotes the number of parameters to be estimated.

Traditionally, statisticians maximize the log-likelihood function to compute the estimators. But, as pointed out by [Kooperberg *et al.*, 1997], the introduction of a small penalty term allows numerical stability and guarantees the existence of a finite maximum. Moreover, for a reasonably small value of ε, the penalty term would not affect the value of the estimators that we could obtain without the penalty term. Numerical maximization of the penalized log-likelihood function is achieved by a Newton-Raphson algorithm.

Neural network

Neural networks have a great adaptability to any statistical problems and especially to overcome the difficulties of non linear problems even if the predictors are highly correlated; thus it is not surprising to find them used in the chronological series prediction ([Bishop, 1995], [Lai and Wong, 2001] and [Parlitz and Merkwirth, 2000]). The main interest of neural networks is their ability to approximate any function with the desired precision (universal approximation): see, for instance, [Hornik, 1991].

Here we propose to estimate, in model (2.1), the function f_k in the form of a feedforward neural network with one hidden layer; we call "feedforward neural network with one hidden

layer" (see Figure 2.4) the ψ function from $\mathbb{R}^q$ to $\mathbb{R}$ that can be written: for all x in $\mathbb{R}^q$

$$\psi_w(x) = \sum_{i=1}^{q_2} w_i^{(2)} g\left(< x, w_i^{(1)} > + w_{i,0}^{(1)} \right)$$

where q_2 in $\mathbb{N}$ is the number of neurons on the hidden layer, $(w_i^{(1)})_{i=1,\dots,q_2}$ (respectively $(w_i^{(2)})_{i=1,\dots,q_2}$, $(w_{i,0}^{(1)})_{i=1,\dots,q_2}$) are in $\mathbb{R}^q$ (resp. $\mathbb{R}$) and are called weights of the first layer (resp. weights of the second layer, bias) and where g, the activation function, is a sigmoïd; for example, $g(x) = \frac{1}{1+e^{-x}}$.

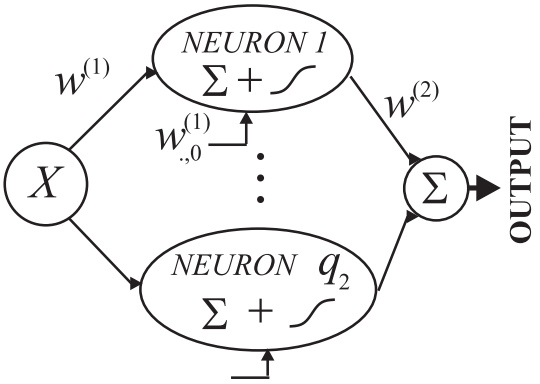

Figure 2.4: Feedforward neural network with one hidden layer

Then, the output of the neural network is a smooth function (here it is indefinitly continuous and derivable) of its input. As we have already said, this property ensures that the neural network takes into account the spatial aspect of the data set, since two neighbouring pixels have "close" values for their predictor variables.

To determine the optimal value for weights $w = ((w_i^{(1)})_i, (w_i^{(2)})_i, (w_{i,0}^{(1)})_i)$, we usually minimize the quadratic error on the training sample: for all $k = 1, \dots, K$, we choose

$$w_{opt}^k = \arg\min_{w \in \mathbb{R}^{q_2(q+2)}} \sum_{n=1}^{N} \left[c_k^{(n)} - \psi_w^k(X^{(n)}) \right]^2, \tag{2.6}$$

where $c^{(n)}$ and the categorical data in $X^{(n)}$ are written on a disjonctive form. This can be performed by classical numerical methods of the first or the second order (such as gradient descent or conjugate gradients, ...). [White, 1989] gives many results that ensure the convergence of the empirical parameters to the optimal parameters.

2.3.2 Geographic Information System

To establish and calibrate a GIS model for prospective land cover changes, we use available GIS software components and a restrictive list of criteria so that the methodology is easy to apply to other areas. The chosen approach may be considered as a "supervised" model (manual establishment of the knowledge base) in comparison to the two other "automatic" approaches.

The model for land cover calls on a chain of different tools: Markovian chain analysis (MCA, time transition probabilities), multi-criteria evaluation (MCE, relationship between

the land cover changes and relevant criteria to perform the land cover suitability which assists the spatial implementation of the predicted transitions), multi-objective evaluation (MOE, resolution between the predicted land cover and the concurrent land cover) and a cellular automata (CA, introducing the principle of spatial autocorrelation).

- *Suitability of land cover dynamics in time and space*
  Knowledge on recent past dynamics is considered as essential to apprehend the future evolution. Knowledge means statistically improved measures of the land cover behavior in space and time related to criteria which are able to explain a part of their variability; it is performed by classic geographical analyses. The criteria are split up into Boolean constraints and factors which express a suitability variable in space. The constraints will simply mask a part of the area while factors may be weighted and allow to take a tradeoff into account.

  The constraints may be the same for all land cover captions or specific (*e.g.* elevation limits are specific constraints for forests).

  The rules of behavior in time and space for each land cover type are determined by an analysis of the chronological set of land cover maps and by evaluation of geographical roughness. We note significant differences between real land cover location and a theoretical distribution (isotropic and homogeny space). This geographic roughness, or friction, is made by a set of known and mapped variables (elevation, slope, orientation, cost-distance accessibility, proximity to land cover categories, particular status like domain forest, pastoral management, ...) on a scale compatible with the resolution of the land cover maps.

  Some criteria, like proximity to existing land cover, are analyzed in terms of the probability that a type of land cover may occur (for instance forest spreading on the edges). The analysis of the recent dynamics gives an idea of the model (linear, sigmoid, etc.) to employ and about the parameters of suitability.

  Once the factors are calculated and standardized, they are weighted by pairs using the Saaty matrix ([Saaty, 1977]) to calculate the eigenvector of each factor.

  Finally, the chosen multi-criteria evaluation ([Eastman *et al.*, 1993]) includes a lot of order weights (ordered weighted averaging) allowing the choice of the level of risk and tradeoff. The number of order weights is equal to the number of factors; the weights sum is equal to 1. Order weights are calculated for each pixel. For each pixel, the first order weight is assigned to the factor which has the lowest suitability in the weighted factor set; the last order weight is assigned to the highest suitability among the weighted factors.

  Allowing entire weight to the first order weight means a risk adverse decision without a factor tradeoff, while the opposite means a high risk strategy (picking only the factor with the highest suitability and not considering the others), also without tradeoff ([Yager, 1988]).

- *Computing time transition probabilities*
  To perform the land cover extrapolation, we use Markov chain analysis (MCA), a discrete process with discrete time for which values on date t depend on values on dates $t-1$ and $t-2$ (order 2 Markov). The prediction is given as an estimation of the transition probabilities. The results are expressed as a transition matrix recording the probability that a given land cover type will change to each other type; we also

get the number of pixels expected to change. The algorithm generates conditional probability maps for each land cover type showing the probability for a pixel to be covered by a given land cover type. The set of the Markovian probability maps may be integrated in a single map by stochastic choice ([Flamm and Turner, 1994]). To do so, the algorithm evaluates the conditional probabilities of each land cover for each pixel against a random probability distribution; the land cover category exceeding random threshold is assigned to the evaluated pixel ([Eastman, 2001]). The single stochastic integration generally gives a rather poor idea of predicted land cover because it neither includes suitability knowledge nor spatial contiguity.

- *Spatial allocation of predicted land cover probabilities using suitability maps*
  The land cover probabilities (MCA) are transformed into spatial distributions using the knowledge on the likely spatial distribution (MCE). To integrate the set of the predicted land cover maps in a single one, we use a multi-objective evaluation (MOE). MOE is like MCE but uses various and concurrent objectives. The prioritization of objectives ([Rosenthal, 1985]) takes into account both probability scores and suitability. This combination is assisted by a cellular automata adding spatial contiguity.

The entire modelling steps are summed up in Figure 2.5.

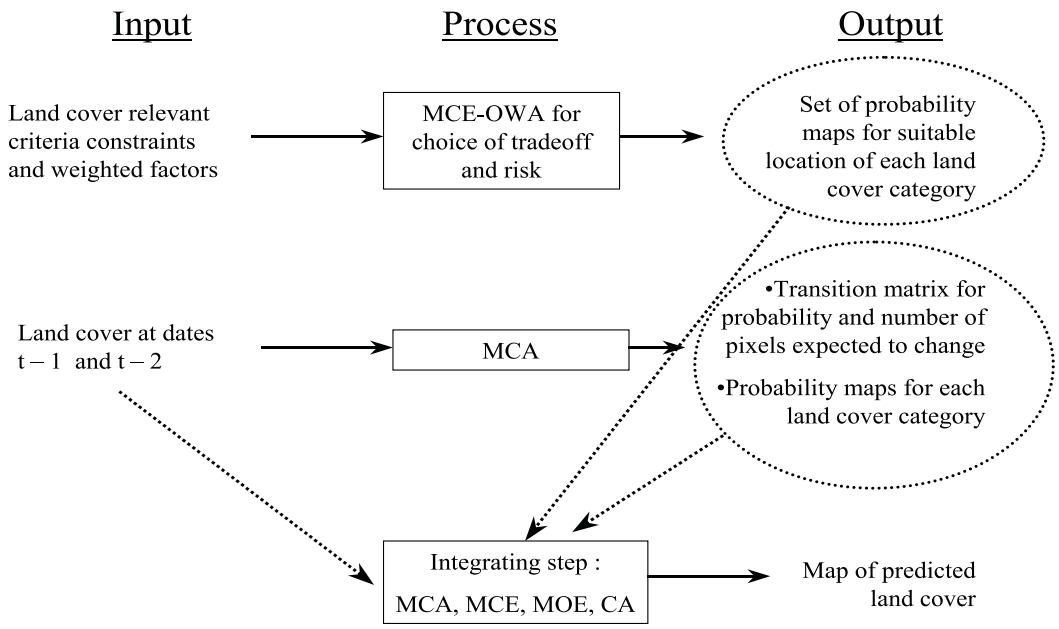

Figure 2.5: Methodological overview for the GIS approach

2.4 Practical application on the Garrotxes data set

In order to compare the three models, we apply the same methodology on the Garrotxes data set. We proceed in two stages: at first, we "train" the three models, that is we calibrate the parameters for each model. To that end, we used the maps on dates 1980 and 1990 with targets 1990 and 2000 for statistical approaches (time series of order 1) and the maps on dates 1980 and 1990 with target 2000 for GIS approach (time series of order 2). Secondly, using the estimated parameters, we build the predicted map on dates $t = 2000$ and confront it with the real map; this allows us to compare the various approaches. In the following sections, we give the pratical applications for the three models.

2.4.1 Statistical approaches

As usual in statistical methods, there are two stages: the *estimation step* and the *validation step*.

- The *estimation step*, based only on the maps on dates 1980 and 1990, consists in estimating the parameters of the models (either for the polychotomous regression or the neural network). For this step, we put the urban pixels aside because they are constant (in these mountain area, nothing was constructed or destroyed during the last decades);
- The *validation step* allows us to choose, for both methodologies, the best neighbourhood, for polychotomous regression, the penalization parameter and, for neural network, the number of neurons on the hidden layer. Concerning the neighbourhood, we only consider square shapes so choosing a neighbourhood leads us to determine its size.

Polychotomous regression

- The *estimation step* produces the estimated parameter vector $\widehat{\delta}_\varepsilon$ of the parameters δ_ε of model (2.3) for fixed neighbourhood and penalization parameter ε. This step is repeated for various values concerning both neighbourhood and penalization parameter.
- *Validation step:* Once given an estimated parameter vector $\widehat{\delta}_\varepsilon = (\widehat{\alpha}_1, \ldots, \widehat{\alpha}_{K-1}, \widehat{\beta}_{1,1}, \ldots, \widehat{\beta}_{1,K}, \widehat{\beta}_{2,1}, \ldots, \widehat{\beta}_{2,K}, \ldots, \widehat{\beta}_{K-1,1}, \ldots, \widehat{\beta}_{K-1,K}, \widehat{\gamma}_{1,1}, \ldots, \widehat{\gamma}_{1,p}, \ldots, \widehat{\gamma}_{K-1,1}, \ldots, \widehat{\gamma}_{K-1,p})$, it is easy to estimate, for $k = 1, \ldots, K$, the quantities

$$\widehat{P}\left(c_{i,j}(t) = \mathcal{C}_k | X_{i,j}(t)\right) = \frac{\exp \widehat{\theta}\left(\mathcal{C}_k | X_{i,j}(t)\right)}{\sum_{k'=1}^{K} \exp \widehat{\theta}\left(\mathcal{C}_{k'} | X_{i,j}(t)\right)},$$

where

$$\widehat{\theta}\left(\mathcal{C}_k | X_{i,j}(t)\right) = \widehat{\alpha}_k + \sum_{c \in V_{i,j}(t)} \sum_{l=1}^{K} \widehat{\beta}_{kl} \mathbb{1}_{[c=\mathcal{C}_l]} + \sum_{r=1}^{p} \widehat{\gamma}_{kr} Y_{i,j}^r.$$

At each pixel (i, j) for the predicted map on date t, we affect the most probable vegetation type namely the $\mathcal{C}_k$ which maximizes

$$\left\{ \widehat{P}\left(c_{i,j}(t) = \mathcal{C}_k | X_{i,j}(t)\right) \right\}_{k=1,\ldots,K}.$$

Note that here we consider $t - 1 = 1990$, so this procedure achieves a prediction of the landcover for the map on date $t = 2000$. Knowing the 2000 map, we choose the

neighbourhood and the penalization parameter in order to minimize the misclassification rate between the predicted 2000 map and the observed one. Finally, the optimal neighbourhood is given by a square of size 7 (number of pixels for one side) and the optimal penalization parameter is given by $\varepsilon = 10$.

Neural network

We use a neural network with 19 neurons as input, one hidden layer with q_2 neurons (where q_2 is a parameter to be calibrated) and 7 neurons as outputs. The inputs of the neural network are:

- For the *time series*, 7 neurons for the disjonctive form of the value of the pixel (urban is left out of the model, as said before);
- For the *spatial aspect*, 8 neurons for the weighted frequency of each type of land cover in the neighbourhood of the pixel;
- 4 neurons for the environmental variables.

The output is the estimation of the probabilities (2.1).

The estimation is also made in two stages:

- The *estimation step* produces the estimated weights as described in (2.6) for a fixed number of neurons (q_2) and a fixed neighbourhood. For this step, the neural network has been trained by cross-validation ([Bishop, 1995]) only with a part of the pixels: we saw that large areas are constant, thus we only used the pixels for which one neighbour, at least, has a different land cover. These pixels will be called "frontier pixels"; the others are considered as constant for the decade.
  This step is repeated for various values of both neighbourhood and q_2.
- *Validation step:* once an estimation of the optimal weights is given, we choose q_2 and the size of neighbourhood, using, as for the previous model, the 2000 map to minimize the misclassification rate.
  Finally, q_2 is set at 8 neurons and the optimal size for the neighbourhood is 7 pixels.

Programs have been made using Matlab ([Beale and Demuth, 1998]) and are avalaible on request.

2.4.2 GIS

The software implementation of the GIS methodology uses a decision support modules into Idrisi 32, release 2.

- *Computing suitability*
  Finally, six factors are used in multi-criteria evaluation (MCE, Table 2.1). The nature and number of constraints are specific for each land cover type.

  The approach used may be considered as low risk taking with some tradeoff as shown in Figure 2.6. Results of multi-criteria evaluation with ordered weighted averaging (MCE-OWA) are expressed as a probability map for each land cover type.

Factor	Technique to process suitability
Elevation Slope aspect Distance to roads and villages	Manual recoding based on significant (99 % and 99.9 % level) differences between real and theoretical distributions
Proximity to existing land cover features (distance)	Fuzzy function based on the observed distance for border and spontaneous appearances
Probability for land cover change	Manual recoding based on the cross-tabulation transitions between the training land cover types

Table 2.1: Table of used factors and involved techniques to process suitability

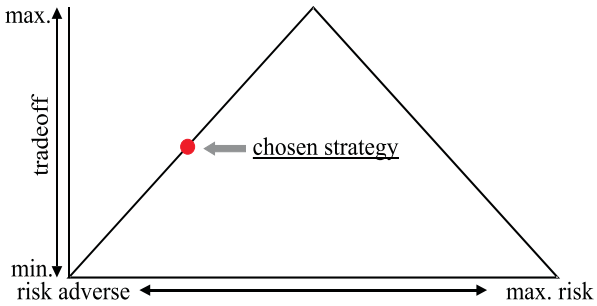

Figure 2.6: Decision strategy space and chosen approach into multi-criteria evaluation (ordered weighted averaging). Used order weights: 0.45, 0.20, 0.15, 0.10, 0.07, 0.03

- *Computing time transition probabilities*
  In order to test this modelling step, it is applied first to simulate land cover on date 2000 (t), based on land cover maps on dates 1980 ($t-2$) and 1990 ($t-1$).

- *Spatial allocation of the predicted land cover probabilities using suitability maps*
  The integrating step, combining the knowledge about the likely spatial distribution (MCE), the time transition probabilities (MCA) as well as the multi-objective land allocation, is performed by the chosen software by CA_Markov, an aggregated module. The applied cellular automata is based on a standard contiguity 5×5 filter. The algorithm is iterative so as to match the time distance between $t-2$ and $t-1$ and between $t-1$ and t.
  The inputs are: the land cover maps on dates $t-2$ (1980) and $t-1$ (1990), land cover probability (suitability) maps resulting from multi-criteria evaluation and transition probabilities (Markovian output). The output is a prospective land cover map at date t (2000 is tested to calibrate the model by comparison with the real map).

2.5 Comparison and discussion

After the three models have been trained, we build the predicting map on date 2000. The performances of the three models are summarized in Table 2.2: the frequency of error for each land cover type is made on the pixels which are really of this land cover type and we focus on the 6 more frequent land cover types, since the number of agriculture pixels tends to zero. In Figure 2.7, we can see the three predictive maps that can be confronted with the real map (Figure 2.8). Several remarks can be done:

Land cover types	Frequency in the area	Poly. Regression error rate	Neural Network error rate	GIS error rate
Coniferous forests	40.9 %	11.9 %	10.6 %	11.4 %
Deciduous forests	11.7 %	51.7 %	45.8 %	55.3 %
Scrubs	15.1 %	57.1 %	54.5 %	51.9 %
Broom lands	21.6 %	14.4 %	16.2 %	17.1 %
Grass pastures	5.7 %	59.2 %	59.4 %	54.4 %
Grasslands	4.8 %	25.6 %	19.3 %	30.4 %
Overall		**27.2 %**	**25.7 %**	**27.2 %**

Table 2.2: Missclassification rates for the various approaches and for each land cover type

- The predictive maps are coherent, smooth and close to reality. This can also be shown through the good error rate (about 25 % - 27 % for the three approaches) which is clearly a good performance considering the poverty of the data (we only have 3 dates to train the models).
  The neural network is the model which gives the best results (best error rate for the whole map and 3 best error rates by land cover type). But the striking fact is that the "automatic" statistical approaches do as well as the guided GIS approach. This is an interesting point in order to help improving the classical geographical approach to predicting land cover, and better understand the environmental changes in time and space.

- We can see in Table 2.2 that the performances are quite different depending on the land cover type: we can observe that frequent land cover types (such as coniferous forests or broomlands) have quite good error rates whereas the rare land cover types are hardly predicted. This can be explained by the fact that the three approaches are based on a training step which is better for frequent land cover types, by definition.
  Another fact can also explain bad performances for some land cover types: some land cover types, such as scrubs or deciduous forests or grass pastures, are particularly unstable or subject to random changes. A forest fire transforms a forest into scrubs; scrubs become higher and are then classified in the forest pixels (for example, this is the case in an area on the south east of the Garroxtes). These changes are impossible to predict due to a lack of knowledge: the number of land cover categories was intentionally limited to minimize interpretation failures because the source of the land cover data is mainly panchromatic aerial photographs. The disadvantage is then a certain variability into categories. To improve the models, we can think of using a semi-quantitative land cover variable, like cover rates.

- The results given by the three models are similar: Table 2.3 shows the number of pixels which are truly predicted by the 3 models, by only 2 models or by only 1 model.

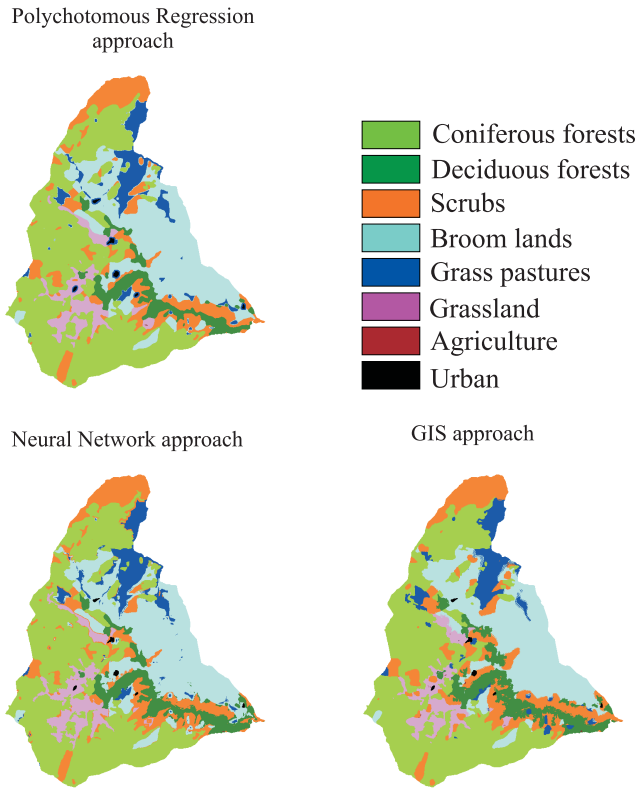

Figure 2.7: Predictive maps for the various approaches on date 2000

For example, 3.8 % of the pixels are correctly predicted by both Neural Network and Polychotomous Regression and incorrectly predicted by GIS. We can see that 66.5

3 models	2 models only			1 model only			no model
	NN + PR	GIS + PR	GIS + NN	GIS	NN	PR	
66.5 %	3.8 %	1.4 %	1.5 %	3.2 %	2.3 %	0.9 %	20.3 %

Table 2.3: Rates of pixels correctly predicted by various combinations of models

% of the pixels are correctly predicted by the three models and an other 20.3 % are incorrectly predicted by the three models. But this table shows another important fact: there is a great similarity between the two statistical models since the pixels which are the most often correctly predicted by two models are the pixels predicted by the neural network and by the polychotomous regression modelling (3.8 %). This fact is confirmed by the pixels correctly predicted by only one model: GIS approach has the greatest rate (3.2 %) which shows that this model is slightly apart from the others. On the contrary, the polychotomous regression modelling has a very low rate (0.9 %) of pixels correctly predicted by only one model: we can then suppose that this approach doesn't give a big improvement compared to the neural network and the GIS.

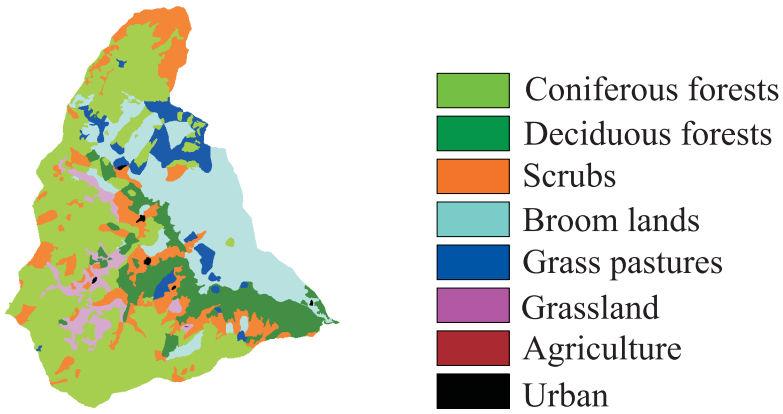

Figure 2.8: Real map on date 2000

If we combine this remark on the differentiation of the GIS approach from the two other models, with the results given in Table 2.2, we can see that GIS approach has better performances than statistical models on the land cover types which are more unstable such as scrubs or grass pastures. This can suggest a way to improve the prediction: a statistical approach such as a neural network or a polychotomous regression modelling could be guided in the same way as a GIS approach, giving good results both on stable and unstable land cover types.

2.6 Conclusion

Finally, this work shows the great potential of the two statistical models in predictive prospection on geographical data. These models have as good performances as GIS approach and we can hope that a combination of the two points of view (statistics and GIS) can improve the land cover predictions. Another aspect that has to be worked on is the form of the data: for example, we underlined that an information on the density of the scrubs is needed to better understand their evolution. Then, this new methodology will also have to be tested on various data sets and various areas in order to test its ability to be generalized and limit its singularity effects. That is the next step of the work as another area, located in the Sierra Nevada, will now be treated with a similar methodology.

Chapitre 3

Modélisations prospectives de données géoréférencées par approches croisées SIG et statistiques

Application à l'occupation du sol en milieu montagnard méditerranéen.
(Prospective modelling of georeferenced data by crossed GIS and statistic approaches applied to land cover in Mediterranean mountain areas.)

Martin Paegelow
GEODE UMR 5602 CNRS, Université Toulouse Le Mirail, France
Nathalie Villa
GRIMM, Equipe d'Accueil 3686, Université Toulouse Le Mirail, France
Laurence Cornez
ONERA, Toulouse, France
Frédéric Ferraty
GRIMM, Equipe d'Accueil 3686, Université Toulouse Le Mirail, France
Louis Ferré
GRIMM, Equipe d'Accueil 3686, Université Toulouse Le Mirail, France
Pascal Sarda
GRIMM, Equipe d'Accueil 3686, Université Toulouse Le Mirail, France

Référence : Modélisations prospectives de données géoréférencées par approches croisées SIG et statistiques. Application à l'occupation du sol en milieu montagnard méditerranéen. (2004), *Cybergéo*, **295 (6 décembre 2004)**, pages 1-19.

Résumé :

Les auteurs mettent en œuvre trois méthodes de modélisation prospective appliquées à des données géoréférencées haute résolution portant sur l'occupation du sol en milieu montagnard méditerranéen : approche SIG, modèle linéaire généralisée et réseaux neuronaux. Une validation des modèles est entreprise par la prédiction de l'occupation du sol à la dernière date connue. Les résultats obtenus sont, dans le contexte de la dynamique spatio-temporelle de systèmes ouverts encourageants et comparables. Les scores de prédiction correcte se situent autour de 73 %. L'analyse des résultats porte notamment sur la localisation géographique, les types d'occupation du sol concernés et les écarts à la réalité des résidus. Un croisement des trois modèles souligne le degré élevé de convergence et une relative similitude des résultats issus des deux approches statistiques comparée au modèle SIG supervisé. Des travaux en cours concernent la mise en œuvre des modèles sur d'autres sites et le repérage des points forts respectifs afin de développer un modèle intégré.
Mots clés : *modélisation, modèle linéaire généralisé, prévision, réseaux de neurones, SIG*

Abstract:

The authors apply three methods of prospective modelling to high resolution georeferenced land cover data in a Mediterranean mountain area: GIS approach, non linear parametric model and neuronal network. Land cover prediction to the latest known date is used to validate the models. In the frame of spatial-temporal dynamics in open systems results are encouraging and comparable. Correct prediction scores are about 73 %. The results analysis focuses on geographic location, land cover categories and parametric distance to reality of the residues. Crossing the three models show the high degree of convergence and a relative similitude of the results obtained by the two statistic approaches compared to the GIS supervised model. Steps under work are the application of the models to other test areas and the identification of respective advantages to develop an integrated model.
Key words: *forecast, GIS, modelling, neuronal network, non linear parametric model*

3.1 Problématique et objectifs

L'objet de notre recherche est la modélisation de dynamiques environnementales dans le cadre de systèmes complexes et ouverts. Dans ce cadre, la variable étudiée - ainsi que les variables d'environnement, susceptibles d'expliquer son évolution dans l'espace et dans le temps - contient une part d'incertitude ou d'aléa ce qui exclut, de fait, une approche déterministe. Ainsi, nous utilisons une approche stochastique (probabiliste) tenant compte de la dépendance dans le temps (effet mémoire) et dans l'espace : les outils probabilistes utilisés sont notamment la distribution multinomiale et l'analyse de Markov. En outre, notre approche fait également appel à la logique floue et à un automate cellulaire. L'ensemble de ces méthodes est mis en œuvre dans trois modèles à but prévisionnel différents afin de comparer leurs performances respectives. Plus précisément, il s'agit de comparer un modèle géomatique combiné de simulation prospective dont la mise en œuvre est possible en utilisant les fonctions de logiciels SIG disponibles sur le marché à deux modèles statistiques dont la mise en œuvre, plus longue, est extérieure au SIG. En contrepartie, l'intérêt des deux approches statistiques réside dans leur caractère automatique tandis que le modèle SIG nécessite une analyse thématique experte.

Notre choix en matière de modélisation statistique a porté sur deux approches classiques, l'une basée sur le maximum de vraisemblance (modèle linéaire généralisé) et l'autre utilisant un réseau de neurones. Ces deux méthodes sont proches du point de vue de la modélisation et différent essentiellement en ce qui concerne les algorithmes de mise en œuvre.

Un des défis actuels de la recherche en géomatique est celui de la modélisation prospective de données géographiques à haute résolution. Des méthodes géostatistiques éprouvées pour l'interpolation et l'extrapolation spatiales existent depuis plusieurs décennies et sont implémentées dans nombre de logiciels géomatiques commercialisés. Par contre, des outils de modélisation temporelle et d'aide à la décision ne furent implémentés dans les SIG que récemment et doivent être considérés plutôt comme des algorithmes expérimentaux intéressants que de techniques opérationnelles.

Depuis les années 1990 la demande sociale en outils d'aide à la décision et de modélisation capables d'assister différentes tâches de gestion environnementale (notamment la prévention de risques) et d'aménagement des territoires s'est fortement accrue.

Cet article illustre les premiers résultats d'une étude comparative de trois méthodes de modélisation prospective appliquée à l'occupation du sol dans des anthroposystèmes montagnards de l'Europe du sud. Nous considérons l'occupation du sol comme un indicateur pertinent, disponible à haute résolution, d'une combinaison d'activités humaines que les sociétés déploient dans l'espace - et auxquelles l'occupation du sol réagit avec une certaine inertie - et de facteurs naturels. Les montagnes méditerranéennes font l'objet d'une profonde restructuration socio-économique qui se manifeste, entre autres, dans de spectaculaires changements paysagers. Cette réorganisation commença dans les Garrotxes (Pyrénées Orientales, zone d'études) à la fin de la première moitié du XIX$^{\text{ème}}$ siècle par le déclin du système agropastoral traditionnel provoquant l'exode rural.

Une base de données géoréférencées matérialise les connaissances des dynamiques passées et actuelles de l'occupation du sol ainsi que des facteurs d'environnement potentiellement explicatifs. Elle alimente trois méthodes de modélisation prospective : l'une, supervisée, met en œuvre des algorithmes implémentés dans des logiciels SIG ; les deux autres approches - modèle linéaire généralisé et réseau neuronal - peuvent être qualifiées de non supervisées dans la mesure où les règles du comportement spatio-temporel sont automatiquement détectées par l'outil. L'objectif principal étant la mise en œuvre et l'optimisation de chacune des trois approches sur le même jeu de données. L'interprétation critique des résultats, notamment des résidus de la prédiction, permet de cerner avantages et inconvénients respectifs. A partir de cette analyse comparative, peuvent être envisagées la possibilité de construction d'un modèle optimisé intégrant les points forts de chacune des méthodes ainsi que les modalités de transposition et les limites de généralisation.

Afin de valider et d'optimiser les modèles ceux-ci sont appliqués, dans un premier temps, à prédire l'occupation du sol à la dernière date connue avant de proposer des scénarii prospectifs. Cette étude est menée dans le cadre d'une coopération entre trois équipes de recherche travaillant sur deux sites [1] : les Garrotxes dont nous présentons ici les premiers résultats et la Alta Alpujarra Granadina (Andalousie, Espagne - travaux en cours).

[1] GEODE UMR 5602 CNRS, Groupe SMASH -EA 3686 GRIMM, UTM et Instituto de Désarrollo Régional - Universidad de Granada

3.2 Zone d'études et base de données

3.2.1 Les Garrotxes

Les Garrotxes, situées dans le département des Pyrénées Orientales, à l'extrémité NO du Conflent forment un ensemble géographique constitué de cinq communes et d'une taille de 8 570 ha. Ce bassin versant présente une rive droite granitique, à modelé géomorphologique relativement lourd, où sont localisées la quasi-totalité des anciennes terrasses de cultures et des forêts de pins à crochet (Pinus uncinata) et de pins sylvestre (Pinus sylvestris) ; un espace à dynamique végétale très rapide. La rive gauche du Cabrils, cours d'eau collecteur ce jetant dans la Têt à Olette, est un large soulane orientée SO sur substrat schisteux avec un métamorphisme de contact dans les zones les plus basses et occupée par des landes majoritairement ligneuses (à base de Genista purgans et, dans une moindre mesure, de Calluna vulgaris), fortement embroussaillée aux altitudes les plus basses par des chênes verts (Quercus ilex). La particularité des Garrotxes est leur enclavement : le bassin versant, à l'écart des grandes routes, est délimité au nord par le massif du Madrès (2469 m), à l'ouest (Cami Ramader) et au sud (Puig de la Tossa, Serrat del Cortal) par des chaînes culminant entre 1600 et 2000 m d'altitude et à l'est par la crête (Lloumet) de la soulane rejoignant le Madrès. La vallée du Cabrils présente une dégradation progressive du climat méditerranéen ; la remontée de l'influence méditerranéenne au cœur des Pyrénées Orientales étant assurée par la vallée de la Têt modifiant ainsi la rudesse du climat montagnard.

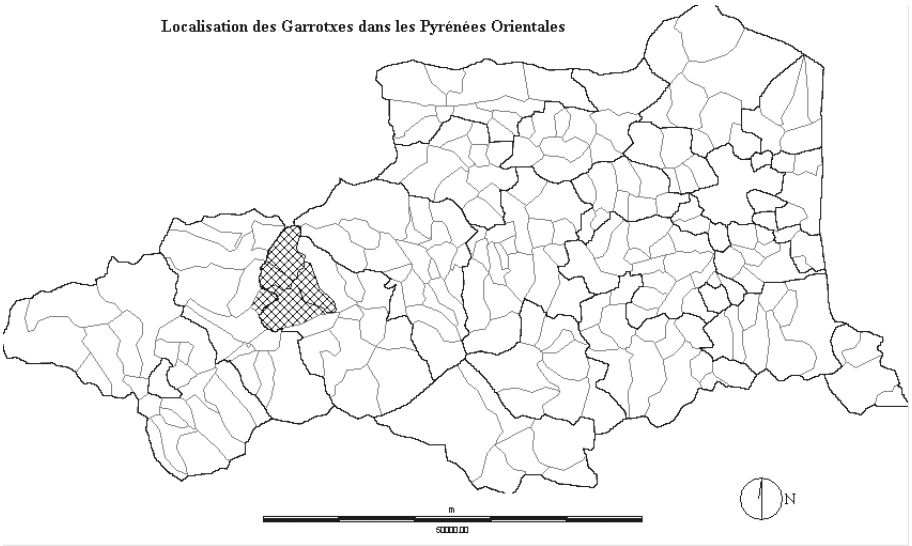

Fig. 3.1 – Localisation des Garrotxes à l'intérieur du département des Pyrénées Orientales

Autrefois un modèle d'organisation agropastorale traditionnelle, de nos jours l'agriculture a quasiment disparu tandis que l'activité pastorale, longtemps en déclin, donne des signes de renouveau suite à une profonde réorganisation entamée durant les années 1980 ([Métailie and Paegelow, 2004], [Paegelow and Camacho Olmedo, 2003]). Le maximum démographique au début du XIX^{ème} siècle se traduisait par une mise en valeur de toutes les ressources montagnardes traditionnelles mobilisables (agriculture, élevage, sylviculture). Ainsi en 1826 (cadastre napoléonien) un quart de la surface totale était cultivé. Le sup-

port quasi exclusif de l'agriculture a été les terrasses de cultures (feixtes). Le déclin démographique (de 1 832 habitants en 1830 à 90 en 1999) et la reconversion des terrasses de culture en pâturages, broussailles et forêts allaient de pair.

Parmi les agents externes considérés responsables du déclin de cette société locale à faible degré d'insertion dans l'économie nationale on peut citer, outre les processus d'industrialisation et de mise en valeur agricole de plaines au cours du XIX$^{\text{ème}}$ siècle, une variabilité interannuelle accrue des précipitations, observée au milieu du XIX$^{\text{ème}}$ siècle ([Tabeaud *et al.*, 2003]), qui eût pu contribuer à la rupture d'un système poussé à bout par la pression anthropique sur le milieu. Deux évènements ponctuels - l'arrivée du chemin de fer à Olette (1911) et la Première Guerre Mondiale - ont accéléré l'exode rural. Ainsi il est probable que l'avenir proche se jouera en termes de gestion - ou de non gestion - pastorale se matérialisant par divers moyens de blocage, voire d'inversion, de l'embroussaillement et du reboisement spontané des espaces pastoraux (berger guidant le bétail, clôtures, écobuage). L'instauration de groupements pastoraux (GP) et d'associations foncières et pastorales (AFP) à partir des années 1980 a effectivement conduit à une reprise de l'activité pastorale avec un remplacement partiel du cheptel ovin par des bovins et des équins. Les signes de reconversion économique sont récents (années 1990) mais d'une portée limitée (ouverture d'un gîte d'étape à Sansa, tentatives de valorisation en tourisme vert) malgré la concrétisation prévue prochainement du Projet de Parc Naturel Régional des Pyrénées Catalanes.

3.2.2 La base de données et l'évolution de l'occupation du sol

La base de données géoréférencées consiste en une série de cartes d'occupation du sol assorties de plans d'information représentant des facteurs environnementaux et sociaux. Les cartes existent - selon les traitements envisagés - soit en mode image (résolution du pixel d'environ 18 m), soit en mode objet. Ainsi les principaux traitements pour la modélisation font appel à une logique image (analyse spatiale) tandis que le mode objet offre une plus grande souplesse pour réaliser des requêtes attributaires.

Les cartes d'occupation du sol ont la même résolution spatiale mais ont des origines et des légendes variables. La première carte est basée sur le cadastre napoléonien (1826) - un support permettant de distinguer entre forêts, espaces pastoraux, prairies, champs agricoles et le bâti (villages). La première mission aérienne disponible (1942) rend possible le renseignement de la catégorie broussailles (landes très embroussaillées contenant de groupes d'arbres ou un nombre important d'arbres isolés) - maillon manquant en 1826 entre les formations arborescentes denses et arbustives (landes ligneuses). La carte de 1962 conserve la même légende tandis que l'échelle et la qualité des missions aériennes plus récentes (1980 et 1989), également panchromatiques, facilitent la distinction entre forêts de conifères et formations boisées de type feuillus. Il en va de même pour une meilleure discrimination des espaces pastoraux : landes ligneuses (notamment à base de Genista purgans) et landes à graminées. La carte d'occupation du sol la plus récente (2000) est basée sur des observations de terrain et est, par conséquent, nettement plus détaillée (20 catégories).

En raison de la nature des sources, la classification de l'occupation du sol, sous forme de trois nomenclatures emboîtées, est surtout d'ordre physionomique.

L'évolution de l'occupation du sol (cf. Figure 3.2) est classique. Les terres labourées délaissées ont été d'abord utilisées comme zones de pâture avant un embroussaillement menant souvent à la reconquête par la forêt. La base de données géoréférencées contient nombre de plans d'information ayant trait à l'occupation du sol :

- Plans issus du MNT : carte altitudinale, carte des pentes, carte d'occupation du sol ;
- Plans d'accessibilité calculés à partir du réseau routier et l'emplacement des villages (habitat groupé) : indice d'accessibilité (analyse de distance-coût) selon la date ;

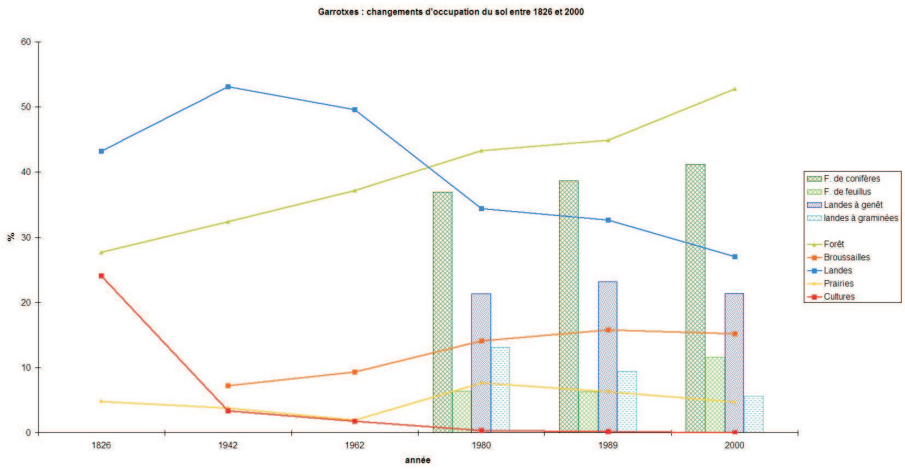

FIG. 3.2 – Changements de l'occupation du sol entre 1826 et 2000 (Garrotxes)

- Plans relatifs à la gestion pastorale : unités pastorales (UP), associations foncières et pastorales (AFP), pression pastorale ;
- Plans tenant compte du statut particulier de certains zones : forêts domaniales et zone militaire ;
- Limites administratives ;
- Réseau hydrographique.

Les données purement attributaires (recensement de la population, recensement général agricole, ...) sont, selon leur degré de confidentialité, connus, mais apportent peu de connaissances compte tenu de leur unité spatiale de rattachement (la commune) incompatible avec une analyse haute résolution.

3.3 Méthodologie et mise en œuvre

Il s'agit de construire trois modèles prédictifs de l'occupation du sol (variable qualitative) à haute résolution, de les calibrer par un test sur la dernière date connue en utilisant la même base de données afin de comparer leurs efficacités respectives. La première approche fait appel aux techniques disponibles dans les SIG et peut être qualifiée de supervisée dans la mesure où l'analyse du géographe conduit à l'implémentation des règles nécessaire au calcul des cartes de probabilité. Bien que les deux autres approches, statistiques, soient également supervisées, le rôle du mathématicien, peu familier avec la thématique, consiste plutôt à optimiser l'algorithme que d'y introduire les conclusions de sa propre analyse du comportement spatio-temporel du milieu considéré.

3.3.1 Approche supervisée par SIG

Le modèle que nous présentons se veut être simple à deux égards : simple dans le lever des données d'entrée (limitation à quelques données facilement disponibles, cf. paragraphe 3.2.2) et simple dans sa mise en œuvre informatique (recours à des algorithmes implémentés dans un logiciel SIG commercialisé). Ce modèle géomatique combiné :

- Fait appel à la logique floue afin d'ajuster les données environnementales dans l'évaluation multicritère.
- Est stochastique pour l'aspect prédictif de la simulation à événements discrets et états finis - chaînes de Markov avec mémoire (deux dates initiales).
- Remédie aux limites de l'analyse markovienne en recourant à une évaluation multicritère optimisant l'affectation spatiale des probabilités markoviennes (prise en compte de la rugosité de l'espace) par constitution d'une base de connaissances et de règles d'inférence (variables d'environnement) relatives à la phase d'apprentissage (calibration) du modèle.
- Utilise un automate cellulaire simple pour favoriser l'émergence de zones de probabilités d'états à extension surfacique réaliste.

Mise en œuvre sous le logiciel Idrisi 32, la modélisation se découpe en trois phases :

- La constitution de la base de connaissances de la dynamique spatio-temporelle de l'occupation du sol par évaluation multicritère (EMC) des variables d'environnement. Les variables d'environnement d'origine sont transformées, pour chacun des types d'occupation du sol, par traitements statistiques et par logique floue en plans de probabilité d'occurrence de chacune des catégories d'occupation du sol. Ces plans de probabilité résultants, se basant sur la période d'apprentissage (1980 à 1989) servent à l'allocation spatiale des probabilités de transition.
- Le calcul des probabilités de transition par analyse de chaînes de Markov (ACM) entre les dates de la phase d'apprentissage et la date simulée (2000 - dernière date connue).
- L'allocation spatiale des probabilités de transition markoviennes : cette dernière étape utilise les résultats catégoriels de l'EMC. Ceux-ci sont intégrés, par évaluation multiobjectif (EMO), en une seule carte de l'occupation du sol simulée laquelle est traitée par un automate cellulaire (AC) basé sur un filtre de contiguïté spatiale.

La calibration du modèle est obtenue en modélisant l'état de l'occupation du sol à la dernière date connue (2000) sur la base de l'information provenant d'une période d'apprentissage englobant les deux dates précédentes (1980 et 1989). Bien que nous disposons d'une connaissance historique plus approfondie, il est évident que les conditions socio-économiques actuelles ne s'appliquent pas aux états de l'occupation du sol du XIX$^{\text{ème}}$ et du XX$^{\text{ème}}$ siècle jusque dans les années 1960.

Construction d'une base de connaissances des dynamiques de l'occupation du sol

La connaissance des dynamiques récentes est essentielle pour appréhender l'évolution future et sa modélisation. Nous entendons par connaissance des mesures statistiquement significatives du comportement spatial et temporel de l'occupation du sol en relation avec des critères environnementaux considérés explicatifs d'une partie de sa variabilité. Dans une évaluation multicritère (EMC), on distingue entre des critères binaires, les contraintes, et les critères ayant une aptitude variable dans l'espace, les facteurs. Les contraintes booléennes masquent certaines zones (occurrence possible ou non) ; ils peuvent s'appliquer à toutes formes d'occupation du sol (exclusion des espaces bâtis) ou être spécifique à certaines formations (limite altitudinale des conifères). Les facteurs traduisent une connaissance graduée pour l'objectif en question (une forme d'occupation du sol) - ils peuvent être pondérés et on peut définir leur degré de compensation.

Pour chaque catégorie d'occupation du sol la connaissance de son comportement spatio-temporel provient d'une analyse diachronique des dynamiques et de la friction géographique en comparant la répartition théorique (espace homogène) et la répartition réelle (niveaux de probabilité 99% et 99.9%). La rugosité géographique est exprimée par les facteurs d'envi-

ronnement cartographiés (altitude, pente, exposition, accessibilité, proximité aux entités de même nature, statut de gestion particulière de certaines zones et probabilité de changement) et disponibles à la même résolution spatiale.

Les facteurs sont standardisés par recodage manuel ou par l'emploi de fonctions fuzzy et pondérés par l'emploi de la matrice de Saaty ([Saaty, 1977]) qui renvoie le vecteur propre de chaque facteur. L'approche de l'EMC développée ([Eastman *et al.*, 1993]) inclut des poids d'ordre (ordered weighted averaging - OWA) permettant le choix du niveau de risque et de compensation entre facteurs. Ces poids d'ordre classent les aptitudes par rang croissant et aboutissent à un classement spécifique à chaque pixel. Nous avons opté pour une approche conservatrice (peu de risques et niveau de compensation limité) comme l'exprime la Figure 3.3.

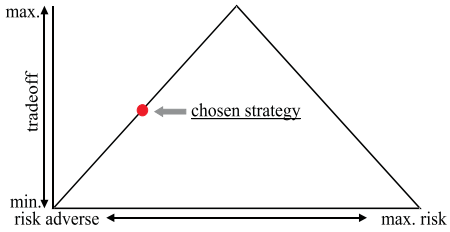

FIG. 3.3 – Espace de décision et approche EMC-OWA choisie

Calcul des probabilités de transition

Le calcul prédictif de l'occupation du sol est opéré par une analyse des chaînes de Markov, un processus discret avec des pas temporels discrets et dont les valeurs à la date prédite dépendent des valeurs à des dates antérieures. La prédiction est exprimée par une estimation des probabilités de transition. Le test consiste à prédire l'occupation du sol en 2000 (dernière date connue) sur la base de 1980 et de 1989. Le résultat se présente sous forme d'une matrice dans laquelle sont codées les probabilités de changement de chaque catégorie d'occupation du sol ainsi que le nombre de pixels affectées entre la dernière date d'apprentissage et la date projetée. La fonction calcule également une carte de probabilité conditionnelle pour chaque catégorie d'occupation du sol indiquant la probabilité markovienne par pixel de la modalité en question à la date projetée. Une intégration stochastique de l'ensemble des cartes par modalité en une seule est possible mais aboutit à un résultat relativement éloigné de la réalité (image bruitée) car cette procédure purement statistique ne tient pas compte ni des règles de connaissances établies par EMC, ni de la continuité spatiale.

Allocation spatiale des probabilités prédites

L'allocation spatiale des probabilités markoviennes intègre la connaissance sur la répartition probable de l'occupation du sol (EMC), une évaluation multiobjectif (EMO) tenant compte des objectifs concurrents (chaque modalité d'occupation du sol étant un objectif) et un automate cellulaire, basé sur un filtre de contiguïté spatiale. La fonction utilisée sous Idrisi est CA_Markov dont l'algorithme est itératif afin de tenir compte des distances temporelles entre les deux dates d'apprentissage et la dernière date d'apprentissage et la date de projection. Elle donne en sortie une carte prospective de l'occupation du sol pro-

bable. La Figure 3.4 résume les principales étapes de modélisation par SIG ([Paegelow, 2003], [Paegelow *et al.*, 2004a]).

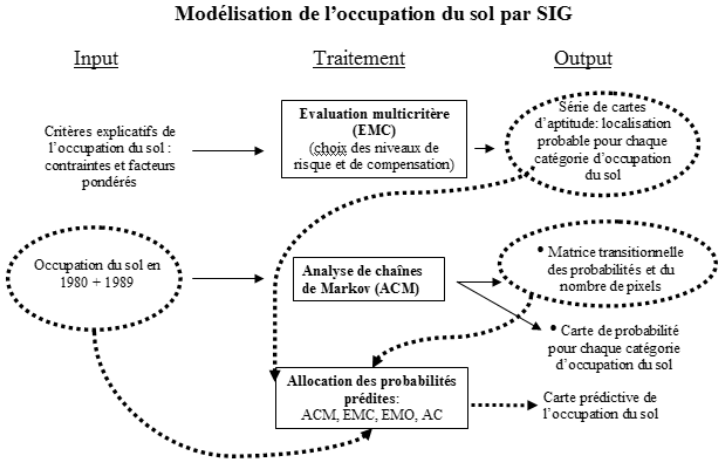

FIG. 3.4 – Modélisation de l'occupation du sol par SIG : aperçu des principales fonctions et de leur enchaînement

3.3.2 Approche par réseaux de neurones

L'idée de l'utilisation de réseaux de neurones a été principalement motivée par leurs remarquables capacités d'adaptation et de souplesse face à un très grand nombre de problèmes, notamment lorsque ceux-ci présentent des aspects non linéaires ou lorsque les variables explicatives sont fortement corrélées ; ces deux aspects se retrouvent dans le travail de prédiction que nous cherchons à effectuer. Aussi, les réseaux de neurones ont récemment connu une grande popularité et ont très favorablement concurrencé les méthodes statistiques classiques. On les retrouve notamment dans la prédiction de séries chronologiques (cf. [Bishop, 1995], [Lai and Wong, 2001] et [Parlitz and Merkwirth, 2000]). Le cadre dans lequel nous sommes amenés à travailler est encore plus étendu puisqu'il s'agit ici d'un processus spatio-temporel auquel s'ajoutent des variables explicatives comme nous le détaillerons plus loin.

Réseaux de neurones multi-couches (perceptrons)

Nous avons travaillé avec une classe particulière de réseaux de neurones, les réseaux multi-couches ou perceptrons. Ceux-ci ont été les premiers à connaître un essor important ; leur création est issue des premières tentatives de modélisation des principes de base régissant le fonctionnement du cerveau même si leur champ d'application s'est, depuis, considérablement élargi, notamment au traitement de données statistiques (pour plus de détails, consulter [Davaloe and Naïm, 1969]). Lorsque l'on parle de réseaux à couches, le nombre de couches est à définir par l'utilisateur mais doit comprendre au minimum une couche d'entrée et une couche de sortie ; les autres couches dont le nombre varie sont appelées couches cachées ; pour leurs remarquables propriétés d'approximation, nous avons choisi d'utiliser un réseau à une couche cachée dont l'architecture générale est celle de la Figure 3.5.

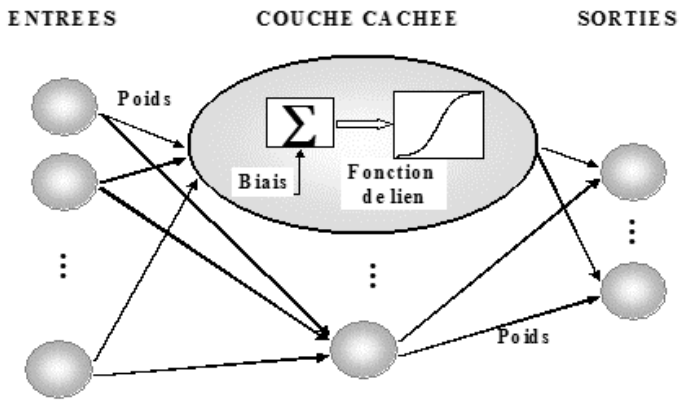

FIG. 3.5 – Architecture d'un réseau à une couche cachée

Détaillons un peu le fonctionnement de ce réseau : en entrée, la valeur des neurones est celle des variables explicatives du modèle ; chacune de ces valeurs numériques est multipliée par un certain nombre de poids pour être, finalement, additionnée et transformée par une fonction de lien au niveau des neurones de la couche cachée. Enfin, les valeurs numériques des neurones de la couche cachée subissent à leur tour une multiplication par des poids et leur addition donne la valeur des neurones de sortie qui modélisent la variable expliquée. Les poids, généralement notés w, sont choisis lors d'une phase dite d'apprentissage sur un jeu de données test et minimisent l'erreur quadratique de ce jeu de données. Finalement, les réseaux de neurones à une couche cachée sont les fonctions de la forme :

$$\Psi_w(x) = \sum_{i=1}^{q} w_i^{(2)} g\left(x^T w_i^{(1)} + w_{i,0}^{(1)}\right)$$

où x est le vecteur des variables explicatives du modèle, q_2 le nombre de neurones sur la couche cachée, g la fonction de lien de la couche cachée (typiquement g est la sigmoïde $g : x \to \frac{1}{1+e^{-x}}$), $w^{(1)}$ sont les poids entre la couche d'entrée et la couche cachée et $w^{(2)}$ les poids entre la couche cachée et la couche de sortie. L'intérêt de ce type de réseau est expliqué par le résultat suivant ([Hornik, 1993]) : les réseaux de neurones à une couche cachée permettent d'approcher, avec la précision souhaitée, n'importe quelle fonction continue (ou d'autres fonctions qui ne sont pas nécessairement continues) : c'est ce que l'on appelle la capacité d'approximateur universel et c'est aussi ce qui leur permet de s'appliquer avec une grande efficacité à un grand nombre de modèles.

Modèle pour les données de Garrotxes

Dans l'exemple des données de Garrotxes, plusieurs facteurs ont été retenus comme pouvant influencer l'occupation du sol d'un pixel donné à la date t :

- *pour l'aspect temporel (processus d'ordre 1)* : la valeur du pixel considéré à la date précédente (t - 1) que l'on exprime sous forme disjonctive (par exemple, si l'on dispose de 8 catégories d'occupation du sol, la première sera codée sous la forme d'un vecteur à 8 coordonnées : (1 0 0 0 0 0 0 0), la seconde : (0 1 0 0 0 0 0 0), etc) ;

- *pour l'aspect spatial* : la fréquence de chaque type d'occupation du sol dans le voisinage du pixel considéré à la date précédente $(t-1)$. Se pose alors le problème du choix du voisinage (taille et forme) : pour la forme, diverses possibilités s'offrent à nous, de la plus simple (voisinage carré ou en étoile) à des voisinages plus sophistiquées (voisinage suivant la pente pour mieux tenir compte des influences morphologiques du terrain). Quant à la taille du voisinage, il s'agira de déterminer jusqu'à quelle distance un pixel est susceptible d'influencer le pixel considéré. Afin de respecter la spatialisation de la carte, on pondèrera l'influence d'un pixel par une fonction décroissante de la distance au pixel considéré (cf. Figure 3.6) ;

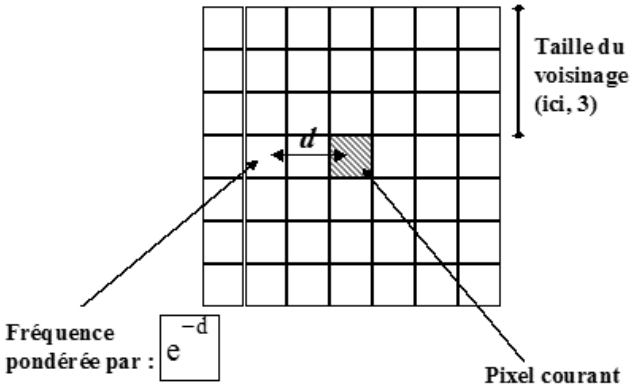

FIG. 3.6 – Un exemple de voisinage

- des variables environnementales (pente, altitude, ...).

Mise en œuvre

A l'issue d'une phase exploratoire qui nous a permis de cerner les variables pertinentes et divers paramètres du modèle comme la forme du voisinage (que, dans un soucis de simplicité, nous avons choisi carré), sa taille (finalement fixée à 3 ; cf. Figure 3.5) ou le nombre de neurones optimal sur la couche cachée, l'architecture choisie compte, en entrée, 19 neurones :

- *pour l'aspect temporel* : 7 neurones pour le codage disjonctif de la valeur du pixel (le bâti, constant, ayant été retiré de l'étude) à la date précédente $(t-1)$;
- *pour l'aspect spatial* : 8 neurones pour la fréquence des divers types d'occupation du sol dans le voisinage (fréquence pondérée par une fonction décroissante de la distance) ;
- 4 neurones pour les variables d'environnement pente, altitude, exposition et distance aux infrastructures (préalablement centrées et réduites).

Le réseau dispose aussi de 8 neurones sur la couche cachée et de 7 neurones en sortie, chacun estimant la probabilité d'appartenance du pixel à un type d'occupation du sol (hors bâti).

Pour la phase d'apprentissage, nous avons utilisé comme jeu de données la carte de 1980 avec comme cible à prévoir la carte de 1989 et la carte de 1989 avec comme cible à prévoir la carte de 2000. Après avoir repéré de larges zones dans lesquelles l'occupation du sol était stable, nous avons considéré uniquement les pixels dont un voisin au moins avait une valeur d'occupation du sol différente (exploitant ainsi pleinement la spatialisation des données) :

ces pixels seront, dans la suite, nommés pixels frontières. Les autres pixels ont été considérés comme stables dans un intervalle de temps de 10 ans.

D'un point de vue calculatoire, les programmes ont été réalisés à l'aide du logiciel Matlab (cf [Beale and Demuth, 1998]) et sont disponibles sur demande.

3.3.3 Approche par modèle linéaire généralisé

Le *modèle de régression logistique* est un modèle linéaire généralise (et donc paramétrique) dans lequel la variable réponse est qualitative et qui permet d'obtenir une prédiction de celle-ci en tenant compte d'un ensemble d'informations issues de variables explicatives. Lorsque la réponse possède plus de deux modalités, on parle de *modèle de régression logistique multiple* ou *modèle de régression polychotomique* ([Hosmer and S., 1989]). D'autres développements plus récents concernant ce modèle ont été réalises par [Kooperberg *et al.*, 1997]. Ce type de modèle logistique est particulièrement bien adapte à notre problématique puisqu'il s'agit de prédire pour chaque pixel de la carte un type d'occupation du sol (variable réponse ayant 8 modalités codées par un entier ν, $\nu = 1, 2, \ldots, 8$). La spécificité de notre étude vient du fait que, outre la nature topographique du pixel (pente, altitude, ...), le modèle choisi doit tenir compte d'un effet spatial (état de la végétation dans l'environnement du pixel) et d'un effet temporel (évolution du type d'occupation du sol du pixel et de son environnement). En ce sens il s'agit d'adapter le modèle de régression logistique à notre cadre, un des enjeux les plus importants étant le choix de la forme et de la taille de l'environnement pris en compte par le modèle.

De façon générale, le modèle de régression logistique permet de modéliser, en fonction d'un certain nombre de paramètres, la probabilité pour que le type d'occupation du sol d'un pixel au temps t (c'est-à-dire la variable réponse) soit égal à un des 8 catégories d'occupation du sol. Il s'agit donc d'estimer les paramètres inconnus du modèle, et ensuite les probabilités a posteriori de type d'occupation du sol sachant les valeurs des différentes variables explicatives. On utilise ensuite une règle de type *bayésien* consistant à affecter au temps t à un pixel donné l'indice de végétation ayant la plus forte probabilité *a posteriori*.

Régression logistique multiple spatio-temporelle

Indexons par $i = 1, 2, \ldots, N$ les pixels de la carte d'occupation du sol et notons $\mathcal{I}_i$ l'ensemble des informations dont on dispose concernant le pixel numéro i. D'un point de vue formel, le modèle de régression logistique multiple que nous adoptons peut se présenter sous la forme générale suivante :

$$\log\left(\frac{Prob(\text{pixel}_i = \nu|\mathcal{I}_i)}{Prob(\text{pixel}_i = 8|\mathcal{I}_i)}\right) = \alpha_\nu + \gamma_{\nu,\mathcal{I}_i},\ i = 1, \ldots, N, \tag{3.1}$$

où α_ν est un paramètre associé au type d'occupation du sol ν que l'on souhaite prédire pour le $i^{\text{ème}}$ pixel et $\gamma_{\nu,\mathcal{I}_i}$ un ensemble de paramètres lies à ν ainsi qu'aux informations concernant toujours ce pixel numéro i. Ainsi, le nombre total de paramètres mis en jeu dans ce modèle dépend uniquement du nombre de types d'occupation du sol et du nombre de variables explicatives. Dans l'expression (3.1), $Prob(\text{pixel}_i = \nu|\mathcal{I}_i)$ représente la probabilité que l'occupation du sol du pixel i soit du type ν lorsque les variables explicatives prennent les valeurs décrites par l'ensemble $\mathcal{I}_i$. Notons que l'expression (3.1) modélise le rapport (son logarithme) de la probabilité qu'un pixel prenne la modalité v sur la probabilité que ce pixel prenne la modalité codée 8 ce qui permet d'intégrer la contrainte que la somme des huit probabilités est égale à 1. Dans l'expression (3.1), nous devons intégrer l'*effet temporel* : celui-ci est pris en compte en faisant dépendre le type d'occupation du sol du pixel i du temps

t c'est-à-dire en posant $\text{pixel}_i = \text{pixel}_i(t)$. Par ailleurs l'information (ou plus exactement une partie de cette information) dépend du temps $t-1$: $\mathcal{I}_i = \mathcal{I}_i(t-1)$. L'idée consiste donc à calculer la probabilité qu'un pixel prenne un type d'occupation du sol ν à l'instant t en fonction de l'information que l'on possède sur ce même pixel à l'instant précédent $t-1$; on répète cette procédure pour tous les pixels de la carte. Connaissant les cartes 1980 $(t-1)$ et 1989 (t), on peut estimer l'ensemble des paramètres de notre modèle de sorte à ajuster au mieux la carte 1989. Il suffit alors d'incrémenter le temps dans notre modèle pour prédire la carte à l'instant futur $t+1$ (2000) à partir de la carte observée à l'instant t (1989). Quant à l'*effet spatial*, il est pris en compte de la même façon que dans l'approche par réseau de neurones. Il est en effet naturel de penser que l'évolution de l'occupation du sol du pixel i dépend de celle des pixels environnants. Pour cela on considère un voisinage carré V_i autour du pixel numéro i que l'on souhaite prédire et on extrait comme information du voisinage V_i le nombre de pixels prenant le type numéro 1 d'occupation du sol, le type numéro 2 d'occupation du sol, ... Cette façon de procéder revient à supposer une *invariance isotrope*, c'est-à-dire que le type d'occupation du sol autour du pixel i ne dépend pas de la direction. Dans la mise en oeuvre de la méthode, nous avons privilégié la simplicité de la forme du voisinage (carré). Lors de développements ultérieurs, on pourrait envisager d'autres formes que le carré (étoile, rectangle, ...) et varier s'il en résulte un gain ou pas en terme de prédiction. On peut également envisager une modélisation privilégiant certaines directions c'est-à-dire rompant avec l'hypothèse d'invariance isotrope. Notons cependant qu'il en résulterait un modèle avec un plus grand nombre de paramètres et que de ce point de vue on doit également composer avec la capacité à bien estimer un modèle qui serait trop complexe.

En combinant effet temporel et effet spatial, on est finalement amené à considérer le modèle suivant

$$\log\left(\frac{Prob(\text{pixel}_i(t)=\nu|\mathcal{I}_i(t-1))}{Prob(\text{pixel}_i(t)=8|\mathcal{I}_i(t-1))}\right) = \alpha_\nu + \gamma_{\nu,\mathcal{I}_i(t-1)},$$

où $\mathcal{I}_i(t-1)$ englobe l'information extraite du voisinage $V_i(t-1)$, c'est-à-dire tient compte de l'occupation du sol observée autour du pixel numéro i à l'instant $t-1$. Enfin, $\mathcal{I}_i(t-1)$ comprend également l'information issue des variables telles que la pente ou l'altitude décrites plus haut.

Mise en oeuvre

D'un point de vue pratique, la mise en oeuvre se décompose en deux étapes : une étape d'estimation et une étape de validation.

Etape d'estimation : On estime les paramètres du modèle (α_ν et ceux contenus dans $\gamma_{\nu,\mathcal{I}_i(t-1)}$). La procédure d'estimation est basée sur la maximisation de la *vraisemblance pénalisée*, critère bien connu en statistique pour la stabilité des solutions obtenues. L'algorithme d'optimisation utilise est de type *Newton-Raphson*. Remarquons que la pénalisation introduit un nouveau paramètre, appelle paramètre de pénalisation et note ϵ, qu'il faudra choisir. Comme cela a été dit précédemment, on utilise les cartes 1980 et 1989 pour estimer les paramètres, ceci pour différentes tailles de voisinage et valeurs ϵ.

Etape de validation : Il s'agit de déterminer la taille de voisinage et le paramètre de pénalisation optimaux en ce sens que ces choix fourniront une prédiction de la carte 2000 la plus proche possible de celle observée Plus précisément, à l'aide de étape précédente, il est possible de construire plusieurs prédictions de la carte 2000, chacune correspondant à différentes tailles de voisinage et valeurs ϵ. En comparant les cartes prédîtes pour 2000 avec la carte réelle de 2000, on repère la carte qui possède le plus petit nombre de pixels

mal prédits ; la taille de voisinage et le paramètre de pénalisation correspondants seront considérés comme étant optimaux.

3.4 Résultats et interprétation

Les trois approches ont été testées par la modélisation de l'occupation du sol à la dernière date connue (2000).

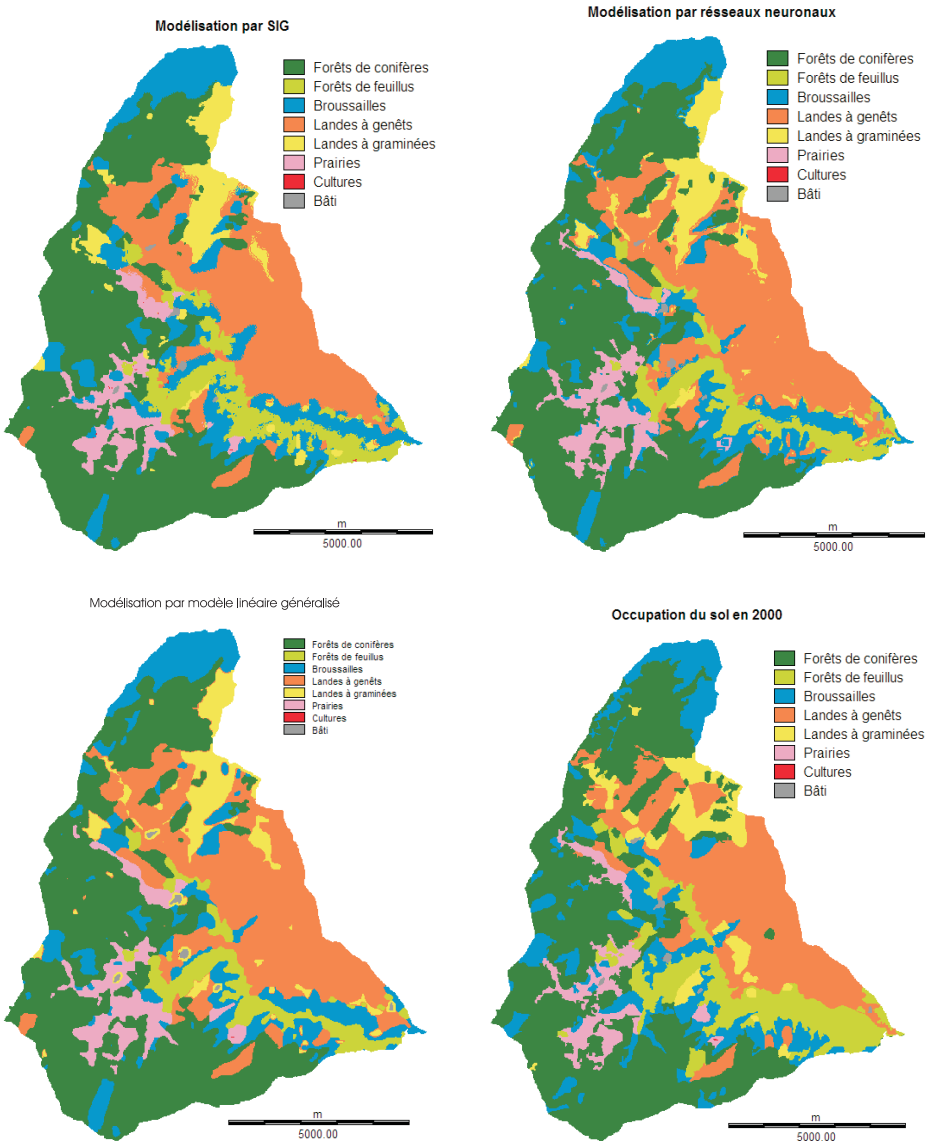

FIG. 3.7 – Résultats des modélisations de l'occupation du sol en 2000 et occupation du sol réelle

Les résultats globaux (pourcentage de surface) de ce test sont proches de la réalité (cf. Figure 3.7 et Tableau 3.1).

Occupation du sol en 2000	Réalité	Modélisation		
		SIG	Réseaux neuronaux	Modèle linéaire généralisé
Forêt de conifères	40,9	40,7	41,4	41,2
Forêt de feuillus	11,7	6,0	7,2	6,3
Broussailles	15,1	16,9	14,1	14,6
Landes à genêts	21,6	23,3	25,1	25,9
Landes à graminées	5,7	7,6	6,0	6,2
Prairies	4,8	5,2	6,0	5,7
Cultures	0,01	0,1	0	0

TAB. 3.1 – Surface en pourcentage de l'occupation du sol en 2000, réelle et modélisée

Cependant il convient d'analyser la répartition spatiale de ces sommes de surface prédites par catégorie à haute résolution (coté du pixel environ 18 m). Le Tableau 3.2 compare les résidus par catégorie en pourcentage de la surface réelle de chaque catégorie mise à part les cultures dont le nombre de pixels tend vers zéro. Le premier enseignement de ce tableau est la relative concordance des résultats des trois approches. On remarque également que la modélisation de modalités ayant une grande superficie (forêt de conifères, landes à genêt) est plus facile que celle des catégories d'occupation du sol de faible ampleur. Ainsi, quel que soit le modèle, moins d'un pixel sur deux a été correctement prédit pour la catégorie broussailles. Parmi les modalités de faible surface, on note cependant de grandes différences selon leur stabilité spatio-temporelle : les prairies étant plus stables que les landes à graminées, leur taux de prédiction est meilleur.

Les taux de prédiction globale des trois méthodes sont très proches : 72.8 % (SIG), 74.3 % (réseaux neuronaux) et 72.8 % (modèle linéaire généralisé).

Occupation du sol en 2000 Surface (%)	Forêt de conifères 40,9	Forêt de feuillus 11,7	Brous--sailles 15,1	Landes à genêts 21,6	Landes à graminées 5,7	Prairies 4,8
Résidus (%) de modélisation SIG (27,2%)	11,42	55,28	51,92	17,13	54,39	30,35
Réseaux de neurones (25,7%)	10,60	45,84	54,54	16,23	59,38	19,26
Modèle linéaire généralisé (27,2 %)	11,88	51,65	57,07	14,35	59,24	25,57

TAB. 3.2 – Pourcentage de résidus par catégorie d'occupation du sol et approche modélisatrice

Les modélisations n'ont pas vocation à prédire la réalité mais peuvent nous aider à mieux comprendre des changements spatio-temporels environnementaux et sociaux complexes. Dans ce sens, l'interprétation des résultats des modélisations doit tenir compte des limites des modèles. La modélisation de l'occupation du sol signifie une simulation de ce que la réalité pourrait être, un scénario raisonné et quantifiable dans le contexte d'aide à la décision.

Cependant une interprétation minutieuse des résultats devrait nous permettre à améliorer le modèle et, par conséquent, le taux de prédiction. Dans ce sens, l'analyse focalise surtout sur les résidus.

La catégorie d'occupation du sol la plus représentée (les conifères) obtient un très bon

score de prédiction alors que les broussailles, relativement présentes sur le territoire, obtiennent un très mauvais score (plus de la moitié de mal prédits). Diverses remarques permettent d'expliquer ces phénomènes et de penser à des stratégies d'amélioration de la prédiction. Tout d'abord, on peut constater que les broussailles sont la catégorie naturellement la plus dynamique sur un territoire caractérisé par un équilibre entre espace forestier et pastoral régi notamment par la gestion pastorale. Les broussailles sont également soumises le plus à des effets aléatoires : un feu de forêt, une coupe ou bien l'abandon de pâturage transforment en l'espace de 10 ans une parcelle en broussailles ; ce sont des phénomènes complètement incontrôlables.

Ecart de prédication	SIG	Réseaux neuronaux	Modèle linéaire généralisé
1 catégorie	12,9	12,5	13,0
2 catégories	9,1	8,5	9,2
3 catégories	3,2	2,9	3,1
4 ou 5 catégories	1,9	1,8	1,9
Total résidus	27,2	25,2	27,2

TAB. 3.3 – Analyse des résidus de la modélisation par l'écart de catégorie entre la réalité et la modélisation (données en pourcentage de la surface totale)

Bien que l'occupation du sol soit décrite de manière qualitative, ses différentes catégories s'échelonnent entre des formations fermées (forêt de conifères, forêt de feuillus) et ouvertes (cultures). Ces rangs « paysagers » permettent de quantifier l'erreur de prédiction, exprimée en nombre de catégories (cf. Tableau 3.3). Ainsi, quelle que soit la méthode de modélisation, pour environ la moitié des pixels mal prédits, l'erreur de prédiction n'est que d'une catégorie à une résolution spatiale élevée. Le nombre de résidus décroît fortement avec l'augmentation de l'écart entre la réalité et la projection.

	3 modèles	2 modèles			1 modèle			Aucun modèle
Prédiction correcte par :		RN + MLG	SIG + MLG	SIG + RN	SIG	RN	MLG	
Forêt de conifères	85,35	1,65	0,40	0,68	0,96	1,75	0,76	8,53
Forêt de feuillus	46,26	0,90	0,57	3,11	4,98	3,93	0,50	39,75
Brous--sailles	32,38	5,92	2,64	2,50	7,75	3,89	1,03	43,89
Lande à genêts	76,98	4,99	2,45	0,74	2,82	0,70	0,93	10,39
Lande à graminées	26,30	7,63	3,27	2,24	7,71	2,91	2,32	47,62
Prairies	59,14	11,64	1,66	4,70	1,06	5,26	1,99	14,55
Total	**66,49**	**3,75**	**1,42**	**1,54**	**3,23**	**2,33**	**0,92**	**20,32**

TAB. 3.4 – Mise en relation des scores de prédiction correctes des trois modèles avec l'occupation du sol en 2000. Données en % de la surface totale. RN = modèle par réseaux neuronaux ; MLG = modèle linéaire généralisé ; SIG = modèle SIG

Un autre aspect intéressant est la grande concordance des trois modèles (cf. Figure 3.8

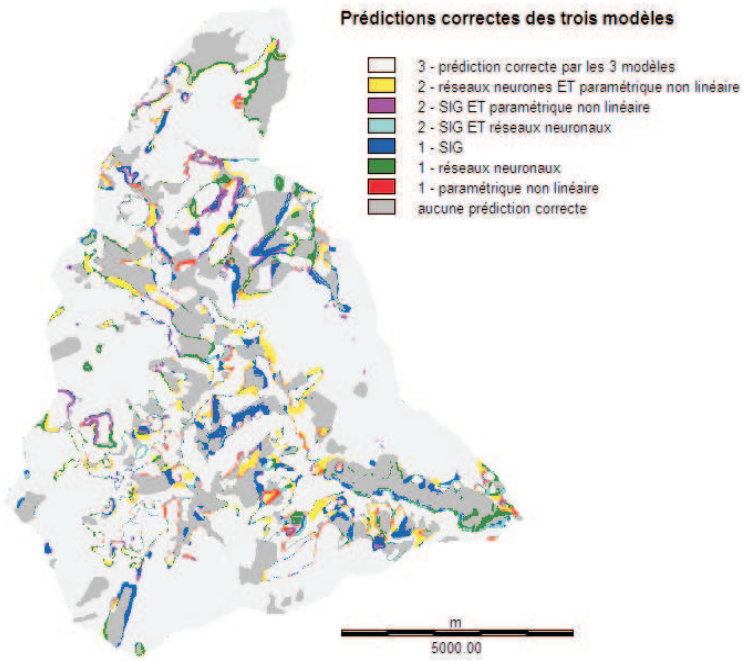

FIG. 3.8 – Prédictions correctes croisées par les 3 modèles

et Tableau 3.4). Ainsi 66.5 % de la surface totale est correctement prédite par chacun des modèles, 20.3 % par aucune.

Un autre enseignement du croisement des pixels correctement prédits par les différents modèles est la similitude des résultats des deux modèles statistiques. Globalement, les surfaces correctement prédites par deux modèles sur trois le sont le plus souvent par les réseaux neuronaux et le modèle linéaire généralisé (3.75 %). Le taux de prédiction combiné de la méthode SIG avec chacun des modèles statistiques est nettement plus faible. La relative distinction du modèle SIG est corroborée par les surfaces correctement prédites par seulement une des approches (3.23 %). Une analyse catégorielle souligne ce constat. Ainsi, pour les surfaces correctement prédites par uniquement deux modèles, cinq catégories d'occupation du sol sur six le sont par les approches statistiques. A l'inverse, les surfaces uniquement prédites correctement par un seul modèle le sont le plus souvent par le modèle SIG (quatre catégories d'occupation du sol sur six). En outre, le modèle SIG se distingue par le fait que son taux de prédiction est meilleur pour des zones affectées par un changement d'état, soit entre la dernière date de la phase d'apprentissage et la date simulée, soit durant la phase d'apprentissage. Les deux méthodes statistiques, au contraire, aboutissent à des résultats légèrement plus proches de l'observation sur les secteurs stables. Ce comportement spécifique sur les marges, s'explique par les procédures d'affectation spatiale différentes des transitions temporelles où le modèle SIG fait appel à une analyse géographique de la rugosité de l'espace par évaluation multicritère. Il s'agit, en l'occurrence, d'indices à approfondir dans la perspective de l'intégration des trois modèles en un seul.

Les résultats font également ressortir certaines limites des modélisations :

- La variable modélisée est décrite par un nombre limité de catégories. Ce choix, intentionnel, est motivé par la qualité et l'origine des données source afin de minimisez

les erreurs d'interprétation. L'inconvénient est une certaine variabilité à l'intérieur de chaque catégorie qui n'est pas prise en compte par les modèles. Ainsi peut-on voir dans les trois cartes de modélisation (Figure 3.8) une vaste zone prédite en broussailles dans le secteur sud-est qui est réellement occupée par des bois de feuillus. Pendant la période d'apprentissage cette zone a été photointerprétée comme broussailles en 1980 et en 1989. Cependant les broussailles se sont densifiées, leur composition floristique a changé au profit de Quercus ilex formant une strate arborescente dominante en 2000 où la zone a été classée forêt de feuillus.

- La période d'apprentissage est peu fournie. Les modélisations ne se basent que sur deux dates connues. Ceci pose des problèmes par rapport à la source de données à utiliser. En l'occurrence il s'agit de données haute résolution : des missions de photographies aériennes espacées dans le temps au point que l'utilisation de missions plus anciennes nous semble problématique sachant que le contexte socio-économique a beaucoup évolué.
- Le taux d'explication de la variabilité spatio-temporelle de l'occupation du sol par les variables d'environnement est inégal selon les catégories d'occupation du sol.
- Finalement chaque modèle est affecté par un bruit aléatoire. Des effets aléatoires comme des incendies de forêt, du chablis ou des programmes de reforestation sont difficilement prévisibles.

A cela s'ajoutent des limites spécifiques à chacun des modèles. Pour le modèle SIG deux facteurs limitants sont à mentionner. Des zones stables pendant la période d'apprentissage sont prédites stables par l'analyse des chaînes de Markov. En outre, les procédures EMC, EMO et l'automate cellulaire ne gèrent que ra répartition spatiale des scores de probabilités calculés par l'ACM. Cette restriction est, en partie, aussi valable pour les approches statistiques.

3.5 Perspectives

Les résultats exposés sont les premiers au sein d'un projet de recherche portant sur trois sites d'études. Les mêmes modèles seront appliqués à la Montagne de Lure en Haute Provence ainsi qu'à la Alta Alpujarra Granadina formant la partie occidentale du versant sud de la Sierra Nevada (Espagne).

Cette comparaison croisée entre plusieurs approches appliquées sur plusieurs sites devrait, d'une part, limiter l'effet de singularité propre à chaque terrain d'études et, d'autre part, nous donner à terme des indications afin de proposer un modèle intégré, issu des trois méthodes mises en œuvre actuellement parallèlement. En même temps, les modèles restent évolutifs au sens où l'interprétation des premiers résultats réoriente les prochaines étapes. A ce propos il nous semble intéressant de remédier au problème de la variabilité interne à chaque catégorie d'occupation du sol par une approche semi quantitative. Aux modalités qualitatives décrites s'ajoutent des données ordonnées sur le taux de recouvrement de la strate arborescente.

Enfin, les approches mises en œuvre sans intervention « spécialiste » (du type SIG) donnent des résultats aussi bons que la méthode « supervisée ». La combinaison de méthodes purement mathématiques avec un guidage expert pourrait permettre de gommer les imperfections du modèle. Cette voie est encore à explorer afin de progresser vers un modèle intégré de la simulation prospective de l'occupation du sol.

Deuxième partie

Réseaux de neurones et SVM en Analyse des Données Fonctionnelles

Résumé :

Nous présentons, dans cette partie, divers modèles de discrimination et de régression fonctionnelles basés sur des perceptrons multi-couches ou des Support Vector Machine (SVM).

Dans un premier temps, le Chapitre 4 est une synthèse sur les diverses approches de Régression Inverse lorsque la variable explicative est fonctionnelle et que le problème à traiter est une discrimination à plusieurs classes. Les méthodes de régression inverse ont été, en premier lieu, introduites par [Li, 1991] pour faire face au fléau de la dimension dans le cas de la régression multivariée. Elles ont ensuite été étendues au cadre hilbertien dans [Ferré and Yao, 2003] et dans ([Dauxois *et al.*, 2001]). Dans l'article de ce chapitre, nous explorons diverses approches, par régularisation et par filtrage, de discrimination par régression inverse fonctionnelle et nous les appliquons sur deux problèmes, l'un réel et l'autre simulé.

L'efficacité des méthodes de régression inverse dans la résolution de problèmes en grande dimension, et notamment, en dimension infinie, nous pousse à les utiliser, dans le Chapitre 5, comme base de projection pertinente pour des perceptrons multi-couches à entrées fonctionnelles. Utiliser l'espace EDR comme un espace de projection exhaustif consitue un pré-traitement efficace aux méthodes de régression et classification classiques. Le modèle présenté ici allie donc la souplesse d'un réseau de neurones (qui permet de fournir une solution non linéaire) et la rapidité d'une méthode linéaire de réduction des données comme la SIR. Nous présentons également, dans ce chapitre, un résultat de consistance pour l'estimation des paramètres de cette méthode neuronale par régression inverse et illlustrons sa capacité par des expériences sur deux problèmes réels. On trouvera dans l'Annexe A.1 l'intégralité des preuves des théorèmes de ce chapitre. Enfin, nous signalons que ce travail est à mettre en parallèle avec les travaux de [Rossi and Conan-Guez, 2005a], [Rossi and Conan-Guez, 2005c] et [Rossi and Conan-Guez, 2005d] qui traitent aussi de perceptrons multi-couches à entrées fonctionnelles mais en utilisant une base de projection déterministe (base Spline, par exemple) et avec les travaux de [Thodberg, 1996] qui utilise une base de projection issue de l'Analyse en Composante Principale qui ne tient donc pas compte de la variable cible. Les Chapitres 4 et 5 sont issus d'un travail mené en collaboration avec Louis Ferré [2].

Par la suite, dans le Chapitre 6, nous présentons un article traitant de l'utilisation des SVM (Support Vector Machine) pour le traitement de données fonctionnelles. Ici, nous nous sommes limités à l'étude de problèmes de discrimination dans lesquels la variable explicative est fonctionnelle. Nous définissons ainsi des noyaux simples qui prennent en compte la nature fonctionnelle des données (projection sur des bases fonctionnelles adaptées, pré-traitement fonctionnel par différentiation ou par régression inverse). Les SVM utilisant de tels noyaux constituent des modèles fonctionnels non linéaires pour la discrimination. Nous montrons l'intérêt d'une telle approche sur une série de problèmes réels et démontrons, pour certains de ces noyaux, un résultat de consistance. La preuve de ce résultat est donnée en Annexe A.2. Ce travail a été mené en collaboration avec Fabrice Rossi [3]

Finalement, le Chapitre 7 présente brièvement une approche par régression inverse fonctionnelle qui est le pendant des travaux effectués pour la construction de réseaux de neurones à entrées fonctionelles (Chapitre 5). Cette approche est encore en voie d'exploration.

[2] Professeur à l'université Toulouse Le Mirail

[3] Maître de Conférence à l'Université Paris Dauphine, Chercheur détaché du projet AxIS, INRIA, Rocquencourt.

Chapitre 4

Discrimination de courbes par régression inverse fonctionnelle

Louis Ferré
GRIMM, Equipe d'Accueil 3686, Université Toulouse Le Mirail, France
Nathalie Villa
GRIMM, Equipe d'Accueil 3686, Université Toulouse Le Mirail, France

Référence : Discrimination de courbes par régression inverse fonctionnelle (2005), *Revue de Statistique Appliquée*, **LIII(1)**, pages 39-57.

Résumé :

Les méthodes de régression inverse telles que la SIR ([Li, 1991]) ont été développées dans le domaine de la régression multivariée pour éviter le célèbre fléau de la dimension. Elles ont été récemment étendues aux données fonctionnelles. Plusieurs approches ont été proposées et nous présentons ici un article de synthèse et de comparaison en abordant le cas où la variable réponse est un vecteur d'indicatrice d'appartenance à des classes. Nous montrons qu'alors la régression inverse conduit à une méthode de discrimination dont la pertinence est établie sur des données réelles et simulées.
Mots clés : *discrimination, données fonctionnelles, régression inverse, régression non paramétrique.*

4.1 Introduction

L'analyse discriminante est une méthode éprouvée qui a été largement étudiée et étendue à différents contextes depuis sa découverte par Fisher. Les domaines d'application variés dans lesquels les problèmes de discrimination se rencontrent expliquent sans doute son succès. Elle se définit comme une méthode de classification supervisée qui consiste à classer des individus sur la base de variables explicatives et d'un échantillon d'apprentissage pour lequel à la fois ces variables et l'affectation aux classes sont connues.

Notons X la variable explicative, J le nombre de classes, C la variable identifiant les classes et $\mu_j = E(X|C = j)$ pour $j = 1, ..., J$. Le problème essentiel est celui de l'affectation des individus aux classes. Schématiquement, l'affectation peut s'opérer soit en utilisant des

arguments géométriques, soit des arguments probabilistes. Dans le premier cas, affecter x à la classe j signifie que la distance de x au centre de gravité de la classe j est minimale, i.e., $d^2(x, \mu_j)$ est minimale en j pour une certaine distance d, habituellement la métrique de Mahalanobis. Cette règle peut s'appliquer directement aux données ou bien après réduction de la dimension par Analyse Factorielle Discriminante. Dans le deuxième cas, il s'agit de maximiser la probabilité $P(C = j|x)$ parmi toutes les valeurs de j. Au niveau statistique, tout le travail consiste à estimer cette probabilité. D'un point de vue paramétrique, la formule de Bayes fournit une réponse pour des modèles gaussiens et il est bien connu que cette règle d'affection est alors une version pénalisée de la règle géométrique. Mais on peut également estimer cette probabilité conditionnelle de façon non-paramétrique, voir e.g. [Hand, 1982]. Cela bien sûr lorsque le régresseur est multivarié. Mais que se passe-t-il si le régresseur est fonctionnel ?

Comme la plupart des méthodes multivariées (voir, e.g., [Dauxois and Pousse, 1976] ou plus récemment [Ramsay and Silverman, 1997]), l'analyse discriminante a été étendue au cas fonctionnel, moyennant cependant quelques adaptations. En particulier, si on considère le problème de l'affectation probabiliste, [James and Hastie, 2001] considérent le problème sous des hypothèses de normalité alors que [Ferraty and Vieu, 2003] aborde le problème sous l'angle de la régression non paramétrique.

Nous proposons ici une méthode sans hypothèse de loi. Elle est basée sur l'estimation du vecteur des probabilités $P = (P(C = j|X))_{j=1,...,J}$ à partir d'un modèle semi-paramétrique. Si on note Y le vecteur aléatoire de R^j tel que $Y = (Y_1, ..., Y_J)$ avec $Y_j = I_{[C=j]}$ où I est la fonction indicatrice de $[C = j]$ et X une variable aléatoire fonctionnelle, on obtient très simplement que :

$$P = E(Y|X). \tag{4.1}$$

On pose alors le modèle suivant :

$$P = f(< \beta_1, X >, ..., < \beta_d, X >) \tag{4.2}$$

où f est une fonction de R^d dans R^J et les β sont d fonctions définies sur le même ensemble que X. En fait, X est un processus stochastique continu $X(t)$ défini sur un intervalle I de R. On supposera que les fonctions X sont de carré intégrable et on considérera le produit scalaire sur L^2_I, ce qui signifie que $< \beta_k, X >= \int_I X(t)\beta_k(t)dt$.

Ce modèle est un modèle de réduction de dimension ; c'est une façon d'écrire que Y dépend de X uniquement au travers de sa projection sur un sous-espace d dimensionnel de L^2_I, engendré par les d vecteurs linéairement indépendants, $\beta_1, ..., \beta_d$. C'est, en ce sens, un espace "exhaustif" qui porte le nom d'espace EDR (pour Effective Dimension Reduction) dans la littérature sur la régression inverse ([Li, 1991]) ou central dans [Cook and Yin, 2001]. S'agissant de données fonctionnelles, cet espace EDR va notamment permettre d'exhiber une base "optimale" au sens de la régression sur laquelle seront projetées les données avant de procéder à une autre analyse. L'estimation de P va dépendre de la façon dont est estimé cet espace. Dans la Section 4.2, nous verrons que si X admet un opérateur de covariance, la solution dérive directement des résultats de [Dauxois *et al.*, 2001] et que l'estimation de l'espace repose alors sur la décomposition spectrale de l'opérateur $\Gamma_X^{-1}\Gamma_{E(X|Y)}$, où Γ_Z désigne l'opérateur de covariance d'une variable fonctionnelle Z.

Pour un échantillon i.i.d. de taille n du couple (Y, X), $(Y_i, X_i)_{i=1,...,n}$, on déduit aisément des estimateurs convergents de Γ_X et $\Gamma_{E(X|Y)}$. Malheureusement, le premier opérateur n'est pas borné de sorte que son estimateur empirique est mal conditionné. Cette situation est bien connue en statistique fonctionnelle et plusieurs solutions ont été proposées pour contourner ce problème. Pour l'essentiel, il s'agit de méthodes de *régularisation* ou

de *filtrage*. Dans le premier cas, on s'applique à charger la diagonale de l'estimateur de l'opérateur de covariance. Par exemple, la Ridge-regression ([Hoerl and Kennard, 1970b] et [Hoerl and Kennard, 1970a]) est une méthode ancienne qui a été appliquée à l'analyse discriminante par [DiPillo, 1979] et [Friedman, 1989]. Une alternative consiste à utiliser un critère pénalisé comme dans l'analyse discriminante "fléxible", [Hastie *et al.*, 1994], ou dans l'analyse discriminante pénalisée, [Hastie *et al.*, 1995]. Notons au passage que ces deux approches reposent sur la méthode d'« optimal scoring ». On peut citer également les travaux de [Leurgans *et al.*, 1993] sur l'analyse canonique pénalisée en raison des liens étroits entre l'analyse canonique et l'analyse discriminante.

Dans [James and Hastie, 2001], une méthode de *filtrage* est utilisée. Elle consiste à projeter les données sur une base préalablement choisie, par exemple, une base d'ondelettes, de polynômes orthogonaux, de polynômes trigonométriques ou de splines. L'avantage de cette approche est qu'elle autorise le traitement d'observations faites à des instants différents. L'inconvénient est que le choix des éléments de la base n'est pas toujours aisé.

Concernant la régression inverse fonctionnelle, plusieurs solutions ont été envisagées. Ainsi, [Ferré and Yao, 2003] utilisent une méthode de filtrage en projetant les données sur une base formée des premières fonctions propres de l'opérateur de covariance de X. Elle reprend l'idée développée par [Bosq, 1991] pour des modèles AR1. [Ferré and Villa, 2005b] utilisent une méthode de régularisation. Enfin, pour éviter l'inversion, [Ferré and Yao, 2005] déduisent l'espace EDR des vecteurs propres de l'opérateur $\Gamma^+_{E(X|Y)}\Gamma_X$. Après avoir rappellé en Section 4.2 ce qu'est la régression inverse fonctionnelle, nous présenterons succinctement en Section 4.3 ces différentes méthodes en montrant comment elles s'inscrivent dans le cadre de l'analyse discriminante.

La Section 4.4 est elle consacrée à la régle d'affectation utilisée et nous terminons par la Section 4.5 où les méthodes seront mises en oeuvre et comparées sur des données réelles ou simulées.

4.2 La regression inverse

Soit X une variable aléatoire à valeur dans l'espace des fonctions de carré intégrable L^2_I, où I est un intervalle de R et Y une variable aléatoire à valeur dans R^J.

La régression inverse fonctionnelle s'appuie sur le modèle suivant :

$$Y = f(<\beta_1, X>, ..., <\beta_d, X>) + \epsilon \ , \tag{4.3}$$

où $\beta_1, ..., \beta_d$ sont des vecteurs de L^2_I linéairement indépendants, f est une fonction, inconnue, de R^d dans R^J et ϵ est une variable aléatoire dans R^J non-corrélée avec X. D'un côté, ce modèle (4.3) est un cas particulier du modèle semi-parametrique pour variables hilbertiennes présenté dans [Dauxois *et al.*, 2001] dont nous allons exploiter les propriétés. D'un autre côté, il admet comme cas particulier la situation où Y est un vecteur aléatoire de R^j tel que $Y = (Y^{(1)}, ..., Y^{(J)})$ avec $Y^{(j)} = I_{[C=j]}$, où I est la fonction indicatrice de $[C = j]$, et il sera donc un modèle pertinent pour l'analyse discriminante.

Si notre approche ne repose sur aucune hypothèse de loi, il est cependant nécesssaire de supposer que :

Hypothèse H-1 pour tout $b \in L^2_I$, si on pose $B' = (<\beta_1, X>, ..., <\beta_K, X>)$, alors $E(<b, X> | B)$ est linéaire en B ;

Hypothèse H-2 $E(\| X \|^4) < \infty$).

L'hypothèse H-1 est à la fois un cas particulier de l'hypothèse *H-1* de [Dauxois *et al.*, 2001] et la version fonctionnelle de l'hypothèse 1.6 de [Li, 1991]. Notons

qu'elle est vérifiée notamment si X est une variable fonctionnelle gaussienne ou plus généralement elliptique.

L'hypothèse H-2 assure l'existence de l'espérance de X, $E(X)$, notée μ par la suite et de son opérateur de covariance noté Γ_X. Cela permet également de définir $\mu_j = E(X|C = j)$ pour tout $j = 1, ..., J$ et $\Gamma_{E(X|Y)}$, l'opérateur de covariance de $E(X|Y)$. Par ailleurs, on supposera tout au long de l'article que :

Hypothèse H-3 Γ_X est définie positive.

Soient η_k les vecteurs propres de l'opérateur $\Gamma_X^{1/2}\Gamma_{E(X|Y)}\Gamma_X^{1/2}$, on pose $b_k = \Gamma_X^{-1/2}\eta_k$. Pour garantir, l'existence des vecteurs b_k ainsi définis, nous supposerons que (voir [Ferré and Yao, 2005]) :

Hypothèse H-4 Si $X = \sum_{i=1}^{\infty} \xi_i u_i$ est la décomposition de Karhunen-Loève de X, alors $\sum_{i=1}^{\infty}\sum_{j=1}^{\infty} \frac{E(\xi_i|Y)E(\xi_j|Y)}{\delta_i^2\delta_j} < \infty$ où $\delta_i = E(\xi_i)$.

En utilisant les résultats de [Dauxois *et al.*, 2001], on vérifie aisément que :

Théorème 4.1. *Sous les hypothèses H-1, H-2, H-3, H-4, $sp\{b_1, ..., b_d\}$ est inclus dans l'espace EDR.*

On en déduit alors que l'estimation de l'espace EDR s'obtient à partir de celle de $sp\{b_1, ..., b_d\}$. L'analogie avec la SIR et en particulier avec la SIR fonctionnelle est évidente mais deux points méritent d'être précisés. Tout d'abord, l'estimation de $\Gamma_{E(X|Y)}$ ne nécessite pas de tranchage et s'obtient directement par la matrice de covariance inter-groupe. Ensuite, la variable Y est ici multivariée alors qu'en SIR ou FSIR, elle est généralement univariée (voir cependant, pour la SIR, [Hsing, 1999] et [Li *et al.*, 2003]).

Si nous considérons ici le cas fonctionnel, notre approche s'applique également au cas multivarié. L'utilisation des méthodes de réduction de dimension a été récemment considérée en analyse discriminante multivariée. En effet, [Cook and Yin, 2001] considèrent un modèle comparable au modèle (4.3), mais dans lequel P est remplacé par la coordonnée maximale de P, revenant ainsi à un modèle univarié. Ils utilisent ensuite la méthode SIR ou SAVE ([Cook and Weisberg, 1991]) pour estimer ce sous-espace. De même, [Hernandez and Velilla, 2001] estiment l'espace central qui maximise la règle de Bayes. Leur technique repose sur la maximisation d'un critère basé sur l'entropie et conduit à une procédure beaucoup plus lourde que la méthode présentée ici dans le cas multivarié et elle n'a pas, à ce jour, d'équivalent dans le cas fonctionnel.

Les estimateurs des vecteurs de base de la méthode par réduction de dimension sont identiques aux fonctions discriminantes de l'analyse discriminante linéaire. Même si, *a priori,* ces deux problèmes ne sont pas à la base semblables, la relation entre l'analyse discriminante linéaire et la régression inverse provient du fait que chacune d'elle se ramène, pour la première direction, au problème de maximisation du critère de Rayleigh :

$$max \frac{< \Gamma_{E(X|Y)} b, b >}{< \Gamma_X b, b >}, \tag{4.4}$$

les autres directions étant solutions de problèmes identiques sous contraintes d'orthogonalité. Ceci est vrai dans le cas fonctionnel ou multivarié.

4.3 Estimation des paramètres

A partir d'un échantillon i.i.d. de taille n, (X_i, Y_i), pour $i = 1, ..., n$, les estimateurs de l'espace central se déduisent des estimateurs suivants. On estime $\Gamma_{E(X|Y)}$ par la matrice de

covariance inter groupes,

$$\Gamma^n_{E(X|Y)} = \sum_{j=1}^{J} \frac{n_i}{n} (\hat{\mu}_j - \bar{X}) \otimes (\hat{\mu}_j - \bar{X}),$$

où $n_j = \sum_{i=1}^n Y_i^{(j)}$ et $\hat{\mu}_j = \frac{1}{n_j} \sum_{i=1}^n X_i Y_i^{(j)}$. L'opérateur Γ_X est estimé par l'opérateur empirique, $\Gamma^n_X = \frac{1}{n} \sum_{i=1}^n (X_i - \bar{X}) \otimes (X_i - \bar{X}))$, où $\bar{X}$ est la moyenne empirique de X. Cependant, sous l'hypothèse H-2, Γ_X est un opérateur de Hilbert-Schmidt est n'est donc pas inversible dans L^2_I. De plus, si l'on restreint Γ_X à son image, l'opérateur ainsi obtenu est inversible, mais son inverse n'est pas borné. Ainsi, la matrice Γ^n_X sera mal-conditionnée et il convient d'utiliser des stratégies pour contourner ce problème. Nous présentons ci-dessous plusieurs solutions proposées dans le cadre de la régression inverse.

4.3.1 Une solution de filtrage

Nous présentons ici l'approche de [Ferré and Yao, 2003] : la "Fonctional Sliced Inverse regression", FSIR. Elle consiste à projeter les données dans une base des vecteurs propres de l'opérateur Γ_X. Cela conduit à un choix "objectif" de la base de projection alors que l'utilisation de bases orthogonales comme celle de Fourier, de fonctions splines, d'ondelettes, ne permettent pas des choix toujours pertinents des éléments de la base. Soit $(k_n)_{n \in N}$ une suite non convergente d'entiers. Pour tout n, on note Π_{k_n} le projecteur propre associé aux k_n plus grandes valeurs propres de Γ^n_X. L'estimation de l'espace EDR s'obtient par décomposition spectrale de l'opérateur $((\Pi_{k_n} \Gamma^n_X \Pi_{k_n})^+)^{1/2} \Gamma^n_{E(X|Y)} ((\Pi_{k_n} \Gamma^n_X \Pi_{k/_n})^+)^{1/2}$ où la notation "+" est utilisée pour représenter l'inverse généralisé d'une matrice. La consistance de l'estimateur de l'espace EDR a été étudiée dans [Ferré and Yao, 2003] lorsque Y est une variable aléatoire réelle. La convergence de $\Gamma^n_{E(X|Y)}$ vers $\Gamma_{E(X|Y)}$ permet d'étendre directement ce résultat au cas où Y est un vecteur d'indicatrice. Ces résultats reposent sur des hypothèses sur les valeurs propres de Γ_X. Grossièrement, celles-ci ne doivent pas tendre vers 0 trop rapidement. Précisons que ces hypothèses sont vérifiées, en particulier, si X est un mouvement brownien. La mise en oeuvre de cette méthode nécessite de sélectionner une valeur convenable pour k_n.

4.3.2 Une solution basée sur un inverse généralisé de $\Gamma^n_{E(X|Y)}$

Cette solution repose sur le fait que sous le modèle (4.3), $\Gamma_{E(X|Y)}$ est un opérateur de rang fini. Ainsi, $(\Gamma_X)^{-1/2} \Gamma_{E(X|Y)} (\Gamma_X)^{-1/2}$ est lui-même de rang fini et l'espace propre engendré par ses vecteurs propres associés aux valeurs propres non nulles est identique à celui celui de son inverse généralisé. Le problème qui se pose ici est que cela nécessite la connaissance *a priori* de la dimension d de l'espace EDR. Si on note $(\Gamma^n_{E(X|Y)})^{+d}$ l'inverse généralisé de $\Gamma^n_{E(X|Y)}$ tronqué aux d plus grandes valeurs propres, l'espace EDR est estimé à partir de la diagonalisation de la matrice $((\Gamma^n_X)^{1/2} (\Gamma^n_{E(X|Y)})^{+d} (\Gamma^n_X)^{1/2})^+$.

Cette méthode a été proposée par [Ferré and Yao, 2005] dans le cadre de la régression inverse lorsque Y est une variable aléatoire réelle et les auteurs établissent la consistance de l'estimateur de l'espace EDR sous des hypothèses assez faibles. Ces résultats s'entendent là encore à notre contexte. L' avantage principal de cette approche est qu'elle est particulièrement simple à mettre en oeuvre. Cependant, elle nécessite l'estimation de d pour calculer $(\Gamma^n_{E(X|Y)})^{+d}$. Si dans [Ferré and Yao, 2005], un critère lié à l'estimation de $(\Gamma^n_{E(X|Y)})$ est proposé, nous estimons ici d à partir d'un critère prenant en compte l'ensemble de la procédure. Nous développerons ce point dans les applications.

4.3.3 Une approche par régularisation

Les méthodes de régularisation sont très communes pour le traitement des données fonctionnelles. Elles sont présentées comme plus efficaces que les méthodes de filtrage. L'idée principale est de pénaliser l'opérateur de covariance en introduisant des contraintes de régularité sur les fonctions estimées par modification du critère optimisé lors de l'estimation.

Ici, rappelons-le, la première direction, b_1, est la solution du problème $max \frac{<\Gamma_{E(X|Y)}b,b>}{<\Gamma_X b,b>}$, la deuxième, b_2, celle de $max \frac{<\Gamma_{E(X|Y)}b,b>}{<\Gamma_X b,b>}$, sous la contrainte $< \Gamma_X b_1, b >= 0$, etc...

Pour prendre en compte les contraintes de régularité, nous introduisons un paramètre de lissage λ et nous considérons la procédure suivante :

- déterminer $\hat{b}_1$ tel que $\hat{b}_1$ maximise le critère $\frac{<\Gamma^n_{E(X|Y)}b,b>}{<\Gamma^n_X b,b>+\lambda<D^2b,D^2b>}$;
- puis déterminer $\hat{b}_2$ tel que $\hat{b}_2$ maximise $\frac{<\Gamma^n_{E(X|Y)}b,b>}{<\Gamma^n_X b,b>+\lambda<D^2b,D^2b>}$, sous la contrainte $< \Gamma^n_X b, \hat{b}_1 > +\lambda < D^2 b, D^2 \hat{b}_1 >= 0$;
- les autres vecteurs s'obtiennent par optimisation du critère sous des contraintes semblables.

La solution à ce problème est donnée par $(\hat{b}_1, ..., \hat{b}_d)$, vecteurs propres associés aux d plus grandes valeurs propres de la matrice $(\Gamma^n_X + \lambda D^4)^{-1}\Gamma^n_{E(X|Y)}$ et vérifiant $< (\Gamma^n_X + \lambda D^4)\hat{b}_i, \hat{b}_j >= \delta_{ij}$. La matrice D^4 est un estimateur de l'opérateur de différentiation d'ordre 4 et est calculée à partir d'une base de fonctions splines.

La consistance de ces estimateurs a été démontrée dans [Ferré and Villa, 2005b] pour une variable réelle, mais elle reste également valable dans le cadre étudié ici.

4.4 Régle de classification

Dans le modèle (4.2), f est une fonction de lien définie de R^d dans R^J. Elle s' écrit $f = (f_1, ..., f_J)$ où, pour chaque u dans R^d et chaque j, $f_j(u) = P(C = j|U = u)$, avec U est la projection de X sur l'espace EDR et $f(x) = E(Y|X = x)$. Une fois l'espace EDR estimé, le problème de l'estimation de f se ramène à un problème de régression multivariée et l'estimation des probabilités d'appartenance aux groupes sachant X va s'obtenir par régression non paramétrique.

Toute méthode non paramétrique peut être, bien sûr, utilisée, mais nous emploierons ici des estimateurs à noyaux. On considère l'estimateur de Nadaraya-Watson :

$$\hat{f}_j(u) = \frac{\Sigma^n_{i=1} Y_i^{(j)} \Pi^d_{k=1} K(\frac{<b_k, X_i> - u}{h})}{\Sigma^n_{i=1} \Pi^d_{k=1} K(\frac{<b_k, X_i> - u}{h})} \tag{4.5}$$

où K est un noyau vérifiant $\int K(v)dv = 1$ et h est la fenêtre.

En fait, ce critère est une version non paramétrique de la règle de Bayes appliquée aux données projectées sur l'espace EDR. En effet, $\frac{\Sigma^n_{i=1} Y_i^{(j)} \Pi^d_{k=1} K(\frac{<b_k, X_i> - u}{h})}{nh}$ est l'estimateur à noyau de la densité de $(< b_1, X >, ..., < b_d, X >)$ conditionnelle à $Y^{(j)}$ et $\frac{\Sigma^n_{i=1} \Pi^d_{k=1} K(\frac{<b_k, X_i> - u}{h})}{nh}$ est l'estimateur de la densité conjointe de $(< b_1, X >, ..., < b_d, X >)$. Ainsi, la règle de classification conduit à maximiser en j un estimateur non paramétrique de $P(Y^{(j)}|(< b_1, X >, ..., < b_d, X >))$ dans l'esprit de ceux présentés dans [Devroye *et al.*, 1996].

L'utilisation d'estimateurs à noyau peut s'avérer, dans le cas de groupes trop nombreux, inefficace en raison du "fléau de la dimension". Si tel est le cas, il est possible de remplacer ces estimateurs à noyau par des réseaux de neurones, insensibles à ce problème, et dont la

pertinence de leur association avec les méthodes de régression inverse est démontrée dans [Ferré and Villa, 2005b]. Cependant dans la pratique, le nombre de groupes est généralement raisonnablement faible ce qui explique le choix effectué ici.

4.5 Applications

4.5.1 Méthode

Cette partie est consacrée à la mise en oeuvre des méthodes ci-dessus et à leur comparaison avec des méthodes concurrentes. Que les données soient réelles ou simulées, le mode opératoire présenté ci-après leur sera commun. Nous désignerons par SIR-Np la méthode par projection, SIR2-N celle par inverse généralisé et SIR-Nr celle par régularisation. Nous les comparerons avec RPDA (Ridge Penalized Discriminant Analysis) de [Hastie *et al.*, 1995] et avec la méthode à noyaux de [Ferraty and Vieu, 2003] désignée ici par NPCD-PCA.

Pour obtenir les estimateurs en pratique, nous proposons de diviser l'échantillon en trois et de procéder de la façon suivante :

- tout d'abord estimer l'espace EDR sur le premier échantillon, dit échantillon d'apprentissage ;
- puis, déterminer la fenêtre et les paramètres des différents modèles par validation croisée sur un échantillon de contrôle ;
- enfin de déterminer le pourcentage de mal classés sur un échantillon test.

Cette façon de procéder présente les avantages suivants : tout d'abord, les deux premières étapes s'effectuant sur des échantillons indépendants, il est possible d'utiliser les résultats de convergence des estimateurs à noyau pour obtenir la convergence de l'estimateur de f. Ensuite, il est possible de considérer la dimension d comme un paramètre du modèle pour l'approche 2 et de réitérer la procédure avec différentes valeurs de d pour retenir celle qui minimise le taux de mal classés. Le paramètre d est naturellement borné, par le rang de l'opérateur $\Gamma_{\mathbb{E}(X|Y)}$, c'est-à-dire, par $J-1$.

Dans le cadre de notre étude, nous avons, tout d'abord et comme décrit ci-dessus, déterminé par validation croisée les paramètres de chaque méthode. Ces paramètres (λ, d, h, k_n ou k_N, selon la méthode) ont été calés dans un premier temps afin d'éviter les explosions de temps de calcul liés au volume important des données. Pour étudier la variabilité des résultats, nous considérons ensuite cinquante segmentations de l'échantillon en deux parties par répartition aléatoire. En utilisant les paramètres calculés précédemment, le premier échantillon conduit à estimer les espaces de projection (espace EDR pour les SIR, espace discriminant pour la RPDA ou espace principal pour la NPCD-PCA) tandis que le second sert à mettre en œuvre la règle de classement (maximisation de la probabilité d'appartenance calculée par estimateur à noyau pour SIR et NPCD-PCA ou minimisation de la distance aux groupes pour la RPDA) et donc à déterminer le taux de mal classés. Grâce aux 50 répétitions, nous obtenons ainsi une distribution empirique de ce taux.

4.5.2 Données simulées : les "waveform data"

Nous considérons ici un jeu de données simulées qui est une sorte d'étalon pour la comparaison des méthodes de discrimination fonctionnelle.

La base de donnnées est composée de 3000 courbes discrétisées en 21 points ($t = 1, 2, \ldots, 21$) qui sont issues de 3 familles différentes (1000 courbes par classes) :

1. $t \to uh_1(t) + (1-u)h_2(t) + \epsilon(t)$;
2. $t \to uh_1(t) + (1-u)h_3(t) + \epsilon(t)$;

3. $t \rightarrow uh_2(t) + (1-u)h_3(t) + \epsilon(t)$

où u est une variable aléatoire de loi uniforme sur $[0;1]$, $\epsilon(t)$ est une variable aléatoire de loi normale centrée réduite et

$$h_1(t) = max(6 - |t-11|, 0),\ h_2(t) = h_1(t-4) \text{ et } h_3(t) = h_1(t+4)\ .$$

Des représentations de ces courbes sont données en Figure 4.1.

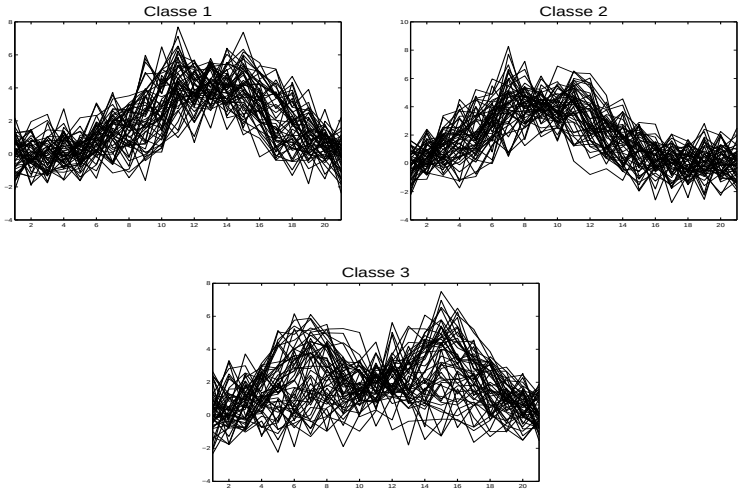

FIG. 4.1 – Un échantillon de 50 courbes par classe

La base de données a été partagée en deux de manière aléatoire : un échantillon de 1500 individus (500 par classe) constituait la base d'apprentissage et un échantillon de 1500 individus (500 par classe) la base de test.

Les valeurs optimales pour chacunes des méthodes employées sont données dans le Tableau 4.1.

	Paramètre 1	Paramètre 2	Paramètre 3
SIR-Nr	$\lambda = 1$ (régularisation de Γ_X)	$d = 2$ (dimension SIR)	$h = 0{,}75$ (fenêtre du noyau)
SIR-Np	$k_n = 2$ (dimension ACP)	$d = 2$ (dimension SIR)	$h = 3$ (fenêtre du noyau)
SIR2-N	$d = 2$ (dimension SIR)	$h = 3$ (fenêtre du noyau)	
RPDA	$\lambda = 2$ (régularisation de Γ_X)	$d = 2$ (dimension AFD)	
NPCD-PCA	$k_N = 16$ (dimension ACP)	$h = 6$ (fenêtre du noyau)	

TAB. 4.1 – Valeurs optimales

Les fonctions qui engendrent de l'espace EDR et l'espace dicriminant dans la RPDA sont données Figure 4.2. Ces fonctions propres sont très proches en ce qui concerne SIR2-N et

SIR-Np et toutes deux sont assez voisines de celles de la RPDA. On peut observer aussi que seule SIR-Nr conduit à des fonctions lisses ce qui est cohérent avec cette méthode.

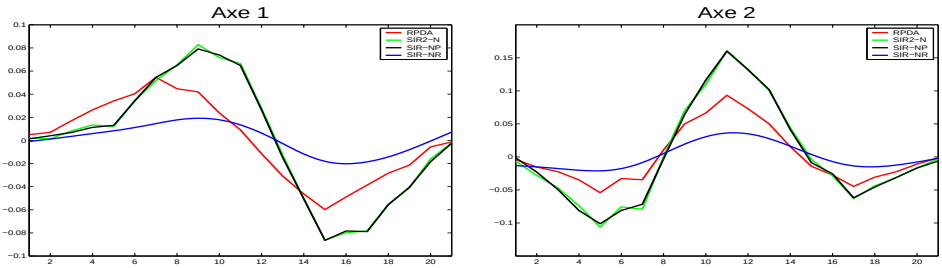

FIG. 4.2 – Vecteurs de l'espace EDR

La projection du nuage de points sur ces espaces est donnée en Figure 4.3. Il est difficile

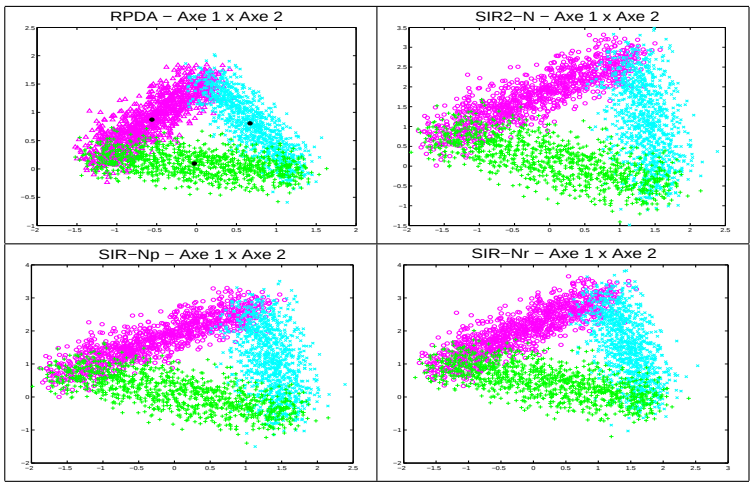

FIG. 4.3 – Projection des données sur l'espace EDR

de distinguer une différence entre les différents graphiques laissant ainsi présager de performances comparables dans les affectations. C'est en effet ce que l'on observe à la Figure 4.4 qui donne les boîtes à moustaches des taux d'erreurs de classements construites à partir des cinquante simulations. On peut constater que la méthode RPDA est celle qui donne les plus mauvais résultats alors que les meilleurs sont obtenus par SIR2-N. La relative médiocrité des résultats de SIR-Nr s'explique sans doute par la difficulté d'interpoler convenablement les 21 points de discrétisation sur une base spline. Le Tableau 4.2 donne les caractéristiques du taux de mal classés. On peut y remarquer que les méthodes reposant sur la régression inverse fonctionnelle fournissent globalement des résultats qui les situent parmi les méthodes les plus performantes si on les compare avec ceux d'études similaires ([Hernandez and Velilla, 2001] indiquent que leur méthode RKDA a un taux d'erreur de 16,2 % et que le meilleur taux est obtenu pour un réseau de neurones avec 15,1 %).

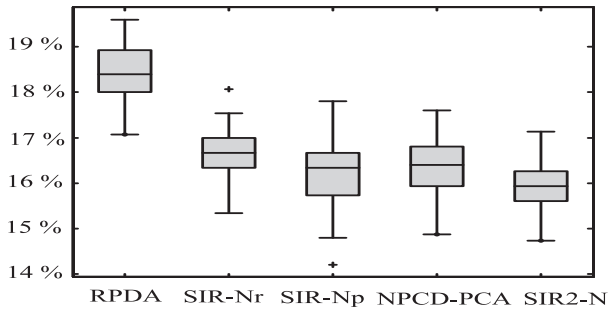

FIG. 4.4 – Comparaison des taux d'erreur pour 50 échantillons

	Moyennes	Médianes	Ecart type	1° quartile	Minimum
SIR-Nr	16,62 %	16,67 %	0,63 %	16,33 %	15,33 %
SIR-Np	16,15 %	16,33 %	0,77 %	15,73 %	14,20 %
SIR2-N	15,92 %	15,93 %	0,55 %	15,60 %	14,73 %
RPDA	18,38 %	18,40 %	0,68 %	18,00 %	17,07 %
NPCD-PCA	16,37 %	16,40 %	0,66 %	15,93 %	14,87 %

TAB. 4.2 – Caractéristiques des taux de mal classés

4.5.3 Reconnaissance de phonèmes

La base de données est composée de 4509 log-périodogrammes (discrétisés en 256 points) qui correspondent aux enregistrements de 5 phonèmes différents dont des représentations

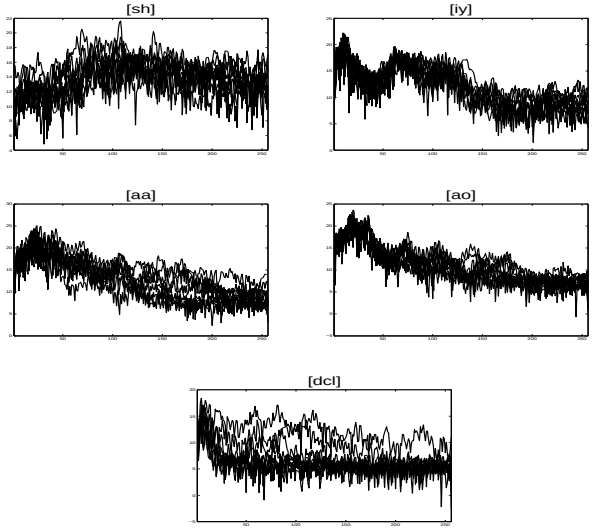

FIG. 4.5 – Un échantillon de 10 log-périodogrammes par classe

sont données en Figure 4.5. Les phonèmes enregistrés sont [sh] (872 log-périodogrammes), [iy] (1163 log-pério-dogrammes), [dcl] (757 log-périodogrammes), [aa] (695 log-périodogrammes) et [ao] (1022 log-périodogrammes).

La base de données a été partagée en deux de manière aléatoire : un échantillon de 1735 individus (347 par classe) constituait la base d'apprentissage et un échantillon de 1735 individus (347 par classe) la base de test sur laquelle était calculée l'erreur correspondant à chaque méthode.

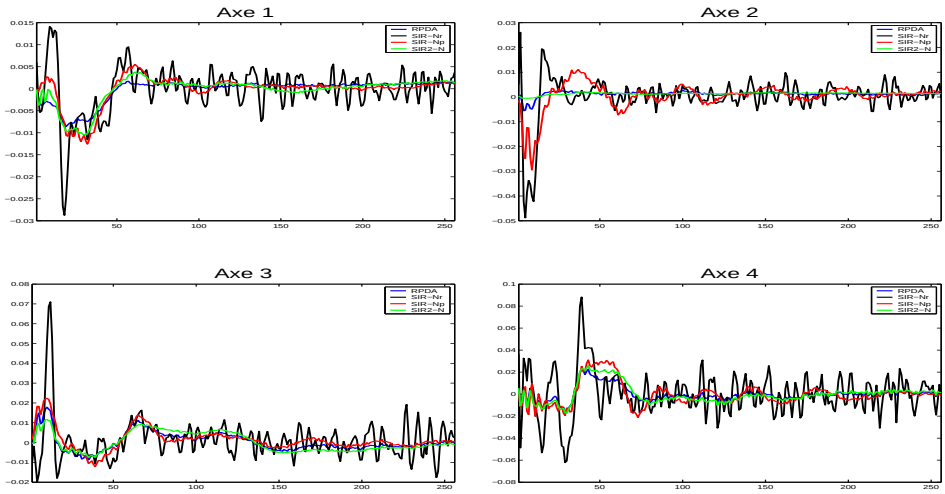

FIG. 4.6 – Vecteurs de l'espace EDR pour les phonèmes.

Les vecteurs engendrant l'espace EDR sont représentés Figure 4.6. Les solutions les moins lisses sont fournies par la méthode SIR-Nr (en raison d'une faible valeur du paramètre optimal de régularisation) alors que les plus lisses sont obtenues par RPDA et SIR2-N. Les solutions fournies par les diverses méthodes testées sont très proches les unes des autres.

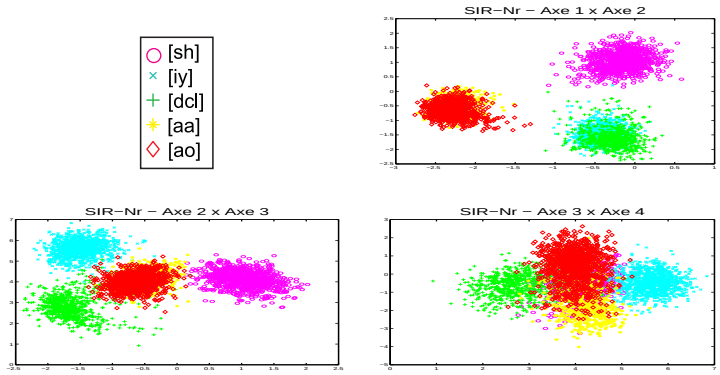

FIG. 4.7 – Projection des données (SIR-Nr)

Les Figures 4.7, 4.8, 4.9 et 4.10 permettent de visualiser la projection des données sur l'espace EDR fournie, respectivement, par les méthodes SIR-Nr, SIR-Np, SIR2-N et RPDA.

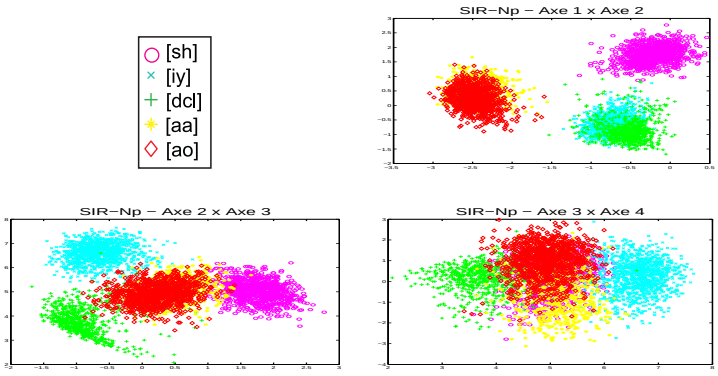

FIG. 4.8 – Projection des données (SIR-Np)

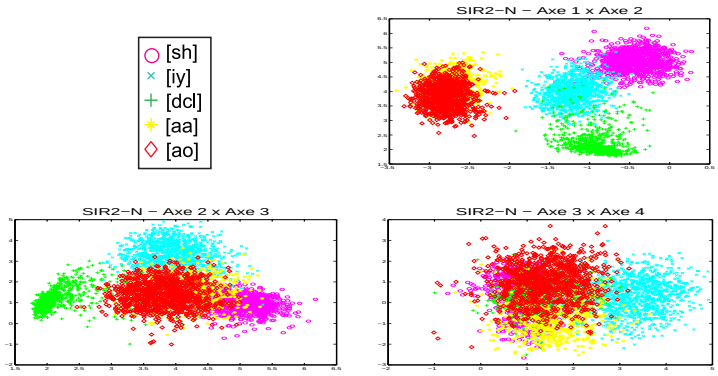

FIG. 4.9 – Projection des données (SIR2-N)

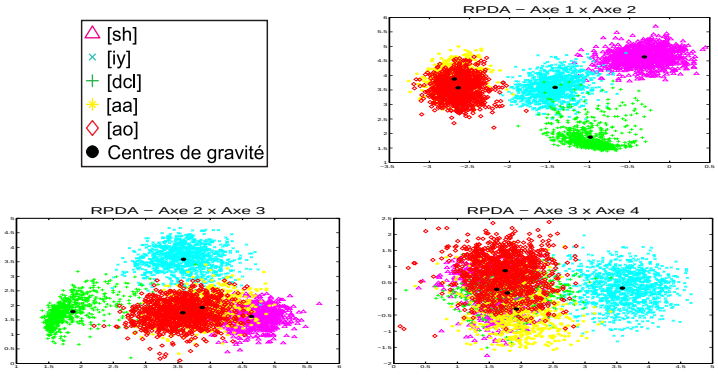

FIG. 4.10 – Projection des données (RPDA)

Elles font apparaître une bonne séparabilité linéaire des données projetées, laissant penser que le simple modèle RPDA fournira des taux d'erreur très satisfaisants. Concernant les méthodes SIR2-N et RPDA, les axes 1 et 2 permettent de séparer les phonèmes [sh] (en

haut à droite), [ly] (au centre), [dcl] (en bas) et les phonèmes en "a", [aa] et [ao] qui eux sont confondus (à gauche). A l'inverse, SIR-Nr et SIR-Np, ne permettent pas de séparer [ly] et [dcl] sur l'axe 2 : en deux dimensions, SIR2-N et RPDA sont les plus discriminantes. Si l'axe 3 donne des résultats comparables d'une méthode à l'autre, en séparant bien [ly] et [dcl], l'axe 4 est le seul qui permet de discriminer les phonèmes [aa] et [ao]. La discrimination y est meilleure pour SIR-Nr et SIR-Np que pour RPDA et SIR2-N, ce qui explique que, globalement, les premières ont des performances supérieures aux secondes comme l'indiquent le Tableau 4.3 et la Figure 4.11.

	Moyennes	Médianes	Ecart type	1° quartile	Minimum
SIR-Nr	8,24 %	8,30 %	0,44 %	7,95 %	7,32 %
SIR-Np	8,45 %	8,44 %	0,51 %	8,01 %	7,32 %
SIR2-N	9,33 %	9,48 %	0,47 %	8,99 %	8,36 %
RPDA	9,04 %	9,05 %	0,52 %	8,65 %	7,84 %
NPCD-PCA	9,89 %	9,83 %	0,60 %	9,39 %	8,47 %

TAB. 4.3 – Caractéristiques des taux de mal classés pour les phonèmes.

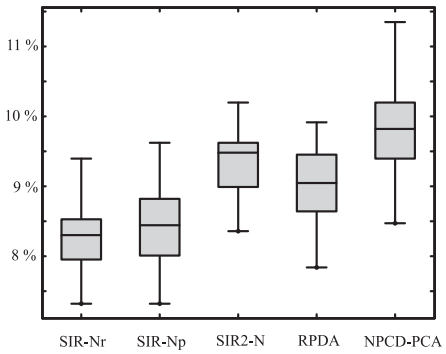

FIG. 4.11 – Comparaison des taux d'erreur pour 50 échantillons

4.6 Conclusion

Dans le cadre de la discrimination ou dans celui de la régression, le modèle (4.2) permet de projeter la variable explicative sur un espace "exhaustif" et les résultats présentés ici démontrent l'apport d'une telle démarche en comparaison aux approches plus classiques. On voit ainsi que les méthodes de régression inverse se comportent mieux que la méthode NPCD-PCA basée sur une projection sur l'espace des vecteurs principaux. La comparaison des méthodes SIR entre elles montre que les méthodes par projection, SIR2-N et celle par régularisation, SIR-Nr donnent ici des résultats très voisins. Cela a été confirmé par d'autres études. Notons que la première donne d'assez bons résultats dans les deux exemples ci-dessus puisque toujours en seconde position. La seconde, elle, semble moins performante dans le premier exemple en raison du faible nombre de points de discrétisation, mais elle se révèle la meilleure dès lors que le caractère fonctionnel des données apparaît plus clairement (256

points de discrétisation dans l'exemple des phonèmes). Notons, enfin pour terminer, qu'il est envisageable d'améliorer les résultats obtenus en faisant d'autres choix d'estimateurs de la fonction de lien.

Chapitre 5

Multi-Layer Neural Network with functional inputs : an inverse regression approach

Louis Ferré
GRIMM, Equipe d'accueil 3686, Université Toulouse Le Mirail, France
Nathalie Villa
GRIMM, Equipe d'accueil 3686, Université Toulouse Le Mirail, France

Référence : Multi-Layer Neural Network with functional inputs : an inverse regression approach (2005), *à paraître dans Scandinavian Journal of Statistics.*

Abstract:

Functional data analysis is a growing research field since more and more pratical applications involve functional data. In this paper, we focus on the problem of regression and classification with functional predictors: the model suggested combines an efficient dimension reduction procedure (functional SIR, first introduced by [Ferré and Yao, 2003]), for which we give a regularized version, with the accuracy of a neural network. Some consistency result are given and the method is successfully confronted to real life data.
Keywords: *classification, dimension reduction, functional data analysis, multi-layer perceptron, prediction.*

5.1 Introduction

Functional regression is now a very important part of statistics as functional variables occur frequently in practical applications. We present two examples that take place in this area. First, we face a regression problem where the regressor are curves (see Figure 5.1): the Tecator data problem consists in predicting the fat content of pieces of meat from a near absorbance spectrum. This data set has already been studied by [Thodberg, 1996] and [Ferré and Yao, 2003]. Secondly, in the phoneme data set, the data are log-periodograms of a 32 ms duration corresponding to recorded speakers and we expect to determine which

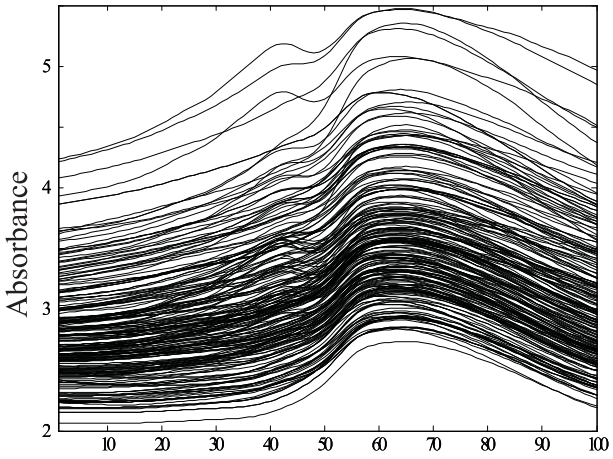

Figure 5.1: The regressor curves

one of the five phonemes, [sh] as in "she", [dcl] as in "dark", [iy] as in "she", [aa] as in "dark" and [ao] as in "water", corresponds to this recording. It has already been described by [Hastie *et al.*, 1995] and by [Ferraty and Vieu, 2003]. Clearly, here, functional data is also involved but we face now a classification problem. However, we will see that both - regression and classification - can be tackled via a common modelling.

An extensive review of the numerous studies developped for functional data analysis can be found in [Ramsay and Silverman, 1997] including regression and classification but also many factorial methods. A particularity of functional regression is that it leads to ill-posed problems because of the infinite dimension of the feature space. Then original solutions have been introduced to overcome this problem for the functional linear regression, see e.g. [Cardot *et al.*, 1999]. At the same time, [Dauxois *et al.*, 2001] and then [Ferré and Yao, 2003], [Ferré and Yao, 2005] have proposed a semi-parametric model for Hilbertian variables which corresponds to the functional version of Li's Sliced Inverse Regression, [Li, 1991].

On a classification point of view, many solutions have been proposed to overcome ill-posed functional problems including the popular penalization methods. [Friedman, 1989] presents the RDA model based on regularization and shrinkage. [Hastie *et al.*, 1994] and [Hastie *et al.*, 1995] propose a discriminant analysis penalized by smoothing functionals. The idea of penalization was first developped by [Ivanov, 1962] and [Tihonov, 1963b], [Tihonov, 1963a] and it has been used by [Pezzulli and Silverman, 1993] and [Silverman, 1996] for smoothed Principal Components Analysis and by [Leurgans *et al.*, 1993] for Canonical Correlation Analysis. Finally, a review of many regularization methods can be found in [Tenorio, 2001].

Nonlinear methods for functional data analysis have also been developped: for instance, neural networks models ([Rossi and Conan-Guez, 2005a] for multilayer perceptrons, [Rossi *et al.*, 2004] for the SOM algorithm), k-nearest neighbour models ([Biau *et al.*, 2005]) or nonparametric estimators ([Ferraty and Vieu, 2002])

In this paper, we propose a new way to achieve functional regression: the idea is to join the efficiency of a dimension reduction method using smoothing penalization, to the strong adaptability of a neural network which can provide highly non linear solutions even if the number of predictors is too large for classical nonparametric methods such as kernels

smoothing. The functional SIR dimension reduction method is first presented in Section 5.2. For this penalized version, consistency results are given in Section 5.3. Section 5.4 discusses Neural Network and gives consistency results for the proposed model combining FSIR and Neural Networks (which will be called SIR-NNr). Section 5.5 is devoted to applications: Section 5.5.1 deals with the Tecator data set and Section 5.5.2 with the phoneme data set. In Appendix 5.6, we give a sketch of the proofs. All programs have been made using Matlab and are available on request.

5.2 Sliced Inverse Regression

Let Y be a real random variable and X be a multivariate variable assumed to have a fourth moment. To overcome the curse of dimensionality in the nonparametric regression of Y on X, [Li, 1991] introduced the Sliced Inverse Regression. He considers the following model

$$Y = f(a_1'X, a_2'X, \ldots, a_q'X, \epsilon),$$

where ϵ is centered and independent of X, f is an unknown function and $(a_j)_{j=1,\ldots,q}$ are lineary independent vectors.

The space spanned by $(a_j)_{j=1,\ldots,q}$ is called EDR (Effective Dimension Reduction) space. SIR deals with the estimation of this EDR space and the aim of sliced inverse regression is to estimate it by means of the eigenvectors of the matrix $Var(X)^{-1}Var(E(X|Y))$.

In the multivariate context, numerous works deal with SIR. [Li, 1991], [Schott, 1994], [Ferré, 1998] and [Velilla, 1998] have worked to determine the dimensionality q. Then, methods have been proposed to improve SIR: different estimates of the covariance of the conditional mean have been built (in [Hsing and Carroll, 1992], [Zhu and Ng, 1995] and [Zhu and Fang, 1996]) while other methods have been proposed to estimate the EDR space (for example, PHD proposed by [Li, 1992], SAVE by [Cook and Weisberg, 1991] or MAVE by [Xia *et al.*, 2002]). The main interest of this model is that, once the EDR space is estimated, the estimation of f is obtained very easily with traditional techniques provided that q is not too large.

5.2.1 Functional SIR

Now consider a real random variable Y and X a random variable taking its values in $\mathcal{L}^2_{\mathcal{T}}$, the space of squared intregrable functions from a compact interval $\mathcal{T}$ into $\mathbb{R}$. With the usual inner product defined by, for all f, g in $\mathcal{L}^2_{\mathcal{T}}$, $< f, g >= \int_{\mathcal{T}} f(t)g(t)dt$, $\mathcal{L}^2_{\mathcal{T}}$ is a Hilbert space. We will assume that the random variable X is centered, without loss of generality, and has a fourth moment. Then, the covariance operator of X exists and is defined by $\Gamma_X = E(X \otimes X)$ where $X \otimes X$ denotes the operator which associates to any f in $\mathcal{L}^2_{\mathcal{T}}$, $< f, X > X$. We also get that $E(X/Y)$ and $\Gamma_{E(X|Y)} = Var(E(X|Y))$ exist. Ferré and Yao (2003) have proposed to investigate the following model for functional inverse regression:

$$Y = f(< X, a_1 >, \ldots, < X, a_q >, \epsilon) \tag{5.1}$$

where f is an unknown function, ϵ a random variable which is centered and independent of X and $(a_j)_{j=1,\ldots,q}$ are lineary independent functions of $\mathcal{L}^2_{\mathcal{T}}$.

The crucial point of functional SIR is that, unlike the multivariate case, Γ_X^{-1} is not defined since we have to assume that Γ_X is a positive definite operator which implies that it is not invertible as defined from $\mathcal{L}^2_{\mathcal{T}}$ to $\mathcal{L}^2_{\mathcal{T}}$. However, if we call $(\delta_i)_{i=1,\ldots,\infty}$ its sequence of eigenvalues and $(u_i)_{i=1,\ldots,\infty}$ those of orthonormed eigenvectors, R_Γ the range of Γ_X and

$R_\Gamma^{-1} = \left\{ h \in \mathcal{L}_\tau^2 : \exists f \in R_\Gamma, h = \sum_i \frac{1}{\delta_i}(u_i \otimes u_i)(f) \right\}$, Γ_X is a one-to-one mapping from R_Γ^{-1} to R_Γ whose inverse, also called Γ_X^{-1}, is defined by $\Gamma_X^{-1} = \sum_i \frac{1}{\delta_i} u_i \otimes u_i$.

We focus on the estimation of the EDR space spanned by the vectors $(a_j)_{j=1,\dots,q}$. Now, the key of the method comes from the following theorem:

Theorem 5.1. *Writing* $A = (< X, a_1 >, \dots, < X, a_q >)^T$, *if*

(A1) *for all* u *in* $\mathcal{L}_\mathcal{T}^2$ *there exists* v *in* $\mathbb{R}^q$ *such that:* $E(< u, X > /A) = v^T A$

then $E(X/Y)$ *belongs to the subspace spanned by* $\Gamma_X a_1, \dots, \Gamma_X a_q$.

Remark: Note that [Cook and Weisberg, 1991] show that elliptically distributed variables satisfy condition **(A1)** in the multidimensional context but this can be transposed in infinite dimensional Hilbert spaces.

By using the result of [Dauxois *et al.*, 2001], a consequence of Theorem 5.1 is that the EDR subspace contains the Γ_X-orthonormed eigenvectors of $\Gamma_X^{-1}\Gamma_{E(X/Y)}$ associated with the q positive eigenvalues. Then, in the following, $(a_j)_{j=1,\dots,q}$ will denote these eigenvectors. This is the generalization of [Li, 1991] on SIR to infinite dimensional case.

A basis of the EDR space is thus given by the eigenvector of $\Gamma_X^{-1}\Gamma_{E(X/Y)}$ but to ensure that these eigenvectors exist in $\mathcal{L}_\mathcal{T}^2$, we have to assume that (see [Ferré and Yao, 2005] for details) $\sum_i \sum_j \frac{1}{\delta_i \delta_j} E(E(\zeta_i/Y)E(\zeta_j/Y))^2 < +\infty$, where $X = \sum_i \zeta_i u_i$ is the Karhunen-Loève decomposition of X.

Let $\{(X^n, Y^n)\}_{n=1,\dots,N}$ be an i.i.d. sample. Thus, in order to estimate the EDR space, we have to choose an estimate for $\Gamma_{E(X/Y)}$. We have two possibilities:

1. A slicing approach. In [Ferré and Yao, 2003], the estimate is obtained by partitionning the domain of Y in $\{I_h\}_{h=1,\dots,H}$:

$$\Gamma_{E(X/Y)}^N = \sum_{h=1}^{H} \frac{N_h}{N} \mu_h \otimes \mu_h - \overline{X} \otimes \overline{X}$$

   where, if $\mathbb{I}$ is the indicator function, $N_h = \sum_{n=1}^N \mathbb{I}_{\{Y^n \in I_h\}}$,
   $\mu_h = \frac{1}{N_h} \sum_{n=1}^N X^n \mathbb{I}_{\{Y^n \in I_h\}}$ and $\overline{X}$ is the empirical mean.

2. A kernel based approach. In [Ferré and Yao, 2005], it is assumed that Y has a probability density; thus a kernel estimate (of the Nadaraya-Watson type) is used:

$$\widehat{E(X/Y = y)} = \sum_{n=1}^{N} \frac{X^n K\left(\frac{Y^n - y}{h}\right)}{\sum_{m=1}^N K\left(\frac{Y^m - y}{h}\right)}$$

   and $\Gamma_{E(X/Y)}^N = \frac{1}{N} \sum_{n=1}^N \widehat{E(X/Y = Y^n)} \otimes \widehat{E(X/Y = Y^n)} - \overline{X} \otimes \overline{X}$.

A usual estimate of Γ_X is $\Gamma_X^N = \frac{1}{N} \sum_{n=1}^N X^n \otimes X^n - \overline{X} \otimes \overline{X}$, but this estimate is ill conditionned (because Γ_X^{-1} is not a bounded operator) so the eigenvectors of $(\Gamma_X^N)^{-1}\Gamma_{E(X/Y)}^N$ do not converge to the eigenvectors of $\Gamma_X^{-1}\Gamma_{E(X/Y)}$. That is the reason why penalization or regularization is needed.

[Ferré and Yao, 2003] suggest to proceed like [Bosq, 1991] by considering, instead of Γ_X, a sequence of finite rank operators with bounded inverses and converging to Γ_X. This leads

to the estimates $(a_j^N)_{j=1,...,q}$ of $(a_j)_{j=1,...,q}$ that, under some conditions, satisfy the following consistency result:

$$\| a_j^N - a_j \| \rightarrow_p 0$$

The authors also suggest a way of estimating the EDR space for functional data without inverting the covariance operator of the regressor ([Ferré and Yao, 2005]).

We propose, in Section 5.3, a regularized approach by penalization.

5.2.2 SIR for classification

Let $\mathcal{C}_1, \ldots, \mathcal{C}_H$ be H groups. When Y is multidimensional, the results of [Dauxois *et al.*, 2001] are still available and by setting $Y = (\mathbb{I}_{\mathcal{C}_1}, \ldots, \mathbb{I}_{\mathcal{C}_H})$, where $\mathbb{I}_{\mathcal{C}_h}$ is the indicator function of the hth group, Model (5.1) remains valid and we get a natural way to include classification problems into FSIR, see [Ferré and Villa, 2005a]. Note that, in the functional case, multivariate methods for discrimination have been extended, mainly inspired from Linear Discriminant Analysis (LDA). In this area, let us mention the works of [Hastie *et al.*, 1994], [Hastie *et al.*, 1995] and [James and Sugar, 2003].

Now, by estimating $\Gamma_{E(X/Y)}$ by

$$\Gamma_{E(X/Y)}^N = \frac{1}{N} \sum_{h=1}^{H} N_h \widehat{E(X/Y = h)} \otimes \widehat{E(X/Y = h)} - \overline{X} \otimes \overline{X}$$

where $N_h = \sum_{n=1}^N \mathbb{I}_{\{Y^n = h\}}$ and $\widehat{E(X/Y = h)} = \frac{1}{N_h} \sum_{n=1}^N X^n \mathbb{I}_{\{Y^n = h\}}$, FSIR leads to a discriminant analysis. The estimation of the EDR space is identical to the discriminant space in linear discriminant analysis. However, the estimation of f leads to a natural classification rule. Indeed, since we have, for all x, $f(x) = E(Y|X = x) = (P(C_1|X = x), ..., P(C_H|X = x))$, the estimation of f coincides with the estimations of the probabilities of the groups conditionally to X.

5.3 Regularized functional SIR

In Section 5.2, we saw that the EDR space contains the eigenvalues of the operator $\Gamma_X^{-1}\Gamma_{E(X/Y)}$. Thus, as it is the case for Discriminant Analysis, the estimator of the first direction of the EDR space can be found by maximizing a Rayleigh criterion: $\max_a \frac{<\Gamma_{E(X/Y)}a, a>}{<\Gamma_X a, a>}$. Unfortunately, as Γ_X^N is ill conditionned, the maximization of the empirical Rayleigh expression does not lead to a good estimate of the EDR space: that is the reason why a regularization is needed.

Provided that we have smooth functions, a relevant method for functional data is to penalize the covariance operator in the Rayleigh expression by introducing smoothing constraints on the estimated functions. This method has already proved its great efficiency (see [Hastie *et al.*, 1995] for an example of the penalized discriminant analysis).

5.3.1 Main result

Let $\mathcal{S}$ be the subspace of $\mathcal{L}_{\mathcal{T}}^2$ of functions with a squared integrable second derivative. We introduce a penalty through a bilinear form defined on $\mathcal{S} \times \mathcal{S}$ by, for all f, g in $\in \mathcal{S}$,

$$[f, g] = \int_{\mathcal{T}} D^2 f(t) D^2 g(t) dt$$

We also define the penalized bilinear form associated with empirical operators Γ_X and Γ_X^N:

$$Q_\alpha(f,g) =< \Gamma_X f, g > +\alpha[f,g] \qquad \text{and} \qquad Q_\alpha^N(f,g) =< \Gamma_X^N f, g > +\alpha[f,g]$$

where α is a regularization parameter. The solutions of the regularized FIR are given by maximizing, under orthogonal constraints, the function

$$\gamma^N(a) = \frac{< \Gamma_{E(X/Y)}^N a, a >}{< \Gamma_X^N a, a > +\alpha[a,a]}.$$

In order to obtain consistency results for the estimate of $(a_j)_{j=1,\ldots,q}$, we make the following assumptions

(A2) $E(\| X \|^4) < +\infty$;

(A3) for all $\alpha > 0$,

$$\inf_{\|a\|=1,\ a\in\mathcal{S}} Q_\alpha(a,a) = \rho_\alpha > 0;$$

(A4) $\Gamma_{E(X/Y)}^N$ is a continuous operator which converges in probability to $\Gamma_{E(X/Y)}$ with $\sqrt{N}$ rate;

(A5) $\lim_{N\to+\infty} \alpha = 0$, $\lim_{N\to+\infty} \sqrt{N}\alpha = +\infty$;

(A6) $(a_j)_{j=1,\ldots,q}$ belong to $\mathcal{S}$ and verify, for all u such that $< \Gamma_X u, a_1 >= 0$ and that $< \Gamma_X u, u >= 1$,

$$< \Gamma_{E(X/Y)} u, u > \leq < \Gamma_{E(X/Y)} a_2, a_2 >= \lambda_2 < \lambda_1.$$

Since $\mathcal{S}$ is not a closed subset, γ^N could not reach a maximum on $\mathcal{S}$. However, the following result holds:

Theorem 5.2. *Under assumptions* **(A1)-(A6)**, *with probability converging to 1, the function γ^N reaches its maximum on $\mathcal{S}$ when N grows to $+\infty$.*
In this case, let then a_1^N be a vector of $\mathcal{S}$ for which γ^N is maximum and which is such that $< \Gamma_X a_1^N, a_1 >= 1$. Then,

$$< \Gamma_X (a_1^N - a_1), a_1^N - a_1 > \to_p 0 \ .$$

Remarks:

- For an understandable presentation, we introduce a particular type of penalization but previous results can be found for other regularization functionals satisfying the assumptions. For example, we can replace the bilinear form $[.,.]$ by another one which is similar to the one used in Ridge-PDA ([Hastie *et al.*, 1995]).
- Assumptions **(A2)**, **(A3)** and **(A4)** are technical assumptions that ensure the existence and convergence for $(a_j^N)_{j=1,\ldots,q}$: **(A2)** implies that Γ_X^N will converge to Γ_X at the $\sqrt{N}$ rate; we can find in [Leurgans *et al.*, 1993] conditions that involve **(A3)**. This assumption shows the purpose of regularization: it controls the scaling of Q_α and, thanks to **(A5)**, ensures that the denominator of γ^N doesn't go too fast to 0. Finally **(A5)** gives a way of choosing regularization parameter α (for pratical aspects see section 5.3.2).

5.3.2 Practical aspects

On a practical point of view, X has been observed at some points t_1, t_2, ..., t_D (for a understandable presentation, we suppose that these observations have been centered). The optimization of the penalized Rayleigh expression described in Section 5.3.1 can be performed by using, for example, B-Splines $(B_i)_i$ to parametrize a_1^N:

$$a_1^N(t) = \sum_i A_{1i} B_i(t) = A_1 B$$

where B is the matrix containing the values of $(B_i(t))_i$ at the points t_1, t_2, ..., t_D.
Similarly, the matrix of observations X can be written in the form of B-Splines:

$$X = CB$$

with $C = \begin{bmatrix} C^1 \\ \vdots \\ C^N \end{bmatrix}$. Let $B^{(2)}$ be the vector containing the values $D^2B(t)$. If we use the slicing estimate of $\Gamma_{E(X/Y)}$ for regression, we introduce,
for all $h = 1, \ldots, H$,

$$Y_h = \begin{bmatrix} \mathbb{I}_{\{Y^1 \in I_h\}} \\ \vdots \\ \mathbb{I}_{\{Y^N \in I_h\}} \end{bmatrix}.$$

Then the problem of maximizing γ^N is equivalent to maximizing

$$\frac{A' M_e A}{A' M_{X,\alpha} A}$$

where M_e is the estimator of $\Gamma_{E(X/Y)}$ obtained by the slicing approach :

$$M_e = \sum_{h=1}^{H} \frac{N_h}{N} BB'C'Y_h Y_h' CBB'$$

and where

$$M_{X,\alpha} = \frac{1}{N} BB'C'CBB' + \alpha B^{(2)\ \prime} B^{(2)}\ .$$

The first solution is the eigenvector, with $M_{X,\alpha}$-norm equal to 1, associated with the largest eigenvalue of the matrix $M_{X,\alpha}^{-1} M_e$. By pursuing the procedure under othogonality constraints, we get that the other solutions are the $M_{X,\alpha}$-orthonormal eigenvectors of $M_{X,\alpha}^{-1} M_e$.

If we deal with classification, the same procedure is achieved by letting

$$Y_h = \begin{bmatrix} \mathbb{I}_{\{Y^1 = h\}} \\ \vdots \\ \mathbb{I}_{\{Y^N = h\}} \end{bmatrix}.$$

Finally we have to find the optimal value for α. This can be done, if the sample is large enough (which is the case in the presented applications), by dividing it into two parts: we apply the previous procedure on the first part to find $(a_j^N)_j$ and evaluate the error committed by Model (5.1) on the second part; the best parameter is then chosen to minimize this error.

Remark : The estimation of $\Gamma_{E(X/Y)}$ can also be made by a kernel approach ; the efficiency of this approach can even be better than those we have with the slicing estimate (see [Ferré and Yao, 2005] for practical comparisons).

5.4 Neural network

5.4.1 Approximation by neural networks

After the EDR space is estimated, the goal is to get an estimation of the function f in (5.1): we propose to use a feedforward neural network with one hidden layer. This method (see, e.g., [Bishop, 1995] for a review on Neural Networks) is an alternative to nonparametric regressions if the dimension of the EDR space is too large. It has the advantage of working in any cases while nonparametric methods, such as kernel smoothing or splines smoothing, face the curse of dimensionality.

The main interest of neural networks is their ability to approximate any function with the desired precision (universal approximation); see, for instance, [Hornik, 1991], [Hornik, 1993] for the multivariate context and [Stinchcombe, 1999] and [Rossi *et al.*, 2002] in the infinite dimensional one.

5.4.2 A consistency result

Neural Network approximations of functionals in infinite dimensional spaces have been studied in [Chen and Chen, 1995], [Sandberg and Xu, 1996], [Rossi *et al.*, 2002], [Conan-Guez and Rossi, 2002] and [Rossi and Conan-Guez, 2005a]. Several strategies are available either by directly using the curves as inputs of the feedforward neural networks or by first projecting the data onto a classical functional basis (such as a spline basis, a Fourier basis, wavelets) or a basis derived from the PCA of X. This latter approach is used by [Thodberg, 1996].

Our approach is similar but, instead of projecting the data onto a fixed basis or a principal component basis, we project them onto the EDR space. The EDR space behaves as an efficient subspace for the regression of Y on X and it is a way to get a basis which takes into account the relationship between Y and X. In fact, the data are projected onto an estimation of the EDR space, so the accuracy of the projection and then the estimation of the optimal weights for the neural network also depend on how good the EDR space is estimated.

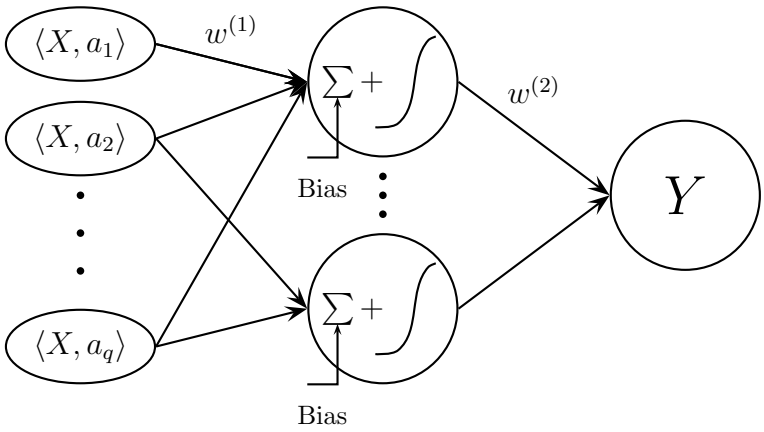

Figure 5.2: Neural network estimating f

We construct a perceptron (see Figure 5.2) with one hidden layer having

- as inputs, the coordinates of the projection of X onto $\text{Span}\{(a_j)_{j=1,\dots,q}\}$: $< X, a_1 >$, $\dots$, $< X, a_q >$;
- q_2 neurons on the hidden layer (where q_2 is a parameter to be estimated);
- as outputs, one neuron for regression and H neurons for classification, representing target Y.

The output of such a neural network is then $\sum_{i=1}^{q_2} w_i^{(2)} g\left(\sum_{j=1}^{q} w_{i,j}^{(1)} < X, a_j > + w_i^{(0)}\right)$ where g is the activation function (for example a sigmoid). The purpose of the training step is then to find w^* which minimizes a loss function L between the output of the neural network with weights $w = \left((w_i^{(2)})_{i=1,\dots,q_2}, (w_{i,j}^{(1)})_{i=1,\dots,q_2}^{j=1,\dots,q}, (w_i^{(0)})_{i=1,\dots,q_2}\right)$, and the target Y:

$$w^* = \arg\min \left\{ E\left[L\left(\sum_{i=1}^{q_2} w_i^{(2)} g\left(\sum_{j=1}^{q} w_{i,j}^{(1)} < X, a_j > + w_i^{(0)} \right), Y \right) \right] \right\}. \tag{5.2}$$

Actually we obtain an estimation w_N^* of w^* by

$$w_N^* = \arg\min \left\{ \sum_{n=1}^{N} L\left(\sum_{i=1}^{q_2} w_i^{(2)} g\left(\sum_{j=1}^{q} w_{i,j}^{(1)} < X^n, a_j^N > + w_i^{(0)} \right), Y^n \right) \right\}.$$

[White, 1989] gives a consistency theorem for the weights of a neural networks estimated by a set of iid observations. Since $(a_j^N)_j$ is an estimation of the EDR space deduced from the whole data set $(X^n, Y^n)_n$, the inputs of our functional perceptron used to determine w_N^* do not satisfy the iid assumption and a proper consistency result is then needed.

Let us introduce some notations: ζ is the function from $\mathcal{O} \times \mathcal{W}$ ($\mathcal{O}$ is an open set of $\mathbb{R}^{q+1}$ and $\mathcal{W}$ is a compact set of $\mathbb{R}^{(q+2)q_2}$) such as for all $z = (u, y)$ in $\mathcal{O}$, $\zeta(z, w) = L\left(\sum_{i=1}^{q_2} w_i^{(2)} g\left(\sum_{j=1}^{q} w_{i,j}^{(1)} u_j + w_i^{(0)}\right), y\right)$; Z is the couple of random variables $(\{< X, a_j >\}_j, Y)$ and $\{Z_n\}_{n=1,\dots,N}$ are observations of Z; finally, $(\tilde{Z}_N^n)_{n=1,\dots,N}$ are the couples of $(\{< X^n, a_j^N >\}_j, Y^n)$. In our context, the consistency of the Multilayer Perceptron is given by the following theorem:

Theorem 5.3. *Under assumptions* **(A1)-(A6)** *and the following assumptions*

(A7) *for all z in $\mathcal{O}$, $\zeta(z, .)$ is continuous;*

(A8) *there is a measurable function $\tilde{\zeta}$ from $\mathcal{O}$ into $\mathbb{R}$ such that, for all z in $\mathcal{O}$, for all w in $\mathcal{W}$, $|\zeta(z, w)| < \tilde{\zeta}(z)$ and $E(\tilde{\zeta}(Z)) < +\infty$;*

(A9) *for all w in $\mathcal{W}$, there exists $C(w) > 0$ such that, for all (x, y) and (x', y') in $\mathcal{O}$, $|\zeta((x, y), w) - \zeta((x', y), w)| \leq C(w) \parallel x - x' \parallel$*

(A10) *for all w in $\mathcal{W}$, $\zeta(., w)$ is measurable.*

If $\mathcal{W}^*$ is the set of minimizers of the problem (5.2) then

$$d(w_N^*, \mathcal{W}^*) \xrightarrow{N \to +\infty}_p 0,$$

where d is defined by: $d(w, \mathcal{W}) = \inf_{\tilde{w} \in \mathcal{W}} \parallel w - \tilde{w} \parallel$ with $\parallel . \parallel$ the usual euclidean distance.

Remarks:

- This list of assumptions is, for example, checked by a perceptron with one hidden layer and a sigmoid function $g(x) = \frac{e^x}{1+e^x}$ on the hidden layer associated with the mean squared error $L(\psi, y) = \| \psi - y \|^2$ provided Y has a second order moment.
- Assumptions **(A1)-(A6)** ensure the convergence of $(a_j^N)_{j=1,\dots,q}$ to $(a_j)_{j=1,\dots,q}$ but they can be replaced by a list of assumptions implying the same result. For example, we would have the same consistency result by projecting the data onto the estimated EDR space found by the functional SIR presented in [Ferré and Yao, 2003] and [Ferré and Yao, 2005].

5.5 Applications

5.5.1 Tecator data

As already said, the Tecator data problem consists in predicting the fat content of pieces of meat from a near infrared absorbance spectrum. We have $N = 215$ observations of (X, Y) where X is the spectrum of absorbance discretized at one hundred points and Y is the lipid rate.

In order to compute the procedure described in section 5.3.2, we project the data onto a cubic Spline basis. Because of their smoothness, these data are very well projected onto a basis with 40 knots (actually, up to 40 knots, the interpolation is exact); then, for simplicity, we used this projection for the computation when needed and used the original data in the other cases. We tried several classical methods in order to test the efficiency of SIR-NNr. The competitors are:

- **SIR-NNr**: the functional SIR regularized by penalization, presented in Section 5.3, precedes a neural network. The neural network training step is made by early stopping procedure: the learning sample is divided into 3 samples (training / validation / test); the training sample is used to train the neural network, the validation sample for the early stopping procedure and this training step is performed 10 times. The best performance of the test samples is kept as the optimal weights;
- **SIR-NNk**: here we use the smoothed functional inverse regression method presented in [Ferré and Yao, 2003] as pre-processing to a neural network; the purpose is to show the benefit of the regularization. The neural network is also trained by early stopping;
- **PCA-NN**: in order to show the advantage of SIR, we compute a principal component analysis (as [Thodberg, 1996]) before a neural network procedure is used (a classical neural network while Thodberg uses a sophisticated bayesian neural network);
- **NNf**: this method is the functional neural network (the Spline projections are used to represent the functional weights and inputs) described by [Rossi and Conan-Guez, 2005a]. In this paper, B-Spline basis projection is selected by cross-validation which leads to a huge computational time: we do not follow this approach and use the cubic basis with 40 knots.
- **SIR-L**: after projecting the data onto the EDR space determined by regularized SIR, we compute a linear regression in order to show the efficiency of a neural network compared to a classical parametric method.

We also have to notice that classical nonparametric methods such as kernel can not be used for this data set as the dimensionality of the EDR space is too large (the value of q is given in Table 5.1).

Before we compare the different methods and in order to limit computational time, we determined the best parameters for each one. Our sample is divided into two parts: on the first one, we determine the values of $(a_j^N)_j$ and of the weights of the neural network for various values of α, q and q_2. On the second part, we determine the standard error of prediction (SEP): the "best" parameters are those which minimize this SEP (see Table 5.1).

	Parameter 1	*Parameter 2*	*Parameter 3*
PCA-NN	$k_n = 25$ (PCA dimension)	$q_2 = 12$ (number of neurons)	
NNf	$q_2 = 18$ (number of neurons)		
SIR-NNr	$\alpha = 5$ (regularization of Γ_X)	$q = 20$ (SIR dimension)	$q_2 = 10$ (number of neurons)
SIR-NNk	$h = 0{,}5$ (kernel window)	$q = 10$ (SIR dimension)	$q_2 = 15$ (number of neurons)
SIR-L	$\alpha = 0{,}5$ (regularization of Γ_X)	$q = 20$ (SIR dimension)	

Table 5.1: Best parameters for the five compared methods

Then, in order to see not only the error made by each method but also its variability, we randomly build 50 samples divided as follows: the learning sample contains 172 observations and the test sample contains 43. All five methods are first trained on the learning sample (with their optimal parameters pre-determined as described above) and the standard error of prediction (SEP) is then performed on the test sample.

Figure 5.3 gives the boxplot of the test errors for the 50 samples and Table 5.2 gives a numerical description of the performances of the different methods.

	Mean	*Median*	*Standard deviation*	*1st quartile*	*Minimum*
PCA-NN	1,74	1,59	1,82	1,14	0,47
NNf	1,55	1,55	1,13	0,90	0,52
SIR-NNr	0,68	0,66	0,16	0,56	0,44
SIR-NNk	1,40	1,24	0,71	0,84	0,54
SIR-L	2,70	2,64	0,48	2,31	1,84

Table 5.2: Tecator data set: Description of the performances

These results show the excellent performances obtained by SIR-NNr: its SEP average over the 50 samples is twice lower than any of the other competitors. Moreover, this method garantees a good stability unlike the others. SIR seems to be a very good pre-processing stage, as SIR-NNk also obtains good performances. Then we have NNf but its rather good results suffer from a very slow computational time. To show this, we give the computational time of each method in Table 5.3. Clearly NNf is very expensive while SIR-L is very fast but works poorly. Actually, it is closely related to the number of inputs: 42 for NNf, 10 for SIR-NNk, 12 for PCA-NN and 20 for SIR-NNr.

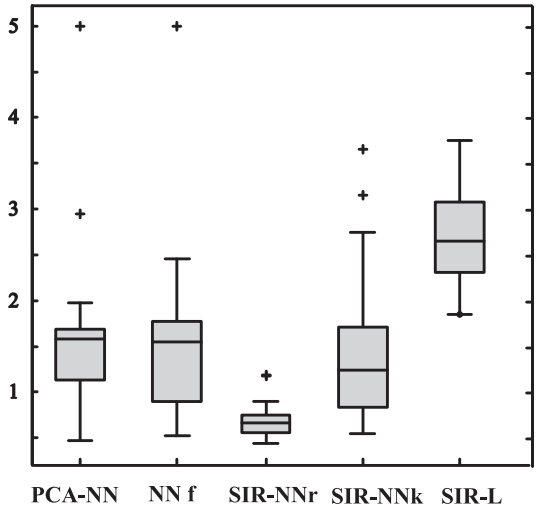

Figure 5.3: Tecator data set: SEP for 50 samples

Methods	PCA-NN	NNf	SIR-NNr	SIR-NNk	SIR-L
Computational time (number of seconds per sample)	50	350	100	50	1

Table 5.3: Computational time for the five compared methods

5.5.2 Phoneme data

In this section, we compare our methodology with other approaches on a classification problem, namely the phoneme data. The data are log-periodograms of a 32 ms duration corresponding to recorded speakers; it deals with the discrimination of five speech frames corresponding to five phonemes transcribed as follow: [sh] as in "she", [dcl] as in "dark", [iy] as in "she", [aa] as in "dark" and [ao] as in "water". Finally, the data consist in 4 509 log-periodograms of a 256 length (see Figure 5.4).

We tried several classical methods in order to test the efficiency of SIR-NNr which is compared with:

- **SIR-NNp**: a classical SIR as presented in [Ferré and Yao, 2003] as preprocessing of a neural network;
- **SIR-K**: a regularized functional SIR where the function f is estimated by a nonparametric kernel method;
- **Ridge-PDA**: the penalized discriminant analysis introduced in [Hastie *et al.*, 1995] which uses ridge penalty;
- **NPCD-PCA**: a nonparametric method using kernels and semi-metrics based on Principal Component Analysis and introduced by [Ferraty and Vieu, 2003].

The optimal parameters for these methods are shown in Table 5.4.

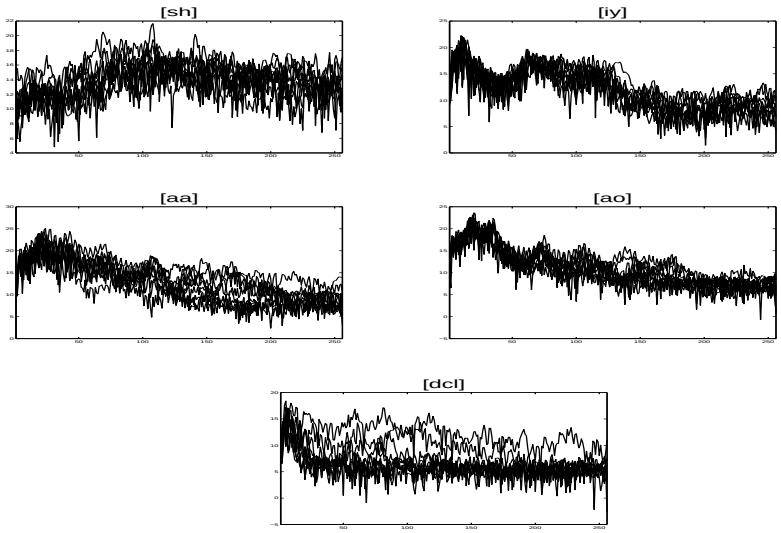

Figure 5.4: A sample of 10 log-periodograms per class

	Parameter 1	*Parameter 2*	*Parameter 3*
SIR-NNr	$\alpha = 10$ (regularization of Γ_X)	$q = 4$ (SIR dimension)	$q_2 = 15$ (number of neurons)
SIR-NNp	$k_n = 17$ (PCA dimension)	$q = 4$ (SIR dimension)	$q_2 = 12$ (number of neurons)
SIR-K	$\alpha = 10^{-3}$ (regularization of Γ_X)	$q = 4$ (SIR dimension)	$h = 1$ (kernel bandwidth)
RPDA	$\alpha = 5$ (regularization of Γ_X)	q= 4 (PDA dimension)	
NPCD-PCA	$k_n = 7$ (PCA dimension)	$h = 25$ (kernel window)	

Table 5.4: Best parameters for the five compared methods

For the SIR stage, the dimension of the EDR space is 4: it is the maximum dimension possible as the operator $\Gamma^N_{E(X/Y)}$ is of rank $H - 1$. We can also see that this dimension is relevant by looking at the projection of the data onto the EDR space (for SIR-NNr, for example, see Figure 5.5). We can see that only the fourth axis is able to separate the phonems [aa] and [ao].

Then we randomly build 50 samples divided as follows: the learning sample contains 1 735 log-periodograms (347 for each class) and the test sample contains also 1 735 (347 for each class). All five methods are first trained on the learning sample and the test error rate is then computed on the test sample. Figure 5.6 proposes the boxplot of the test error rates and Table 5.5 gives a description of the performances of test error rates over the 50 samples.

The results of SIR-NNr, SIR-NNp and SIR-K are very close. The benefit of SIR is highlighted since those three methods work better than others based on different projections of data. The advantage of regularization is also revealed since it leads again to the best results. Then comes RPDA and finally NPCD-PCA which provides the poorest performances. On the contrary, due to a low dimensionality, neural networks seem to be less performant than

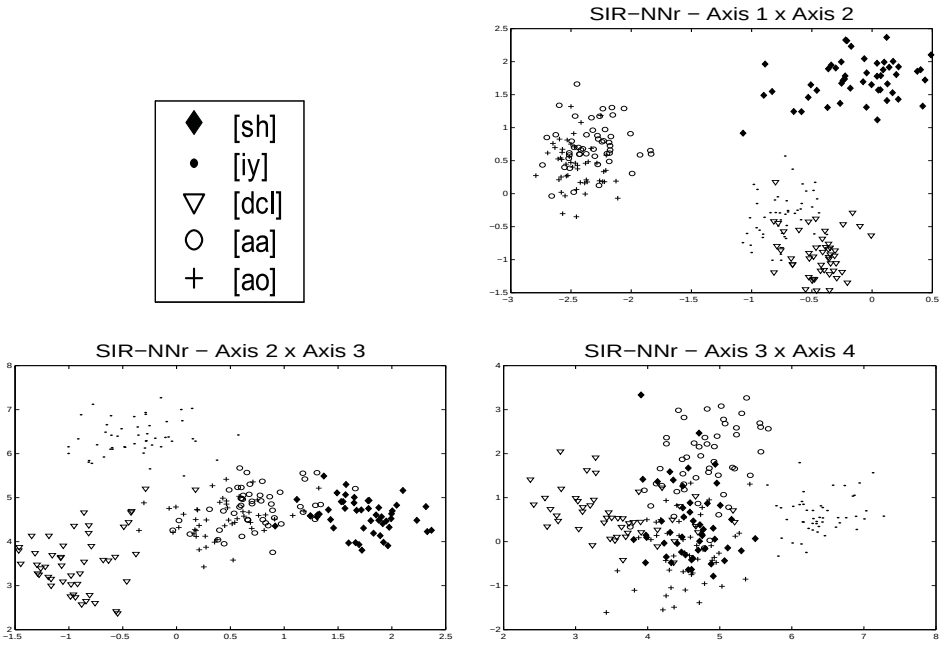

Figure 5.5: Projection onto the EDR space of 50 log-periodograms by class

	Mean	*Median*	*Standard deviation*	*1st quartile*	*Minimum*
SIR-NNr	8,21 %	8,16 %	0,56 %	7,90 %	6,74 %
SIR-K	8,09 %	8,10 %	0,40 %	7,84 %	6,92 %
SIR-NNp	8,38 %	8,24 %	0,59 %	7,95 %	7,20 %
RPDA	8,95 %	8,99 %	0,54 %	8,70 %	7,20 %
NPCD-PCA	9,78 %	9,68 %	0,65 %	9,34 %	8,30 %

Table 5.5: Phonem Data: Test error rates

kernels and have a bigger variability (standard deviation is 0,56 for SIR-NNr and only 0,40 for SIR-K): this problem can be removed by increasing the number of training steps or by using more sophisticated architecture or by a regularization technique (such as weight decay), but at the price of a larger computational cost. Finally, if SIR-K obtains the best mean, SIR-NNr is the method which reaches the best minima which shows its great potential.

In conclusion, both on regression and classification problems, regularized SIR-NN is a competitive solution for functional problems: we can explain these good results by noting that the procedure combines an efficient dimension reduction model and the great accuracy of a neural network, which is able to approximate almost every function. Thus this model can be efficient both for ill-posed problems thanks to the penalized functional and for problems with a large dimensionality thanks to the neural network step. Finally it has another great advantage: computational time is rather short and does not increase too much with the number of observation points for the curves.

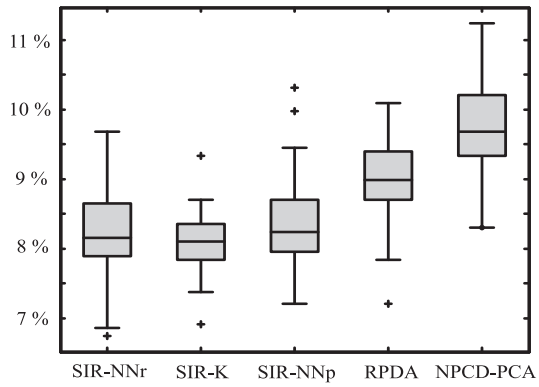

Figure 5.6: Phoneme Data: Test error rates for 50 samples

5.6 Appendix (Proofs)

Here we give the main lines of the proofs of Theorems 5.2 and 5.3.

5.6.1 Theorem 5.2

The proof of this theorem is related to the one of Theorem 1 in [Leurgans *et al.*, 1993] and only sketches are given.

Lemma 1: Using Central Limit Theorem, it is easy to show that if $\delta^N = \max\{|||\Gamma_X^N - \Gamma_X|||; |||\Gamma_{E(X/Y)}^N - \Gamma_{E(X/Y)}|||\}$ and if the sequence $(k_N)_N$ satisfies $\sqrt{N}k_N \to +\infty$ then

$$k_N^{-1}\delta^N \to_p 0.$$

Existence: We have for α in $[0,1]$, $Q_\alpha = (1-\alpha) < \Gamma_X .,. > +\alpha Q_1$ and then, for all u such that $\| u \| = 1$, $\frac{1}{\alpha}Q_\alpha(u,u) > (\frac{1}{\alpha} - 1) < \Gamma_X u, u > + Q_1(u,u) > \rho_1$ by the positiveness of Γ_X. Then, $\sqrt{N}\rho_\alpha > \alpha\sqrt{N}\rho_1$ and we have

$$\sqrt{N}\rho_\alpha \to +\infty \ . \tag{5.3}$$

Then, by Lemma 1, noting $\Delta_1^N = \Gamma_X^N - \Gamma_X$,

$$\lim_{N\to+\infty} P\left(\{\omega \in \Omega \quad ||| \Delta_1^N ||| \leq \frac{1}{2}\rho_\alpha\}\right) = 1.$$

But, we have

$$\{\omega \in \Omega \quad ||| \Delta_1^N ||| \leq \frac{1}{2}\rho_\alpha\} \subset \left\{\omega : \forall\ a \in \mathcal{S},\ \| a \| = 1,\ Q_\alpha^N(a,a) \geq \frac{1}{2}\rho_\alpha > 0\right\}$$

and finally the right hand part of the previous equation has a probability converging to 1 when N converges to $+\infty$.

Let $\overline{\mathcal{B}(0,1)}$ be the weak closure of $\{a \in \mathcal{S}\ \ Q_\alpha^N(a,a) = 1\}$ and ζ be the functional defined on $\{a \in \mathcal{S}\ \ Q_\alpha^N(a,a) = 1\}$ by

$$\zeta(a) = < \Gamma_{E(X/Y)}^N a, a >$$

then ζ can be extended to a uniformly continuous functional $\tilde{\zeta}$ defined on $\overline{\mathcal{B}(0,1)}$ for the weak topology. Finally, provided that $Q_\alpha^N(a,a) \geq \frac{1}{2}\rho_\alpha$, $\tilde{\zeta}$ reaches its maximum on weak compact $\overline{\mathcal{B}(0,1)}$ which concludes the proof of the existence of $(a_j^N)_{j=1,\ldots,q}$.

Consistency: For the following we suppose that we stand on the set where γ^N has a maximum on $\mathcal{S}$ and reaches it.

Let λ_1^N be this maximum and λ_1^α be the maximum of

$$\gamma_\alpha(a) = \frac{< \Gamma_{E(X/Y)} a, a >}{< \Gamma_X a, a > + \alpha[a,a]}$$

on $\mathcal{S}$; λ_1^α is well defined thanks to assumption **(A3)**.

Considering $\frac{\gamma_\alpha(a)}{\gamma_0(a)}$, we easily show that

$$\lambda_1^\alpha \to \lambda_1. \tag{5.4}$$

Then, by proving that $\sup_{a\in\mathcal{S}} |\gamma^N(a) - \gamma_\alpha(a)| \to_p 0$ we can show that

$$\left|\lambda_1^N - \lambda_1^\alpha\right| \to_p 0. \tag{5.5}$$

Finally, by combining (5.4) and (5.5), we conclude that

$$\lambda_1^N \to_p \lambda_1 \tag{5.6}$$

Then, by using (5.6), we demonstrate that

$$\gamma(a_1^N) \to_p \lambda_1 = \gamma(a_1). \tag{5.7}$$

Thanks to the conclusion of Theorem 5.1 we show that

$$\lim_{N\to+\infty} \mathbb{P}(< \Gamma_{E(X/Y)} a_1, a_1^N - a_1 >=< \Gamma_X a_1, a_1^N - a_1 >= 0) = 1.$$

Let μ_N be $< \Gamma_X(a_1^N - a_1), a_1^N - a_1 >$; if $< \Gamma_{E(X/Y)} a_1, a_1^N - a_1 >= 0$, we have

$$\lambda_1^{-1}\gamma(a_1^N) \leq \frac{1 + \lambda_1^{-1}\lambda_2\mu_N}{1+\mu_N} \ .$$

As $\lambda_1^{-1}\lambda_2 < 1$, the right hand side of the previous inequality is less than 1; but $\lambda_1^{-1}\gamma(a_1^N)$ converges in probability to 1 by (5.7) so

$$\frac{1 + \lambda_1^{-1}\lambda_2\mu_N}{1+\mu_N} \to_p 1$$

and then we conclude with $\mu_N \to_p 0$.

5.6.2 Theorem 5.3

The proof of this theorem is close to the one found in [Rossi *et al.*, 2002] and [Conan-Guez and Rossi, 2002]; the main difference is that the projection for the data is a random variable. The proof will be divided into two parts:

We first prove that

$$\sup_{w\in\mathcal{W}} \left| \frac{1}{N} \sum_{n=1}^{N} \zeta(\tilde{Z}_N^n, w) - E(\zeta(Z,w)) \right| \to_p 0. \tag{5.8}$$

Forall w in $\mathcal{W}$, we have

$$\left|\frac{1}{N}\sum_{n=1}^{N}\zeta(\tilde{Z}_N^n,w)-E(\zeta(Z,w))\right|$$

$$\leq\left|\frac{1}{N}\sum_{n=1}^{N}\zeta(\tilde{Z}_N^n,w)-\frac{1}{N}\sum_{n=1}^{N}\zeta(Z_n,w)\right|+\left|\frac{1}{N}\sum_{n=1}^{N}\zeta(Z_n,w)-E(\zeta(Z,w))\right|.$$

By using Dominated Convergence Theorem, the fact that $\mathcal{W}$ is a compact set, that $\zeta(z,.)$ is continuous for all $z\in\mathcal{O}$, and that $\zeta(.,w)$ is mesurable for all $w\in\mathcal{W}$, we can show that, for all $\tilde{w}\in\mathcal{W}$,

$$\lim_{\mu\to 0}E\left(\sup_{w\in\mathcal{W}\cap\mathcal{B}(\tilde{w},\mu)}\zeta(Z,w)\right)=E(\zeta(Z,\tilde{w}))$$

where $\mathcal{B}(\tilde{w},\mu)$ is the ball centered on $\tilde{w}$ and of radius μ. Then let ϵ be a real positive number, for all $\tilde{w}\in\mathcal{W}$, there is a $\mu(\tilde{w})$ such that

$$E\left(\sup_{w\in\mathcal{W}\cap\mathcal{B}(\tilde{w},\mu(\tilde{w}))}\zeta(Z,w)\right)\leq E(\zeta(Z,\tilde{w}))+\frac{\epsilon}{3}\tag{5.9}$$

Similarly, with the function $-\zeta$, we get:

$$E\left(\inf_{w\in\mathcal{W}\cap\mathcal{B}(\tilde{w},\mu(\tilde{w}))}\zeta(Z,w)\right)\geq E(\zeta(Z,\tilde{w}))-\frac{\epsilon}{3}.\tag{5.10}$$

Using the law of large numbers, we can deduce from (5.9) and (5.10) that for all $\tilde{w}\in\mathcal{W}$, almost surely, there is a $N(\tilde{w})\in\mathbb{N}$ such that, for all $N\geq N(\tilde{w})$,

$$\sup_{w\in\mathcal{W}\cap\mathcal{B}(\tilde{w},\mu(\tilde{w}))}\left|\frac{1}{N}\sum_{n=1}^{N}\zeta(Z_n,w)-E(\zeta(Z,w))\right|\leq\epsilon.$$

As $\mathcal{W}$ is a compact set, we can find $\tilde{w}_1,\ldots,\tilde{w}_I$ such as $\{\mathcal{B}(\tilde{w}_i,\mu(\tilde{w}_i))\}_{i=1,\ldots,I}$ re-cover $\mathcal{W}$. Using these sets we conclude that

$$\left|\frac{1}{N}\sum_{n=1}^{N}\zeta(Z_n,w)-E(\zeta(Z,w))\right|\leq\epsilon,$$

Using assumption **(A8)** we see that

$$\begin{aligned}&\left|\tfrac{1}{N}\textstyle\sum_{n=1}^{N}\left(\zeta(\tilde{Z}_N^n,w)-\zeta(Z_n,w)\right)\right|\\&\qquad\leq C(w)\left[\textstyle\sum_{j=1}^{q}<\Gamma_X^N(a_j^N-a_j),a_j^N-a_j>\right]^{1/2}\end{aligned}$$

As $|||\Gamma_X^N-\Gamma_X|||\to_p 0$ and as, for all $j=1,\ldots,q$, $<\Gamma_X(a_j^N-a_j),a_j^N-a_j>\to_p 0$, we then conclude that

$$\sup_{w\in\mathcal{W}}\left|\frac{1}{N}\sum_{n=1}^{N}\left(\zeta(\tilde{Z}_N^n,w)-\zeta(Z_n,w)\right)\right|\to_p 0,$$

which finally implies (5.8).

Secondly, let ϵ be a positive real. According to the Dominated Convergence Theorem, $E(\zeta(Z,.))$ is a continuous function which reaches its minimum m on compact set $\mathcal{W}$. Then we can show that there is a $\eta(\epsilon) > 0$ such that, for all w in $\mathcal{W}$,

$$|E(\zeta(Z,w)) - m| \leq \eta \;\Rightarrow\; d(w, \mathcal{W}^*) \leq \epsilon. \tag{5.11}$$

Then let $\Omega_{\eta,N}$ be the following subset of Ω

$$\left\{ \omega \in \Omega : \sup_{w \in \mathcal{W}} \left| \frac{1}{N} \sum_{n=1}^{N} \zeta(\tilde{Z}_N^n, w) - E(\zeta(Z,w)) \right| \leq \frac{\eta}{3} \right\}.$$

If $\omega \in \Omega_{\eta,N}$ then, as $\mathcal{W}$ is a compact set, we can find, for all $N \in \mathbb{N}$, $w_N^*(\omega) \in \mathcal{W}$ which minimizes $\frac{1}{N}\sum_{n=1}^{N} \zeta(\tilde{Z}_N^n(\omega), w)$. Let w^* be in the closure of $\{w_N^*\}_N$; then by arguments similar to the ones used in the first part of the proof we show that, for all $\omega \in \Omega_{\eta,N}$ and for all $w \in \mathcal{W}$,

$$E(\zeta(Z,w^*)) \leq E(\zeta(z,w)) + \eta,$$

which implies by the use of (5.11) that

$$\Omega_{\eta,N} \subset \{\omega \;\; d(w^*(\omega), \mathcal{W}^*) \leq \epsilon\}$$

and this concludes the proof as $\lim_{N \to +\infty} P(\Omega_{\eta,N}) = 1$.

Chapitre 6

Support Vector Machine For Functional Data Classification

Fabrice Rossi
Projet AxIS, INRIA-Rocquencourt, Le Chesnay, France
Nathalie Villa
GRIMM, Equipe d'accueil 3686, Université Toulouse Le Mirail, France

Référence : Support Vector Machine For Functional Data Classification (2005), *à paraître dans Neurocomputing.*

Abstract:

In many applications, input data are sampled functions taking their values in infinite dimensional spaces rather than standard vectors. This fact has complex consequences on data analysis algorithms that motivate modifications of them. In fact most of the traditional data analysis tools for regression, classification and clustering have been adapted to functional inputs under the general name of Functional Data Analysis (FDA). In this paper, we investigate the use of Support Vector Machines (SVMs) for functional data analysis and we focus on the problem of curves discrimination. SVMs are large margin classifier tools based on implicit non linear mappings of the considered data into high dimensional spaces thanks to kernels. We show how to define simple kernels that take into account the functional nature of the data and lead to consistent classification. Experiments conducted on real world data emphasize the benefit of taking into account some functional aspects of the problems.
Keywords: *Functional Data Analysis, Support Vector Machine, Classification, Consistency*

6.1 Introduction

In many real world applications, data should be considered as discretized functions rather than as standard vectors. In these applications, each observation corresponds to a mapping between some conditions (that might be implicit) and the observed response. A well studied example of those functional data is given by spectrometric data (see section 6.6.3): each

spectrum is a function that maps the wavelengths of the illuminating light to the corresponding absorbances (the responses) of the studied sample. Other natural examples can be found in voice recognition area (see sections 6.6.1 and 6.6.2) or in meteorological problems, and more generally, in multiple time series analysis where each observation is a complete time series.

The direct use of classical models for this type of data faces several difficulties: as the inputs are discretized functions, they are generally represented by high dimensional vectors whose coordinates are highly correlated. As a consequence, classical methods lead to ill-posed problems, both on a theoretical point of view (when working in functional spaces that have infinite dimension) and on a practical one (when working with the discretized functions). The goal of Functional Data Analysis (FDA) is to use, in data analysis algorithms, the underlying functional nature of the data: many data analysis methods have been adapted to functions (see [Ramsay and Silverman, 1997] for a comprehensive introduction to functional data analysis and a review of linear methods). While the original papers on FDA focused on linear methods such as Principal Component Analysis ([Deville, 1974], [Dauxois and Pousse, 1976], [Dauxois *et al.*, 1982] and [Besse and Ramsay, 1986]) and the linear model ([Ramsay and Dalzell, 1991], [Frank and Friedman, 1993] and [Hastie and Mallows, 1993]), non linear models have been studied extensively in the recent years. This is the case, for instance, of most neural network models ([Ferré and Villa, 2005b], [Rossi and Conan-Guez, 2005a], [Rossi *et al.*, 2004], [Rossi *et al.*, 2005]).

In the present paper, we adapt Support Vector Machines (SVMs, see e.g. [Vapnik, 1995], [Cristianini and Shawe-Taylor, 2000]) to functional data classification (the paper extends results from [Rossi and Villa, 2005a] and [Villa and Rossi, 2005]). We show in particular both the practical and theoretical advantages of using functional kernels, which are kernels that take into account the functional nature of the data. On a practical point of view, those kernels allow to take advantage of the expert knowledge on the data. On the theoretical point of view, a specific type of functional kernels allows the construction of a consistent training procedure for functional SVMs.

The paper is organized as follow: section 6.2 presents the functional data classification and why it generally leads to ill-posed problems. Section 6.3 provides a short introduction to SVMs and explains why their generalization to FDA can lead to particular problems. Section 6.4 describes several functional kernels and explains how they can be practically computed while section 6.5 presents a consistency result for some of them. Finally, section 6.6 illustrates the various approaches presented in the paper on real data sets.

6.2 Functional Data Analysis

6.2.1 Functional Data

To simplify the presentation, this article focuses on functional data for which each observation is described by one function from $\mathbb{R}$ to $\mathbb{R}$. Extension to the case of several real valued functions is straightforward. More formally, if μ denotes a finite positive Borel measure on $\mathbb{R}$, an observation is an element of $L^2(\mu)$, the Hilbert space of μ-square-integrable real valued functions defined on $\mathbb{R}$. In some situations, additional regularity assumptions (e.g., existence of derivatives) will be needed.

However, almost all the developments of this paper are not specific to functions and use only the Hilbert space structure of $L^2(\mu)$. We will therefore denote $\mathcal{X}$ an arbitrary Hilbert space and $\langle ., . \rangle$ the corresponding inner product. Additional assumptions on $\mathcal{X}$ will be given

on a case by case basis. As stated above, the most common situation will of course be $\mathcal{X} = L^2(\mu)$ with $\langle u, v \rangle = \int uv \mathrm{d}\mu$.

6.2.2 Data analysis methods for Hilbert spaces

It should be first noted that many data analysis algorithms can be written so as to apply, at least on a theoretical point of view, to arbitrary Hilbert spaces. This is obviously the case, for instance, for distance-based algorithms such as the k-nearest neighbor method. Indeed, this algorithm uses only the fact that distances between observations can be calculated. Obviously, it can be applied to Hilbert spaces using the distance induced by the inner product. This is also the case of methods directly based on inner products such as multi-layer perceptrons (see [Sandberg, 1996], [Sandberg and Xu, 1996], [Stinchcombe, 1999] for a presentation of multi-layer perceptrons with almost arbitrary input spaces, including Hilbert spaces).

However, functional spaces have infinite dimension and a basic transposition of standard algorithms introduces both theoretical and practical difficulties. In fact, some simple problems in $\mathbb{R}^d$ become ill-posed in $\mathcal{X}$ when the space has infinite dimension, even on a theoretical point of view.

Let us consider for instance the linear regression model in which a real valued target variable Y is modeled by $E(Y|X) = H(X)$ where H is a linear continuous operator defined on the input space. When X has values in $\mathbb{R}^d$ (i.e., $\mathcal{X} = \mathbb{R}^d$), H can be easily estimated by the least square method that leads to the inversion of the covariance matrix of X. In practice, problems might appear when d is not small compared to N, the number of available examples, and regularization techniques should be used (e.g., ridge regression [Hoerl and Kennard, 1970b]). When X has values in a Hilbert space, the problem is ill-posed because the covariance of X is a Hilbert-Schmidt operator and thus has no continuous inverse; direct approximation of the inverse of this operator is then problematic as it does not provide a consistant estimate (see [Cardot *et al.*, 1999]).

To overcome the infinite dimensional problem, most of FDA methods so far have been constructed thanks to two general principles: *filtering* and *regularization*. In the filtering approach, the idea is to use representation methods that allow to work in finite dimension (see for instance [Cardot *et al.*, 1999] for the functional linear model and [Biau *et al.*, 2005] for a functional k-nearest neighbor method). In the regularization approach, the complexity of the solution is constrained thanks to smoothness constraints. For instance, building a linear model in a Hilbert space consists in finding a function $h \in L^2(\mu)$ such that $E(Y|X) = \langle h, X \rangle$. In the regularization approach, h is chosen among smooth candidates (for instance twice derivable functions with minimal curvature), see e.g. [Hastie and Mallows, 1993], [Marx and Eilers, 1996], [Cardot *et al.*, 2003]. Other examples of the regularization approach include smooth Principal Component Analysis [Pezzulli and Silverman, 1993] and penalized Canonical Component Analysis [Leurgans *et al.*, 1993]. A comparison of filtering and regularization approaches for a semi-parametric model used in curve discrimination can be found in [Ferré and Villa, 2005a].

Using both approaches, a lot of data analysis algorithms have been successfully adapted to functional data. Our goal in the present paper is to study the case of Support Vector Machines (SVM), mainly thanks to a filtering approach.

6.3 Support Vector Machines for FDA

6.3.1 Support Vector Machines

We give, in this section, a very brief presentation of Support Vector Machines (SVMs) that is needed for the definition of their functional versions. We refer the reader to e.g. [Cristianini and Shawe-Taylor, 2000] for a more comprehensive presentation. As stated in section 6.2.1, $\mathcal{X}$ denotes an arbitrary Hilbert space. Our presentation of SVM departs from the standard introduction because it assumes that the observations belong to $\mathcal{X}$ rather than to a $\mathbb{R}^d$. This will make clear that the definition of SVM on arbitrary Hilbert spaces is not the difficult part in the construction of functional SVM. We will discuss problems related to the functional nature of the data in section 6.3.2.

Our goal is to classify data into two predefined classes. We assume given a learning set, i.e. N examples $(x_1, y_1), \ldots, (x_N, y_N)$ which are i.i.d. realizations of the random variable pair (X, Y) where X has values in $\mathcal{X}$ and Y in $\{-1, 1\}$, i.e. Y is the class label for X which is the observation.

Hard margin SVM

The principle of SVM is to perform an affine discrimination of the observations with maximal margin, that is to find an element $w \in \mathcal{X}$ with a minimum norm and a real value b, such that $y_i(\langle w, x_i\rangle + b) \geq 1$ for all i. To do so, we have to solve the following quadratic programming problem:

$$(P_0) \quad \min_{w,b} \langle w, w\rangle, \text{ subject to } y_i(\langle w, x_i\rangle + b) \geq 1, \ 1 \leq i \leq N.$$

The classification rule associated to (w, b) is simply $f(x) = \text{sign}(\langle w, x\rangle + b)$. In this situation (called hard margin SVM), we request the rule to have zero error on the learning set.

Soft margin SVM

In practice, the solution provided by problem (P_0) is not very satisfactory. Firstly, perfectly linearly separable problems are quite rare, partly because non linear problems are frequent, but also because noise can turn a linearly separable problem into a non separable one. Secondly, choosing a classifier with maximal margin does not prevent overfitting, especially in very high dimensional spaces (see e.g. [Hastie *et al.*, 2004] for a discussion about this point).

A first step to solve this problem is to allow some classification errors on the learning set. This is done by replacing (P_0) by its soft margin version, i.e., by the problem:

$$\begin{array}{ll} (P_C) & \min_{w,b,\xi} \langle w, w\rangle + C \sum_{i=1}^N \xi_i, \\ & \text{subject to} \quad y_i(\langle w, x_i\rangle + b) \geq 1 - \xi_i, \ 1 \leq i \leq N, \\ & \qquad\qquad \xi_i \geq 0, \ 1 \leq i \leq N. \end{array}$$

Classification errors are allowed thanks to the slack variables ξ_i. The C parameter acts as an inverse regularization parameter. When C is small, the cost of violating the hard margin constraints, i.e., the cost of having some $\xi_i > 0$ is small and therefore the constraint on w dominates. On the contrary, when C is large, classification errors dominate and (P_C) gets closer to (P_0).

Non linear SVM

As noted in the previous section, some classification problems don't have a satisfactory linear solution but have a non linear one. Non linear SVMs are obtained by transforming the original data. Assume given an Hilbert space $\mathcal{H}$ (and denote $\langle .,. \rangle_{\mathcal{H}}$ the corresponding inner product) and a function ϕ from $\mathcal{X}$ to $\mathcal{H}$ (this function is called a *feature map*). A linear SVM in $\mathcal{H}$ can be constructed on the data set $(\phi(x_1), y_1), \ldots, (\phi(x_N), y_N)$. If ϕ is a non linear mapping, the classification rule $f(x) = \text{sign}(\langle w, \phi(x) \rangle_{\mathcal{H}} + b)$ is also non linear.

In order to obtain the linear SVM in $\mathcal{H}$ one has to solve the following optimization problem:

$$\begin{array}{rl} (P_{C,\mathcal{H}}) & \min_{w,b,\xi} \langle w, w \rangle_{\mathcal{H}} + C \sum_{i=1}^N \xi_i, \\ & \text{subject to} \quad y_i(\langle w, \phi(x_i) \rangle_{\mathcal{H}} + b) \geq 1 - \xi_i, \ 1 \leq i \leq N, \\ & \qquad\qquad \xi_i \geq 0, \ 1 \leq i \leq N. \end{array}$$

It should be noted that this feature mapping allows to define SVM on almost arbitrary input spaces.

Dual formulation and Kernels

Solving problems (P_C) or $(P_{C,\mathcal{H}})$ might seem very difficult at first, because $\mathcal{X}$ and $\mathcal{H}$ are arbitrary Hilbert spaces and can therefore have very high or even infinite dimension (when $\mathcal{X}$ is a functional space for instance). However, each problem has a dual formulation. More precisely, (P_C) is equivalent to the following optimization problem (see [Lin, 2001]):

$$\begin{array}{rl} (D_C) & \max_{\alpha} \sum_{i=1}^N \alpha_i - \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle, \\ & \text{subject to} \quad \sum_{i=1}^N \alpha_i y_i = 0, \\ & \qquad\qquad 0 \leq \alpha_i \leq C, \ 1 \leq i \leq N. \end{array}$$

This result applies to the original problem in which data are not mapped into $\mathcal{H}$, but also to the mapped data, i.e., $(P_{C,\mathcal{H}})$ is equivalent to a problem $(D_{C,\mathcal{H}})$ in which the x_i are replaced by $\phi(x_i)$ and in which the inner product of $\mathcal{H}$ is used. This leads to:

$$\begin{array}{rl} (D_{C,\mathcal{H}}) & \max_{\alpha} \sum_{i=1}^N \alpha_i - \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j \langle \phi(x_i), \phi(x_j) \rangle_{\mathcal{H}}, \\ & \text{subject to} \quad \sum_{i=1}^N \alpha_i y_i = 0, \\ & \qquad\qquad 0 \leq \alpha_i \leq C, \ 1 \leq i \leq N. \end{array}$$

Solving $(D_{C,\mathcal{H}})$ rather than $(P_{C,\mathcal{H}})$ has two advantages. The first positive aspect is that $(D_{C,\mathcal{H}})$ is an optimization problem in $\mathbb{R}^N$ rather than in $\mathcal{H}$ which can have infinite dimension (the same is true for $\mathcal{X}$).

The second important point is linked to the fact that the optimal classification rule can be written $f(x) = \text{sign}(\sum_{i=1}^N \alpha_i y_i \langle \phi(x_i), \phi(x) \rangle_{\mathcal{H}} + b)$. This means that both the optimization problem and the classification rule do not make direct use of the transformed data, i.e. of the $\phi(x_i)$. All the calculations are done through the inner product in $\mathcal{H}$, more precisely through the inner product in $\mathcal{H}$, more precisely through the values $\langle \phi(x_i), \phi(x_j) \rangle_{\mathcal{H}}$. Therefore, rather than choosing directly $\mathcal{H}$ and ϕ, one can provide a so called *Kernel function* K such that $K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle_{\mathcal{H}}$ for a given pair $(\mathcal{H}, \phi)$.

In order that K corresponds to an actual inner product in a Hilbert space, it has to fulfill some conditions. K has to be symmetric and positive definite, that is, for every N, $x_1, \ldots, x_N$ in $\mathcal{X}$ and $\alpha_1, \ldots, \alpha_N$ in $\mathbb{R}$, $\sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j K(x_i, x_j) \geq 0$. If K satisfies those conditions, according to Moore-Aronszajn theorem [Aronszajn, 1950], there exists a Hilbert space $\mathcal{H}$ and feature map ϕ such that $K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle_{\mathcal{H}}$.

6.3.2 The case of functional data

The short introduction to SVM proposed in the previous section has clearly shown that defining linear SVM for data in a functional space is as easy as for data in $\mathbb{R}^d$, because we only assumed that the input space was a Hilbert space. By the dual formulation of the optimization problem (P_C), a software implementation of linear SVM on functional data is even possible, by relying on numerical quadrature methods to calculate the requested integrals (inner product in $L^2(\mu)$, cf section 6.4.3).

However, the functional nature of the data has some effects. It should be first noted that in infinite dimensional Hilbert spaces, the hard margin problem (P_0) has always a solution when the input data are in general positions, i.e., when N observations span a N dimensional subspace of $\mathcal{X}$. A very naive solution would therefore consists in avoiding soft margins and non linear kernels. This would not give very interesting results in practice because of the lack of regularization (see [Hastie *et al.*, 2004] for some examples in very high dimension spaces, as well as section 6.6.1).

Moreover, the linear SVM with soft margin can also lead to bad performances. It is indeed well known (see e.g. [Hastie *et al.*, 2001]) that problem (P_C) is equivalent to the following unconstrained optimization problem:

$$(R_\lambda) \min_{w,b} \frac{1}{N} \sum_{i=1}^{N} \max\left(0, 1 - y_i(\langle w, x_i\rangle + b)\right) + \lambda\langle w, w\rangle,$$

with $\lambda = \frac{1}{CN}$. This way of viewing (P_C) emphasizes the regularization aspect (see also [Smola and Schölkopf, 1998b], [Smola and Schölkopf, 1998a], [Evgeniou *et al.*, 2000]) and links the SVM model to ridge regression [Hoerl and Kennard, 1970b]. As shown in [Hastie *et al.*, 1995], the penalization used in ridge regression behaves poorly with functional data. Of course, the loss function used by SVM (the *hinge loss*, i.e., $h(u, v) = \max(0, 1-uv)$) is different from the quadratic loss used in ridge regression and therefore no conclusion can be drawn from experiments reported in [Hastie *et al.*, 1995]. However they show that we might expect bad performances with the linear SVM applied directly to functional data. We will see in sections 6.6.1 and 6.6.2 that the efficiency of the ridge regularization seems to be linked with the actual dimension of the data: it does not behave very well when the number of discretization points is very big and thus leads to approximate the ridge penalty by a dot product in a very high dimensional space (see also section 6.4.3).

It is therefore interesting to consider non linear SVM for functional data, by introducing adapted kernels. As pointed out in e.g. [Evgeniou *et al.*, 2000], $(P_{C,\mathcal{H}})$ is equivalent to

$$(R_{\lambda,\mathcal{H}}) \min_{f\in\mathcal{H}} \frac{1}{N} \sum_{i=1}^{N} \max\left(0, 1 - y_i f(x_i)\right)) + \lambda\langle f, f\rangle_{\mathcal{H}}.$$

Using a kernel corresponds therefore both to replace a linear classifier by a non linear one, but also to replace the ridge penalization by a penalization induced by the kernel which might be more adapted to the problem (see [Smola and Schölkopf, 1998a] for links between regularization operators and kernels). The applications presented in 6.6 illustrate this fact.

6.4 Kernels for FDA

6.4.1 Classical kernels

Many standard kernels for $\mathbb{R}^d$ data are based on the Hilbert structure of $\mathbb{R}^d$ and can therefore be applied to any Hilbert space. This is the case for instance of the Gaussian kernel (based

on the norm in $\mathcal{X}$: $K(u,v) = e^{-\sigma\|u-v\|^2}$) and of the polynomial kernels (based on the inner product in $\mathcal{X}$: $K(u,v) = (1+\langle u,v\rangle)^D$). Obviously, the only practical difficulty consists in implementing the calculations needed in $\mathcal{X}$ so as to evaluate the chosen kernel (the problem also appears for the plain linear "kernel", i.e. when no feature mapping is done). Section 6.4.3 discusses this point.

6.4.2 Using the functional nature of the data

While the functional version of the standard kernels can provide an interesting library of kernels, they do not take advantage of the functional nature of the data (they use only the Hilbert structure of $L^2(\mu)$). Kernels that use the fact that we are dealing with functions are nevertheless quite easy to define.

A standard method consists in introducing kernels that are made by a composition of a simple feature map with a standard kernel. More formally, we use a transformation operator P from $\mathcal{X}$ to another space $\mathcal{D}$ on which a kernel K is defined. The actual kernel Q on $\mathcal{X}$ is defined as $Q(u,v) = K(P(u),P(v))$ (if K is a kernel, then so is Q).

Functional transformations

In some application domains, such as chemometrics, it is well known that the shape of a spectrum (which is a function) is sometimes more important than its actual mean value. Several transformations can be proposed to deal with this kind of data. For instance, if μ is a finite measure (i.e., $\mu(\mathbb{R}) < \infty$), a centering transformation can be defined as the following mapping from $L^2(\mu)$ to itself:

$$C(u) = u - \frac{1}{\mu(\mathbb{R})}\int u \mathrm{d}\mu.$$

A normalization mapping can also be defined:

$$N(u) = \frac{1}{\|C(u)\|}C(u).$$

If the functions are smooth enough, i.e., if we restrict ourselves to a Sobolev space $W^{s,2}$, then some derivative transformations can be used: the Sobolev space $W^{s,2}$, also denoted H^s, is the Hilbert space of functions which have L^2 derivatives up to the order s (in the sense of the distribution theory). For instance, with $s \geq 2$, we can use the second derivative that allows to focus on the curvature of the functions: this is particularly useful in near infrared spectrometry (see e.g., [Rossi and Conan-Guez, 2005a], [Rossi *et al.*, 2005], and section 6.6.3).

Projections

Another type of transformations can be used in order to define adapted kernels. The idea is to reduce the dimensionality of the input space, that is to apply the standard filtering approach of FDA. We assume given a d-dimensional subspace V_d of $\mathcal{X}$ and an orthonormal basis of this space denoted $\{\Psi_j\}_{j=1,\dots,d}$. We define the transformation P_{V_d} as the orthogonal projection on V_d,

$$P_{V_d}(x) = \sum_{j=1}^{d}\langle x,\Psi_j\rangle\Psi_j.$$

$(V_d, \langle .,. \rangle_{\mathcal{X}})$ is isomorphic to $(\mathbb{R}^d, \langle .,. \rangle_{\mathbb{R}^d})$ and therefore one can use a standard $\mathbb{R}^d$ SVM on the vector data $(\langle x, \Psi_1 \rangle, \ldots, \langle x, \Psi_d \rangle)$. This means that K can be any kernel adapted to vector data.

Obviously, this approach is not restricted to functional data, but the choice of V_d can be directed by expert knowledge on the considered functions and we can then consider that it takes advantage of the functional nature of the data. We outline here two possible solutions based on orthogonal basis and on B-spline basis.

If $\mathcal{X}$ is separable, it has a Hilbert basis, i.e., a complete orthonormal system $\{\Psi_j\}_{j \geq 1}$. Therefore one can define V_d as the space spanned by $\{\Psi_j\}_{j=1,\ldots,d}$. The choice of the basis can be based on expert considerations. Good candidates include Fourier basis and wavelet basis. If the signal is known to be non stationary, a wavelet based representation might for instance give better results than a Fourier representation. Once the basis is chosen, an optimal value for d can be derived from the data, as explained in section 6.5, in such a way that the obtained SVM has some consistency properties. Moreover, this projection approach gives good results in practice (see section 6.6.1).

Another solution is to choose a projection space that has interesting practical properties, for instance a spline space with its associated B-spline bases. Spline functions regularity can be chosen *a priori* so as to enforce expert knowledge on the functions. For instance, near infrared spectra are smooth because of the physical properties of the light transmission (and reflection). By using a spline representation of the spectra, we replace original unconstrained observations by C^k approximations (k depends on what kind of smoothness hypothesis can be done). This projection can also be combined with a derivative transformation operation (as proposed in section 6.4.2).

6.4.3 Functional data in practice

In practice, the functions $(x_i)_{1 \leq i \leq N}$ are never perfectly known. It is therefore difficult to implement exactly the functional kernels described in this section.

The best situation is the one in which d discretization points have been chosen in $\mathbb{R}$, $(t_k)_{1 \leq k \leq d}$, and each function x_i is described by a vector of $\mathbb{R}^d$, $(x_i(t_1), \ldots, x_i(t_d))$. In this situation, a simple solution consists in assuming that standard operations in $\mathbb{R}^d$ (linear combinations, inner product and norm) are good approximations of their counterparts in the considered functional space. When the sampling is regular, this is equivalent to applying standard SVMs to the vector representation of the functions (see section 6.6 for real world examples of this situation). When the sampling is not regular, integrals should be approximated thanks to a quadrature method that will take into account the relative position of the sampling points.

In some application domains, especially medical ones (e.g., [James and Hastie, 2001]), the situation is not as good. Each function is in general badly sampled: the number and the location of discretization points depend on the function and therefore a simple vector model is not anymore possible. A possible solution in this context consists in constructing a approximation of x_i based on its observation values (thanks to e.g., B-splines) and then to work with the reconstructed functions (see [Ramsay and Silverman, 1997] and [Rossi *et al.*, 2005] for details).

The function approximation tool used should be simple enough to allow easy implementation of the requested operations. This is the case for instance for B-splines that allow in addition derivative calculations and an easy implementation of the kernels described in section 6.4.2. It should be noted that spline approximation is different from projection on a spline subspace. Indeed each sampled function could be approximated on a different B-spline basis, whereas the projection operator proposed in section 6.4.2 requests an unique

projection space and therefore the same B-spline basis for each input function. In other words, the spline approximation is a convenient way of representing functions (see section 6.6.3 for an application to real world data), whereas the spline projection corresponds to a data reduction technique. Both aspects can be combined.

6.5 Consistency of functional SVM

6.5.1 Introduction

In this section we study one of the functional kernel described above and show that it can be used to define a consistent classifier for functional data. We introduce first some notations and definitions.

Our goal is to define a training procedure for functional SVM such that the asymptotic generalization performances of the constructed model is optimal. We define as usual the generalization error of a classifier f by the probability of misclassification:

$$Lf = \mathbb{P}(f(X) \neq Y).$$

The minimal generalization error is the Bayes error achieved by the optimal classifier f^* given by

$$f^*(x) = \begin{cases} 1 & \text{when } \mathbb{P}(Y = 1 \mid X = x) > 1/2 \\ -1 & \text{otherwise.} \end{cases}$$

We denote $L^* = Lf^*$ the optimal Bayes error. Of course, the closer the error of a classifier is from L^*, the better its generalization ability is.

Suppose that we are given a learning sample of size N defined as in section 6.3.1. A learning procedure is an algorithm which allows the construction, from this learning sample, of a classification rule f_N chosen in a set of admissible classifiers. This algorithm is said to be consistent if

$$Lf_N \xrightarrow{N \to +\infty} L^*.$$

It should be noted that when the data belong to $\mathbb{R}^d$, SVMs don't always provide consistent classifiers. Some sufficient conditions have been given in [Steinwart, 2002]: the input data must belong to a compact subset of $\mathbb{R}^d$, the regularization parameter (C in $(P_{C,\mathcal{H}})$) has to be chosen in specific way (in relation to N and to the type of kernel used) and the kernel must be *universal* [Steinwart, 2001]. If ϕ is the feature map associated to a kernel K, the kernel is universal if the set of all the functions of the form $x \mapsto \langle w, \phi(x) \rangle$ for $w \in \mathcal{H}$ is dense in the set of all continuous functions defined on the considered compact subset. In particular, the Gaussian kernel with any $\sigma > 0$ is universal for all compact subsets of $\mathbb{R}^d$ (see [Steinwart, 2002] for futher details and the proof of Theorem 6.1 for the precise statement on C).

6.5.2 A learning algorithm for functional SVM

The general methodology proposed in [Biau *et al.*, 2005] allows to turn (with some adaptations) a consistent algorithm for data in $\mathbb{R}^d$ into a consistent algorithm for data in $\mathcal{X}$, a separable Hilbert space. We describe in this section the adapted algorithm based on SVM.

The methodology proposed in [Biau *et al.*, 2005] is based on projection operators described in section 6.4.2, more precisely on the usage of a Hilbert basis of $\mathcal{X}$. In order to build a SVM classifier based on N examples, one need to choose from the data several parameters (in addition to the weights $\{\alpha_i\}_{1 \leq i \leq N}$ and b in problem $(D_{C,\mathcal{H}})$):

1. the projection size parameter d, i.e., the dimension of the subset V_d on which the functions are projected before being submitted to the SVM (recall that V_d is the space spanned by $\{\Psi_j\}_{j=1,\dots,d}$);

2. C, the regularization parameter;

3. the fully specified kernel K, that is the type of the universal kernel (Gaussian, exponential, etc.) but also the parameter of this kernel such as σ for the Gaussian kernel $K(u,v) = e^{-\sigma^2\|u-v\|^2}$.

Let us denote $\mathcal{A}$ the set of lists of parameters to explore (see section 6.5.3 for practical examples). Following [Biau *et al.*, 2005] we use a validation approach to choose the best list of parameters $a \in \mathcal{A}$ and in fact the best classifier on the validation set.

The data are split into two sets: a training set $\{(x_i, y_i), i = 1, \dots, l_N\}$ and a validation set $\{(x_i, y_i), i = l_N + 1, \dots, N\}$. For each fixed list a of parameters, the training set $\{(x_i, y_i), i = 1, \dots, l_N\}$ is used to calculate the SVM classification rule $f_a(x) = \text{sign}\sum_{i=1}^{l_N} \alpha_i^* y_i K(P_{V_d}(x), P_{V_d}(x_i)) + b^*$ where $(\{\alpha_i^*\}_{1\le i\le l_N}, b^*)$ is the solution of $(D_{C,\mathcal{H}})$ applied to the projected data $\{P_{V_d}(x_i), i = 1, \dots, l_N\}$ (please note that everything should be indexed by a, for instance one should write K_a rather than K).

The validation set is used to select the optimal value of a in $\mathcal{A}$, a^*, according to estimation of the generalization error based on a penalized empirical error, that is, we define

$$a^* = \arg\min_{a\in\mathcal{A}} \widehat{L} f_a + \frac{\lambda_a}{\sqrt{N - l_N}},$$

where

$$\widehat{L} f_a = \frac{1}{N - l_N} \sum_{n=l_N+1}^{N} \mathbb{1}_{\{f_a(x_n)\neq y_n\}},$$

and λ_a is a penalty term used to avoid the selection of the most complex models (i.e., the one with the highest d in general). The classifier f_N is then chosen as $f_N = f_{a^*}$.

6.5.3 Consistency

Under some conditions on $\mathcal{A}$, the algorithm proposed in the previous section is consistent. We assume given a fixed Hilbert basis of the separable Hilbert space $\mathcal{X}$, $\{\Psi_j\}_{j\ge1}$. When the dimension of the projection space V_d is chosen, a fully specified kernel K has to be chosen in a finite set of kernels, $\mathcal{J}_d$. The regularization parameter C can be chosen in a bounded interval of the form $[0, \mathcal{C}_d]$, for instance thanks to the algorithm proposed in [Hastie *et al.*, 2004] that allows to calculate the validation performances for all values of C in a finite time. Therefore, the set $\mathcal{A}$ can be written $\bigcup_{d\ge1}\{d\} \times \mathcal{J}_d \times [0, \mathcal{C}_d]$. An element of $\mathcal{A}$ is a triple $a = (d, K, C)$ that specifies the projection operator P_{V_d}, the kernel K (including all its parameters) and the regularization constant C.

Let us first define, for all $\epsilon > 0$, $\mathcal{N}(\mathcal{H}, \epsilon)$ the covering number of the Hilbert space $\mathcal{H}$ which is the minimum number of balls with radius ϵ that are needed to cover the whole space $\mathcal{H}$ (see e.g., chapter 28 of [Devroye *et al.*, 1996]). Note that in SVM, as $\mathcal{H}$ is induced by a kernel K, this number is closely related to the kernel; in this case, we will then denote the covering number $\mathcal{N}(K, \epsilon)$. For example, Gaussian kernels are known to induce feature spaces with covering number of the form $\mathcal{O}(\epsilon^{-d})$ where d is the dimension of the input space (see [Steinwart, 2002]).

Then we have:

Theorem 6.1. *We assume that X takes its values in a bounded subspace of the separable Hilbert space $\mathcal{X}$. We suppose that,*

$$\begin{array}{lll} \forall d \geq 1, & \mathcal{J}_d \text{ is a finite set,} & \\ & \exists K_d \in \mathcal{J}_d \text{ such that:} & K_d \text{ is universal,} \\ & & \exists \nu_d > 0 : \ \mathcal{N}(K_d, \epsilon) = \mathcal{O}(\epsilon^{-\nu_d}), \\ & \mathcal{C}_d > 1, & \end{array}$$

and that

$$\sum_{d \geq 1} |\mathcal{J}_d| e^{-2\lambda_d^2} < +\infty,$$

and finally that

$$\lim_{N \to +\infty} l_N = +\infty \qquad \lim_{N \to +\infty} N - l_N = +\infty$$
$$\lim_{N \to +\infty} \frac{l_N \log(N - l_N)}{N - l_N} = 0.$$

Then, the functional SVM $f_N = f_{a^*}$ chosen as described in section 6.5.2 (where a^* is optimal in $\mathcal{A} = \bigcup_{d \geq 1} \{d\} \times \mathcal{J}_d \times [0, \mathcal{C}_d]$) is consistent that is:

$$Lf_N \xrightarrow{N \to +\infty} L^*.$$

The proof of this result is given in Appendix 6.8. It is close from the proof given in [Biau *et al.*, 2005] except that in [Biau *et al.*, 2005] the proof follows from an oracle inequality given for a finite grid search model. The grid search is adapted to the classifier used in the paper (a k-nearest neighbor method), but not to our setting. Our result includes the search for a parameter C which can belong to an infinite and non countable set; this can be done by the use of the shatter coefficient of a particular class of linear classifiers which provides the behavior of the classification rule on a set of $N - l_N$ observations (see [Devroye *et al.*, 1996]).

As pointed out before, the Gaussian kernel satisfies the hypothesis of the theorem. Therefore, if $\mathcal{I}_d$ contains a Gaussian kernel for all d, then consistency of the whole procedure is guaranteed. Other non universal kernels can of course be included in the search for the optimal model.

Remark 6.1. Note that, in this theorem, the sets $\mathcal{J}_d$ and $[0, \mathcal{C}_d]$ depend on d: this does not influence the consistency of the method. In fact, one could have chosen the same set for every d, and $\mathcal{J}_d$ could also contain a single Gaussian kernel with any parameter $\sigma > 0$. In practice however, this additional flexibility is very useful to adapt the model to the data, for instance by choosing on the validation set an optimal value for σ with a Gaussian kernel.

6.6 Applications

We present, in this section, several applications of the functional SVM models described before to real world data. The first two applications illustrate the consistent methodology introduced in section 6.5.2: one has an input variable with a high number of discretization points and the second have much less discretization points. Those applications show that more benefits are obtained from the functional approach when the data can be reasonably

considered as functions, that is when the number of discretization points is higher than the number of observations.

The last application deals with spectrometric data and allows to show how a functional transformation (derivative calculation) can improve the efficiency of SVMs. For this application, we do not use the consistent methodology but a projection on a spline space that permits easy derivative calculations.

6.6.1 Speech recognition

We first illustrate in this section the consistent learning procedure given in section 6.5. We compare it to the original procedure based on k-nn described in [Biau *et al.*, 2005]. In practice, the only difference between the approaches is that we use a SVM whereas [Biau *et al.*, 2005] uses a k-nn.

The problems considered in [Biau *et al.*, 2005] consist in classifying speech samples[1]. There are three problems with two classes each: classifying "yes" against "no", "boat" against "goat" and "sh" against "ao". For each problem, we have 100 functions. Table 6.1 gives the sizes of the classes for each problem.

Problem	Class 1	Class -1
yes/no	48	52
boat/goat	55	45
sh/ao	52	48

Table 6.1: Sizes of the classes

Each function is described by a vector in $\mathbb{R}^{8192}$ which corresponds to a digitized speech frame. The goal of this benchmark is to compare data processing methods that make minimal assumptions on the data: no prior knowledge is used to preprocess the data.

In order to directly compare to results from [Biau *et al.*, 2005], performances of the algorithms are assessed by a leave-one-out procedure: 99 functions are used as the learning set (to which the split sample procedure is applied to choose SVM) and the remaining function provides a test example.

While the procedure described in 6.5.2 allows to choose most of the parameters, both the basis $\{\Psi_j\}_{j\geq 1}$ and the penalty term λ_d can be freely chosen. To focus on the improvement provided by SVM over k-nn, we have used the same elements as [Biau *et al.*, 2005]. As the data are temporal patterns, [Biau *et al.*, 2005] relies on the Fourier basis (moreover, the Fast Fourier Transform allows an efficient calculation of the coordinates of the data on the basis). The penalty term is 0 for all d below 100 and a high value (for instance 1000) for $d > 100$. This allows to only evaluate the models for $d \leq 100$ because the high value of λ_d for higher d prevents the corresponding models to be chosen, regardless of their performances. As pointed out in [Biau *et al.*, 2005], this choice appears to be safe as most of the dimensions then selected are much smaller than 50.

The last free parameter is the split between the training set and the validation set. As in [Biau *et al.*, 2005] we have used the first 50 examples for training and the remaining 49 for validation. We report the error rate for each problem and several methods in tables 6.2 and 6.3.

Table 6.2 has been reproduced from [Biau *et al.*, 2005]. QDA corresponds to Quadratic Discriminant Analysis performed, as for k-nn, on the projection of the data onto a finite dimensional subspace induced by the Fourier basis. Table 6.3 gives results obtained with

[1]Data are available at `http://www.math.univ-montp2.fr/~biau/bbwdata.tgz`

Problem	k-nn	QDA
yes/no	10%	7%
boat/goat	21%	35%
sh/ao	16%	19%

Table 6.2: Error rate for reference methods (leave-one out)

Problem/Kernel	linear (direct)	linear (projection)	Gaussian (projection)
yes/no	58%	19%	10%
boat/goat	46%	29%	8%
sh/ao	47%	25%	12%

Table 6.3: Error rate for SVM based methods (leave-one out)

SVMs. The second column, "linear (direct)", corresponds corresponds to the direct application of the procedure described in 6.3.1, without any prior projection. This is in fact the plain linear SVM directly applied to the original data. The two other columns corresponds to the SVM applied to the projected data, as described in section 6.5.2.

The most obvious fact is that the plain linear kernel gives very poor performances, especially compared to the functional kernels on projections: its results are sometimes worse than the rule that affects any observation to the dominating class. This shows that the ridge regularization of problem (R_λ) is not adapted to functional data, a fact that was already known in the context of linear discriminant analysis [Hastie *et al.*, 1995]. The projection operator improves the results of the linear kernel, but not enough to reach the performance levels of k-nn. It seems that the projected problem is therefore non linear.

As expected, the functional Gaussian SVM performs generally better than k-nn and QDA, but the training times of the methods are not comparable. On a mid range personal computer, the full leave-one-out evaluation procedure applied to Gaussian SVM takes approximately one and half hour (using LIBSVM [Chang and Lin, 2001] embedded in the package e1071 of the R software [Team, 2005]), whereas the same procedure takes only a few minutes for k-nn and QDA.

The performances of SVM with Gaussian kernel directely used on the raw data (in $\mathbb{R}^{8192}$) are not reported here as they are quite meaningless. The results are indeed extremely sensitive to the way the grid search is conducted, especially for the value of C, the regularization parameter. On the "yes/no" data set for instance, if the search grid for C contains only values higher than 1, then the leave-one-out gives 19% of error. But in each case, the value $C = 1$ is selected on the validation set. When the grid search is extended to smaller values, the smallest value is always selected and the error rate increases up to 46%. Similar behaviors occur for the other data sets. On this benchmark, the performances depend in fact on the choice of the search grid for C. This is neither the case of the linear kernel on raw data, nor the case for the projection based kernels. This is not very surprising as Gaussian kernels have some locality problems in very high dimensional spaces (see [François *et al.*, 2005]) that makes them difficult to use.

6.6.2 Using wavelet basis

In order to investigate the limitation of the direct use of the linear SVM, we have applied them to another speech recognition problem. We studied a part of TIMIT database which

was used in [Hastie *et al.*, 1995][2]. The data are log-periodograms corresponding to recording phonemes of 32 ms duration (the length of each log-periodogram is 256). We have chosen to restrict ourselves to classifying "aa" against "ao", because this is the most difficult sub-problem in the database. The database is a multi-speaker database. There are 325 speakers in the training set and 112 in the test set. We have 519 examples for "aa" in the training set (759 for "ao") and 176 in the test set (263 for "ao"). We use the split sample approach to choose the parameters on the training set (50% of the training examples are used for validation) and we report the classification error on the test set.

Here, we do not use a Fourier basis as the functions are already represented in a frequency form. As the data are very noisy, we decided to use a hierarchical wavelet basis (see e.g., [Mallat, 1989]). We used the same penalty term as in 6.6.1. The error rate on the test set is reported in table 6.4. It appears that functional kernels are not as useful here as

Functional Gaussian SVM	Functional linear SVM	Linear SVM
22%	19.4%	20%

Table 6.4: Error rate for all methods on the test set

in the previous example, as a linear SVM applied directly to the discretized functions (in $\mathbb{R}^{256}$) performs as well as a linear SVM on the wavelet coefficients. A natural explanation is that the actual dimension of the input space (256) is smaller than the number of training examples (639) which means that evaluating the optimal coefficients of the SVM is less difficult than in the previous example. Therefore, the additional regularization provided by reducing the dimension with a projection onto a small dimensional space is not really useful in this context.

6.6.3 Spectrometric data set

We study in this section spectrometric data from food industry[3]. Each observation is the near infrared absorbance spectrum of a meat sample (finely chopped), recorded on a Tecator Infratec Food and Feed Analyser (we have 215 spectra). More precisely, an observation consists in a 100 channel spectrum of absorbances in the wavelength range 850–1050 nm (see figure 6.1). The classification problem consists in separating meat samples with a high fat content (more than 20%) from samples with a low fat content (less than 20%).

It appears on figure 6.1 that high fat content spectra have sometimes two local maxima rather than one: we have therefore decided to focus on the curvature of the spectra, i.e., to use the second derivative. The figure 6.2 shows that there is more differences between the second derivatives of each class than between the original curves.

The data set is split into 120 spectra for learning and 95 spectra for testing. The problem is used to compare standard kernels (linear and Gaussian kernels) to a derivative based kernel. We do not use here the consistent procedure as we choose a fixed spline subspace to represent the functions so as to calculate their second derivative. However, the parameters C and σ are still chosen by a split sample approach that divides the 120 learning samples into 60 spectra for learning and 60 spectra for validation. The dimension of the spline subspace is obtained thanks to a leave-one-out procedure applied to the whole set of input functions, without taking into account classes (see [Rossi *et al.*, 2005] for details).

[2]Data are available at `http://www-stat.stanford.edu/~tibs/ElemStatLearn/datasets/phoneme.data`
[3]Data are available on statlib at `http://lib.stat.cmu.edu/datasets/tecator`

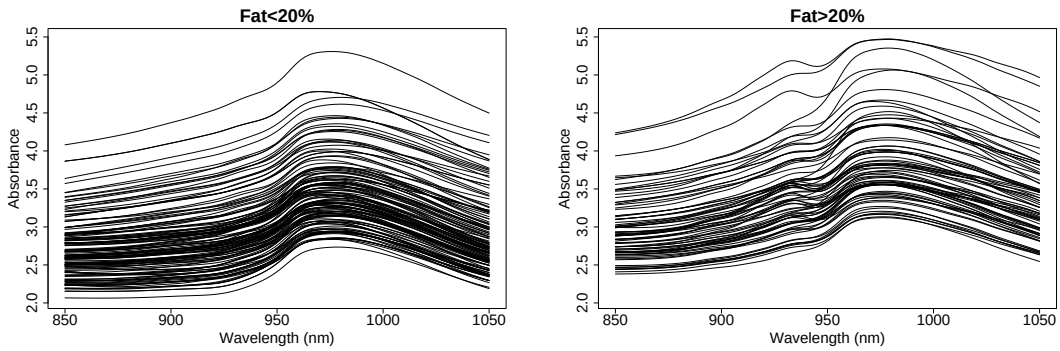

Figure 6.1: Spectra for both classes

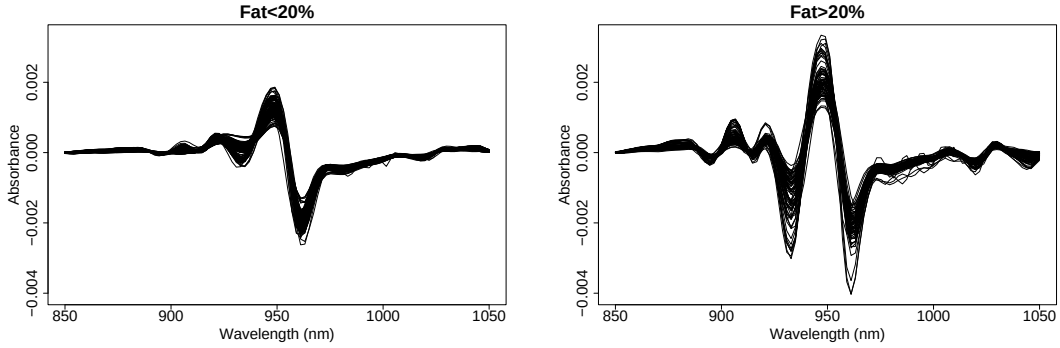

Figure 6.2: Second derivatives of the spectra for both classes

The performances depend of course on the random split between learning and test. We have therefore repeated this splitting 250 times (as we do not select an optimal projection dimension, the procedure is much faster than the one used for both previous experiments). Table 6.5 gives the mean error rate of those experiments on the test set.

Kernel	mean test error
Linear	3.38%
Linear on second derivatives	3.28%
Gaussian	7.5%
Gaussian on second derivatives	2.6%

Table 6.5: Mean test error rate for all methods

The results show that the problem is less difficult that the previous ones. Nevertheless, it also appears that a functional transformation improves the results: the use of a Gaussian kernel on second derivatives gives significantly better results than the use of an usual kernel (linear or Gaussian) on the original data (t-test results). The relatively bad performances of the Gaussian kernel on plain data can be explained by the fact that a direct comparison of spectra based on their $L^2(\mu)$ norm is in general dominated by the mean value of those spectra which is not a good feature for classification in spectrometric problems. The linear kernel is less sensitive to this problem and is not really improved by the derivative operator. In the Gaussian case, the use of a functional transformation introduces expert knowledge (i.e., curvature is a good feature for some spectrometric problems) and allows to overcome most of the limitations of the original kernel.

6.7 Conclusion

In this paper, we have shown how to use Support Vector Machines (SVMs) for functional data classification. While plain linear SVMs could be used directly on functional data, we have shown the benefits of using adapted functional kernels. We have indeed define projection based kernels that provide a consistent learning procedure for functional SVMs. We have also introduced transformation based kernels that allow to take into account expert knowledge (such as the fact that the curvature of a function can be more discriminant than its values in some applications). Both type of kernels have been tested on real world problems. The experiments gave very satisfactory results and showed that for some types of functional data, the performances of SVM based classification can be improved by using kernels that make use of the functional nature of the data.

6.8 Proofs

In order to simplify the notations, we denote $l = l_N$ when N is obvious. We also denote $X^{(d)} = P_{V_d}(X)$ and $x_i^{(d)} = P_{V_d}(x_i)$.

The proof of the consistency result of [Biau *et al.*, 2005] is based on an oracle. We

demonstrate a similar inequality: for N large enough,

$$Lf_{a^*} - L^* \leq \inf_{d\geq 1}\left[L_d^* - L^* + \inf_{C\in\mathcal{I}_d,\ K\in\mathcal{J}_d}(Lf_a - L_d^*) + \frac{\lambda_d}{\sqrt{m}}\right] + \sqrt{\frac{32(l+1)\log m}{m}} + 128\Delta\sqrt{\frac{1}{32m(l+1)\log m}} \quad (6.1)$$

where $m = N - l$, $\Delta \equiv \sum_{d\geq 1}|\mathcal{J}_d|e^{-\lambda_d^2/32} < +\infty$ and L_d^* is the Bayes error for the projected problem, i.e. $L_d^* = \inf_{f:\mathbb{R}^d\to\{-1;1\}}\mathbb{P}(f(X^{(d)}) \neq Y)$.

Following [Biau *et al.*, 2005], we see that the definition of $a^* = (d^*, K^*, C^*)$ leads to,

$$\widehat{L}f_{a^*} + \frac{\lambda_{d^*}}{\sqrt{m}} \leq \widehat{L}f_a + \frac{\lambda_d}{\sqrt{m}}$$

for all $a = (d, C, K)$ in $\mathcal{A} = \bigcup_{d\geq 1}\{d\}\times\mathcal{J}_d\times[0,\mathcal{C}_d]$. Then, for all $\epsilon > 0$,

$$\begin{aligned}\mathbb{P}\left(Lf_{a^*} - \widehat{L}f_a > \frac{\lambda_d}{\sqrt{m}} + \epsilon\right) &\leq \mathbb{P}\left(Lf_{a^*} - \widehat{L}f_{a^*} > \frac{\lambda_{d^*}}{\sqrt{m}} + \epsilon\right)\\ &\leq \sum_{d\geq 1}\mathbb{P}\left(Lf_{(d,C^*,K^*)} - \widehat{L}f_{(d,C^*,K^*)} > \frac{\lambda_d}{\sqrt{m}} + \epsilon\right)\\ &\leq \sum_{d\geq 1,\ K\in\mathcal{J}_d}\mathbb{P}\left(Lf_{(d,C^*,K)} - \widehat{L}f_{(d,C^*,K)} > \frac{\lambda_d}{\sqrt{m}} + \epsilon\right) \quad (6.2)\end{aligned}$$

In [Biau *et al.*, 2005], the right part of the inequality is bounded by the use of the union bound on $\mathcal{A}$. Here, $[0,\mathcal{C}_d]$ is not countable and thus we can not do the same. We will then use the generalization capability of a set of linear classifiers via its shatter coefficient. Actually, when d and K are set, $f_{(d,C^*,K)}$ is an affine discrimination function built from the observation projections and the kernel K. More precisely, we have:

$$\text{for all } x \text{ in } \mathcal{X}, \qquad f_a(x^{(d)}) = \sum_{n=1}^{l}\alpha_n^* y_n K(x_n^{(d)}, x^{(d)}) + b^*.$$

Then, f_a has the form $b + f$ where f is chosen in the set of functions spanned by $\{K(x_1^{(d)},.),\ldots,K(x_l^{(d)},.)\}$. Let us denote by $\mathcal{F}_K(x_1^{(d)},\ldots,x_l^{(d)})$ this set of classifiers and, for all f in $\mathcal{F}_K(x_1^{(d)},\ldots,x_l^{(d)})$, we introduce $L^l f = \mathbb{P}(f(X^{(d)}) \neq Y \mid (x_1,y_1),\ldots,(x_l,y_l))$. By Theorem 12.6 in [Devroye *et al.*, 1996], we then have, for all $\nu > 0$,

$$\mathbb{P}\left(\sup_{f\in\mathcal{F}_K(x_1^{(d)},\ldots,x_l^{(d)})}|\widehat{L}f - L^l f| > \nu \,\middle|\, (x_1,y_1),\ldots,(x_l,y_l)\right) \leq 8\mathcal{S}(\mathcal{F}_K(x_1^{(d)},\ldots,x_l^{(d)}),m)e^{-m\nu^2/32},$$

where $\mathcal{S}(\mathcal{F}_K(x_1^{(d)},\ldots,x_l^{(d)}),m)$ is the shatter coefficient of $\mathcal{F}_K(x_1^{(d)},\ldots,x_l^{(d)})$, that is the maximum number of different subsets of m points that can be separated by the set of classifiers $\mathcal{F}_K(x_1^{(d)},\ldots,x_l^{(d)})$. This set is a vector space of dimension less or equal to $l+1$, therefore according to chapter 13 of [Devroye *et al.*, 1996], $\mathcal{S}(\mathcal{F}_K(x_1^{(d)},\ldots,x_l^{(d)}),m) \leq m^{l+1}$.

This implies that, for all $(d, K) \in \mathbb{N}^* \times \mathcal{J}_d$,

$$\begin{aligned}
&\mathbb{P}\left(Lf_{(d,C^*,K)} - \widehat{L}f_{(d,C^*,K)} > \frac{\lambda_d}{\sqrt{m}} + \epsilon\right) \\
&= \mathbb{E}\left[\mathbb{P}\left(Lf_{(d,C^*,K)} - \widehat{L}f_{(d,C^*,K)} > \frac{\lambda_d}{\sqrt{m}} + \epsilon \middle| (x_1, y_1), \ldots, (x_l, y_l)\right)\right] \\
&\leq \mathbb{E}\left[\mathbb{P}\left(\sup_{f \in \mathcal{F}_K(x_1^{(d)}, \ldots, x_l^{(d)})} |\widehat{L}f - L^l f| > \frac{\lambda_d}{\sqrt{m}} + \epsilon \middle| (x_1, y_1), \ldots, (x_l, y_l)\right)\right] \\
&\leq 8m^{l+1} e^{-\lambda_d^2/32} e^{-m\epsilon^2/32}. \qquad (6.3)
\end{aligned}$$

Combining (6.2) and (6.3), we finally see that

$$\mathbb{P}\left(Lf_{a^*} - \widehat{L}f_a > \frac{\lambda_d}{\sqrt{m}} + \epsilon\right) \leq 8\Delta m^{l+1} e^{-m\epsilon^2/32}.$$

If Z is a positive random variable, we have obviously

$$\mathbb{E}(Z) \leq \mathbb{E}(Z \mathbb{1}_{\{Z>0\}}) = \int_0^{+\infty} \mathbb{P}(Z \geq \epsilon)\, d\epsilon.$$

For $Z = Lf_{a^*} - \widehat{L}f_a - \frac{\lambda_d}{\sqrt{m}}$, this leads, for all a in $\cup_d \{d\} \times \mathcal{I}_d \times \mathcal{J}_d$, to

$$Lf_{a^*} \leq \mathbb{E}(\widehat{L}f_a) + \frac{\lambda_d}{\sqrt{m}} + \int_0^{+\infty} \mathbb{P}\left(Lf_{a^*} - \widehat{L}f_a > \frac{\lambda_d}{\sqrt{m}} + \epsilon\right) d\epsilon.$$

Finally, following [Biau *et al.*, 2005], for all $u > 0$,

$$\begin{aligned}
\int_0^{+\infty} \mathbb{P}\left(Lf_{a^*} - \widehat{L}f_a > \frac{\lambda_d}{\sqrt{m}} + \epsilon\right) d\epsilon &\leq \int_0^u 1\, d\epsilon + \int_u^{+\infty} 8\Delta m^{l+1} e^{-m\epsilon^2/32}\, d\epsilon \\
&\leq u + 128\Delta m^{l+1} \int_u^{+\infty} \left(\frac{1}{16} + \frac{1}{m\epsilon^2}\right) e^{-m\epsilon^2/32}\, d\epsilon
\end{aligned}$$

and then

$$Lf_{a^*} \leq \mathbb{E}(\widehat{L}f_a) + \frac{\lambda_d}{\sqrt{m}} + u + \frac{128\Delta m^l}{u} e^{-mu^2/32};$$

if we set $u = \sqrt{\frac{32(l+1)\log m}{m}}$ and by the equality $\mathbb{E}(\widehat{L}f_a) = Lf_a$, we deduce that, for all a in $\mathcal{A}$,

$$Lf_{a^*} \leq Lf_a + \frac{\lambda_d}{\sqrt{m}} + \sqrt{\frac{32(l+1)\log m}{m}} + 128\Delta\sqrt{\frac{1}{32(l+1)\log m}}$$

which finally proves oracle (6.1).

We conclude thanks to the following steps:

1. $\lim_{m\to+\infty} \sqrt{\frac{32(l+1)\log m}{m}} + 128\Delta\sqrt{\frac{1}{32m(l+1)\log m}} = 0$ from the assumptions of Theorem 6.1;

2. Lemma 5 in [Biau *et al.*, 2005] shows that $L_d^* - L^* \xrightarrow{d\to+\infty} 0$;

3. Let $\epsilon > 0$. If we take a d_0 such that, for all $d \geq d_0$, $L^*_d - L^* \leq \epsilon$. To conclude, we finally have to prove that

$$\inf_{(C,K)\in\mathcal{I}_{d_0}\times\mathcal{J}_{d_0}} Lf_{(d_0,C,K)} - L^*_{d_0} \xrightarrow{N\to+\infty} 0.$$

This is a direct consequence of Theorem 2 in [Steinwart, 2002]. Let us show that the hypotheses of this theorem are fulfilled:

(a) Theorem 2 in [Steinwart, 2002] is valid for universal kernels that satisfy some requirements on their covering numbers.

As we focus on $\inf_{(C,K)\in\mathcal{I}_{d_0}\times\mathcal{J}_{d_0}} Lf_{(d_0,C,K)}$, we can choose freely the kernel and the regularization parameter in $\mathcal{I}_{d_0}\times\mathcal{J}_{d_0}$. Therefore, we choose K_{d_0} an universal kernel with covering number of the form $\mathcal{O}(\epsilon^{-\nu_{d_0}})$ for some $\nu_{d_0} > 0$ (this is possible according to our hypotheses).

(b) Theorem 2 in [Steinwart, 2002] asks for $X^{(d)}$ to take its values in a compact set of $\mathbb{R}^d$.

Actually, X is bounded in $\mathcal{X}$ so, by definition of $x \to x^{(d)}$, $X^{(d)}$ takes its values in a bounded set of $\mathbb{R}^d$ which is included in a compact set of $\mathbb{R}^d$;

(c) Finally, Theorem 2 in [Steinwart, 2002] requests a particular behavior for C_l, the regularization parameter used for l examples: C_l is such that $lC_l \to +\infty$ and $C_l = \mathcal{O}(l^{\beta-1})$ for some $0 < \beta < \frac{1}{\nu_{d_0}}$.

Let β_{d_0} be any number in $\left]0, \frac{1}{\nu_{d_0}} \wedge 1\right[$ (where $a \wedge b$ denotes the infimum between a and b). Then, let C_l be $l^{\beta_{d_0}-1}$. This defines a sequence of real numbers included in $]0,1[$ which fulfills the requirements stated above. As $\mathcal{C}_{d_0} \geq 1$ for all $l \geq 2$, we have $C_l \in [0, \mathcal{C}_{d_0}]$ therefore such choice of the regularization parameters is compatible with the hypothesis of our theorem.

This allows to apply Theorem 2 in [Steinwart, 2002] which implies that $Lf_{(d_0,(C_l),K_{d_0})}$ converges to $L^*_{d_0}$ and finally to obtain the conclusion.

Chapitre 7

SVM pour la discrimination de courbes : une approche par régression inverse

7.1 Régression inverse pour la discrimination de courbes

En préambule, rappelons rapidement le principe de la régression inverse. Dans [Ferré and Villa, 2005a], nous montrons que la régression inverse, introduite par [Li, 1991], peut être étendue au cadre fonctionnel pour la discrimination de courbes. De manière plus formelle, on considère un couple de variables aléatoires (X, Y) dans lequel X est un élément d'un espace de Hilbert séparable $(\mathcal{H}, \langle ., . \rangle)$ et Y prend ses valeurs dans $\{-1; 1\}$. On cherche donc à estimer la variable aléatoire $P = 2\mathbb{P}(Y = 1|X) - 1$ en posant le modèle suivant :

$$P = f(\langle a_1, X \rangle, \ldots, \langle a_q, X \rangle), \tag{7.1}$$

où les $(a_j)_{j=1,\ldots,q}$ sont linéairement indépendants. Le Théorème 4.1 du Chapitre 4 met en évidence que l'estimation de l'espace engendré par les $(a_j)_{j=1,\ldots,q}$, l'espace EDR, se déduit de l'estimation des valeurs propres de l'opérateur $\Gamma_X^{-1}\Gamma_{\mathbb{E}(X/Y)}$. Dans les espaces de Hilbert de dimension infinie, l'opérateur Γ_X n'est pas inversible et l'estimation des valeurs propres de $\Gamma_X^{-1}\Gamma_{\mathbb{E}(X/Y)}$ nécessite donc une procédure de filrage ou de régularisation : nous explicitons plusieurs approches possibles pour l'estimation de l'espace EDR dans le Chapitre 4. Toutes ces approches conduisent à des estimateurs $(\hat{a}_j^N)_{j=1,\ldots,q}$ des $(a_j)_j$ dont on peut démontrer la consistance.

Celle qui est mise en œuvre dans l'application présentée en Section 7.3 est l'approche par inverse généralisé : elle repose sur le fait que $\Gamma_{\mathbb{E}(X|Y)}$ est un opérateur de rang fini. Ainsi, $(\Gamma_X)^{-1/2}\Gamma_{\mathbb{E}(X|Y)}(\Gamma_X)^{-1/2}$ est également de rang fini : son inverse généralisé, qui engendre le même sous-espace propre associé aux valeurs propres non nulles, est donc facilement déterminé. En conclusion, si on note $(\Gamma^N_{\mathbb{E}(X|Y)})^{+q}$ l'inverse généralisé de $\Gamma^N_{\mathbb{E}(X|Y)}$ tronqué aux q premiers vecteurs propres, une estimation consistance des vecteurs $(a_j)_j$ est fournie par la diagonalisation de l'opérateur $((\Gamma^N_X)^{1/2}(\Gamma^N_{\mathbb{E}(X|Y)})^{+q}(\Gamma^n_X)^{1/2})^+$; c'est la méthode développée par [Ferré and Yao, 2005].

7.2 SVM fonctionnels : une approche par régression inverse

Une fois les observations projetées sur l'estimateur de l'espace EDR, la règle de classification est déterminée par l'estimation de la fonction f du modèle (7.1). Si dans [Ferré and Villa, 2005b] nous proposons une estimation de f par perceptron multi-couche, nous explorons ici l'estimation par Support Vector Machine. Ici, f est estimé par, $\forall\, x \in \text{Vect}\left\{\hat{a}_1^N, \ldots, \hat{a}_q^N\right\}$,

$$\hat{f}(x) = \sum_{n=1}^{N} \alpha_n^* y_n K(x, \mathcal{P}_{\hat{a}}(x_n)) + b^*,$$

où K est un noyau q-dimensionnel quelconque, $\mathcal{P}_{\hat{a}}$ désigne la projection sur $\text{Vect}\left\{\hat{a}_1^N, \ldots, \hat{a}_q^N\right\}$ et $(\{\alpha_n^*\}_{n=1,\ldots,N}, b^*)$ sont les solutions du problème d'optimisation :

$$\begin{array}{l} \max_\alpha \sum_{n=1}^N \alpha_n - \sum_{n,m=1}^N \alpha_n \alpha_m K(\mathcal{P}_{\hat{a}}(x_n), \mathcal{P}_{\hat{a}}(x_m)), \\ \text{sous les contraintes} \quad \sum_{n=1}^N \alpha_n y_n = 0 \\ \qquad\qquad\qquad\quad \text{et } 0 \leq \alpha_n \leq N, \text{ pour tout } n = 1, \ldots, N \end{array}$$

et $b^* = \frac{1}{|\{n:0<\alpha_n^*<C\}|} \sum_{n:0<\alpha_n^*<C} \left(y_n - \sum_{m=1}^N \alpha_m^* y_m K(\mathcal{P}_{\hat{a}}(x_m), \mathcal{P}_{\hat{a}}(x_n))\right)$. La règle de classification est ensuite déterminée par $\text{sign}(\hat{P})$ où $\hat{P}$ est l'estimateur de P construit par SVM comme décrit ci-dessus.

On voit que le modèle (7.1) est équivalent, comme pour les modèles développés dans le Chapitre 6, à la construction d'un SVM avec un noyau fonctionnel qui prend la forme suivante :

$$\forall\, x, x' \in \mathcal{H},\ \mathcal{K}(x, x') = K(\mathcal{P}_{\hat{a}}(x), \mathcal{P}_{\hat{a}}(x')).$$

Cette procédure se généralise au cas où le problème de discrimination comporte plus de 2 classes ; plusieurs approches de SVM multi-classes ont été developpées et on peut se rapporter à [Hsu and Lin, 2001] pour une description et une comparaison de ces méthodes. Certaines de ces méthodes sont des approches "tous ensemble" qui traitent globalement du problème de classification multi-classes : c'est le cas, par exemple, de la méthode développée par [Crammer and Singer, 2001] qui a récemment été améliorée, avec des résultats intéressants, par [Aiolli and Sperduti, 2005]. Pour notre part, nous avons choisi une méthode qui ne considère par le problème dans sa globalité mais qui présente l'avantage de sa grande simplicité : il s'agit de l'approche de [Vapnik, 1995], le "un contre tous" : on procède à l'estimation de chaque composant du vecteur $P = (2P(Y = \mathcal{C}_1|X) - 1, \ldots, 2P(Y = \mathcal{C}_K|X) - 1)$ en itérant K fois la procédure ci-dessus. L'estimation de la classe d'appartenance est alors déterminée par :

$$\mathcal{C}_{\hat{k}} = \arg\max_{(\mathcal{C}_k)_{k=1,\ldots,K}} \hat{P}_k.$$

Nous signalons enfin qu'une méthode de "un contre un" existe aussi mais qu'elle nécessite la construction de $K(K-1)/2$ SVM (au lieu de K pour la méthode que nous avons choisie). Dans [Hsu and Lin, 2001], les comparaisons effectuées sur un grand nombre de jeux de données montre que les diverses approches donnent des résultats similaires en terme de performance.

Dans la section suivante, nous appliquons cette méthode à la résolution d'un problème de discrimination à trois classes.

7.3 Application

Dans cette Section, nous revenons sur un jeu de données simulées qui fait figure d'étalon pour la discrimination de courbes. La base de données est composée de courbes discrétisées en 21 points ($t = 1, 2, \ldots, 21$) qui sont issues de 3 familles différentes :

Classe 1 : $t \to uh_1(t) + (1-u)h_2(t) + \epsilon(t)$;

Classe 2 : $t \to uh_1(t) + (1-u)h_3(t) + \epsilon(t)$;

Classe 3 : $t \to uh_2(t) + (1-u)h_2(t) + \epsilon(t)$.

où u est une variable aléatoire de loi uniforme sur $[0; 1]$, $\epsilon(t)$ est une variable aléatoire indépendante de u de loi normale centrée réduite et

$$h_1(t) = \max(6 - |t - 11|, 0), \qquad h_2(t) = h_1(t - 4) \qquad \text{et} \qquad h_3(t) = h_1(t + 4).$$

Ce problème a déjà été présenté dans le Chapitre 4 sous le nom de "waveform data set".

Nous choisissons une méthodologie proche de celle de [Hastie *et al.*, 1994] en déterminant les paramètres des modèles comparés par validation croisée sur un échantillon indépendant de 500 courbes générées au hasard dans une des trois classes. L'apprentissage est effectué sur 300 courbes (100 pour chaque classe) et l'erreur est estimée sur un troisième échantillon (de test) de 500 courbes. Afin de démontrer l'intérêt de l'utilisation conjointe d'une approche par régression inverse et d'un SVM, nous comparons les méthodes suivantes :

SIR-SVM est la méthode décrite ci-dessus où le noyau K est le noyau gaussien usuel. Les paramètres à déterminer dans ce modèle sont la dimension de l'espace EDR (q), le paramètre du noyau (σ) et le paramètre de régularisation du SVM (C) ;

SVM est la méthode consistant à traiter directement les données discrétisées par un SVM à noyau gaussien ;

R-PDA est le modèle "Ridge-PDA", modèle d'analyse discriminante régularisé par pénalisation, décrit dans [Hastie *et al.*, 1995] ;

SIR-N est un modèle de régression inverse fonctionnel tel qu'il est décrit dans la Section 7.1 mais ici la fonction f n'est pas estimée par un SVM mais par un noyau. C'est aussi l'approche développée dans [Ferré and Villa, 2005a] (Chapitre 4).

L'expérience est réalisée 10 fois ; à chaque itération, les paramètres optimaux sont déterminés par validation croisée dans une grille de recherche (on trouvera dans le tableau 7.1 les valeurs des grilles de recherche.

Méthodes	Paramètres	Grille de recherche
SVM et SIR-SV	C (paramètre de régularisation)	$10^{0\ldots5}$
	σ (paramètre du noyau gaussien)	$10^{-1\ldots2}$
SIR-N	h (fenêtre du noyau)	$10^{-3\ldots2}$
	q (dimension SIR)	2
R-PDA	α (paramètre de régularisation)	$10^{-3\ldots2}$
	q (dimension AFD)	2

TAB. 7.1 – Valeurs des grilles de recherche pour les méthodes testées

Les résultats obtenus sont résumés dans le tableau 7.2.

La première remarque que l'on peut faire au vu de ces résultats est que les quatre méthodes obtiennent des résultats proches : la différence entre la moyenne de SIR-SVM

	SIR-SVM	SVM	R-PDA	SIR-N
Moyenne (test)	13,70	15,46	15,62	14,16
Ecart type (test)	2,25	3,04	2,05	2,01
Minimum (test)	10,20	12,20	12,60	12,00
Moyenne (apprentissage)	11,73	10,17	12,47	12,37

TAB. 7.2 – Statistiques sur les pourcentages de mauvais classements obtenus sur 10 échantillons aléatoires

et celle de SIR-N s'explique par seulement 2,3 courbes mal classées d'écart (en moyenne). On peut néanmoins remarquer le bon comportement expérimental de SIR-SVM qui obtient la meilleure moyenne et également le plus petit minimum (avec cette fois-ci 9 courbes mal classées de différences avec le second). C'est surtout la SIR qui apparaît comme une méthode efficace de pré-traitement des données fonctionnelles puisque SIR-N a également un bon comportement expérimental ; cependant, par rapport à une méthode non paramétrique d'estimation de f par un noyau, le principe des SVM permet un gain de performance. Enfin, on peut remarquer que le SVM utilisé sur données brutes n'obtient pas des performances remarquables et surtout, c'est la méthode qui connaît le plus grand sur-apprentissage (différence entre l'erreur en apprentissage et l'erreur en test).

Conclusion et perspectives

Conclusion et perspectives

7.4 Synthèse du travail effectué

Le but de mon travail de thèse était l'extension d'outils non linéaires performants (SVM et réseaux de neurones) à l'analyse des données fonctionnelles.

7.4.1 Intérêts du problème

Nous avons mis en évidence, à l'instar des précédents travaux dans le domaine, que ce type de données nécessitait un traitement particulier à cause, d'un point de vue théorique, de la dimension infinie des espaces de définition des variables aléatoires, et, d'un point de vue pratique, du très grand nombre de points de discrétisation des observations fonctionnelles, de leur irrégularité de mesure et des corrélations importantes qu'il existe entre divers points de discrétisation d'une même observation. Nous avons également montré que les applications dans ce domaine étaient riches en présentant des applications en reconnaissance de voix et en chimie (spectrométrie).

Dans la partie I, nous avons montré l'intérêt de l'utilisation d'un type particulier de réseaux de neurones (perceptrons multi-couches) dans un problème en grande dimension où les variables explicatives, de natures multiples, sont fortement corrélées. Ce travail, mené en collaboration avec des géographes de l'université Toulouse II, permettait la mise en évidence des bonnes qualités explicatives de cette méthode neuronale.

Enfin, l'étude des Support Vector Machine était motivée par leurs bonnes capacités de généralisation mises en évidence dans les travaux de V. Vapnik. L'idée était de profiter de celles-ci pour obtenir des résultats de consistance pour des outils fonctionnels non linéaires.

7.4.2 Approches développées

Finalement, notre travail s'est orienté autour de deux aspects de la question :

- utilisation de perceptrons multi-couches en régression et discrimination pour des variables explicatives fonctionnelles ;
- utilisation de SVM en discrimination pour des variables explicatives hilbertiennes.

Nous avons développé des approches différentes dans ces deux cas :

- Le temps d'optimisation des perceptrons multi-couches est fortement lié au nombre de neurones et donc au nombre d'entrées. Nous avons donc choisi de développer, dans ce cas-ci, une approche semi-paramétrique qui intègre une phase linéaire de réduction de la dimension (Chapitre 5).
  Ce modèle utilise une approche par SIR (Sliced Inverse Regression) qui est une méthode linéaire de réduction des données : l'idée est la recherche d'un espace de projection exhaustif (espace EDR) par décomposition spectrale d'un opérateur tenant compte des variables explicatives et de la cible. Dans ce travail, nous avons tiré profit des

résultats existants dans le domaine de la régression inverse pour données fonctionnelles (FIR). Les approches précédentes en FIR utilisaient une régularisation par filtrage des données fonctionnelles : nous avons développé une méthodologie supplémentaire utilisant une régularisation par pénalisation de l'opérateur de variance. Les avantages respectifs de ces diverses approches de FIR ont été étudiés dans le Chapitre 4 où nous les appliquons à la discrimination de courbes. Les bénéfices de l'approche par filtrage se font particulièrement sentir dans le cas où le caractère fonctionnel des données est plus marqué : dans ce cas, le fait de choisir comme base de projection des fonctions régulières est une alternative efficace au filtrage.
Notre modèle de réseau de neurones fonctionnel combine donc un pré-traitement linéaire (FIR) qui permet la projection des données sur un espace de faible dimension et un perceptron multi-couche ordinaire. Les avantages du pré-traitement sont donc multiples : il permet la construction d'un perceptron de taille raisonnable (et donc facilement optimisé) et il autorise l'utilisation d'outils informatiques déjà développés pour l'utilisation d'un perceptron multi-couches dans le cadre multidimensionnel.
Enfin, notre approche s'applique aussi bien pour des problèmes de régression que des problèmes de classification. Dans ce dernier cas toutefois, la dimension de l'espace EDR est majorée par le nombre de classes moins un ; la dimension de l'espace EDR peut donc être particulièrement faible dans le cas où le nombre de classes est peu important : notre méthode sera alors assez peu pertinante.

- Pour la mise en œuvre de l'algorithme Support Vector Machine (SVM), le nombre de points de discrétisation joue un rôle moins important que le nombre d'observations puisqu'il n'entre en compte que lors de l'évaluation des produits scalaires et non dans la phase d'optimisation même. Nous avons donc ici développé une approche plus directe que dans le cas des perceptrons multi-couches.
  Nous montrons l'intérêt de la construction de noyaux spécialement conçus pour le traitement des données fonctionnelles et mettons en évidence le rôle crucial de la régularisation (effectuée par relaxation des contraintes sur les erreurs de classification) dans le cas de problèmes où la dimension de l'espace initial est infinie. Plusieurs noyaux sont proposés ; ils sont basés sur l'utilisation, à l'intérieur du noyau, d'un pré-traitement qui tient compte de la nature fonctionelle des données. Nous développons des noyaux utilisant des opérations fonctionnelles (notamment la dérivation) et des noyaux qui sont basés sur une approche par filtrage, c'est-à-dire par projection des données sur une base hilbertienne tronquée. Enfin, nous développons une procédure de sélection des paramètres du modèle par validation croisée qui permet d'optimiser le type de noyau, les paramètres du pré-traitement fonctionnel (et notamment la dimension de projection) ainsi que le paramètre de régularisation du SVM.

7.4.3 Résultats théoriques

Nous nous sommes basés sur les travaux précédents existants, dans le cadre multidimensionnel, sur les réseaux de neurones et les SVM pour développer des résultats similaires de convergence.

Nous démontrons, tout d'abord, (Chapitre 5) que l'approche par régularisation de régression inverse fonctionnelle est consistante : l'estimation des vecteurs de l'espace EDR converge vers une base du véritable espace EDR.

Ce résultat permet ensuite (Chapitre 5) de prouver que la procédure d'estimation des paramètres du réseau de neurones fonctionnel construit par projection sur l'espace EDR estimé est consistante. De manière plus précise, nous montrons que les poids obtenus par minimisation de l'erreur empirique convergent en probabilité vers les poids optimaux théoriques.

Enfin, en ce qui concerne les SVM, nous montrons la consistance universelle des SVM avec un noyau qui incorpore une projection sur une base hilbertienne tronquée : nous démontrons que l'erreur de l'estimateur sélectionné par la procédure de validation croisée, avec ce type de noyaux, converge vers l'erreur de Bayes qui est l'erreur optimale.

7.5 Ouvertures et projets en cours

Le but de ce dernier paragraphe est de donner des idées (succinctes) sur les développements futurs de notre travail.

7.5.1 Interaction avec les sciences humaines

Deux projets tournés vers les sciences humaines et impliquant l'équipe GRIMM-SMASH sont actuellement en cours. Ceux-ci orienteront une partie de mes activités prochaines.

Tout d'abord, la collaboration avec le laboratoire GEODE, présentée en Partie I, a été prolongée par une collaboration élargie avec des équipes de trois universités différentes : l'université Toulouse le Mirail (laboratoire GEODE et équipe GRIMM-SMASH), l'université de Grenade (Espagne) et l'université de Jaen (Espagne). Ce projet, soutenu par le ministère espagnol "de Ciencia y Tecnologia" (n° BIA2003-01499) porte sur la modélisation d'anthroposystèmes montagnards méditerranéens : nous nous concentrons particulièrement sur une partie de la Sierra Nevada et confrontons diverses approches, certaines statistiques et d'autres issues des SIG.

Un projet avec des historiens issus des équipes UTAH (UMR Toulouse II / CNRS) et FRAMESPA (UMR Toulouse II / CNRS) démarre. La problématique posée par les historiens concerne l'analyse de l'espace et de la sociabilité paysanne au Moyen-Age ; nous nous proposons d'étudier cette problématique sous l'angle de la comparaison de grands graphes et de la recherche de motifs dans ceux-ci. Dans cette perspective, l'utilisation des SVM peut s'avérer tout à fait pertinente : par le biais d'un noyau (noyau de diffusion, par exemple, voir [Kondor and Lafferty, 2002]), il est en effet possible de plonger le graphe dans un espace de Hilbert de grande dimension (qui est en fait un RKHS) et d'effectuer des analyses statistiques linéaires dans cet espace (ACP, AFC, ... ; voir [Schölkopf *et al.*, 2004] et [Shawe-Taylor and Cristianini, 2004] pour des exemples de ce type d'utilisation). Ces méthodes peuvent être appliquées aussi bien pour la simplification de graphes que pour leur comparaison. Cette thématique récente est assez nouvelle par rapport aux précédents travaux effectués mais elle présente avec ceux-ci des similarités : le but est en effet d'adapter des outils statistiques récents et reconnus pour leur efficacité à des données de nature un peu inhabituelle (ici des graphes) et qui sont de grandes tailles.

7.5.2 Perspectives théoriques

Nous envisageons également une série de prolongements plus directs à ce travail de thèse. Voici quelques idées des développements envisagés pour le futur.

Tout d'abord, nous n'avons pas démontré de résultat de consistance pour un classifieur obtenu après projection sur un base FIR. En effet, le résultat de consistance présenté dans le Chapitre 5 (perceptron multi-couches : approche par régression inverse) est un résultat sur la consistance de la méthode d'optimisation qui ne nous permet pas de déduire quelque chose sur l'erreur commise. De même, le résultat de consistance démontré dans le Chapitre 6 (SVM à entrées fonctionnelles) s'applique uniquement pour une projection sur une base hilbertienne déterministe. Une des clés de ce résultat est la convergence de $\mathbb{E}(Y/\mathcal{P}_d(X))$

vers $\mathbb{E}(Y/X)$. Or (voir [Biau *et al.*, 2005]), ce résultat est obtenu grâce à la propriété de martingale de la suite $(\mathbb{E}(Y/\mathcal{P}_d(X)))_d$ (pour la filtration naturelle $\sigma(\mathcal{P}_d(X))$) ; en utilisant une base estimée, comme c'est le cas dans les méthodes où l'on effectue un pré-traitement par régression inverse, les projections $\mathcal{P}_d$ et $\mathcal{P}_{d+1}$ ne dépendent pas l'une de l'autre au travers uniquement de la projection sur le dernier vecteur estimé a_{d+1}^N : la projection est elle-même une variable aléatoire et dépend de l'ensemble des observations et la suite $(\mathbb{E}(Y/\mathcal{P}_d(X)))_d$ n'est donc pas une martingale. Par ailleurs, pour la même raison, la base même de cette preuve est mise en défaut puisqu'elle est fondée sur la comparaison entre l'erreur commise par le classifieur et l'erreur de Bayes sur l'espace de projection, qui n'est pas ici déterministe. Cette difficulté théorique est un de nos projets de travail.

Dans la même idée, nous souhaiterions pouvoir étendre le résultat de consistance pour les SVM fonctionnels aux différents noyaux que nous avons introduits. Pour les pré-traitement par projection, des problèmes similaires à ceux rencontrés par la FIR se posent : une projection sur une base ACP, par exemple, connaîtra le problème de l'estimation de la base de projection qui dépend donc des données. Si l'on considère le cas d'une projection sur une base Spline, le problème peut alors paraître plus simple : la base est déterministe mais les espaces engendrés ne sont pas emboîtés et la propriété de martingale de la suite $(\mathbb{E}(Y/\mathcal{P}_d(x)))_d$ n'est donc pas, là non plus, vérifiée. Une alternative à l'approche proposée par [Biau *et al.*, 2005] peut être trouvée dans les travaux de [Guo *et al.*, 2002] qui relient les nombres couvertures des SVM à l'opérateur de Mercer défini par le noyau du SVM : ce résultat permet de contrôler la capacité de généralisation des SVM en tenant compte uniquement de la nature du noyau et de sa décomposition spectrale. Cette approche pourrait, en outre, permettre de relier la dimension de projection d avec le nombre d'observations N : la procédure de validation croisée, qui sélectionne d de manière optimale, ne permet pas, en effet, d'éclaircir le lien existant entre ces deux quantités.

Enfin, les SVM fonctionnels n'ont été abordés que sous l'angle de la discrimination de courbes. Nous souhaiterions pouvoir les étudier dans le cadre de problèmes de régression où la variable explicative est fonctionnelle. Cette étude nous conduira à nous intéresser de plus près à la nature des RKHS induits par les différents types de noyaux et à leurs propriétés de régularisation respectives.

Annexes

Annexe A

Preuves

A.1 Preuves complètes des théorèmes du Chapitre 5

A.1.1 Démonstration du théorème 5.2 page 92

Nous commencerons la preuve de la proposition par un lemme technique :

Lemme A.1. *Notons* $\Delta_X^N = \Gamma_X^N - \Gamma_X$ *et* $\Delta_{E(X/Y)}^N = \Gamma_{E(X/Y)}^N - \Gamma_{E(X/Y)}$. *Si* $\delta^N = max\{\| \Delta_X^N \|; \| \Delta_{E(X/Y)}^N \|\}$ *et si* $(k_N)_N$ *est une suite telle que* $\sqrt{N}k_N \xrightarrow{N\to+\infty} +\infty$ *alors*

$$k_N^{-1}\delta^N \xrightarrow{\mathbb{P}, N\to+\infty} 0.$$

Preuve : Comme X a un moment d'ordre 4, par le théorème de la limite centrale,

$$\sqrt{N}(\Gamma_X^N - \Gamma_X) \xrightarrow{\mathcal{L},\ N\to+\infty} \mathcal{N}(0, \sigma^2)$$

donc, $k_N^{-1}\Delta_X^N = (k_N\sqrt{N})^{-1} \times \sqrt{N}(\Gamma_X^N - \Gamma_X) \xrightarrow{\mathbb{P},\ N\to+\infty} 0$.
Le même raisonnement appliqué à l'opérateur $\Delta_{E(X/Y)}^N$ (cf **(A3)**) conduit au résultat. □

Preuve de l'existence :

Montrons ensuite que la suite ρ_α vérifie

$$\sqrt{N}\rho_\alpha \xrightarrow{N\to+\infty} +\infty\ . \tag{A.1.1}$$

En effet, $\forall\ u \in \mathcal{S}$ tel que $\| u \| = 1$ et $\forall\ \alpha \in]0;1]$,

$$\alpha^{-1}Q_\alpha(u,u) = (\alpha^{-1} - 1) < \Gamma_X u, u > + Q_1(u,u)\ \geq \rho_1$$

puisque Γ_X est positif. Ainsi, en prenant l'infimum sur u, il vient $\alpha^{-1}\rho_\alpha \geq \rho_1 > 0$ d'où

$$0 < \sqrt{N}\alpha\rho_1 \leq \sqrt{N}\rho_\alpha$$

ce qui, via l'hypothèse **(A4)** finit de prouver le résultat annoncé.

Par la suite, on notera Ω^N l'ensemble

$$\{\omega \in \Omega\ :\ \| \Delta_X^N \| \leq \frac{1}{2}\rho_\alpha\}$$

on constate, par le **lemme A.1** que $\lim_{N\to+\infty}\mathbb{P}(\Omega^N)=1$. Or, $\forall\ \omega\in\Omega^N$, on a

$$\forall\ a\in\mathcal{S},\ \parallel a\parallel=1,\qquad \frac{1}{2}\rho_\alpha\geq\ Q_\alpha(a,a)-Q_\alpha^N(a,a)\ \geq\rho_\alpha-Q_\alpha^N(a,a)$$

d'où il vient aisément que *avec une probabilité qui tend vers 1 lorsque N tend vers $+\infty$,*

$$\forall\ a\in\mathcal{S},\ \parallel a\parallel=1,\qquad Q_\alpha^N(a,a)\geq\frac{1}{2}\rho_\alpha>0\ . \tag{A.1.2}$$

Notons alors $\mathcal{B}(0,1)=\{a\in\mathcal{S}\ :\ Q_\alpha^N(a,a)=1\}$. On remarque alors que, $\forall\ \omega\in\Omega^N$,

- par (A.1.2), $\mathcal{B}(0,1)$ est borné dans $\mathcal{L}^2_\tau$: $\mathcal{B}(0,1)$ est donc relativement compact pour la topologie faible de $\mathcal{L}^2_\tau$. On notera $\overline{\mathcal{B}(0,1)}$ l'adhérence de $\mathcal{B}(0,1)$ pour la topologie faible : $\overline{\mathcal{B}(0,1)}$ est donc faiblement compact ;
- $\overline{\mathcal{B}(0,1)}$ muni de la topologie faible est métrisable ;
- comme $\Gamma^N_{E(X/Y)}$ est de rang fini et que $\mathcal{B}(0,1)$ est borné, l'application

$$\xi\ :\ a\in\mathcal{B}(0,1)\to<\Gamma^N_{E(X/Y)}a,a>$$

est uniformément continue pour la topologie affaiblie : en effet, si $(f_i)_{i=1..r}$ sont les vecteurs propres de $\Gamma^N_{E(X/Y)}$ de valeurs propres respectives (l_i),

$$\forall\ a,b\in\mathcal{B}(0,1),\qquad |\xi(a)-\xi(b)|\leq\sum_{i=1}^{r}|l_i|\ |<f_i,a+b>|\ |<f_i,a-b>|\ .$$

Elle se prolonge donc en une application, $\tilde{\xi}$, uniformément continue sur $\overline{\mathcal{B}(0,1)}$ muni de la topologie faible. $\tilde{\xi}$ atteint ses bornes sur le compact $\overline{\mathcal{B}(0,1)}$.

Finalement, la conclusion de la proposition vient facilement : γ^N est définie pour tout $a\in\mathcal{S}$ et majorer γ^N sur $\mathcal{S}$ est équivalent à majorer $a\in\mathcal{S}\to<\Gamma^N_{E(X/Y)}a,a>$ sur $\{a\in\mathcal{S}\ :\ Q_\alpha^N(a,a)=1\}$.

Preuve de la consistance :

Dans la suite, on posera $\forall\ a\in\mathcal{S}$,

$$\begin{aligned}&\gamma_\alpha(a)=\tfrac{<\Gamma_{E(X/Y)}a,a>}{<\Gamma_X a,a>+\alpha[a,a]}\ ,\\&\gamma(a)=\gamma_0(a),\\\text{et}\qquad&\lambda_1^\alpha=\sup_{a\in\mathcal{S}}\gamma_\alpha(a)\end{aligned}$$

(λ_1^α est bien défini d'après l'hypothèse **(A2)**).
Décomposons la preuve du théorème en deux parties :

1. Montrons d'abord que $\lambda_1^N\xrightarrow{\mathbb{P},N\to+\infty}\lambda_1$:

$$\forall\ a,\qquad\frac{\gamma_\alpha(a)}{\gamma(a)}=\frac{<\Gamma_Xa,a>}{<\Gamma_Xa,a>+\alpha[a,a]}\leq1.$$

Ainsi, $\lambda_1^\alpha\leq\lambda_1$.
Or, $\forall\ a\in\mathcal{S}$, $\gamma_\alpha(a)\xrightarrow{\alpha\to0}\gamma(a)$ donc $\lambda_1\geq\lambda_1^\alpha\geq\gamma_\alpha(a_1)\to\gamma(a_1)=\lambda_1$ et finalement,

$$\lambda_1^\alpha\xrightarrow{N\to+\infty}\lambda_1. \tag{A.1.3}$$

D'autre part, $\forall\ a \in \mathcal{S}$,

$$\left| \frac{< \Gamma_X^N a, a > + \alpha[a,a]}{< \Gamma_X a, a > + \alpha[a,a]} - 1 \right| \leq \sup_{\|u\|=1,\ u \in \mathcal{S}} \left| \frac{< \Delta_X^N u, u >}{\rho_\alpha} \right| \leq \frac{\delta^N}{\rho_\alpha}$$

or, par le lemme A.1 et la relation (A.1.1), le terme de droite tend vers 0 en probabilité lorsque N tend vers $+\infty$. On en déduit donc que

$$\gamma^N(a) = \frac{< \Gamma_{E(X/Y)}^N a, a >}{< \Gamma_X a, a > + \alpha[a,a]} \left(1 + o_{\mathbb{P}}(1)\right)$$

où $o_{\mathbb{P}}(1)$ désigne une quantité qui converge uniformément en a et en probabilité vers 0.
De même, $\forall\ a \in \mathcal{S}$,

$$\begin{aligned} \left| \frac{< \Gamma_{E(X/Y)}^N a, a > - < \Gamma_{E(X/Y)} a, a >}{< \Gamma_X a, a > + \alpha[a,a]} \right| &\leq \sup_{\|u\|=1,\ u \in \mathcal{S}} \left| \frac{< \Delta_{E(X/Y)}^N u, u >}{\rho_\alpha} \right| \\ &\leq \frac{\delta^N}{\rho_\alpha} \end{aligned}$$

où le terme de droite tend vers 0 en probabilité uniformément en $a \in \mathcal{S}$. Des deux résultats précédents, on déduit que, $\forall\ a \in \mathcal{S}$,

$$\gamma^N(a) = \left(\gamma_\alpha(a) + o_{\mathbb{P}}(1)\right)\left(1 + o_{\mathbb{P}}(1)\right)$$

En remarquant finalement que γ_α est borné sur $\mathcal{S}$ par λ_1, il vient

$$\sup_a \left| \gamma^N(a) - \gamma_\alpha(a) \right| \xrightarrow{\mathbb{P}, N \to +\infty} 0. \tag{A.1.4}$$

Soit alors $\epsilon > 0$ et Ω_ϵ^N l'ensemble

$$\left\{ \sup_{a \in \mathcal{S}} |\gamma^N(a) - \gamma_\alpha(a)| \leq \epsilon \right\} \bigcap \left\{ \gamma^N \text{ atteint son maximum } \lambda_1^N \text{ sur } \mathcal{S} \right\}$$

Pour tout $\omega \in \Omega_\epsilon^N$, on a :
• $\forall\ a \in \mathcal{S}$, $\gamma^N(a) \geq \gamma_\alpha(a) - \epsilon$ d'où $\lambda_1^N \geq \lambda_1^\alpha - \epsilon$;
• $\forall\ a \in \mathcal{S}$, $\gamma^N(a) \leq \gamma_\alpha(a) + \epsilon$ d'où $\lambda_1^N \leq \lambda_1^\alpha + \epsilon$.
Ainsi, $\Omega_\epsilon^N \subset \left\{ |\lambda_1^N - \lambda_1^\alpha| \leq \epsilon \right\}$ ce qui finit de prouver que

$$\left| \lambda_1^N - \lambda_1^\alpha \right| \xrightarrow{\mathbb{P}, N \to +\infty} 0. \tag{A.1.5}$$

Ainsi, en combinant les résultats de (A.1.3) et (A.1.5), on obtient le résultat annoncé au début de la première partie de la preuve.

2. Déduisons alors du résultat précédent que

$$< \Gamma_X \left(a_1^N - a_1\right), a_1^N - a_1 > \xrightarrow{\mathbb{P}, N \to +\infty} 0$$

Soit $\epsilon > 0$. Sur l'ensemble $\Omega_{\frac{\epsilon}{2}}^N \cap \left\{ |\lambda_1^N - \lambda_1| \leq \frac{\epsilon}{2} \right\}$ (où Ω_ϵ^N est défini comme précédemment),

$$\begin{aligned} \lambda_1 \geq \gamma(a_1^N) \geq \gamma_\alpha(a_1^N) &\geq \gamma^N(a_1^N) - \frac{\epsilon}{2} = \lambda_1^N - \frac{\epsilon}{2} \\ &\geq \lambda_1 - \epsilon\ ; \end{aligned}$$

comme la probabilité de l'ensemble énoncé plus haut converge vers 1 lorsque N tend vers $+\infty$, il vient

$$\gamma(a_1^N) \xrightarrow{\mathbb{P}, N\to+\infty} \lambda_1 = \gamma(a_1). \tag{A.1.6}$$

Si l'on se place sur l'ensemble où γ^N atteint sa borne et qu'on définit $c_1^N = a_1^N - a_1$ où a_1^N est défini comme dans l'énoncé du théorème. c_1^N est donc Γ_X-orthogonal à a_1 et la conclusion du Théorème 5.1 page 90 implique que

$$\lim_{N\to+\infty} \mathbb{P}(< \Gamma_{E(X/Y)} a_1, c_1^N >=< \Gamma_X a_1, c_1^N >= 0) = 1.$$

Enfin, posons $\mu_N =< \Gamma_X c_1^N, c_1^N >$. On a donc, $\forall \omega \in \left\{< \Gamma_{E(X/Y)} a_1, c_1^N >= 0\right\}$,

$$\begin{aligned}
\lambda_1^{-1}\gamma(a_1^N) &= \frac{\lambda_1^{-1} < \Gamma_{E(X/Y)} a_1^N, a_1^N >}{1+\mu_N} \\
&= \frac{\lambda_1^{-1}\left(\lambda_1 + < \Gamma_{E(X/Y)} c_1^N, c_1^N >\right)}{1+\mu_N} \\
&= \frac{1+\lambda_1^{-1}\mu_N\gamma(c_1^N)}{1+\mu_N} \\
&\leq \frac{1+\lambda_1^{-1}\lambda_2\mu_N}{1+\mu_N}
\end{aligned}$$

Comme $\lambda_1^{-1}\lambda_2 < 1$, le terme de droite de l'inégalité précédente est inférieur à 1. Or, par (A.1.6), $\lambda_1^{-1}\gamma(a_1^N)$ converge en probabilité vers 1, ce qui prouve que

$$\frac{1+\lambda_1^{-1}\lambda_2\mu_N}{1+\mu_N} \xrightarrow{\mathbb{P}, N\to+\infty} 1$$

et, par suite, que $\mu_N \xrightarrow{\mathbb{P},\ N\to+\infty} 0$ □.

A.1.2 Démonstration du Théorème 5.3 page 95

La démonstration du théorème se fera en deux temps.
Dans un premier temps, nous allons montrer que :

$$\sup_{w\in\mathcal{W}} \left| \frac{1}{N}\sum_{n=1}^{N} \zeta(\tilde{Z}_N^n, w) - \mathbb{E}(\zeta(Z,w)) \right| \xrightarrow{\mathbb{P},\ N\to+\infty} 0. \tag{A.1.7}$$

En effet, $\forall\ w \in \mathcal{W}$, on a l'inégalité suivante :

$$\left| \frac{1}{N}\sum_{n=1}^{N} \zeta(\tilde{Z}_n^N, w) - \mathbb{E}(\zeta(Z,w)) \right|$$

$$\leq \left| \frac{1}{N}\sum_{n=1}^{N} \zeta(\tilde{Z}_n^N, w) - \frac{1}{N}\sum_{n=1}^{N} \zeta(Z_n, w) \right| + \left| \frac{1}{N}\sum_{n=1}^{N} \zeta(Z_n, w) - \mathbb{E}(\zeta(Z,w)) \right|.$$

Nous allons montrer le résultat de convergence attendu pour les différents termes du membre de droite de l'inégalité ; la démonstration du résultat annoncé se fera donc en 2 étapes :

1. Montrons tout d'abord que :
$\sup_{w\in\mathcal{W}} \left| \frac{1}{N}\sum_{n=1}^{N} \zeta(Z_n, w) - \mathbb{E}(\zeta(Z,w)) \right| \xrightarrow{ps,\ N\to+\infty} 0$:
Fixons $\tilde{w} \in \mathcal{W}$. Par convergence dominée, on montre facilement que

$$\lim_{\mu\to 0} \mathbb{E}\left(\sup_{w\in\mathcal{W}\cap\mathcal{B}(\tilde{w},\mu)} \zeta(Z,w) \right) = \mathbb{E}(\zeta(Z,\tilde{w}))$$

où $\mathcal{B}(\tilde{w},\mu)$ est la boule centrée en $\tilde{w}$ de rayon μ. Ce résultat est une conséquence des remarques suivantes :
- Comme $\mathcal{W}$ est compact, il existe une partie dénombrable, $\{\omega_i\}_{i\in\mathbb{N}}$, partout dense dans $\mathcal{W}$; ainsi, on montre facilement que, puisque $\forall\ z \in \mathcal{O}$, $\zeta(z,.)$ est continue,

$$\sup_{w\in\mathcal{W}\cap\mathcal{B}(\tilde{w},\mu)} \zeta(.,w) = \sup_{w\in\{\omega_i\}_i\cap\mathcal{B}(\tilde{w},\mu)} \zeta(.,w).$$

  Comme $\forall\ w \in \mathcal{O}$, $\zeta(.,w)$ est mesurable, il vient que $\sup_{w\in\mathcal{W}\cap\mathcal{B}(\tilde{w},\mu)} \zeta(.,w)$ est mesurable.
- $\forall\ z \in \mathcal{O}$, $\zeta(z,.)$ est continue donc

$$\lim_{\mu\to 0} \sup_{w\in\mathcal{W}\cap\mathcal{B}(\tilde{w},\mu)} \zeta(z,w) = \zeta(z,\tilde{w}).$$

- Par l'hypothèse **(A7)**, $\forall\ z \in \mathcal{O}$,

$$\left| \sup_{w\in\mathcal{W}\cap\mathcal{B}(\tilde{w},\mu)} \zeta(z,w) \right| \leq \tilde{\zeta}(z)$$

  où $\mathbb{E}(\tilde{\zeta}(Z)) < +\infty$.

Soit alors $\epsilon > 0$. $\forall\ \tilde{w} \in \mathcal{W}$, $\exists \mu(\tilde{w})$:

$$\mathbb{E}\left(\sup_{w\in\mathcal{W}\cap\mathcal{B}(\tilde{w},\mu(\tilde{w}))} \zeta(Z,w) \right) \leq \mathbb{E}(\zeta(Z,\tilde{w})) + \frac{\epsilon}{3}\ ; \tag{A.1.8}$$

$$\mathbb{E}\left(\inf_{w\in\mathcal{W}\cap\mathcal{B}(\tilde{w},\mu(\tilde{w}))} \zeta(Z,w) \right) \geq \mathbb{E}(\zeta(Z,\tilde{w})) - \frac{\epsilon}{3}. \tag{A.1.9}$$

D'autre part, par la loi forte des grands nombres, $\forall\, \tilde{w} \in \mathcal{W}$, presque sûrement, $\exists\, N(\tilde{w}) \in \mathbb{N} : \forall\ N \geq N(\tilde{w})$,

$$\frac{1}{N}\sum_{n=1}^{N} \sup_{w\in\mathcal{W}\cap\mathcal{B}(\tilde{w},\mu(\tilde{w})} \zeta(Z^n,w) \leq \mathbb{E}\left(\sup_{w\in\mathcal{W}\cap\mathcal{B}(\tilde{w},\mu(\tilde{w}))} \zeta(Z,w) \right) + \frac{\epsilon}{3}\ ;$$

Par (A.1.8), il vient alors,

$$\frac{1}{N}\sum_{n=1}^{N} \sup_{w\in\mathcal{W}\cap\mathcal{B}(\tilde{w},\mu(\tilde{w}))} \zeta(Z^n,w) \leq \mathbb{E}(\zeta(Z,w)) + \frac{2\epsilon}{3}.$$

Combiné avec (A.1.9), le résultat précédent devient : $\forall\, \tilde{w} \in \mathcal{W}$, presque sûrement, $\exists\, N(\tilde{w}) \in \mathbb{N} : \forall\ N \geq N(\tilde{w})$,

$$\sup_{w\in\mathcal{W}\cap\mathcal{B}(\tilde{w},\mu(\tilde{w}))} \left\{ \frac{1}{N}\sum_{n=1}^{N} \zeta(Z^n,w) - \mathbb{E}(\zeta(Z,w)) \right\} \leq \epsilon;$$

on peut facilement déduire, de manière symétrique, que

$$\sup_{w\in\mathcal{W}\cap\mathcal{B}(\tilde{w},\mu(\tilde{w}))}\left|\frac{1}{N}\sum_{n=1}^{N}\zeta(Z^n,w)-\mathbb{E}(\zeta(Z,w))\right|\leq\epsilon.$$

Or, $\{\mathcal{B}(\tilde{w},\mu(\tilde{w}))\}_{\tilde{w}\in\mathcal{W}}$ forment un recouvrement ouvert de $\mathcal{W}$; ainsi, $\exists\,\tilde{w}_1,\ldots,\tilde{w}_I$ tels que

$$\mathcal{W}\subset\bigcup_{i=1}^{I}\mathcal{B}(\tilde{w}_i,\mu(\tilde{w}_i)).$$

Alors, si on fixe $\epsilon>0$ et si on note $\Omega_i(\epsilon)$ l'ensemble des $\omega\in\Omega$ tels qu'il existe $N(\tilde{w}_i)\in\mathbb{N}$ pour lequel $\forall\ N\geq N(\tilde{w}_i)$,

$$\left|\frac{1}{N}\sum_{n=1}^{N}\zeta(Z^n,w)-\mathbb{E}(\zeta(Z,w))\right|\leq\epsilon,$$

on a

$$\bigcap_{i=1}^{I}\Omega_i(\epsilon)\subset$$

$$\left\{\omega\in\Omega\ :\exists\,N_0:\forall N\geq N_0,\sup_{w\in\mathcal{W}}\left|\tfrac{1}{N}\sum_{n=1}^{N}\zeta(Z^n,w)-\mathbb{E}(\zeta(Z,w))\right|\leq\epsilon\right\}$$

(Il suffit de poser $N_0=\max_{i=1,\ldots,I}\{N(\tilde{w}_i)\}$.)
Comme l'événement du terme de gauche est presque sûr, on obtient finalement le résultat attendu :

$$\sup_{w\in\mathcal{W}}\left|\frac{1}{N}\sum_{n=1}^{N}\zeta(Z^n,w)-\mathbb{E}(\zeta(Z,w)))\right|\xrightarrow{ps,\ N\to+\infty}0.$$

2. Montrons, à présent, que

$$\sup_{w\in\mathcal{W}}\left|\frac{1}{N}\sum_{n=1}^{N}\Big(\zeta(\tilde{Z}_N^n,w)-\zeta(Z^n,w)\Big)\right|\xrightarrow{\mathbb{P},\ N\to+\infty}0,$$

ce qui finira pour démontrer le résultat (A.1.7) annoncé plus haut.
Par l'hypothèse **(A8)**,

$$\begin{aligned}\Big|\tfrac{1}{N}\textstyle\sum_{n=1}^{N}\Big(\zeta(\tilde{Z}_N^n,w)-\zeta(Z^n,w)\Big)\Big|&\\ \leq\Big[\tfrac{1}{N}\textstyle\sum_{n=1}^{N}\big(\zeta(\{<X^n,a_j^N>\}_j,Y^n,w)&\\ -\zeta(\{<X^n,a_j>\}_j,Y^n,w)\big)^2\Big]^{1/2}&\\ \leq C(w)\Big[\textstyle\sum_{j=1}^{q}<\Gamma_X^N(a_j^N-a_j),a_j^N-a_j>\Big]^{1/2}&\end{aligned}$$

Or, $\|\ \Gamma_X^N-\Gamma_X\ \|\xrightarrow{\mathbb{P},\ N\to+\infty}0$ et $\forall\ j=1,\ldots,q$, $<\Gamma_X(a_j^N-a_j),a_j^N-a_j>\xrightarrow{\mathbb{P},\ N\to+\infty}0$ (ceci implique notamment, puisque Γ_X est bijectif et continu, que $\{\|\ a_j^N-a_j\ \|\}_N$ est bornée) donc, $\forall\ j=1,\ldots,q$,

$$\begin{aligned}|<\Gamma_X^N(a_j^N-a_j),a_j^N-a_j>|\leq\|\ \Gamma_X^N-\Gamma_X\ \|\cdot\|\ a_j^N-a_j\ \|^2&\\ +|<\Gamma_X(a_j^N-a_j),a_j^N-a_j>|,&\end{aligned}$$

cette dernière quantité tendant vers 0 lorsque N tend vers $+\infty$.

Enfin, on conclut par le même argument de continuité et de compacité que dans la partie 1.

Montrons, à présent, comment obtenir la conclusion du Théorème 5.3 :

Par le théorème de convergence dominée, la fonction $w \rightarrow \mathbb{E}(\zeta(Z,w))$ est continue et atteint sont minimum m sur le compact $\mathcal{W}$ (en un point $w_{\min}$). On montre alors que :

$$\forall\, \epsilon > 0,\ \exists\, \eta(\epsilon) > 0\ :\ |\mathbb{E}(\zeta(Z,w)) - m| \leq \eta \qquad \Rightarrow \qquad d(w, \mathcal{W}^*) \leq \epsilon. \tag{A.1.10}$$

En effet, sinon, il existe $\epsilon > 0$ tel que $\forall\, N \geq 1$, $\exists\, w_N \in \mathcal{W}$ avec $|\mathbb{E}(\zeta(Z,w)) - m| \leq 1/N$ et $d(w_N, \mathcal{W}^*) > \epsilon$. Par compacité de $\mathcal{W}$, quitte à extraire une sous-suite, on peut supposer que $\{w_N\}_N$ converge vers $\tilde{w}$ avec $f(\tilde{w}) = m$ (ce qui est équivalent à $\tilde{w} \in \mathcal{W}^*$) et $d(\tilde{w}, \mathcal{W}^*) > \epsilon$. On arrive donc à une contradiction qui prouve donc (A.1.10).

Fixons $\eta > 0$ et notons $\Omega_{\eta,N}$ le sous-ensemble suivant de Ω :

$$\left\{ \omega \in \Omega\ :\ \sup_{w \in \mathcal{W}} \left| \frac{1}{N} \sum_{n=1}^{N} \zeta(\tilde{Z}_N^n, w) - \mathbb{E}(\zeta(Z,w)) \right| \leq \frac{\eta}{3} \right\}.$$

Si $\omega \in \Omega_{\epsilon,N}$ alors, par compacité de $\mathcal{W}$, il existe, pour tout $N \in \mathbb{N}$, $w_N^*(\omega) \in \mathcal{W}$ minimisant $\frac{1}{N}\sum_{n=1}^{N} \zeta(\tilde{Z}_N^n(\omega), w)$. Montrons alors que toute valeur d'adhérence, w^* de $\{w_N^*\}_N$, vérifie

$$\Omega_{\epsilon,N} \subset \{\omega\ :\ d(w^*(\omega), \mathcal{W}^*) \leq \epsilon\}$$

ce qui, puisque $\lim_{N \rightarrow +\infty} \mathbb{P}(\Omega_{\epsilon,N}) = 1$, finira pour conclure la preuve.

Il suffit pour cela, d'après la relation (A.1.10) de démontrer que

$$\mathbb{E}(\zeta(Z,w^*)) - m \leq \eta.$$

Or, soit $\{w_\phi(N)^*\}_N$ une sous-suite de $\{w_N^*\}_N$ qui converge vers w^*,

$$\begin{aligned} \mathbb{E}(\zeta(Z,w^*)) - m \quad = \quad & \mathbb{E}(\zeta(Z,w^*)) - \mathbb{E}(\zeta(Z,w_{\phi(N)}^*) + \mathbb{E}(\zeta(Z,w_{\phi(N)}^*) - \frac{1}{N}\zeta(\tilde{Z}_N^n, w_{\phi(N)}^*) \\ & + \frac{1}{N}\zeta(\tilde{Z}_N^n, w_{\phi(N)}^*) - \frac{1}{N}\zeta(\tilde{Z}_N^n, w_{\min}) + \frac{1}{N}\zeta(\tilde{Z}_N^n, w_{\min}) - m\ ; \end{aligned}$$

ainsi,

- De même que dans la partie 1. précédente, le théorème de convergence dominée et la continuité de $\zeta(z,.)$ pour tout $z \in \mathcal{O}$ permettent de dire qu'il existe N_0 tel que $\forall\ N \geq N_0$,
$$\left| \mathbb{E}(\zeta(Z,w^*)) - \mathbb{E}(\zeta(Z,w_{\phi(N)}^*)) \right| \leq \frac{\epsilon}{3}\ ;$$
ceci, par définition de $\Omega_{\eta,N}$, implique que
$$\left| \mathbb{E}(\zeta(Z,w^*)) - \frac{1}{N}\sum_{n=1}^{N} \zeta(\tilde{Z}_N^n, w_{\phi(N)}^*) \right| \leq \frac{2\epsilon}{3}\ ;$$
- D'autre part, par définition de la suite $\{w_{\phi(N)}\}_N$,
$$\frac{1}{N}\sum_{n=1}^{N} \left(\zeta(\tilde{Z}_N^n, w_{\phi(N)}^*) - \zeta(\tilde{Z}_N^n, w_{\min}) \right) \leq 0;$$
- Et enfin, par définition de l'ensemble $\Omega_{\eta,N}$,
$$\left| \frac{1}{N}\sum_{n=1}^{N} \zeta(\tilde{Z}_N^n, w_{\min}) - \mathbb{E}(\zeta(Z,w_{\min})) \right| \leq \frac{\epsilon}{3}.\ \square$$

A.2 Preuve du Théorème 1.8 page 26

La preuve du Théorème de consistance de [Biau *et al.*, 2005] est basée sur une inégalité "oracle". Nous démontrons ici une inégalité similaire : pour N suffisamment grand,

$$\begin{array}{rcl} L\phi^l_{a^*} - L^* & \leq & \inf_{d\geq 1}\left[L^*_d - L^* + \inf_{C\in\mathcal{I}_d,\ K\in\mathcal{J}_d}(L\phi^l_a - L^*_d) + \frac{\lambda_d}{\sqrt{m}}\right] \\ & & +\sqrt{\frac{32(l+1)\log m}{m}} + 128\Delta\sqrt{\frac{1}{32m(l+1)\log m}} \end{array} \tag{A.2.1}$$

où $m = N - l$, $\Delta \equiv \sum_{d\geq 1}|\mathcal{J}_d|e^{-\lambda_d^2/32} < +\infty$ et $L^*_d = \inf_{\phi:\mathbb{R}^d\to\{-1;1\}}\mathbb{P}(\phi(X^{(d)}) \neq Y)$ (erreur de Bayes du SVM d-dimensionnel).

En suivant la démarche de [Biau *et al.*, 2005], nous remarquons que, par définition de a^*,

$$\widehat{L}_m\phi^l_{a^*} + \frac{\lambda_{d^*}}{\sqrt{m}} \leq \widehat{L}_m\phi^l_a + \frac{\lambda_d}{\sqrt{m}}$$

pour tout $a = (d, C, K) \in \cup_{d\geq 1}\{d\}^* \times \mathcal{I}_d \times \mathcal{J}_d$. Ainsi, pour tout $\epsilon > 0$,

$$\begin{array}{rcl} \mathbb{P}\left(L\phi^l_{a^*} - \widehat{L}_m\phi^l_a > \frac{\lambda_d}{\sqrt{m}} + \epsilon\right) & \leq & \mathbb{P}\left(L\phi^l_{a^*} - \widehat{L}_m\phi^l_{a^*} > \frac{\lambda_{d^*}}{\sqrt{m}} + \epsilon\right) \\ & \leq & \sum_{d\geq 1}\mathbb{P}\left(L\phi^l_{(d,C^*,K^*)} - \widehat{L}_m\phi^l_{(d,C^*,K^*)} > \frac{\lambda_d}{\sqrt{m}} + \epsilon\right) \\ & \leq & \sum_{d\geq 1,\ K\in\mathcal{J}_d}\mathbb{P}\left(L\phi^l_{(d,C^*,K)} - \widehat{L}_m\phi^l_{(d,C^*,K)} > \frac{\lambda_d}{\sqrt{m}} + \epsilon\right) \end{array} \tag{A.2.2}$$

Dans [Biau *et al.*, 2005], le terme de droite est borné par utilisation de l'union sur l'ensemble des paramètres ; ici, l'ensemble des paramètres, $\mathcal{I}_d$, n'est pas dénombrable et il n'est donc pas possible de procéder de cette manière. Nous utilisons donc une autre stratégie qui fait intervenir la capacité de généralisation des classifieurs linéaires via leur coefficient de pulvérisation. En fait, lorsque d et K sont fixés, $\phi^l_{(d,C^*,K)}$ est une fonction de discrimination affine construite à partir des observations et du noyau K. De manière précise, nous avons :

$$\forall\, x \in \mathcal{H}, \qquad \phi^l_a(x^{(d)}) = \sum_{n=1}^{l}\alpha^*_n y_n K(x^{(d)}_n, x^{(d)}) + b^*.$$

Ainsi, ϕ^l_a est de la forme $b + \Phi$ où Φ est choisie dans l'ensemble des fonctions engendrées par $\{K(x^{(d)}_1, .), \ldots, K(x^{(d)}_l, .)\}$. Notons alors $\mathcal{F}_K(x^{(d)}_1, \ldots, x^{(d)}_l)$ cet ensemble de classifieurs et, $\forall\,\phi \in \mathcal{F}_K(x^{(d)}_1, \ldots, x^{(d)}_l)$, $L^l(\phi) = \mathbb{P}(\phi(X^{(d)}) \neq Y | (x_1, y_1), \ldots, (x_l, y_l))$. Par le Théorème 12.6 de [Devroye *et al.*, 1996], on obtient alors, $\forall\,\nu > 0$,

$$\mathbb{P}\left(\sup_{\phi\in\mathcal{F}_K(x^{(d)}_1,\ldots,x^{(d)}_l)}|\widehat{L}_m\phi - L^l\phi| > \nu \,\middle|\, (x_1, y_1), \ldots, (x_l, y_l)\right) \leq$$

$$8\mathcal{S}(\mathcal{F}_K(x^{(d)}_1, \ldots, x^{(d)}_l), m)e^{-m\nu^2/32},$$

où $\mathcal{S}(\mathcal{F}_K(x^{(d)}_1, \ldots, x^{(d)}_l), m)$ est le coefficient de pulvérisation de $\mathcal{F}_K(x^{(d)}_1, \ldots, x^{(d)}_l)$. Or, $\mathcal{S}(\mathcal{F}_K(x^{(d)}_1, \ldots, x^{(d)}_l), m) \leq m^{l+1}$ ce qui nous conduit naturellement à $\forall\,(d, K) \in \cup_{d\geq 1}\{d\} \times$

$\mathcal{J}_d$,

$$\begin{aligned}
&\mathbb{P}\left(L\phi^l_{(d,C^*,K)} - \widehat{L}_m\phi^l_{(d,C^*,K)} > \tfrac{\lambda_d}{\sqrt{m}} + \epsilon\right) \\
&\quad = \mathbb{E}\left[\mathbb{P}\left(L\phi^l_{(d,C^*,K)} - \widehat{L}_m\phi^l_{(d,C^*,K)} > \tfrac{\lambda_d}{\sqrt{m}} + \epsilon \middle| (x_1,y_1),\dots,(x_l,y_l)\right)\right] \\
&\quad \leq \mathbb{E}\left[\mathbb{P}\left(\sup_{\phi\in\mathcal{F}_K(x_1^{(d)},\dots,x_l^{(d)})} |\widehat{L}_m\phi - L^l\phi| > \tfrac{\lambda_d}{\sqrt{m}} + \epsilon \middle| (x_1,y_1),\dots,(x_l,y_l)\right)\right] \\
&\quad \leq 8m^{l+1}e^{-\lambda_d^2/32}e^{-m\epsilon^2/32}.
\end{aligned} \tag{A.2.3}$$

En combinant (A.2.2) et (A.2.3), on montre finalement que

$$\mathbb{P}\left(L\phi^l_{a^*} - \widehat{L}_m\phi^l_a > \frac{\lambda_d}{\sqrt{m}} + \epsilon\right) \leq 8\Delta m^{l+1}e^{-m\epsilon^2/32}.$$

En appliquant l'inégalité classique suivante, valable pour toute variable aléatoire Z,

$$\mathbb{E}(Z) \leq \mathbb{E}(Z\mathbb{1}_{\{Z>0\}}) = \int_0^{+\infty} \mathbb{P}(Z \geq \epsilon)\, d\epsilon$$

à $Z = L\phi^l_{a^*} - \hat{L}_m\phi^l_a - \frac{\lambda_d}{\sqrt{m}}$, on obtient, pour tout $a \in \cup_{d\geq 1}\{d\}\times\mathcal{I}_d\times\mathcal{J}_d$,

$$L\phi^l_{a^*} \leq \mathbb{E}(\widehat{L}_m\phi^l_a) + \frac{\lambda_d}{\sqrt{m}} + \int_0^{+\infty} \mathbb{P}\left(L\phi^l_{a^*} - \widehat{L}_m\phi^l_a > \frac{\lambda_d}{\sqrt{m}} + \epsilon\right) d\epsilon.$$

Or, suivant la démarche de [Biau *et al.*, 2005], pour tout $u > 0$,

$$\begin{array}{rcl}
\int_0^{+\infty} \mathbb{P}\left(L\phi^l_{a^*} - \widehat{L}_m\phi^l_a > \frac{\lambda_d}{\sqrt{m}} + \epsilon\right) d\epsilon & \leq & \int_0^u 1\, d\epsilon + \int_u^{+\infty} 8\Delta m^{l+1}e^{-m\epsilon^2/32}\, d\epsilon \\
& \leq & u + 128\Delta m^{l+1}\int_u^{+\infty}\left(\frac{1}{16} + \frac{1}{m\epsilon^2}\right)e^{-m\epsilon^2/32}\, d\epsilon
\end{array}$$

d'où l'on déduit que

$$L\phi^l_{a^*} \leq \mathbb{E}(\widehat{L}_m\phi^l_a) + \frac{\lambda_d}{\sqrt{m}} + u + \frac{128\Delta m^l}{u}e^{-mu^2/32}\ ;$$

en choisissant $u = \sqrt{\frac{32(l+1)\log m}{m}}$ et en remarquant que $\mathbb{E}(\widehat{L}_m\phi^l_a) = L\phi^l_a$, on obtient finalement que, $\forall\, a \in \cup_{d\geq 1}\{d\}\times\mathcal{I}_d\times\mathcal{J}_d$,

$$L\phi^l_{a^*} \leq L\phi^l_a + \frac{\lambda_d}{\sqrt{m}} + \sqrt{\frac{32(l+1)\log m}{m}} + 128\Delta\sqrt{\frac{1}{32(l+1)\log m}}$$

ce qui implique, finalement, l'oracle (A.2.1).

Pour conclure à partir de cette inégalité, remarquons que :

1. $\lim_{m\to+\infty}\sqrt{\frac{32(l+1)\log m}{m}} + 128\Delta n_{\mathcal{I}}\sqrt{\frac{1}{32m(l+1)\log m}} = 0$ d'après les hypothèses du Théorème 1.8 ;
2. le Lemme 5 dans [Biau *et al.*, 2005] montre que $L^*_d - L^* \xrightarrow{d\to+\infty} 0$;

3. Soit $\epsilon > 0$. Choisissons d_0 tel que, pour tout $d \geq d_0$, $L_d^* - L^* \leq \epsilon$. Pour finir, il reste donc à prouver que

$$\inf_{(C,K)\in\mathcal{I}_{d_0}\times\mathcal{J}_{d_0}} Lf_{(d_0,C,K)} - L_{d_0}^* \xrightarrow{N\to+\infty} 0.$$

Ceci est une conséquence directe du Théorème 2 in [Steinwart, 2002]. Montrons donc que les hypothèses de ce théorème sont vérifiées :

(a) Le Théorème 2 de [Steinwart, 2002] est valable pour des noyaux universels admettant des nombres couvertures qui satisfont certaines conditions.

Comme nous majorons $\inf_{(C,K)\in\mathcal{I}_{d_0}\times\mathcal{J}_{d_0}} Lf_{(d_0,C,K)}$, nous pouvons choisir librement le noyau et le paramètre de régularisation dans $\mathcal{I}_{d_0} \times \mathcal{J}_{d_0}$. Soit donc K_{d_0} un noyau universel avec un nombre couverture de la forme $\mathcal{O}(\epsilon^{-\nu_{d_0}})$ pour un $\nu_{d_0} > 0$ (ceci est toujours possible d'après nos hypothèses).

(b) Le Théorème 2 de [Steinwart, 2002] nécessite que $X^{(d)}$ prennent ces valeurs dans un ensemble compact de $\mathbb{R}^d$.

En fait, X est bornée dans $\mathcal{X}$ donc, par définition de $x \to x^{(d)}$, $X^{(d)}$ prend ses valeurs dans un ensemble borné de $\mathbb{R}^d$ qui est inclus dans un compact de $\mathbb{R}^d$;

(c) Enfin, le Théorème 2 de [Steinwart, 2002] impose une forme particulière pour C_l, le paramètre de régularisation utilisé pour l observations : C_l est tel que $lC_l \to +\infty$ et $C_l = \mathcal{O}(l^{\beta-1})$ pour un $0 < \beta < \frac{1}{\nu_{d_0}}$.

Soit β_{d_0} un nombre dans $\left]0, \frac{1}{\nu_{d_0}} \wedge 1\right[$ (où $a \wedge b$ est le minimum entre a et b). Alors, soit C_l la suite $l^{\beta_{d_0}-1}$. Ceci définit une suite de réels inclus dans $]0,1[$ qui satisfont les contraintes ci-dessus. Comme $\mathcal{C}_{d_0} \geq 1$ pour tout $l \geq 2$, nous avons $C_l \in [0, \mathcal{C}_{d_0}]$ et donc de tels paramètres de régularisation sont compatibles avec les hypothèses de notre théorème.

Ceci nous permet donc d'appliquer le Théorème 2 in [Steinwart, 2002] qui implique que $Lf_{(d_0,(C_l),K_{d_0})}$ converge vers $L_{d_0}^*$ et permet, finalement, d'obtenir la conclusion.

Annexe B

Programmes et simulations

Les programmes ci-dessous ont été écrits pour le logiciel Matlab©.

B.1 Programmes de détermination de l'espace EDR par SIR régularisée

Les programmes ci-dessous permettent d'effectuer l'estimation de l'espace EDR comme décrite dans [Ferré and Villa, 2005b] (Chapitre 5). Les données sont préalablement projetées sur une base de B-Splines pour permettre le calcul des dérivées d'ordre 2.

Fonction pour régression par FIR régularisée

```
function [vSIR,valSIR]=SIR(coef,ValeursB,ValeursD2B,y,alpha,q,H)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ESTIMATION DE L'ESPACE EDR PAR SIR REGULARISEE POUR LA REGRESSION
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Entrées :  coef : Coefficients de l'expansion des données
%initiales sur une base B-Spline (N lignes)
% ValeursB : Valeurs des fonctions de la base B-Spline
%aux points de discrétisation
% ValeursD2B : Valeurs des dérivées secondes des
%fonctions de la base B-Spline aux points de discrétisation
% y : Valeurs de la cible
% alpha : Valeur de alpha, paramètre de régularisation
% q : Valeur de q, dimension de l'espace EDR
% H : Nombre de tranches (l'estimateur de variance
%empirique de E(Y/X) est déterminé par tranchage du support
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Sorties : vSIR : estimation des vecteurs engendrant l'espace EDR
% valSIR : estimation de leurs valeurs propres associées
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% Construction de la matrice de variance de X, régularisée par pénalisation
% et inversion (inverse généralisé)
N=size(coef,1);
T=size(ValeursB,1);
pcentre=coef*ValeursB'-ones(N,1)*mean(coef*ValeursB',1);
Gamma_Xreg=ValeursB'*cov(pcentre)*ValeursB+alpha*T*(ValeursD2B'*ValeursD2B);
Gamma_Xreg=(Gamma_Xreg+Gamma_Xreg')/2;
InvGamma=InvG(Gamma_Xreg);

% Tranchage du support du régresseur et construction de la matrice de variance
% de E(X/Y)
m=min(rtrain) ; M=max(rtrain);
vinf=m:(M-m)/H:M-(M-m)/H;
vsup=m+(M-m)/H:(M-m)/H:M; vsup(H)=M+1;
Indic=(y*ones(1,H)>=ones(N,1)*vinf) & (y*ones(1,H)<ones(N,1)*vsup);
NH=sum(Indic,1);
muH=Indic'*pcentre;
for h=1:H
    if NH(h) > 0
        muH(h,:)=muH(h,:)/sqrt(NH(h));
    else
        muH(h,:)=0;
    end;
end;
Gamma_E=muH'*muH/N;
Gamma_E=ValeursB'*Gamma_E*ValeursB;

% Sous-espace SIR
Mat=InvGamma*Gamma_E*InvGamma;
Mat=(Mat+Mat')/2;
[V D]=eig(Mat);
valSIR=diag(D(end-q+1:end,end-q+1:end));
vSIR=ValeursB*InvGamma*V(:,end-q+1:end);
```

Fonction pour discrimination par FIR régularisée

```
function [vSIR,valSIR]=SIR(coef,ValeursB,ValeursD2B,classes,alpha,q)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ESTIMATION DE L'ESPACE EDR PAR SIR REGULARISEE POUR LA DISCRIMINATION
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Entrées :  coef : Coefficients de l'expansion des données
%initiales sur une base B-Spline (N lignes)
% ValeursB : Valeurs des fonctions de la base B-Spline
%aux points de discrétisation
```

```
% ValeursD2B : Valeurs des dérivées secondes des
%fonctions de la base B-Spline aux points de discrétisation
% classes : Codage disjonctif des classes
% alpha : Valeur de alpha, paramètre de régularisation
% q : Valeur de q, dimension de l'espace EDR
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Sorties : vSIR : estimation des vecteurs engendrant l'espace EDR
% valSIR : estimation de leurs valeurs propres associées
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


% Construction de la matrice de variance de X, régularisée par pénalisation
% et inversion (inverse généralisé)
N=size(coef,1);
T=size(ValeursB,1);
nbclasses=size(classes,2);
pcentre=coef*ValeursB'-ones(N,1)*mean(coef*ValeursB',1);
Gamma_Xreg=ValeursB'*cov(pcentre)*ValeursB+alpha*T*(ValeursD2B'*ValeursD2B);
Gamma_Xreg=(Gamma_Xreg+Gamma_Xreg')/2;
InvGamma=InvG(Gamma_Xreg);

% Construction de la matrice de variance de E(X/Y)
EXcondY=classes*pcentre;
Gamma_E=0;
for k=1:nbclasses
    Gamma_E=Gamma_E+(sum(classes(:,k)==1)/N*((EXcondY(k,:)'*EXcondY(k,:))));
end;
Gamma_E=ValeursB'*Gamma_E*ValeursB;

% Sous-espace SIR
Mat=InvGamma*Gamma_E*InvGamma;
Mat=(Mat+Mat')/2;
[V D]=eig(Mat);
valSIR=diag(D(end-q+1:end,end-q+1:end));
vSIR=ValeursB*InvGamma*V(:,end-q+1:end);
```

Fonction de calcul d'un inverse généralisé

```
function [InvGamma]=InvG(Gamma)

% Calcul de l'inverse généralisée à la puissance 1/2 de Gamma

% Décomposition spectrale de Gamma
T=size(Gamma,1);
R=rank(Gamma);
[vect val]=eig(Gamma);
vect=vect(:,end-R+1:end);
val=diag(val(end-R+1:end,end-R+1:end));
```

```
% Construction de la matrice
InvGamma=zeros(T,T);
for i=1:R
    InvGamma=InvGamma+val(i)^(-0.5)*vect(:,i)*vect(:,i)';
end;
```

B.2 Programmes pour SVM à entrées fonctionnelles

Le programme ci-dessous, utilisant la boite à outil svm.zip [1], met en œuvre la procédure consistante de classification de courbes par SVM fonctionnel comme décrite dans [Rossi and Villa, 2005a]. Ici, C est déterminé dans une grille de recherche mais ce programme, couplé avec les programmes développés par Trévor Hastie [2], autorise la recherche de C dans un intervalle entier.

Programme pour l'apprentissage d'un SVM fonctionnel

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% PROGRAMME D'APPRENTISSAGE D'UN SVM FONCTIONNEL AVEC VALIDATION CROISEE
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Entrees : donnees : matrice des données transformée par Fourier (fft)
% classes : matrice des classes (-1 ou 1)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


global p1

Clist=["Liste des valeurs à tester pour C"];
p1list=["Liste des valeurs à tester pour sigma"];

erreur.valid=1;

for d=1:100
    % Projection sur la base trigonométrique de dimension d
    proj.train=[real(donnees.train(1:ceil((d+1)/2),:))'
imag(donnees.train(2:floor((d+1)/2),:))'];
    proj.valid=[real(donnees.valid(1:ceil((d+1)/2),:))'
imag(donnees.valid(2:floor((d+1)/2),:))'];
    proj.test=[real(donnees.test(1:ceil((d+1)/2),:))'
imag(donnees.test(2:floor((d+1)/2),:))'];

    % SVM sur cette projection
    for ind1=1:length(Clist)
     C=Clist(ind1)
     for ind2=1:length(p1list)
        p1=p1list(ind2)
        [nsv,alpha,b] = svc(proj.train,classes.train,'rbf',C);
        taux.train{d}(ind1,ind2)=svcerror(proj.train,classes.train,proj.train,
```

[1] disponible sur `http://www.isis.ecs.soton.ac.uk`, développée par Steve Gun

[2] `http://www-stat.stanford.edu/~hastie/`

```
classes.train,'rbf',alpha,b)/length(classes.train);
        taux.valid{d}(ind1,ind2)=svcerror(proj.train,classes.train,proj.valid,
classes.valid,'rbf',alpha,b)/length(classes.valid);;
        taux.test{d}(ind1,ind2)=svcerror(proj.train,classes.train,proj.test,
classes.test,'rbf',alpha,b)/length(classes.test);
        if taux.valid{d}(ind1,ind2)<erreur.valid
            erreur.valid=taux.valid{d}(ind1,ind2);
            erreur.test=taux.test{d}(ind1,ind2);
            alpha_opt=alpha; b_opt=b; nsv_opt=nsv;
            C_opt=C; p1_opt=p1; d_opt=d;
        end
     end
    end
end
```

Programme pour le calcul de l'erreur par procédure "Leave one out"

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% PROCEDURE LEAVE ONE OUT DE CALCUL DE L'ERREUR POUR SVM FONCTIONNEL
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Entrées : donnees : matrice des données (N individus x T points de
% discrétisation
% classes : matrice des classes (-1 ou 1)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Détermination de la transformée de Fourier
fdonnees=fft(donnees')';
N=size(donnees,1);
nbapp="Nombre de courbes en apprentissage";

% Leave one out
for ind=1:100
    donnees.test=fdonnees(ind,:);
    d=fdonnees(setdiff(1:N,ind),:);
    donnees.train=fdonnees(1:nbapp,:);
    donnees.valid=fdonnees(nbapp+1:end,:);
    c.test=classes(ind);
    cla=classes(setdiff(1:N,ind));
    c.train=cla(1:nbapp);
    c.valid=cla(nbapp+1:end);

    SVM_trigo

    erreurf.valid(ind)=erreur.valid;
    erreurf.test(ind)=erreur.test;
end;
mean(erreurf)
```

Bibliographie

Bibliographie

[Aguilera *et al.*, 1997] A.M. Aguilera, F.A. Ocaña, and M.J. Valderrama. An approximated principal component prediction for continuous time stochastic processes. *Applied Stochastic Models and Data Analysis*, 13 : 61–72, 1997.

[Aiolli and Sperduti, 2005] F. Aiolli and A. Sperduti. Multiclass classification with multi-prototype support vector machines. *Journal of Machine Learning Research*, 6 : 817–850, 2005.

[Aragon and Saracco, 1997] Y. Aragon and J. Saracco. Sliced inversed regression (SIR) : an appraisal of small sample alternatives to slicing. *Computational Statistics*, 12 : 109–130, 1997.

[Aronszajn, 1950] N. Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68(3) : 337–404, 1950.

[Barron, 1994] A. Barron. Approximation and estimation bounds for artificial neural networks. *Machine Learning*, 14 : 115–133, 1994.

[Baum and Haussler, 1989] E.B. Baum and D. Haussler. What size net gives valid generalization ? *Neural Computation 1*, 151–160, 1989.

[Beale and Demuth, 1998] M. Beale and H. Demuth. *Neural network toolbox user's guide*. The Matworks Inc., version 3 edition, 1998.

[Berlinet and Thomas-Agnan, 2004] A. Berlinet and C. Thomas-Agnan. *Reproducing kernel Hilbert spaces in probability and statistics*. Kluwer Academic Publisher, 2004.

[Besse and Ramsay, 1986] P. Besse and J.O. Ramsay. Principal component analysis of sampled curves. *Psychometrica*, 51 : 285–311, 1986.

[Besse, 1991] P. Besse. Approximation spline de l'analyse en composantes principales d'une variable aléatoire hilbertienne. *Annales de la Faculté des Sciences de Toulouse*, XII(3) : 329–349, 1991.

[Biau *et al.*, 2005] G. Biau, F. Bunea, and M. Wegkamp. Functional classification in Hilbert spaces. *IEEE Transactions on Information Theory*, 51 : 2163–2172, 2005.

[Bishop, 1995] C.M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, New York, 1995.

[Boser *et al.*, 1992] B. Boser, I. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. In 5^{th} *annual ACM Workshop on COLT*, 144–152. D. Haussler Editor, ACM Press, 1992.

[Bosq, 1991] D. Bosq. Modelization, non-parametric estimation and prediction for continuous time processes, In G. Roussas, editor, *Nonparametric functional estimation and related topics, Nato ASI Series C*, volume 335, 509–529. Kluwer Academic Publishers, Dortrecht, 1991.

[Cardot *et al.*, 1993] H. Cardot, R. Faivre, and M. Goulard. Functional approaches for predicting land use with the temporal evolution of coarse resolution remote sensing data. *Journal of Applied Statistics*, 30 : 1185–1199, 1993.

[Cardot *et al.*, 1999] H. Cardot, F. Ferraty, and P Sarda. Functional Linear Model. *Statistics and Probability Letter*, 45 : 11–22, 1999.

[Cardot *et al.*, 2003] H. Cardot, F. Ferraty, and P. Sarda. Spline estimators for the functional linear model. *Statistica Sinica*, 13 : 571–591, 2003.

[Chang and Lin, 2001] C.C. Chang and C.J. Lin. *LIBSVM : a library for support vector machines*. 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[Chen and Chen, 1995] T. Chen and H. Chen. Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems. *IEEE Transactions on Neural Networks*, 6(4) : 911–917, 1995.

[Conan-Guez and Rossi, 2002] B. Conan-Guez and F. Rossi. Multi-layer perceptrons for functional data analysis : a projection based approach. In *ICANN 2002*, 667–672, Madrid, Spain, 2002.

[Cook and Weisberg, 1991] R.D. Cook and S. Weisberg. Comment on sliced inverse regression for dimension reduction by K.C. Li. *Journal of the American Statistical Association*, 86 : 328–332, 1991.

[Cook and Yin, 2001] R.D. Cook and X. Yin. Dimension reduction and visualization in discriminant analysis. *Australian & New-Zealand Journal of Statistics*, 43 : 147–199, 2001.

[Crammer and Singer, 2001] K. Crammer and Y. Singer. On the algorithmic implementation of multiclass kernel-based machines. *Journal of Machine Learning Research*, 2 : 265–292, 2001.

[Cristianini and Shawe-Taylor, 2000] N. Cristianini and J. Shawe-Taylor. *An Introduction to support vector machines*. Cambridge University Press, Cambridge, UK, 2000.

[Dauxois and Pousse, 1976] J. Dauxois and A. Pousse. *Les analyses factorielles en calcul des probabilités et en statistique : essai d'étude synthétique*. Thèse, Université Toulouse III, 1976.

[Dauxois *et al.*, 1982] J. Dauxois, A. Pousse, and Y. Romain. Asymptotic theory for the principal component analysis of a vector of random function : some applications to statistical inference. *Journal of Multivariate Analysis*, 12 : 136–154, 1982.

[Dauxois *et al.*, 2001] J. Dauxois, L. Ferré, and A.F. Yao. Un modèle semi-paramétrique pour variable aléatoire hilbertienne. *C.R. Acad. Sci. Paris*, 327(I) : 947–952, 2001.

[Davaloe and Naïm, 1969] E. Davaloe and P. Naïm. *Des réseaux de neurones*. Ed. Eyrolles, 1969.

[Deville, 1974] J.C. Deville. Méthodes statistiques et numériques de l'analyse harmonique. *Annales de l'INSEE*, 15(Janvier–Avril) : 3–97, 1974.

[Devroye *et al.*, 1996] L. Devroye, L. Györfi, and G. Lugosi. *A probabilistic theory for pattern recognition*. Springer-Verlag, New York, 1996.

[DiPillo, 1979] P. DiPillo. Biased discriminant analysis : evaluation of the optimum probability of classification. *Comm. Statist. Theory Methods*, 8 : 1447–1458, 1979.

[Eastman *et al.*, 1993] J.R. Eastman, P.A.K. Kyem, J. Toledano, and Jin W. *Explorations in geographic systems technology volume 4 : GIS and Decision Making*. UNITAR, Geneva, 1993.

[Eastman, 2001] J.R. Eastman. *Idrisi32, release 2 tutorial*. Clarklabs, Worcester, 2001.

[Evgeniou *et al.*, 2000] T. Evgeniou, M. Pontil, and T. Poggio. Regularization networks and support vector machines. *Advances in Computational Mathematics*, 13(1) : 1–50, 2000.

[Farago and Lugosi, 1993] A. Farago and G. Lugosi. Strong universal consistency of neural network classifiers. *IEEE Transactions on Information Theory*, 39(4) : 1146–1151, 1993.

[Ferraty and Vieu, 2002] F. Ferraty and P. Vieu. The functional nonparametric model and application to spectrometric data. *Computational Statistics*, 17 : 515–561, 2002.

[Ferraty and Vieu, 2003] F. Ferraty and P. Vieu. Curves discrimination : a non parametric approach. *Computational and Statistical Data Analysis*, 44 : 161–173, 2003.

[Ferré and Villa, 2005a] L. Ferré and N. Villa. Discrimination de courbes par régression inverse fonctionnelle. *Revue de Statistique Appliquée*, LIII(1) : 39–57, 2005.

[Ferré and Villa, 2005b] L. Ferré and N. Villa. Multi-layer neural network with functional inputs. *Scandinavian Journal of Statistics*, 2005. A paraître.

[Ferré and Yao, 2003] L. Ferré and A.F. Yao. Functional sliced inverse regression analysis. *Statistics*, 37 : 475–488, 2003.

[Ferré and Yao, 2005] L. Ferré and A.F. Yao. Smoothed functional inverse regression. *Statistica Sinica*, 15(3) : 665–683, 2005.

[Ferré, 1998] L. Ferré. Determining the dimension in sliced inverse regression and related methods. *Journal of the American Statistical Association*, 93 : 132–140, 1998.

[Flamm and Turner, 1994] R.O. Flamm and M.G. Turner. Alternative model formulations for stochastic simulation of landscape change. *Landscape Ecology*, 9(1) : 251–257, 1994.

[François *et al.*, 2005] D. François, V. Wertz, and M. Verleysen. About the locality of kernels in high-dimensional spaces. In *ASMDA 2005 proceedings*, 238–245, Brest, France, 2005.

[Frank and Friedman, 1993] I.E. Frank and J. H. Friedman. A statistical view of some chemometrics regression tools. *Technometrics*, 35 : 109–148, 1993.

[Friedman, 1989] J. Friedman. Regularized discriminant analysis. *Journal of the American Statistical Association*, 84 : 165–175, 1989.

[Girosi, 1997] F. Girosi. An equivalence between sparse approximation and support vector machines. *A.I. Memo No. 1606, MIT*, 1997.

[Guo *et al.*, 2002] Y. Guo, P.L. Bartlett, J. Shawe-Taylor, and R.C. Williamson. Covering numbers for support vector machines. *IEEE Transactions on Information Theory*, 48(1) : 239–250, 2002.

[Hand, 1982] D.J. Hand. *Kernel Discriminant Analysis*. Research Studies Press / Wiley, 1982.

[Hastie and Mallows, 1993] T. Hastie and C. Mallows. A discussion of "a statistical view of some chemometrics regression tools" by I.E. Frank and J.H. Friedman. *Technometrics*, 35 : 140–143, 1993.

[Hastie *et al.*, 1994] T. Hastie, R. Tibshirani, and A. Buja. Flexible discriminant analysis by optimal scoring. *Journal of the American Statistical Association*, 89 : 1255–1270, 1994.

[Hastie *et al.*, 1995] T. Hastie, A. Buja, and R. Tibshirani. Penalized discriminant analysis. *Annals of Statistics*, 23 : 73–102, 1995.

[Hastie *et al.*, 2001] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning : Data Mining, Inference and Prediction*. Springer-Verlag, 2001.

[Hastie *et al.*, 2004] T. Hastie, S. Rosset, R. Tibshirani, and J. Zhu. The entire regularization path for the support vector machine. *Journal of Machine Learning Research*, 5 : 1391–1415, 2004.

[Hernandez and Velilla, 2001] A. Hernandez and S. Velilla. Dimension reduction in nonparametric discriminant analysis. *Technical Report*, 85 : 54–77, 2001.

[Hoerl and Kennard, 1970a] A. E. Hoerl and R. W. Kennard. Ridge regression : Application to Nonorthogonal Problems. *Technometrics*, 12(1) : 69–82, 1970.

[Hoerl and Kennard, 1970b] A. E. Hoerl and R. W. Kennard. Ridge regression : biased estimation for nonorthogonal problems. *Technometrics*, 12(1) : 55–67, 1970.

[Hornik, 1991] K. Hornik. Approximation capabilities of multilayer feedfoward networks. *Neural Networks*, 4(2) : 251–257, 1991.

[Hornik, 1993] K. Hornik. Some new results on neural network approximation. *Neural Networks*, 6(8) : 1069–1072, 1993.

[Hosmer and S., 1989] D. Hosmer and Lemeshow S. *Applied logistic regression*. Wiley, New York, 1989.

[Hsing and Carroll, 1992] T. Hsing and R.J. Carroll. An asymptotic theory for sliced inverse regression. *Annals of Statistics*, 20 : 1040–1061, 1992.

[Hsing, 1999] T. Hsing. Nearest neighbor inverse regression. *Annals of Statistics*, 697–731, 1999.

[Hsu and Lin, 2001] C.-W. Hsu and C.-J. Lin. A comparison of methods for multi-class support vector machines. *Technical Report*, 2001.

[Ivanov, 1962] V.K. Ivanov. On linear problems which are not well-posed. *Soviet. Math. Docl.*, 145(2), 1962.

[James and Hastie, 2001] G.M. James and T.J. Hastie. Functional linear discriminant analysis for irregularly sampled curves. *Journal of the Royal Statistical Society, Series B*, 63 : 533–550, 2001.

[James and Sugar, 2003] G.M. James and C.A. Sugar. Clustering for sparsely sampled functional data. *Journal of the American Statistical Association*, 98 : 397–408, 2003.

[Kondor and Lafferty, 2002] R.I. Kondor and J. Lafferty. Diffusion kernels on graphs and other discrete structures. In *Proceedings of the 19th International Conference on Machine Learning*, 315–322, 2002.

[Kooperberg *et al.*, 1997] C. Kooperberg, S. Bose, and J. Stone. Polychotomous regression. *Journal of the American Statistical Association*, 92 : 117–127, 1997.

[Lai and Wong, 2001] T.L. Lai and S. Wong. Stochastic neural networks with applications to nonlinear time series. *Journal of the American Statistical Association*, 96(455) : 968–981, 2001.

[Leurgans *et al.*, 1993] S.E. Leurgans, R.A. Moyeed, and B.W. Silverman. Canonical correlation analysis when the data are curves. *Journal of the Royal Statistical Society, Series B*, 55 : 725–740, 1993.

[Li *et al.*, 2003] K.C. Li, Y. Aragon, K. Shedden, and Thomas-Agnan C. Dimension reduction for multivariate data. *Journal of the American Statistical Association*, 98 : 99–109, 2003.

[Li, 1991] K.C. Li. Sliced inverse regression for dimension reduction. *Journal of the American Statistical Association*, 86 : 316–342, 1991.

[Li, 1992] K.C. Li. On principal Hessian directions for data visualisation and dimension reduction : another application of Stein's lemma. *Annals of Statistics*, 87 : 1025–1039, 1992.

[Lin, 2001] C.J. Lin. Formulations of support vector machines : a note from an optimization point of view. *Neural Computation*, 2(13) : 307–317, 2001.

[Lugosi and Zeger, 1990] G. Lugosi and K. Zeger. Nonparametric estimation via empirical risk minimization. *IEEE Transaction on Information Theory*, 41(3) : 677–687, 1990.

[Macintyre and Sontag, 1993] A. Macintyre and E. Sontag. Finiteness results for sigmoïdal "neural" networks. In *Proceedings of the 25th Annual ACM symposium on the Theory of Computing*, 325–334, New York, 1993. Association of Computing Machinery.

[Mallat, 1989] Stéphane Mallat. Multiresolution approximation and wavelet orthonormal bases of L2. *Transaction of the American Mathematical Society*, 315 : 69–87, September 1989.

[Marx and Eilers, 1996] B.D. Marx and P.H. Eilers. Generalized linear regression on sampled signals with penalized likelihood. In *Statistical Modelling. Proceedings of the 11th International Workshop on Statistical Modelling*. R. Hatzinger, A. Forcina, G.M. Marchetti and G. Galmacci editors, 1996.

[Métailie and Paegelow, 2004] J.P. Métailie and M. Paegelow. La dynamique du pin â crochet (Pinus uncinata Ram.) dans l'Est des Pyrénées Françaises : le retour de la forêt en montagne pastorale et métallurgique, In *ouvrage collectif*. Ed. Casa de Velasquez, 2004. A paraître.

[Paegelow and Camacho Olmedo, 2003] M. Paegelow and M.T. Camacho Olmedo. *Le processus d'abandon des cultures et la dynamique de reconquête végétale en milieu montagnard méditerranéen : L'exemple des Garrotxes (P.O., France) et de la Alta Alpujarra Granadina (Sierra Nevada, Espagne)*, volume 16. Sud Ouest Européen, 2003.

[Paegelow *et al.*, 2004a] M. Paegelow, M.T. Camacho Olmedo, and J. Menor Toribio. Modelizacion prospectiva del paisaje mediante Sistemas de Informacion Geografica. *GEOFOCUS*, 3 : 22–24, 2004.

[Paegelow *et al.*, 2004b] M. Paegelow, N. Villa, L. Cornez, F. Ferraty, L. Ferré, and P. Sarda. Modélisations prospectives de l'occupation du sol. Le cas d'une montagne méditerranéenne. *Cybergéo*, 295, 06 décembre 2004.

[Paegelow, 2003] M. Paegelow. Prospective modelling with GIS of land cover in Mediterranean mountain regions. In *6th AGILE Conference on GIScience. 24-26 avril 2003*, Lyon, 2003.

[Paegelow, 2004] M. Paegelow. *Géomatique et géographie de l'environnement. De l'analyse spatiale â la modélisation prospective.* Habilitation â diriger des recherches, Université Toulouse II (Le Mirail), 2004.

[Parlitz and Merkwirth, 2000] U. Parlitz and C. Merkwirth. Nonlinear prediction of spatiotemporal time series. In *ESANN'2000 proceedings*, 317–322, Bruges, 2000.

[Pezzulli and Silverman, 1993] S. Pezzulli and B.W. Silverman. Some properties of smoothed principal components analysis for functional data. *Computational Statistics*, 8 : 1–16, 1993.

[Pinkus, 1999] A. Pinkus. Approximation theory of the MLP model in neural networks. *Acta Numerica*, 8 : 143–195, 1999.

[Pollard, 1984] D. Pollard. *Convergence of stochastic processes*. Springer Verlag, New York, 1984.

[Preda and Saporta, 2002] C. Preda and G. Saporta. Régression PLS sur un processus stochastique. *Revue de statistique appliquée*, L(2), 2002.

[Preda and Saporta, 2005a] C. Preda and G. Saporta. Clusterwise PLS regression on a stochastic process. *Computational Statistics and Data Analysis*, 49(1) : 99–108, 2005.

[Preda and Saporta, 2005b] C. Preda and G. Saporta. PLS discriminant analysis for functional data. In *ASMDA 2005 proceedings*, 653–661, Brest, France, 2005.

[Ramsay and Dalzell, 1991] J.O. Ramsay and C.J. Dalzell. Some tools for functional data analysis (with discussion). *Journal of the Royal Statistical Society, Series B*, 53 : 539–572, 1991.

[Ramsay and Silverman, 1997] J.O. Ramsay and B.W. Silverman. *Functional Data Analysis*. Springer Verlag, New York, 1997.

[Ripley, 1994] B.D. Ripley. Neural networks and related methods for classification. *Journal of the Royal Statistical Society, Series B*, 56(3) : 409–456, 1994.

[Rosenthal, 1985] R.E. Rosenthal. Concepts, theory and techniques : principles of multiobjective optimization. *Decision Sciences*, 16(2) : 133–152, 1985.

[Rossi and Conan-Guez, 2005a] F. Rossi and B. Conan-Guez. Functional multi-layer perceptron : a nonlinear tool for functional data anlysis. *Neural Networks*, 18(1) : 45–60, 2005.

[Rossi and Conan-Guez, 2005b] F. Rossi and B. Conan-Guez. Theoretical properties of projection based multilayer perceptrons with functional inputs. *Neural Processing Letters*, 2005. A paraître.

[Rossi and Conan-Guez, 2005c] F. Rossi and B. Conan-Guez. Un modèle neuronal pour la régression et la discrimination sur données fonctionnelles. *Revue de Statistique Appliquée*, 2005. A paraître.

[Rossi and Conan-Guez, 2005d] Fabrice Rossi and Brieuc Conan-Guez. Estimation consistante des paramètres d'un modèle non linéaire pour des données fonctionnelles discrétisées aléatoirement. *Comptes rendus de l'Académie des Sciences - Série I*, 340(2) : 167–170, January 2005.

[Rossi and Villa, 2005a] F. Rossi and N. Villa. Classification in Hilbert spaces with support vector machines. In *ASMDA 2005 proceedings*, 635–642, Brest, France, 2005.

[Rossi and Villa, 2005b] F. Rossi and N. Villa. Support vector machine for functional data classification. 2005. Soumis â publication.

[Rossi *et al.*, 2002] F. Rossi, B. Conan-Guez, and F. Fleuret. Functional data analysis with multi layer perceptrons. In *IJCNN 2002 (part of WCCI) proceedings*, 2843–2848, 2002.

[Rossi *et al.*, 2004] F. Rossi, B. Conan-Guez, and A. El Golli. Clustering functional data with the som algorithm. In *ESANN'2004 proceedings*, 305–312, Bruges, Belgique, 2004.

[Rossi *et al.*, 2005] F. Rossi, N. Delannay, B. Conan-Guez, and M. Verleysen. Representation of functional data in neural networks. *Neurocomputing*, 64 : 183–210, 2005.

[Saaty, 1977] T.L. Saaty. A scaling method for priorities in hierarchical structures. *Journal of Mathematics and Psychology*, 15 : 234–281, 1977.

[Sandberg and Xu, 1996] I.W. Sandberg and L. Xu. Network approximation of input-output maps and functionals. *Circuits Systems Signal Processing*, 15(6) : 711–725, 1996.

[Sandberg, 1996] I.W. Sandberg. Notes on weighted norms and network approximation of functionals. *IEEE Transactions on Circuits and Systems-I : Fundamental Theory and Applications*, 43(7) : 600–601, 1996.

[Saporta, 1981] G. Saporta. Méthodes exploratoires d'analyses des données temporelles. *Cahiers du BURO*, (37-39), 1981.

[Schölkopf *et al.*, 2004] B. Schölkopf, K. Tsuda, and J.P. Vert. *Kernel methods in computational biology*. MIT Press, London, 2004.

[Schott, 1994] J.R. Schott. Determining the dimensionality in sliced inverse regression. *Journal of the American Statistical Association*, 89 : 141–148, 1994.

[Setodji and Cook, 2004] C.M. Setodji and R.D. Cook. K-means inverse regression. *Technometrics*, 46(4) : 421–429, 2004.

[Shawe-Taylor and Cristianini, 2004] J. Shawe-Taylor and N. Cristianini. *Kernel methods for pattern analysis*. Cambridge University Press, Cambridge, UK, 2004.

[Silverman, 1996] B.W. Silverman. Smoothed functional principal components analysis by choice of norm. *Annals of Statistics*, 24 : 1–24, 1996.

[Smola and Schölkopf, 1998a] A. Smola and K.R. Schölkopf, Bernhard B. Müller. The connection between regularization operators and support vector kernels. *Neural Networks*, 11 : 637–649, 1998.

[Smola and Schölkopf, 1998b] Alexander Smola and Bernhard Schölkopf. On a kernel-based method for pattern recognition, regression, approximation and operator inversion. *Algorithmica*, 22(1064) : 211–231, 1998.

[Steinwart, 2001] I. Steinwart. On the influence of the dernel on the consistency of support vector machines. *Journal of Machine Learning Research*, 2 : 67–93, 2001.

[Steinwart, 2002] I. Steinwart. Support vector machines are universally consistent. *J. Complexity*, 18 : 768–791, 2002.

[Stinchcombe, 1999] M.B. Stinchcombe. Neural network approximation of continuous functionals and continuous functions on compactifications. *Neural Network*, 12(3) : 467–477, 1999.

[Tabeaud *et al.*, 2003] M. Tabeaud, P. Pech, and L. Simon. Weather hazards, vulnerabilities and risks in Mediterranean hinterlands from the 19th century - the Lure Mountain (France). In *IAG Working Group on Geochaeology ; colloque international "Dynamiques environnementales et histoire en domaines méditerranéens"*, 143–149, 2003.

[Team, 2005] R Development Core Team. *R : a language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria, 2005.

[Tenorio, 2001] L. Tenorio. Statistical regularization of inverse problems. *Society for Industrial and Applied Mathematics*, 43(2) : 347–366, 2001.

[Thodberg, 1996] H.H. Thodberg. A review of Bayesian Neural Network with an application to near infrared spectroscopy. *IEEE Transaction on Neural Networks*, 7(1) : 56–72, 1996.

[Tihonov, 1963a] A.N. Tihonov. Regularization of incorrectly posed problems. *Soviet Math. Docl.*, 4 : 1624–1627, 1963.

[Tihonov, 1963b] A.N. Tihonov. Solution of incorrectly formulated problems and the regularization method. *Soviet Math. Docl.*, 4 : 1036–1038, 1963.

[Vapnik, 1995] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer Verlag, New York, 1995.

[Vapnik, 1998] V. Vapnik. *Statistical Learning Theory*. Wiley, New York, 1998.

[Velilla, 1998] S. Velilla. Assessing the number of linear component in a general regression problem. *Journal of the American Statistical Association*, 93 : 1088–1098, 1998.

[Villa and Rossi, 2005] N. Villa and F. Rossi. Support vector machine for functional data classification. In *ESANN proceedings*, 467–472, 2005.

[Villa *et al.*, 2005] N. Villa, M. Paegelow, L. Cornez, F. Ferraty, L. Ferré, and P. Sarda. Various approaches to predicting land cover in mediterranean mountain areas. 2005. Soumis â publication.

[White, 1989] H. White. Learning in artificial neural network : a statistical perspective. *Neural Computation*, 1 : 425–464, 1989.

[White, 1990] A. White. Connectionist nonparametric regression : multilayer feedforward networks can learn arbitraty mappings. *Neural Networks*, 3 : 535–549, 1990.

[Williamson *et al.*, 1998] R.C. Williamson, A.J. Smola, and B. Scholkopf. Generalization performance of regularized networks and support vector machines via entropy numbers of compact operators. 1998. NeuroCOLT Technical Report, NC-TR-98-019.

[Xia *et al.*, 2002] Y. Xia, H. Tong, W.K. Li, and L.X. Zhu. An adaptative estimation of dimension reduction space. *Journal of the Royal Statistical Society, Series B*, 64 : 363–410, 2002.

[Yager, 1988] R.R. Yager. On ordered weighted averaging aggregation operators in multicriteria decision making. *IEEE Transactions on Systems, Man, and Cybernetics*, 8(1) : 183–190, 1988.

[Zhu and Fang, 1996] L.X. Zhu and K.T. Fang. Asymptotics for kernel estimate of sliced inverse regression. *Annals of Statistics*, 24 : 1053–1068, 1996.

[Zhu and Ng, 1995] L.X. Zhu and K.W. Ng. Asymptotics of sliced inverse regression. *Statistica Sinica*, 5 : 727–736, 1995.