Function-valued regression with kernels : Improving speed, flexibility and robustness

Dimitri Bouche

INSTITUT POLYTECHNIQUE DE PARIS

NNT : 2023IPPAT001

Thèse de doctorat

Function-valued regression with kernels: Improving speed, flexibility and robustness

Thèse de doctorat de l'Institut Polytechnique de Paris
préparée à Télécom Paris

École doctorale n°626 Ecole Doctorale de l'Institut Polytechnique de Paris (ED IP Paris)
Spécialité de doctorat : Informatique, Données et Intelligence Artificielle

Thèse présentée et soutenue à Palaiseau, le 10 janvier 2023, par

DIMITRI BOUCHE

Composition du Jury :

Gilles Gasso Full Professor, INSA Rouen (LITIS)	Président
Charles Bouveyron Full Professor, Université Côte d'Azur (3IA) and INRIA (MAASAI)	Rapporteur
Hachem Kadri Associate Professor, Université Aix-Marseille (LIS)	Rapporteur
Emilie Devijver Researcher, Université Grenoble Alpes (LIG)	Examinatrice
Massimiliano Pontil Senior Researcher, Istituto Italiano di Tecnologia (CSML) and UCL	Examinateur
Florence d'Alché-Buc Full Professor, Télécom Paris (LTCI)	Directrice de thèse
Marianne Clausel Full Professor, Université de Lorraine (IECL)	Co-directrice de thèse
François Roueff Full Professor, Télécom Paris (LTCI)	Invité

Contents

Remerciements

Je souhaite commencer par remercier ceux et celles qui m'ont accompagné et soutenu durant ces années de doctorat.

Merci tout d'abord à Florence, ma directrice de thèse et Marianne, ma co-directrice. Vous m'avez accueilli chaleureusement et guidé durant ces années de doctorat, toujours dans la bonne humeur. Je vous remercie pour votre implication et je peux dire que j'ai appris énormément à vos côtés. Je tiens également à remercier François, tu as été d'une aide précieuse sur les aspects théoriques en début de thèse. I would also like to thank Zoltan, your incredibly thorough pre-review of my first paper has resulted in great improvements and I also appreciated very much working with you.

I would like to thank Charles Bouveyron and Hachem Kadri for their very thorough review of this manuscript. I also thank Gilles Gasso, Emilie Devijver and Massimiliano Pontil for being part of my PhD committee.

Merci à tous ceux que j'ai eu la chance de côtoyer durant ces années de doctorat. Si j'ai autant apprécié mon passage à Télécom Paris (et que j'ai continué à affronter le RER B jour après jour) c'est beaucoup grâce à vous. Merci aux camarades de stage de la première heure, Luc, Vincent et Rémi. On s'est lancé dans le doctorat ensemble, mais heureusement pour nos thèses, on a arrêté le babyfoot à Saclay. Merci particulièrement à toi Luc, on s'est soutenus et on s'en est sortis (tu y es presque!). Merci à Junjie d'avoir irradié le bureau de bonne humeur, on a bien rigolé et on s'est motivés mutuellement. Ça a été un plaisir de partager le bureau aussi avec Amaury, toujours bienveillant et prêt à discuter musique.

Merci à Emile pour ces moments d'évasion de Saclay vers les spots de surf tantôt de Bretagne nord, tantôt de Vendée, on va se la faire cette session et bientôt! Merci Guillaume d'avoir supporté lesdites discussions et aussi pour tes conseils indispensables pour naviguer le doctorat, tu as décidément su perpétuer l'esprit d'Alexandre G. dans le labo. Merci à Pierre C., le magicien du machine learning, de la montagne, et du cluster. Merci à Marc et Jérémy, ça m'a fait super plaisir de partager avec vous ces divagations rock/métal, j'espère qu'on pourra se croiser un jour au Hellfest. Merci à Joël et Arturo, toujours fidèles au poste, j'ai adoré nos pauses déj et nos cafés.

Merci à Alex, j'ai beaucoup apprécié notre collaboration, ça été rapide et intense mais très fructueux et toujours dans la bonne humeur, même quand la papier faisait 9 pages le jour de la deadline et qu'il nous a fallu recourir à la magie noire LaTeX du Z.

Comme je ne peux pas citer tout le monde, merci aussi à tous les autres, j'ai beaucoup apprécié partager des déjeuners, des cafés, des bières ou des trajets en RER avec vous. Merci Emilia, Tamim, Lucien, Nathan H., Anas A., James, Louis, Nathan N., Myrto, Jayneel, Robin, Mastane, Nidham, Eric, Pierre L., Hamid, Kimia, Kamelia, Anas B., Ariel.

Pour ceux qui n'ont pas encore soutenu, bon courage, vous allez y arriver!

Un grand merci également à Bernard Disdet. Tu as toujours su me rétablir physiquement et me soutenir durant ces longues (longues!) études supérieures.

Je remercie bien sûr ma famille et mes proches, qui m'ont soutenu et permis d'en arriver là. Merci à mon frère qui m'a montré la voie. Merci à mes parents, ma mère pour son soutien et sa compréhension et à mon père pour, entre autres, avoir relu avec grande attention ce manuscrit. Merci à mon oncle Michel. Merci Soraya de partager ma vie et d'avoir vécu cette thèse avec moi, ton soutien a été déterminant.

Abstract

With the increasing ubiquity of data-collecting devices, a great variety of phenomena is monitored with finer and finer accuracy, which constantly expands the scope of Machine Learning applications. Dealing with such volume of data efficiently is however challenging. Fortunately, as measurements get denser, they may become gradually redundant. We can then greatly reduce the burden by finding a representation which exploits properties of the generating process and/or is tailored for the application at hand.

This thesis revolves around an aspect of this idea: functional data. Data indeed consist of discrete measurements, but sometimes thinking of those as functional, we can exploit prior knowledge on smoothness to obtain a better yet lower dimensional representation. The focus is on nonlinear models for functional output regression (FOR), relying on an extension of reproducing kernel Hilbert spaces for vector-valued functions (vv-RKHS), which is the cornerstone of many nonlinear existing FOR methods. We propose to challenge those in two aspects: their computational complexity with respect to the number of measurements per function and their focusing solely on the square loss.

To that end, we introduce the new framework of kernel projection learning (KPL) combining vv-RKHSs and representation of signals in dictionaries. The loss remains functional, however the model predicts only a finite number of representation coefficients. This approach retains the many advantages of vv-RKHSs yet greatly alleviates the computational burden incurred by the functional outputs. We derive two estimators in closed-form using the square loss, one for fully observed output functions and one for discretized ones. We show that both are consistent in terms of excess risk. We demonstrate as well the possibility to use other differentiable and convex losses, to combine this framework with large scale kernel methods and to automatically select the dictionary using a structured penalty.

In another contribution, we propose to solve the regression problem in vv-RKHSs of function-valued functions for the family of convoluted losses which we introduce. These losses can either promote sparsity or robustness with a parameter controlling the degree of locality of these properties. Thanks to their structure, they are particularly amenable to dual approaches which we investigate. We then introduce two representations to overcome the challenges posed by the functional nature of the dual variables and we propose corresponding algorithms to solve each dual problem.

Résumé

L'augmentation du nombre et de la sophistication des appareils collectant des données permet de suivre l'évolution d'une multitude de phénomènes à des résolutions très fines. Cela étend le champ des applications possibles de l'apprentissage statistique. Un tel volume peut néanmoins devenir difficile à exploiter. Cependant quand leur nombre augmente, les données peuvent devenir redondantes. On peut alors chercher une représentation exploitant des propriétés du processus génératif.

Dans cette thèse, nous nous concentrons sur la représentation fonctionnelle. Bien sûr, les données sont toujours des observations discrètes. Néanmoins, si nous pensons que ces suites doivent être par exemple lisses ou de variations bornées, une telle représentation peut être à la fois plus fidèle et de dimension plus faible. Nous nous concentrons sur les modèles non-linéaires de régression à valeurs fonctionnelles (FOR) en utilisant une extension des espaces de Hilbert à noyau reproduisant pour les fonctions à valeurs vectorielles (vv-RKHS) qui constitue la clef de voûte de plusieurs méthodes existantes. Notre objectif est d'en proposer de nouvelles plus performantes sur le plan de la complexité calculatoire liée au caractère fonctionnel et/ou celui du choix de la fonction de perte.

Nous introduisons l'apprentissage de projection kernelisé (KPL) qui combine les vv-RKHSs et la représentation de signaux sur des dictionnaires. La perte demeure fonctionnelle, néanmoins le modèle prédit seulement un nombre fini de coordonnées. Nous bénéficions alors de la flexibilité de l'espace d'hypothèse tout en réduisant nettement la complexité liée aux sorties fonctionnelles. Pour la perte quadratique, nous introduisons deux estimateurs en forme close, l'un est adapté lorsque les fonctions de sortie sont observées totalement, et l'autre l'est lorsqu'elles ne le sont que partiellement. Nous montrons que chacun est consistant en termes d'excès de risque. Nous proposons aussi d'utiliser d'autres fonctions de perte différentiables, de combiner KPL avec les techniques de passage à l'échelle ou encore de sélectionner le dictionnaire via une pénalité structurée.

Une autre partie est dédiée au problème de FOR dans des vv-RKHS de fonctions à valeurs fonctionnelles en utilisant une famille de fonctions de pertes que nous introduisons comme définies à partir d'une convolution infimale. Celles-ci peuvent encourager soit la parcimonie soit la robustesse, le degré de localité de ces propriétés étant contrôlé via un paramètre dédié. Grâce à leur structure, ces pertes se prêtent particulièrement bien à la résolution par dualité lagrangienne. Nous surmontons alors les différents défis que pose la dimension infinie des variables duales en proposant deux représentations pour résoudre chaque problème dual numériquement.

Notation

$:=$	Equal by definition
$\mathbb{N}^*$	Strictly positive integers
$[\![n]\!]$	Set of integers from 1 to n ($\{1, \cdots, n\}$)
$\mathcal{F}(\mathcal{X}, \mathcal{Y})$	Set of functions from a space $\mathcal{X}$ to a Hilbert space $\mathcal{Y}$
$\mathcal{L}(\mathcal{H}, \mathcal{K})$, $\mathcal{L}(\mathcal{H})$	Bounded linear operators between Hilbert spaces $\mathcal{H}$ and $\mathcal{K}$, shortened when $\mathcal{H} = \mathcal{K}$
$\|\cdot\|_{\mathcal{L}(\mathcal{H},\mathcal{K})}$	Operator norm for operators in $\mathcal{L}(\mathcal{H}, \mathcal{K})$
$\mathcal{L}_2(\mathcal{H}, \mathcal{K})$, $\mathcal{L}_2(\mathcal{H})$	Hilbert-Schmidt linear operators between Hilbert spaces $\mathcal{H}$ and $\mathcal{K}$, shortened when $\mathcal{H} = \mathcal{K}$
$\mathcal{X}$	Input space
$\mathcal{Y}$	Output space, at least a Hilbert space
Θ	Domain of definition of output functions
$\mathrm{L}^2(\Theta, \mu, \mathcal{K})$, $\mathrm{L}^2(\Theta, \mu)$	Hilbert space of μ-square-integrable functions from Θ to $\mathcal{K}$, shortened when $\mathcal{K} = \mathbb{R}$
$\mathcal{C}(\Theta)$, $\mathcal{C}^s(\Theta)$	Continuous, continuously s times differentiable functions from Θ to $\mathbb{R}$
$\boldsymbol{\phi}$	Dictionary of vectors $(\phi_l)_{l=1}^d \in \mathcal{Y}^d$. If those are functions on Θ, also vector-valued function $\theta \mapsto (\phi_1(\theta), \phi_2(\theta), \cdots, \phi_d(\theta))^{\mathrm{T}} \in \mathbb{R}^d$
$\phi_l(\boldsymbol{\theta})$	For ϕ_l a function on Θ and $\boldsymbol{\theta} \in \Theta^m$, vector $\phi_l(\boldsymbol{\theta}) = (\phi_l(\theta_1), \cdots, \phi_l(\theta_m))^{\mathrm{T}} \in \mathbb{R}^m$
Φ	Linear operator associated to $\boldsymbol{\phi}$ as: $\mathbf{a} \in \mathbb{R}^d \mapsto \sum_{l=1}^d a_l \phi_l \in \mathcal{Y}$
$\langle \cdot, \cdot \rangle_{\mathcal{Y}}$, $\|\cdot\|_{\mathcal{Y}}$	Scalar product and norm in Hilbert space $\mathcal{Y}$
$\mathrm{A}^{\#}$	Adjoint of operator A
$\mathrm{I}_{\mathcal{H}}$	Identity operator on space $\mathcal{H}$
A^{T}	Transpose of matrix A
A_i, A_l.	i-th column of matrix A, l-th row of matrix A
$[\mathbf{a}_i]_{i=1}^n$	Matrix of $\mathbb{R}^{d \times n}$ which i-th column is the vector $\mathbf{a}_i \in \mathbb{R}^d$
I	Identity matrix, dimension inferred from context
$\otimes$	Kronecker product of matrices, tensor product of Hilbert spaces or their elements
$\mathrm{vec}(\mathrm{A})$	For $\mathrm{A} \in \mathbb{R}^{d \times n}$, vector of $\mathbb{R}^{dn}$ formed by concatenating the columns of A

$\mathrm{A}_{(n)}$	For an operator $\mathrm{A} : \mathbb{R}^d \to \mathcal{Y}$, operator $\mathrm{A}_{(n)} : \boldsymbol{\alpha} \in \mathbb{R}^{d \times n} \mapsto (\mathrm{A}\alpha_i)_{i=1}^n \in \mathcal{Y}^n$.
$k_{\mathcal{X}} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$	Input kernel on $\mathcal{X}$, scalar-valued
$k_\Theta : \Theta \times \Theta \to \mathbb{R}$	Kernel on Θ, scalar-valued
$\mathsf{K} : \mathcal{X} \times \mathcal{X} \to \mathcal{L}(\mathcal{Y})$	Operator-valued kernel
$\mathcal{H}_{\mathsf{K}}$	Vector-valued RKHS associated to K
ev_x	Evaluation map at point x
Trace	Trace of operator or matrix
Im	Range of operator or matrix
Ker	Null space of operator or matrix
Rank	Rank of operator or matrix
Dim	Dimension of space
Sp	Spectrum of operator or matrix
Span	Linear space spanned by a set of vectors
$\|\cdot\|_p$	p-norm for $p \in [1, +\infty]$
$\|\cdot\|_{p,q}$	Mixed norm: q-norm of the p-norms of the columns of a matrix or of the functions from a vector of functions; *e.g.* $\boldsymbol{\alpha} \in \mathcal{Y}^n$, $\|\boldsymbol{\alpha}\|_{p,q} = \left\| (\|\alpha_i\|_p)_{i=1}^n \right\|_q$
$\mathcal{B}_\epsilon$	Ball of radius ϵ for the ambient Hilbert norm
$\mathcal{B}_\epsilon^p$	Ball of radius ϵ for the p-norm
$\mathrm{dom}(f)$	Domain of a function f
$\Gamma_0(\mathcal{H})$	Proper, convex, lower-semicontinuous real-valued functions on Hilbert space $\mathcal{H}$
$f^\star$	Fenchel-Legendre conjugate of function f
$f \square g$	Infimal convolution of functions f and g
$\chi_{\{\mathcal{C}\}}$	Indicator function of set $\mathcal{C}$: 0 on $\mathcal{C}$ and $+\infty$ elsewhere
prox_f	Proximal operator of function f
$\mathrm{Proj}_{\mathcal{C}}$	Orthogonal projection on a closed convex set $\mathcal{C}$
$\|\cdot\|_+$	Positive part: $\|a\|_+ = \max(a, 0)$
$\lfloor \cdot \rfloor$	Floor integer part: $\lfloor a \rfloor = m \Leftrightarrow m \leq a < m + 1$, $m \in \mathbb{N}$

Abbreviation

RKHS	Reproducing kernel Hilbert space
OVK	Operator-valued kernel
vv-RKHS	Vector-valued RKHS
fv-RKHS	Function-valued RKHS
FOR	Functional output regression
FDA	Functional data analysis
RFF	Random Fourier feature
KPL	Kernel projection learning
KRR	Kernel ridge regression
FKRR	Functional kernel ridge regression
3BE	Triple basis estimator
KAM	Kernel additive model
MC	Monte Carlo
APGD	Accelerated proximal gradient descent

1

Motivation and Contributions

Contents

Supervised machine learning describes the process of inferring statistically (*i.e.* from observed examples) a procedure to predict a label from an explanatory variable. A lot of attention has been dedicated to cases where the label consists of a single scalar-valued variable either in regression (labels in a continuous space) or classification (labels in a discrete set). However with the dramatic increase in both the volume and the variety of collected data, the interest for procedures that can deal with more complicated labels, such as high dimensional vectors or structured objects, has grown deeper. While most algorithms can deal with higher dimensional explanatory variables, they may require adjustments to deal with higher dimensional labels in a way that is efficient, and produces predictions which benefit from the outputs' structure or comply with.

In this thesis, we focus on a particular type of high dimensional output data. Thanks to the development of sensors, many phenomena can be monitored at higher and higher resolutions. However, as a corollary, we end up with very high dimensional vectors of measurements which may become redundant. If the underlying generating process is known to exhibit certain properties, we should be able to exploit these to reduce the complexity of the learning algorithms and produce predictions that comply with those observed properties. For instance, if the data are measurements from a spatio-temporal process that is expected to be smooth across space and/or through time, we would ideally include this smoothness as prior knowledge in the learning algorithm. Such data can typically arise in a wide variety of fields. In meteorology, quantities (*e.g.* wind speed, temperature...) are observed at several weather stations through time. At a given time, measurements should be alike for stations close to each other, therefore we expect observations to be smooth through space. In the same way, at a given station, measurements should not vary too quickly over time. This results in a process that is smooth through time as well. The modeling idea stemming from this example is general and relevant in many more fields, ranging from climate science to biomedical imaging or epidemiology monitoring, internet of things, etc. Given signals that we think should be smooth in some way, we can both reduce the dimension and model these signals more accurately if we envision them as sampled

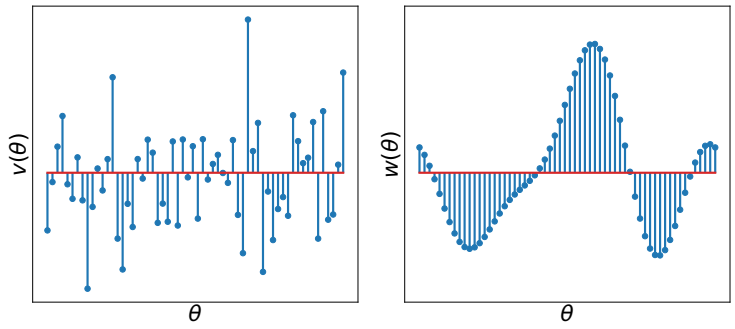


Figure 1.1: A non-smooth vector (v) and a smooth one (w)

versions of an underlying unobserved function-valued process. The vectors displayed in Figure 1.1 are compelling in that regard. On the left, v corresponds to i.i.d. draws from a normal distribution whereas w consists of discrete measurements from a mixture of 6 trigonometric functions. Therefore it is impossible to reduce the dimension of the former whereas the latter can be represented perfectly in dimension 6 instead of 60 if the right functional representation is used.

These real world scenarios have motivated the investigation of statistical procedures that are capable of correctly representing functions and dealing with them. It is the goal of *functional data analysis* (FDA, Ramsay and Silverman 1997). It has been applied successfully to a great variety of fields (Ullah and Finch, 2013) through a rich array of statistical procedures dealing with functional observations (Ramsay and Silverman, 1997; Wang et al., 2016).

Among the possible statistical problems, that of regression involving functional data has been particularly studied. The seminal work of Ramsay and Dalzell (1991) introduced the additive functional linear model already for different kinds of targets. The functional regression problems are then categorized depending on which variables (explanatory ones and/or target) are functional (Ramsay and Silverman, 1997). In this thesis, we focus on the cases where the target is function-valued, namely *functional output regression* (FOR) problems. These problems can however be more challenging as the target takes its values in an infinite-dimensional space. The question of which finite-dimensional representation can be used is of particular importance and it has been central in the models proposed in the FOR literature. Several variations of the functional linear models which smooth the functional variables using functional bases have been proposed (Morris 2015 and references therein). Nonparametric models (Ferraty and Vieu, 2006) have also met a lot of success since they circumvent the need for an explicit representation of the functions. To finish with, the richness of reproducing kernel Hilbert spaces (RKHS) as functional spaces, as well as their possible generalization to model function-valued functions, has made them the cornerstone of many nonlinear FOR methods (Lian, 2007; Kadri et al., 2010; Oliva et al., 2015; Kadri et al., 2016; Reimherr et al., 2018; Laforgue et al., 2020).

These latter methods are indeed attractive in many aspects. They can model efficiently nonlinear relationships between the input variables and the output one. Moreover, since the input variables appear through a kernel, they can lie in any space on which a kernel can be defined; this allows for dealing with complex input data. For all

these reasons, this thesis is centered around tackling the nonlinear FOR problem using reproducing kernels and their operator-valued extensions.

In studying existing nonlinear FOR methods, we identified three main aspects in which improvements could be sought. **(i)** The first challenge is computational complexity, in particular that linked to the functional outputs (the complexity related to the input observations is interesting yet not specific to FOR). The number of discretization points m per output function is bound to be high and generally incurs a high computational cost, typically cubic in m. In the best cases, the complexity related to it and that linked to the number of samples n do not interact in a bad way (they do not multiply one another), however they can do so in some cases. It is therefore desirable for a FOR method to incur a lower computational cost with respect to m. **(ii)** The second challenge is the flexibility in the choice of the functional loss function. Indeed, functional data can be corrupted by outliers or we may want to increase efficiency by enforcing sparsity in the model's coefficients. Therefore, we want to be able to use losses that are robust or can encourage sparsity even when the outputs are functional. **(iii)** The third axis of improvement is the possibility to deal with missing data in the observed output functions. In practice we usually only have access to discrete evaluations. If these evaluations are given at the same locations for all the functions, most methods can be used easily. However, when they vary from one sampled function to another, many methods will require an additional imputation/smoothing step. Then, we want our FOR method to traduce our prior of smoothness by predicting functions, yet we would like it be able to deal directly with discrete observations.

The objective of this thesis is to propose new nonlinear FOR methods based mostly on extensions of RKHSs to model vector-valued or function-valued functions, or improve already existing ones along one or more of these three axes of improvement. This thesis also includes a more applied contribution on using machine learning to predict wind power production in the very short-term for wind farms. We chose however to propose a coherent thesis on function-valued regression, as it is the subject of most of our work and contributions, and we therefore include this applied contribution in a separate part at the end.

This chapter is organized as follows: in Section 1.1, we introduce the general principle of statistical learning and briefly frame the type of problems we will study in this thesis. We then present our propositions to solve those problems and raise the related research questions in Section 1.2. In Section 1.3 we give more details on the content of the manuscript and show how the different chapters address the question we raised; the associated publications are listed in Section 1.4.

1.1 Statistical Learning

1.1.1 Supervised learning

Let $\mathsf{Z} = (\mathsf{X}, \mathsf{Y})$ be a random variable taking its values in a space $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ distributed according to an unknown probability distribution. For this to make sense, we make the minimal assumption that $\mathcal{X}$ and $\mathcal{Y}$ are measurable spaces yet we keep their associated sigma-algebra implicit to lighten notations. The random variable Y is the label (*e.g.* a class in classification, a real value in scalar-valued regression, a function in function-valued regression...) and X is the random variable of features from which we want to predict the label, good features must then encompass information that are

useful to predict it. The goal of supervised machine learning is then to statistically infer from joint observations of Z a way to predict a good value in $\mathcal{Y}$ from an isolated realization of X. To do so, we first need a tool to measure the discrepancy between elements in $\mathcal{Y}$ which is the role of the loss function $\ell : \mathcal{Y} \times \mathcal{Y} \longrightarrow \mathbb{R}$. Ideally, we would want to reach a function f among measurable functions that is optimal in the sense that it minimizes the expected risk:

$$f^* \in \underset{f \text{ measurable}}{\arg\min}\ \mathbb{E}_{(X,Y)}\Big[\ell(f(X), Y)\Big].$$

However, in practice we do not have access to the true distribution, but rather to independently identically distributed (i.i.d.) realizations $(x_i, y_i)_{i=1}^n$ from the random variable Z. Consequently, in most cases we cannot attain a function that is as good as f^*. To find a good candidate from the available data, we then minimize an empirical proxy to the expected risk, namely the empirical risk. Moreover, as optimizing over all measurable functions is infeasible, a set of functions to optimize the empirical risk over must be chosen. Let $\mathcal{G} : \mathcal{X} \to \mathcal{Y}$ be this set of candidate functions, which is called a *hypothesis class*. Moreover, when learning from a discrete sample, one must ensure that the obtained function generalizes well to unknown data, in other words we say it must not *overfit* the training data. To avoid overfitting, the regression function can be encouraged to display certain regularity properties by adding a penalization term to the optimization objective, we denote by $\Omega : \mathcal{G} \to \mathbb{R}$ such a regularization function. The following problem can then be solved:

$$\hat{f} \in \underset{f \in \mathcal{G}}{\arg\min} \frac{1}{n} \sum_{i=1}^{n} \ell(f(x_i), y_i) + \Omega(f).$$

A problem of this form is commonly called a *regularized empirical risk minimization problem*.

Remark 1.1. *When the hypothesis class is included in an inner product vector space, if we denote by $\|\cdot\|$ the corresponding norm, a common choice for the regularization function is $\Omega(f) = \lambda\|f\|^2$ where $\lambda > 0$ is a parameter controlling the intensity of the regularization (Tikhonov and Arsenin, 1977).*

Example 1.2 (Least square regression)**.** *As a well-known example, let us consider that Y takes its values in $\mathbb{R}$, this corresponds to the regression setting. The most commonly used loss function for this task is the square loss. It penalizes the errors as the square of the difference between the prediction and the actual label's value.*

$$f^* \in \underset{f \text{ measurable}}{\arg\min}\ \mathbb{E}_{(X,Y)}\Big[(f(X) - Y)^2\Big].$$

The above coincides with the very definition of the conditional expectation which is indeed the theoretical optimal regression function in that case. It is then defined as $f^* : x \mapsto \mathbb{E}_{Y|X=x}[Y]$. This expression however still does not help us as the distribution of (X, Y) is unknown. Therefore, we formulate the same type of regularized empirical risk minimization problem using the available training data.

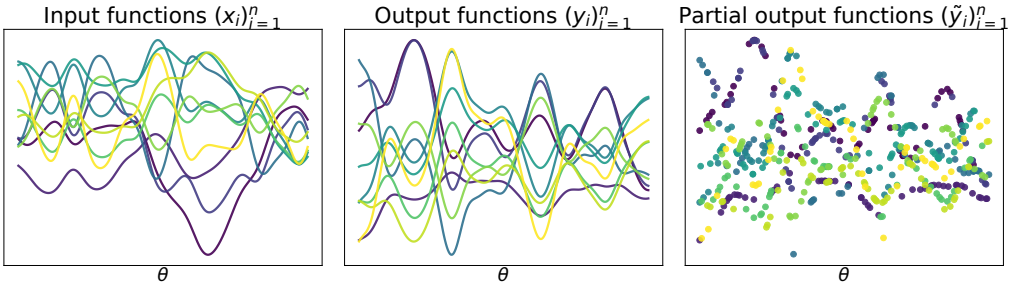


Figure 1.2: Function to function regression with fully or partially observed outputs.

$$\hat{f} \in \operatorname*{arg\,min}_{f \in \mathcal{G}} \frac{1}{n} \sum_{i=1}^{n} (f(x_i) - y_i)^2 + \Omega(f).$$

with $\mathcal{G} \subset \mathcal{F}(\mathcal{X}, \mathbb{R})$ *an hypothesis class and* $\Omega : \mathcal{G} \to \mathbb{R}$ *a regularization function.*

1.1.2 Learning function-valued functions

Functional output regression

When the random variable Y takes its values in a subset of a Hilbert functional space, one must recourse to a hypothesis class of function-valued functions, use a loss which compare two functions and define an appropriate regularization term. Since the output space is essentially infinite dimensional, estimators also cannot be computed exactly.

Thus *functional output regression* (FOR) is challenging in itself. Admittedly, this functional aspect can seem somewhat theoretical since ultimately we are given discrete evaluations of the observed functions. Nevertheless in many applications, the sampling frequencies of the output observations as well their expected smoothness can amply justify modeling the problem as function-valued. This is for instance particularly the case in biomedical signal processing, meteorology or industrial applications (Ullah and Finch, 2013; Ramsay and Silverman, 2007).

A reasonable assumption is then that the random variable Y takes its values in a functional Hilbert space $\mathcal{Y}$. We can suppose further that the vectors in $\mathcal{Y}$ are functions which domain is a set $\Theta \subset \mathbb{R}^b$ for some integer $b \geq 1$. For instance, a possible choice is to take $\mathcal{Y}$ as $\mathrm{L}^2(\Theta)$, the space of square integrable functions on Θ. As an example of a functional loss, the analogous to the classical square loss can be defined as the square of the $\mathcal{Y}$ norm of the residuals. Then, we would ideally want to minimize the following expected risk

$$f^* \in \operatorname*{arg\,min}_{f \,\text{measurable}} = \mathbb{E}_{(\mathrm{X},\mathrm{Y})}\left[\|f(\mathrm{X}) - \mathrm{Y}\|_{\mathcal{Y}}^2\right].$$

where we search for f in the space of measurable functions from $\mathcal{X}$ to $\mathcal{Y}$. In reality, as before we must choose a hypothesis class $\mathcal{G} \subset \mathcal{F}(\mathcal{X}, \mathcal{Y})$ of functions taking their values in the Hilbert space $\mathcal{Y}$, a regularization function on $\mathcal{G}$ and solve a regularized empirical risk minimization problem. This is the approach taken in the seminal works of (Kadri

et al., 2010, 2016) who use a generalization of RKHSs to model function-valued functions (Pedrick, 1957) as a hypothesis class and regularize through the natural norm on these spaces. However, other losses $L : \mathcal{Y} \times \mathcal{Y}$ can be defined on Hilbert spaces. For instance, so as to make their estimator robust, Laforgue et al. (2020) study a possible extension of the Huber loss (Huber, 1964) for Hilbert-valued output data, using vector-valued RKHSs as a hypothesis class and regularizing through their associated norm.

FOR with discrete observations

We now focus on the case where the elements in the Hilbert space $\mathcal{Y}$ are functions defined on a domain Θ and we have access to discrete evaluations of these functions. Even when the sampling rate is not that high, considering the output as functional can still be interesting if we want to include a strong prior on smoothness. For instance, this can have a desirable regularizing effect if the observations from the output functions are noisy. Moreover, this can also help us when data are missing. Indeed, if we are given observations from a phenomena but the locations of observation vary from one observation to the other, it is not possible anymore to solve the problem in a vector-valued fashion. To illustrate the problem of FOR in this setting, we display instances from a synthetic dataset in Figure 1.2. This a function-to-function regression problem and in the bottom panel, we picture the output functions observed only at a low number of randomly chosen locations. To formalize things, we can suppose that instead of getting a sample of the form $(x_i, y_i)_{i=1}^n$, we are rather given one of the form $(x_i, (\theta_i, \tilde{y}_i))_{i=1}^n$. Here, $\theta_i = (\theta_{is})_{s=1}^{m_i} \in \mathbb{R}^{m_i}$ corresponds to the vector of locations at which we observe the i-th output function and $\tilde{y}_i := (\tilde{y}_{is})_{s=1}^{m_i} \in R^{m_i}$ denotes the corresponding observations of the function y_i. In that case, a typical observational model is the following:

$$\tilde{y}_{is} = y_i(\theta_{is}) + \epsilon_{is},$$

where the terms (ϵ_{is}) correspond to noise added to the observations. From there, one can solve one smoothing problem per output observation (n problems in total) to represent each as a function. The obtained functions can then be processed using function-valued learning. In practice, smoothing consists in choosing a set of basis functions and finding a linear combination of those which best represent the function at hand. If the basis is orthogonal, it can boil down to computing the scalar products between the functions and the basis elements. Alternatively, if the basis is not orthogonal and/or if we have few observations per function and/or if those observations are very noisy, least square regression problems can be solved (Ramsay and Silverman, 2005). They give more possibilities to regularize. Yet another possible way to proceed, is to pose a function-valued regularized empirical risk minimization problem but using directly the available discrete observations.

1.2 Research Questions

The goal of this thesis is to propose new ways to solve the nonlinear FOR problem focusing on three key aspects. **(i)** We wish to reduce the computational complexity incurred by the functional nature of the outputs.**(ii)** We also want to go beyond the square loss using for instance robust or sparsity-inducing losses. **(iii)** Finally, we want to be able to apply our functional method directly to discrete observations.

Our first proposition was mostly motivated by points (i) and (iii). To reduce the computational complexity with respect to the functional outputs, we exploit a finite-dimensional representation. Yet we inject these representations directly in a function-valued regularized empirical risk minimization problem. We then learn to predict the expansion coefficients. In doing so we reduce the computational cost associated with the functional nature of the outputs. Moreover, the output functions intervene in the optimization problem only through their functional inner products with the atoms of the dictionary which can be estimated directly from discrete observations. We must then choose a dictionary of functions to represent the outputs and a vector-valued hypothesis class for the function predicting the coefficients. For the latter, we use *vector-valued reproducing kernel Hilbert spaces* (vv-RKHS) as they can model complex nonlinear dynamics and are quite easy to optimize over. Sticking to the functional square loss, the resulting problem in itself raises a series of questions of interest:

- What type of dictionaries of functions can be used to efficiently represent the functional outputs ?
- How can this problem of learning a vector-valued function projected to a functional space through a dictionary be solved efficiently ?
- Can we derive theoretical guarantees on the resulting estimators ?

However, other questions to extend this framework naturally arise:

- How can this framework be adapted to deal with other functional losses ?
- Can we make it scale well with respect to the characteristics of the input data as well ?
- How can we make the selection of the dictionary atoms automatic in the context of this problem ?

Our second contribution is more focused on point (ii). Regression in a function-valued RKHS (fv-RKHS) is a key nonlinear FOR method which is very flexible as it allows to shape the properties of the output functions through the choice of an output kernel. It has been introduced and studied using the square loss (Lian, 2007; Kadri et al., 2010, 2016) as well as possible extensions of the Huber and ϵ-insensitive losses (Laforgue et al., 2020). In the line of this last work, we propose to study this method using losses from the larger family of convoluted ones, which we introduce. We investigate the following questions:

- What type of sparsity and robustness of function-valued estimators can the loss functions we propose promote ?
- How can we tackle the corresponding problems through dual approaches ?
- How to represent the functional dual variables to make the problem amenable for numerical optimization ?

1.3 Organization and contributions

The remainder of the manuscript is organized as follows. Part I is dedicated to background and related works. Part II regroups our contributions on the FOR problem. Part III is a separate applied contribution on machine learning for wind energy.

Part I sets the stage by giving elements of background on which this thesis relies and it presents in some details the main existing nonlinear FOR methods.

- Chapter 2 recalls definitions and results around scalar-valued kernel methods and their vector-valued extensions. An emphasis is put on the rich properties of RKHSs and vv-RKHSs and their use in machine learning. They can be tailored to a precise application through the choice of kernel for the former and operator-valued kernel for the latter. Through the representer theorem or dualization, minimization of empirical risks over those spaces is very practical while their mathematical properties make theoretical analysis of estimators in terms of excess risk possible. We then recall key concepts for convex optimization of possibly non-differentiable functions. A particular focus is put on the tools from Lagrangian duality and proximal optimization we use later on.
- Chapter 3 focuses on how to exploit regularity to represent functions. After presenting some smoothing techniques from functional data analysis, we recall some key notions on conventional families such as Fourier bases, wavelets and splines. Then we investigate the advantageous possibilities of RKHSs to represent functions: the kernel determines the smoothness of the functions they contain, and they can be approximated using a finite dictionary. The last part is dedicated to dictionaries learnt from the observed functions which can be particularly efficient when the regularity of the functions to represent vary within their domain of definition.
- Chapter 4 presents the main approaches to solving the FOR problem nonlinearly. Most leverage RKHS or vv-RKHS in some way. The functional kernel ridge regression is an extension of the kernel ridge regression (KRR) using RKHSs of function-valued functions. It is also possible to represent the input and output functions on truncated orthonormal bases and regress the obtained output coefficients on the input one. The triple basis estimator does this separately for each output coefficient using approximate KRRs. More rooted in the functional data analysis literature, the kernel additive model extends the classic additive linear model by searching for the regression function in a RHKS.

Part II is dedicated to our contributions to solving the FOR problem. The chapters it contains address the questions raised in Section 1.2.

- Chapter 5 introduces the framework of projection learning. It exploits the possibilities to represent functions using a dictionary, but does so directly within a function-valued empirical risk minimization problem. We then focus on vv-RKHSs as hypothesis class to predict the coefficients and call the resulting method kernel projection learning (KPL). For the square loss we derive two ridge-type estimators in closed-form, one for fully observed output functions and one for partially observed ones. We exploit the separability of the operator-valued kernel to compute both estimators efficiently, essentially in $\mathcal{O}(n^3 + d^3)$ time with n

the number of samples and d the number of atoms in the dictionary. We demonstrate as well how to use other losses for KPL and benchmark our different estimators against those presented in Chapter 4.

- Chapter 6 is dedicated to the theoretical study of the two KPL estimators with the square loss. For each, a finite sample bound is derived on the excess risk. The first estimator deals with fully observed functions and therefore the number of samples n is the main quantity of interest, while the second deals with partially observed functions, consequently we also study the effect of the number of observations per function m. For both, consistency is derived from the bound.

- Chapter 7 focuses on regression in function-valued RKHS (fv-RKHS). Extensions of the squared ϵ-insensitive and Huber losses have been used in this context to obtain sparse or robust estimators. However, for functional outputs, these properties can be defined in a richer way. We introduce the family of convoluted losses encompassing a dedicated parameter p to control the degree of locality of these properties. We exploit the structure of these losses and the properties of fv-RKHSs to tackle the empirical risk minimization problem using Lagrangian duality. The dual variables are nevertheless infinite dimensional. To overcome this, we propose two representations in finite dimension which work for particular values of p and derive corresponding numerical algorithms. We show experimentally that the resulting estimators are sparse or robust either locally or globally.

- Chapter 8 corresponds to ongoing work. It extends the scope of KPL in several ways. When the dictionary is too redundant, predicting coordinates in the dictionary is inefficient and can lead to numerical instability. Therefore, we propose a simple effective-rank technique to overcome this issue. To alleviate the cubic complexity of KPL with respect to the number of input observations, we harness the possibilities offered by large scale kernel techniques. To that end, we formulate the feature projection learning (FPL) problem, a linear version of projection learning. For this problem with the square loss, we derive numerically efficient closed-form solutions. Finally, we propose to exploit a structured regularization in order to automatically select the relevant atoms of the output dictionary and propose an associated working set scheme to efficiently conduct the optimization.

Part III is a separate applied contribution on predicting local wind speed and wind power production using machine learning.

- Chapter 9 presents an applied contribution which is independent of the rest of the thesis. It focuses on machine learning for predicting local wind speed and wind power production in the very short term (almost immediate to four hour forecasts). For those time ranges, using both past local observations and predictions from a numerical weather model (NWP) can greatly improve performances. In that context we study the problematic of variable selection using both a linear and a nonlinear technique. Then using the selected variables, we benchmark several models and show that simple ones can perform better than more complex ones. In doing so, we show that most of the important nonlinear and complex dynamics are predicted well enough by the NWP. Across the

chapter, we also compare the direct (predict directly wind power) and the indirect (predict wind speed and pass the predictions through a power curve) ways of predicting.

We finish by highlighting that all the nonlinear FOR algorithms studied or introduced in this thesis are gathered in an open source Python library pyfunreg available on Github.

1.4 Publications

These contributions have resulted in the following peer-reviewed publications and preprints:

- **D. Bouche**, M. Clausel, F. Roueff and F. d'Alché-Buc. Nonlinear Functional Output Regression: A Dictionary Approach. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 235–243, 2021.
- A. Lambert, **D. Bouche**, Z. Szabó, and F. d'Alché-Buc. Functional Output Regression with Infimal Convolution: Exploring the Huber and ϵ-insensitive Losses In *International Conference on Machine Learning (ICML)*, pages 11844–1867, 2022.
- **D. Bouche**, R. Flamary, F. d'Alché-Buc, R. Plougonven, M. Clausel, J. Badosa and P. Drobinski. Wind power predictions from nowcasts to 4-hour forecasts: a learning approach with variable selection *Technical report*, 2022. (https://arxiv.org/abs/2204.09362). (submitted)

Part I

Background and related works

2

Kernel methods and convex optimization

Contents

In this chapter, we introduce several mathematical tools that we rely on throughout this manuscript. Section 2.1 focuses on reproducing kernel Hilbert spaces (RKHS) highlighting some of their key properties which make them a very popular choice for modeling functions. In particular, we show how these properties can be very advantageous in machine learning and have made RKHSs a cornerstone of many algorithms. Then in Section 2.2, we introduce an extension of these spaces to model vector-valued functions. We show some of their applications in machine learning which are central to this thesis and introduce related learning theory tools. Finally, in Section 2.3, we present key tools for convex optimization in general Hilbert spaces with a particular highlight on parametric duality and proximal operators.

2.1 Kernel Methods for scalar-valued outputs

Kernel methods hold a preponderant place in the landscape of machine learning methods. They are an implicit way to apply linear models to projections of the data in a RKHS. These spaces can be high-dimensional (possibly infinite-dimensional), which allows for modeling complex nonlinear dynamics in the original space. On top of that, RKHS combine this expressiveness with a sound mathematical construction and they enjoy many interesting properties. This facilitates both optimization and theoretical analysis of kernelized learning algorithms. Thus, without surprise they occupy an important place in statistical learning theory. We propose in this section to introduce the essential definitions and concepts that are of use for the rest of this thesis, however for more exhaustive accounts of many aspects of machine learning with kernels, we refer the reader to Schölkopf and Smola (2002); Shawe-Taylor and Cristianini (2004); Berlinet and Thomas-Agnan (2004); Steinwart and Christmann (2008).

2.1.1 Scalar-valued kernels and RKHSs

In machine learning, a central problem is that of the minimization of a regularized empirical risk. The choice of the hypothesis space, in which we search for a model, is key. Its richness determines the types of relationship between the input variables and the label that our model will be able to capture. However, in order to be able to derive efficient algorithms, optimization over this space must be practical. Reproducing kernel Hilbert spaces (RKHS, Aronszajn, 1950) are Hilbert spaces of functions which display many favorable characteristics in these regards.

As a first definition, a RKHSs is simply a Hilbert space on which the evaluation functional is continuous at all points of the input domain. This is indeed a highly desirable property in problems where one wants to optimize over h an objective function which involves discrete evaluations of the form $(h(x_i))_{i=1}^n$. Regularized empirical risk minimization problems constitute good examples.

Definition 2.1. *A Hilbert space $\mathcal{H} \subset \mathcal{F}(\mathcal{X}, \mathbb{R})$ is a reproducing kernel Hilbert space if and only if for any $x \in \mathcal{X}$, the following evaluation mapping is continuous*

$$ev_x : \begin{pmatrix} \mathcal{H} & \to & \mathbb{R} \\ h & \mapsto & h(x) \end{pmatrix}.$$

This definition is a bit theoretical, but thankfully the study of spaces verifying this property led to a more practical equivalent definition in terms of *reproducing kernels*.

Definition 2.2. *Let $\mathcal{X}$ be a set , a (scalar-valued) positive-definite reproducing kernel is a function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ that is*

1. *symmetric: $\forall (x_1, x_2) \in \mathcal{X}^2, \quad k(x_1, x_2) = k(x_2, x_1)$, and*
2. *positive definite: $\forall (x_i)_{i=1}^n \in \mathcal{X}^n$, $(\alpha_i)_{i=1}^n \in \mathbb{R}^n, \quad \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j k(x_i, x_j) \geq 0$.*

Remark 2.3. *In the remainder of this thesis, we use the term kernel to denote a scalar-valued positive-definite kernel. However, we deal as well with operator-valued positive-definite kernels further in the manuscript, in which case we always highlight the operator-valued nature explicitly.*

A very attractive aspect of kernels in machine learning is that they imply no assumption on the nature of the set $\mathcal{X}$, which can be of any kind as long as a kernel verifying the axioms from Definition 2.2 can be defined. This assumption is not very restrictive, and indeed kernels have been defined on a variety of very complex objects. To give a few examples, kernels can be used to compare measures (Cuturi et al., 2005) or possibly unaligned time series with different lengths (Cuturi et al., 2007). Another very notable field of application is computational biology where they have been used to compare several types of biological objects and sequences (see *e.g.* Schölkopf et al. 2004). Kernels have also been defined on graphs (see for instance the review of Kriege et al. 2020) or on permutations (Jiao and Vert, 2016). This list is by no means exhaustive but it can give an idea of the many scopes that kernels open regarding the spaces (input or output) that can be considered in machine learning.

Example 2.4 (Gaussian kernel)**.** *As a simple and classic example of kernel, when the space $\mathcal{X}$ is a Hilbert space, the Gaussian kernel is defined for a bandwidth parameter $\gamma > 0$ as:*

$$\forall (x_1, x_2) \in \mathcal{X}^2, \quad k(x_1, x_2) = e^{-\gamma \|x_1 - x_2\|_{\mathcal{X}}^2}. \tag{2.1}$$

The next theorem gives a more explicit understanding of kernels as a scalar product between embeddings of elements of $\mathcal{X}$ in a feature Hilbert space $\mathcal{V}$. This constitutes an alternate definition of a kernel. It originates from (Aronszajn, 1950).

Theorem 2.5. *Let $\mathcal{X}$ be a set. A function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a kernel if and only if there exists a Hilbert space $\mathcal{V}$ and a mapping $\psi : \mathcal{X} \to \mathcal{V}$ such that*

$$\forall (x_1, x_2) \in \mathcal{X}^2, \quad k(x_1, x_2) = \langle \psi(x_1), \psi(x_2) \rangle_{\mathcal{V}} .$$

The next theorem states the fundamental link between RKHSs and kernels. More precisely, a kernel defines a unique RKHS.

Theorem 2.6. *Let $\mathcal{X}$ be a set and let $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a kernel on $\mathcal{X}$. Then there exists a unique Hilbert space $\mathcal{H}_k \subset \mathcal{F}(\mathcal{X}, \mathbb{R})$ such that*

1. $\forall x \in \mathcal{X}, \quad k(., x) \in \mathcal{H}_k$, *and*
2. $\forall (h, x) \in \mathcal{H}_k \times \mathcal{X}, \quad h(x) = \langle h, k(., x) \rangle_{\mathcal{H}_k}$.

We then say that $\mathcal{H}_k$ is the RKHS of the kernel k.

The second property is generally referred to as the *reproducing property*. The equivalence between Definition 2.1 of a RKHS and the characterization through a kernel in Theorem 2.6 can be easily seen. Indeed if one defines a RKHS through a kernel, the use of the reproducing property combined with the Cauchy-Schwarz inequality leads to the continuity of the evaluation functional. Conversely, if the evaluation function is continuous, a kernel can be exhibited by using the Riesz representation theorem.

Remark 2.7. *$x \mapsto k(\cdot, x)$ is indeed a feature map in the sense of Theorem 2.5, the RKHS $\mathcal{H}_k$ itself playing the role of feature space. This feature map is generally referred to as the canonical feature map.*

2.1.2 RKHSs to represent functions

Several works on functional output regression (FOR) presented in this thesis use scalar-valued RKHSs to represent the output functions as well. For that reason, to avoid confusions, in this part we manipulate kernels that are defined on Θ, which denotes later the domain of definition of the output functions. Consequently, we make more restricting assumptions on Θ than we do on $\mathcal{X}$ and we make more assumptions on the kernel as well. However, first to motivate the use of RKHSs to represent functions, it is crucial to highlight their approximation capacities. *Universal kernels* are then a key notion.

Definition 2.8 (Universal kernels, Steinwart 2001)**.** *Let $k : \Theta \times \Theta \to \mathbb{R}$ be a kernel on a compact metric space Θ, k is said to be universal if its RKHS $\mathcal{H}_k$ is dense (with respect to the uniform norm) in the set $\mathcal{C}(\Theta)$ of continuous functions from Θ to $\mathbb{R}$.*

We briefly develop on this definition to make it more explicit. To so, we define the uniform norm for a function in $\mathcal{H}_k$–or $\mathcal{C}(\Theta)$–as:

$$\|h\|_\infty = \sup_{\theta \in \Theta} |h(\theta)| .$$

Then $\mathcal{H}_k$ is dense in $\mathcal{C}(\Theta)$ means that for all $g \in \mathcal{C}(\Theta)$ and for all $\epsilon > 0$, there exists a function $h \in \mathcal{H}_k$ such that $\|g - h\|_\infty \leq \epsilon$.

Remark 2.9. *This property is actually stated for the more general case where Θ is a Hausdorff topological space in Micchelli et al. (2006), and a kernel is then universal if it verifies Definition 2.8 on all compact subsets of Θ. Consequently, all the characterizations of universal kernels that they give remain true for the particular case of Θ compact.*

Many well known kernels can be shown to be universal. For instance it is the case of many shift-invariant kernels (see Definition 2.21) for which universality can be deduced from characteristics of their spectral measures (see *e.g.* Steinwart 2001; Micchelli et al. 2006; Sriperumbudur et al. 2011). As a concrete example, the well-known Gaussian kernel defined in Equation (2.1) is universal.

A particularly interesting class of kernels is that of Mercer kernels. They give rise to RKHSs which can be efficiently understood and approximated through the existence of a spectral decomposition of the integral operator associated to the kernel.

Definition 2.10 (Mercer kernel)**.** *A kernel k on Θ is said to be a Mercer kernel if*

1. *Θ is a compact metric space and,*
2. *$k : \Theta \times \Theta \to \mathbb{R}$ is continuous.*

A key tool to understand the properties of the functional space associated to a Mercer kernel is given by the following integral operator.

Definition 2.11 (Integral Operator)**.** *Let Θ be a compact metric space, let μ be a Borel measure on Θ and let $k : \Theta \times \Theta \to \mathbb{R}$ be a continuous kernel on Θ. The integral operator associated to μ and k is defined as:*

$$\mathrm{T}_{k,\mu} : \begin{pmatrix} \mathrm{L}^2(\Theta,\mu) & \to & \mathrm{L}^2(\Theta,\mu) \\ y & \mapsto & \left(\theta_1 \mapsto \int_\Theta y(\theta_2) k(\theta_1,\theta_2) \mathrm{d}\mu(\theta_2)\right) \end{pmatrix}.$$

Remark 2.12. *In order to lighten the notations, however, when it is possible we keep the measure μ implicit and shorten $\mathrm{T}_{k,\mu}$ to T_k. In the same way, we keep the measure implicit and shorten $\mathrm{L}^2(\Theta,\mu)$ to $\mathrm{L}^2(\Theta)$ when there is no ambiguity.*

To approximate the functions in the RKHS, we can use the spectral decomposition of this operator. Since k is continuous on Θ which is compact, for any $y \in \mathrm{L}^2(\Theta,\mu)$, $\mathrm{T}_k y \in \mathcal{C}(\Theta)$ and T_k is compact (see *e.g.* Cucker and Smale 2001, Chapter III, Proposition 1). However, since $\mathcal{C}(\Theta) \subset \mathrm{L}^2(\Theta,\mu)$, we can define the operator as taking its values in $\mathrm{L}^2(\Theta,\mu)$. Moreover the symmetry and positive definiteness of k respectively imply that T_k is self-adjoint and that it is positive (see *e.g.* Cucker and Smale 2001, Chapter III, Proposition 2). These properties enable us to apply the spectral theorem for self-adjoint, positive and compact operators. It states that there exist an at most countable family of functions $(\phi_l)_{l\in J}$ forming an orthonormal system in $\mathrm{L}^2(\Theta,\mu)$ and a corresponding set of eigenvalues $(\lambda_l)_{l\in J}$ such that

$$\forall y \in \mathrm{L}^2(\Theta,\mu), \quad \mathrm{T}_k y = \sum_{l\in J} \lambda_l \langle y, \phi_l \rangle_{\mathrm{L}^2(\Theta,\mu)} \phi_l. \tag{2.2}$$

Moreover, the eigenfunctions associated to nonzero eigenvalues are continuous since T_k actually maps $L^2(\Theta)$ to $\mathcal{C}(\Theta)$ and by definition, if $\lambda_l \neq 0$, $\phi_l = \frac{1}{\lambda_l} T_k(\phi_l)$. Also, by positivity of the operator, the eigenvalues are positive and without loss of generality we can suppose that they are ordered as $\lambda_1 \geq \lambda_2 \geq \cdots \geq 0$.

Such a spectral decomposition of T_k results in the following spectral representation of the kernel which is the well-known Mercer theorem (see *e.g.* Cucker and Smale 2001, Chapter III, Theorem 1).

Theorem 2.13 (Mercer)**.** *Let Θ be a compact domain, let μ be a Borel measure on Θ and let $k : \Theta \times \Theta \to \mathbb{R}$ be a Mercer kernel. Let $(\lambda_l, \phi_l)_{l=1}^{+\infty}$ be the eigenvalues/eigenfunctions pairs of the operator $T_{k,\mu}$. Then for all $\theta_1, \theta_2 \in \Theta$,*

$$k(\theta_1, \theta_2) = \sum_{l=1}^{+\infty} \lambda_l \phi_l(\theta_1) \phi_l(\theta_2), \tag{2.3}$$

where the convergence is uniform on Θ^2 (limit of the supremum over Θ^2 goes to zero) and absolute (for each $\theta_1, \theta_2 \in \Theta$, the absolute value goes to zero).

Nevertheless, it is not possible to exhibit the eigen decomposition of $T_{k,\mu}$ in closed-form for general μ and k (Rasmussen and Williams, 2006, Section 4.3). It is however possible in some particular cases, which is the object of the two following examples.

Example 2.14 (Gaussian kernel eigendecomposition with Gaussian measure)**.** *When $\Theta = \mathbb{R}$, μ is a Gaussian measure, and k is the Gaussian kernel, it is possible to derive the eigendecomposition in closed-form (Zhu et al. 1997b, Section 4, Rasmussen and Williams 2006, Section 4.3) based on Hermite polynomials. Let $k(\theta_1, \theta_2) = \exp(-\gamma(\theta_1 - \theta_2)^2)$ be a Gaussian kernel, and let μ be a Gaussian measure $\mathcal{N}(0, \sigma^2)$. Then the eigenvalues and eigenfunctions of T_k are given by*

$$\forall l \in \mathbb{N}, \quad \lambda_l = \sqrt{\frac{2a}{A}} B^l, \quad \phi_l(\theta) = \exp(-(c-a)\theta^2) H_l(\sqrt{2c}\theta),$$

where H_l is the l-th order Hermite polynomial (see e.g. *Gradshteyn and Ryzhik 1980, Section 8.95) and we defined the quantities*

$$\frac{1}{a} = 4\sigma^2, \quad c = \sqrt{a^2 + 2a\gamma}, \quad A = a + \gamma + c, \quad B = \frac{\gamma}{A}.$$

Example 2.15 (Laplace kernel eigendecomposition with Lebesgue measure)**.** *Another example in which we can compute the eigenvalues and eigenfunctions in closed-form is when $\Theta = [0, 1]$, k is the Laplace kernel with bandwidth parameter $\gamma = 1$ and μ is the Lebesgue measure– see Hawkins (1989, Section 4) and Kadri et al. (2016). Let $k(\theta_1, \theta_2) = \exp(-|\theta_1 - \theta_2|)$, the eigenvalues and eigenfunctions of T_k are given by*

$$\forall l \in \mathbb{N}, \quad \lambda_l = \frac{2}{1 + c_l^2}, \quad \phi_l(\theta) = c_l \cos(c_l \theta) + \sin(c_l \theta),$$

where $(c_l)_{l \in \mathbb{N}}$ are solutions to the equation $\cot(c) = \frac{1}{2}\left(c - \frac{1}{c}\right)$, cot denoting the cotangent function.

However, in most cases, the eigenvalues and eigenfunctions must be approximated numerically.

Example 2.16 (Approximate eigendecomposition)**.** *Consider that we are given a set of observations $\boldsymbol{\theta} = (\theta_s)_{s=1}^m \in \Theta^m$ which are drawn i.i.d. from the measure μ. We consider the following eigensystem associated to an empirical approximation to the integral operator T_k:*

$$\forall \theta \in \Theta, \quad \frac{1}{m}\sum_{s=1}^{m} k(\theta, \theta_s)\phi(\theta_s) = \lambda\phi(\theta). \tag{2.4}$$

If we evaluate Equation (2.4) for $\theta \in \{\theta_1, \cdots, \theta_m\}$, we get that $(m\lambda, \phi(\boldsymbol{\theta}))$ must be an eigenvalue/eigenvector pair of the kernel matrix $\mathrm{K} \in \mathbb{R}^{m\times m}$ associated to the kernel k and the observations $(\theta_s)_{s=1}^m$, where we use the convention that $\phi(\boldsymbol{\theta}) = (\phi(\theta_l))_{l=1}^d \in \mathbb{R}^m$. It therefore makes sense to perform an eigenvalue decomposition of K. Let then $\mathrm{U} \in \mathbb{R}^{m\times m}$ be the orthonormal matrix which columns $(\mathbf{u}_s)_{s=1}^m$ are the eigenvectors of K and let $(\hat{\lambda}_s)_{s=1}^m \in \mathbb{R}^m$ be the corresponding eigenvalues. So as to obtain approximate eigenfunctions we can evaluate at any $\theta \in \Theta$, a classic trick (see e.g. *Hoegaerts et al. 2005, Section 3.1) is to inject back the discrete eigenvectors into Equation (2.4):*

$$\forall s \in [\![m]\!], \quad \hat{\phi}_s : \theta \mapsto \frac{1}{\hat{\lambda}_s}\boldsymbol{k}(\theta)^{\mathrm{T}}\mathbf{u}_s. \tag{2.5}$$

Nevertheless, in order to obtain an orthonormal family of eigenfunctions in $\mathrm{L}^2(\Theta)$ as well as valid corresponding eigenvalues for the integral operator rather than the kernel matrix, we must do the following scalings:

$$\forall s \in [\![m]\!], \quad \tilde{\phi}_s := \sqrt{m}\hat{\phi}_s, \quad \tilde{\lambda}_s := \frac{1}{m}\hat{\lambda}_s. \tag{2.6}$$

The first comes from the fact that $\hat{\phi}_s(\boldsymbol{\theta}) = \mathbf{u}_s$ which implies

$$\|\hat{\phi}_s(\boldsymbol{\theta})\|^2_{\mathrm{L}^2(\Theta)} \approx \frac{1}{m}\|\mathbf{u}_s\|^2_{\mathbb{R}^m} = \frac{1}{m}.$$

Then, for the eigenfunctions to have approximately unit $\mathrm{L}^2(\Theta)$-norm, we must multiply them by $\sqrt{m}$ for all $s \in [\![m]\!]$. Injecting Equation (2.6) into Equation (2.5), we approximate the eigenfunctions for the integral operator as

$$\forall s \in [\![m]\!], \quad \tilde{\phi}_s : \theta \mapsto \frac{1}{\sqrt{m}\tilde{\lambda}_s}\boldsymbol{k}(\theta)^{\mathrm{T}}\mathbf{u}_s. \tag{2.7}$$

The pairs $(\tilde{\lambda}_s, \tilde{\phi}_s)_{s=1}^m$ can then be used as approximated eigenvalue/eigenfunction pairs for the integral operator T_k.

All the properties we have presented makes RKHSs particularly attractive to use as a hypothesis class in regularized empirical risk minimization problems.

2.1.3 Learning in scalar-valued RKHSs

We place ourselves in the supervised learning setting where we have access to observations $(x_i, y_i)_{i=1}^n \in (\mathcal{X} \times \mathbb{R})^n$ corresponding to i.i.d. realizations from a set of random variables (X, Y) whose distribution is unknown. To statistically infer function on $\mathcal{X}$ producing predictions which are statistically coherent with (X, Y), a classic approach is to formulate an empirical risk minimization problem. To that end, let $\ell : \mathbb{R}^2 \to \mathbb{R}$ be any loss function. Consider a kernel k and its associated RKHS $\mathcal{H}_k$ which we use

as our hypothesis space. However to avoid overfitting of the training data, and to benefit from the representation properties of RKHSs, we add a regularization term. It consists of the RKHS norm multiplied by an intensity parameter $\lambda > 0$. This yields the problem:

$$\min_{h \in \mathcal{H}_k} \frac{1}{n} \sum_{i=1}^{n} \ell(h(x_i), y_i) + \lambda \|h\|^2_{\mathcal{H}_k}. \tag{2.8}$$

It has been widely studied for various continuous convex losses. For the square loss, we obtain the kernel ridge regression estimator. Other founding examples are based on non-differentiable losses. The hinge loss gives rise to the well-known support vector machine (Cortes and Vapnik, 1995, SVM) for classification while the ϵ insensitive loss yields the support vector regression (Drucker et al., 1996, SVR).

However, the space $\mathcal{H}_k$ can be high-dimensional (possibly infinite-dimensional) so at first sight Problem 2.8 is not straightforward to solve. However, a key advantage of RKHSs is that they benefit from a *representer theorem*. It enables the reformulation of the problem in finite dimension.

Theorem 2.17 (Representer theorem)**.** *Let $\mathcal{X}$ be a set, let k be a kernel on this set and let $\mathcal{H}_k$ be its associated RKHS. Consider a set of points $(x_i)_{i=1}^n \in \mathcal{X}^n$. Let $V : \mathbb{R}^{n+1} \to \mathbb{R}$ be a function which is strictly increasing with respect to its last argument. Then any solution $\hat{h}$ to the problem*

$$\min_{h \in \mathcal{H}_k} V\Big(h(x_1), \cdots, h(x_n), \|h\|_{\mathcal{H}_k}\Big),$$

can be written in the form

$$\hat{h} = \sum_{i=1}^{n} k(., x_i)\alpha_i \tag{2.9}$$

for some $(\alpha_i)_{i=1}^n \in \mathbb{R}^n$.

This powerful theorem is a direct consequence of a simple orthogonality argument.

Proof Let $\mathcal{S} = \text{Span}\left\{(k(., x_i))_{i=1}^n\right\}$. This is a finite dimensional subspace of $\mathcal{H}_k$. As a consequence, any $h \in \mathcal{H}_k$ can be written as $h = \hat{h} + h^\perp$ with $\hat{h}, h^\perp \in \mathcal{S} \times \mathcal{S}^\perp$. On the one hand, the reproducing property combined with the orthogonality implies that for all $i \in [\![n]\!]$, $h(x_i) = \hat{h}(x_i)$. On the other hand, Pythagoras' theorem implies that $\|h\|_{\mathcal{H}_k} \geq \|\hat{h}\|_{\mathcal{H}_k}$. Consequently, $h^\perp$ necessarily make the objective V increase, therefore for any minimizer, we should have $h^\perp = 0$ which concludes the proof. ■

Example 2.18 (Kernel ridge regression (KRR))**.** *The square loss $\ell(\hat{y}, y) = \frac{1}{2}(\hat{y} - y)^2$ is a common choice of loss function. Injecting the parametrization given by the representer theorem in Equation (2.9) into Problem 2.8 and setting the gradient to zero, the optimal coefficients $\hat{\alpha} \in \mathbb{R}^n$ can be computed in closed-form. Let us denote by $\mathrm{K} \in \mathbb{R}^{n \times n}$ the kernel matrix associated to the observations $(x_i)_{i=1}^n$ and the kernel k, and by $y \in \mathbb{R}^n$ the vector containing the targets $(y_i)_{i=1}^n$. Then $\hat{\alpha}$ is given by*

$$\hat{\alpha} = (\mathrm{K} + \lambda n I)^{-1} y. \tag{2.10}$$

2.1.4 Large scale learning in RKHS

We see by now that kernel methods are very attractive in many aspects. Nevertheless, they do have a few shortcomings, the most notable one being their computational complexity. Theorem 2.17 implies that the number of parameters of the model is the number of available observations n. Moreover, it is necessary to compute the kernel matrix K which incur a memory complexity of $\mathcal{O}(n^2)$. Regarding time complexity, if we use the square loss, solving Equation (2.10) costs $\mathcal{O}(n^3)$ time . Indeed iterative methods can be used, but those two complexities are an overall good summary. Consequently, when n is low and the dimension of the input data is high (if dimension is relevant), kernel methods are a very good choice as the input dimension is only involved to compute the kernel evaluations. However, when the number of samples is high, kernel methods can no longer be applied in their traditional form. Two main approaches exist to overcome this problem. The first is Nyström's method (Williams and Seeger, 2001) and the second is to use random Fourier features (Rahimi and Recht, 2007).

Nyström method

Williams and Seeger (2001) propose to approximately solve the linear system associated to the KRR problem from Equation (2.10) using a low rank approximation to the kernel matrix based on a random subset of its columns. This approach can be used for any kernel method. More precisely, consider a random subset of the input observations $(\tilde{x}_i)_{i=1}^q$ for $q \leq n$. Let $\mathrm{K}_{nq} \in \mathbb{R}^{n\times q}$ be the matrix with entries $k(x_i, \tilde{x}_j)$ and let $\mathrm{K}_{qq} \in \mathbb{R}^q$ be the matrix with entries $k(\tilde{x}_t, \tilde{x}_j)$, $i \in [\![n]\!]$, $j, t \in [\![q]\!]$. The following low-rank approximation is used

$$\mathrm{K} \approx \mathrm{K}_{nq}\mathrm{K}_{qq}^{\dagger}\mathrm{K}_{nq}^{\mathrm{T}}. \tag{2.11}$$

This boils down to looking for a solution to Problem 2.8 using the parametrization from the representer theorem in Equation (2.9) restricted to a random subset of observations. In other words we search in $\mathrm{Span}\left\{(k(\cdot, \tilde{x}_i))_{i=1}^q\right\}$ instead of $\mathrm{Span}\left\{(k(\cdot, x_i))_{i=1}^n\right\}$.

Example 2.19. *We have the following closed-form solution for the KRR with Nyström approximation (see* e.g. *Rudi et al. 2015):*

$$\tilde{h} = \sum_{i=1}^{q} \tilde{\alpha}_i k(\cdot, \tilde{x}_i), \quad \textit{with} \quad \tilde{\alpha} = (\mathrm{K}_{nq}^{\mathrm{T}}\mathrm{K}_{nq} + \lambda n \mathrm{K}_{qq})^{\dagger}\mathrm{K}_{nq}^{\mathrm{T}} y \in \mathbb{R}^q. \tag{2.12}$$

Drineas and Mahoney W. (2005) shows that the randomized approximation in Equation (2.11) gets very close with high probability and in expectation to the best rank q approximation to K. Then, many developments and refinements have been proposed, including the proposition and study of several sampling techniques (see *e.g.* Kumar et al. 2012 and reference therein). In the context of statistical learning, Rudi et al. (2015) show that KRR with Nyström approximation can achieve optimal learning bounds for a suitable number of sampled observations. Combined with an efficient preconditioning and a stochastic gradient solver (Rudi et al., 2017), Nyström method leads to a very efficient implementation of KRR. While more recently, an implementation optimized for GPUs for this approach allowed to use KRR on dataset encompassing billions of samples (Meanti et al., 2020).

Example 2.20 (Nyström features)**.** *It is possible to derive Nyström features. These can be used to apply the Nyström method to any linear model regularized using the squared Euclidean norm. Yang et al. (2012) highlight that then, this feature-based approach is*

equivalent to the original one. Let $(\tilde{\lambda}_l, \tilde{\mathbf{u}}_l)_{l=1}^r$ *be the eigenvalues/eigenvectors couples of the matrix* K_{qq}*, consider the associated matrices* $\tilde{\Lambda} = \mathrm{diag}((\tilde{\lambda}_l)_{l=1}^r) \in \mathbb{R}^{r\times r}$ *and* $\tilde{\mathrm{U}} = [\mathbf{u}_l]_{l=1}^r \in \mathbb{R}^{q\times r}$*. The Nyström features are then the following*

$$\tilde{\psi}(x) = \tilde{\Lambda}^{-\frac{1}{2}} \tilde{\mathrm{U}}^{\mathrm{T}} (k(x, \tilde{x}_1), \cdots, k(x, \tilde{x}_q))^{\mathrm{T}} \in \mathbb{R}^r.$$

Random Fourier features

Another popular possibility is to learn in an approximate RKHS using a random feature map as proposed in Rahimi and Recht (2007). Let us consider that $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a *shift-invariant* kernel. Such a kernel only depends on the difference between its inputs. Therefore we must make the restriction that $\mathcal{X}$ is a vector space so that the notion of difference is meaningful.

Definition 2.21 (Shift-invariant kernel)**.** *Let* $k : \mathcal{X} \times \mathcal{X} \to$ *be a kernel on a vector space* $\mathcal{X}$*. We say that* k *is shift invariant if there exists a function* k_0 *on* $\mathcal{X}$ *such that for all* $(x_1, x_2) \in \mathcal{X}^2$*,* $k(x_1, x_2) = k_0(x_1 - x_2)$*.*

Remark 2.22. *The positive definiteness of* k *therefore implies that* k_0 *is a positive definite function. Since* k *is real-valued,* k_0 *must be as well, and therefore its positive definiteness implies that it is symmetric, thus for all* $x \in \mathcal{X}$*,* $k_0(-x) = k_0(x)$*.*

We now restrict the input space to $\mathcal{X} = \mathbb{R}^c$. Let $k : \mathbb{R}^c \times \mathbb{R}^c \to \mathbb{R}$ be a continuous kernel and suppose further that it is shift invariant with base function k_0. To build a random feature map, Rahimi and Recht (2007) rely on Bochner's theorem (see for instance Wendland 2004, Theorem 6.6).

Theorem 2.23 (Bochner)**.** *A continuous function* $k_0 : \mathbb{R}^c \to \mathbb{R}$ *is positive definite if and only if it is the Fourier transform of a finite nonnegative Borel measure* ρ *on* $\mathbb{R}^c$*.*

$$k_0(x) = \int_{\mathbb{R}^c} e^{i\omega^{\mathrm{T}} x} \mathrm{d}\rho(\omega) = \int_{\mathbb{R}^c} \cos(\omega^{\mathrm{T}} x) \mathrm{d}\rho(\omega).$$

The second equality is valid since k_0 is real valued and ρ is defined on $\mathbb{R}^c$, therefore we can ignore the imaginary part.

Using Bochner's theorem, for all $(x_1, x_2) \in (\mathbb{R}^c)^2$,

$$k(x_1, x_2) = \int_{\mathbb{R}^c} \cos(\omega^{\mathrm{T}}(x_1 - x_2)) \mathrm{d}\rho(\omega). \tag{2.13}$$

Without loss of generality, we can choose the measure ρ to be a probability measure. Indeed, if ρ does not integrate to 1, we notice that $k_0(0) = \rho(\mathbb{R}^c)$, so we can scale ρ by $\frac{1}{k_0(0)}$. The integral in Equation (2.13) can be approximated by Monte Carlo, yielding a fair approximation of the kernel (see Sriperumbudur and Szabó (2015) for an in depth analysis). Let $(\omega_r)_{r=1}^q$ be i.i.d. draws from the probability measure ρ, the approximate kernel is given by

$$\tilde{k}(x_1, x_2) = \frac{1}{q} \sum_{r=1}^{q} \cos(\omega_r^{\mathrm{T}}(x_1 - x_2)). \tag{2.14}$$

Then linearizing the cosines, we get the feature map:

$$\tilde{\psi} : x \mapsto \frac{1}{\sqrt{q}}(\cos(\omega_1^{\mathrm{T}} x), \cdots, \cos(\omega_q^{\mathrm{T}} x), \sin(\omega_1^{\mathrm{T}} x), \cdots, \sin(\omega_q^{\mathrm{T}} x)). \tag{2.15}$$

It can be used in any kernelizable linear model. The corresponding problem is then solved without paying the high complexity in the number of samples n, but rather in the number of random features q. Using this feature map amounts to learning in the RKHS associated to the approximated kernel $\tilde{k}$.

The study of those random features maps has attracted a lot of attention. On the practical side, (Le et al., 2013) propose to reduce the complexity of computing and storing feature maps, while improvements based on quasi Monte Carlo have been studied extensively in Avron et al. (2016). Zhang et al. (2019) introduce a way to reduce the memory usage of learning with random Fourier features while retaining competitive empirical performances. On the theoretical side, generalization properties of random Fourier features have also been investigated thoroughly, first in the case of the square loss (Rudi and Rosasco, 2017), while a broader and unified framework for theoretical analysis has been introduced later (Li et al., 2021).

A broad class of nonlinear FOR methods as well as new methods introduced in this thesis hinge on an extension of RKHSs to model functions taking their values in a Hilbert space. We give an introduction to these spaces in the following section.

2.2 Kernel Methods for vector-valued outputs

To bring the many advantages of RKHSs for modeling scalar-valued functions to the more general setting where the outputs of the functions lie in a Hilbert space, vector-valued RKHSs (vv-RKHSs) have been introduced (Pedrick, 1957). The vectors may be of finite or infinite dimension depending on the Hilbert space considered, therefore vv-RKHSs allow for modeling multi-output functions as well as function-valued functions.

2.2.1 Operator-valued kernels and vector-valued RKHSs

Let $\mathcal{Y}$ be a separable real Hilbert space. To build the vector-valued extension, the reproducing kernels are no longer real-valued. Now they take their values in the linear space $\mathcal{L}(\mathcal{Y})$ of bounded operators from $\mathcal{Y}$ to $\mathcal{Y}$. We give an overview of the differences between scalar-valued RKHSs and vv-RKHSs in Table 2.1. As in the scalar-valued case, vv-RKHSs can be characterized by the continuity of the evaluation operator at all points of the input space.

Definition 2.24. *A Hilbert space $\mathcal{H} \subset \mathcal{F}(\mathcal{X}, \mathcal{Y})$ is a vector-valued reproducing kernel Hilbert space if and only if for any $x \in \mathcal{X}$, the following evaluation mapping is continuous*

$$ev_x : \begin{pmatrix} \mathcal{H} & \to & \mathcal{Y} \\ h & \mapsto & h(x) \end{pmatrix}.$$

However, a more usable characterization of vv-RKHSs is given through operator-valued kernels (OVK).

Table 2.1: Comparison between scalar (k) and operator-valued kernels (K).

	scalar-valued kernel	operator-valued kernel
kernel	$k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$	$\mathsf{K} : \mathcal{X} \times \mathcal{X} \to \mathcal{L}(\mathcal{Y})$
symmetry	$k(x_1, x_2) = k(x_2, x_1)$	$\mathsf{K}(x_1, x_2) = \mathsf{K}(x_2, x_1)^\#$
positive-definiteness	$\sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j k(x_i, x_j) \geq 0$	$\sum_{i=1}^n \sum_{j=1}^n \langle y_i, \mathsf{K}(x_i, x_j) y_j \rangle_{\mathcal{Y}} \geq 0$
reproducing property	$h(x) = \langle h, k(\cdot, x) \rangle_{\mathcal{H}_k}$	$h(x) = \mathsf{K}_x^\# h$
feature map	$\psi : \mathcal{X} \to \mathcal{V}$	$\Psi : \mathcal{X} \to \mathcal{L}(\mathcal{V}, \mathcal{Y})$
linear parametrization	$h = \langle \psi(\cdot), v \rangle_{\mathcal{V}}$	$h = \Psi(\cdot) v$

Definition 2.25 (Operator-valued kernel)**.** *Let $\mathcal{X}$ be a set, an operator-valued kernel (OVK) is a function $\mathsf{K} : \mathcal{X} \times \mathcal{X} \to \mathcal{L}(\mathcal{Y})$ that is*

1. *symmetric: $\forall (x_1, x_2) \in \mathcal{X}^2, \quad \mathsf{K}(x_1, x_2) = \mathsf{K}(x_2, x_1)^\#$, and*
2. *positive definite: $\forall (x_i)_{i=1}^n \in \mathcal{X}^n$, $(y_i)_{i=1}^n \in \mathcal{Y}^n, \quad \sum_{i=1}^n \sum_{j=1}^n \langle y_i, \mathsf{K}(x_i, x_j) y_j \rangle_{\mathcal{Y}} \geq 0$.*

Remark 2.26 (Meaning of vector-valued)**.** *In the present chapter, we use the term vector-valued RKHS for a RKHS of functions with values in a Hilbert space, regardless of its dimension of this space. However, further in this thesis, we will make a distinction between vector-valued and function-valued RKHSs. We will use the former when $\mathcal{Y}$ is $\mathbb{R}^d$ for some $d \in \mathbb{R}$ and the latter when $\mathcal{Y}$ is a Hilbert space of functions.*

The following theorem is analogous to Theorem 2.6, more precisely it states the uniqueness of the vv-RKHS associated to an OVK and it characterizes it. However, to state it, we first need to define the following linear operators. Let K be an OVK, then for any $x_1 \in \mathcal{X}$, we define $\mathsf{K}_{x_1} \in \mathcal{L}(\mathcal{Y}, \mathcal{F}(\mathcal{X}, \mathcal{Y}))$ as

$$\mathsf{K}_{x_1} : y \mapsto \mathsf{K}_{x_1} y, \quad \textit{with } \mathsf{K}_{x_1} y : x_2 \mapsto \mathsf{K}(x_2, x_1) y. \tag{2.16}$$

Theorem 2.27. *Let $\mathcal{X}$ be a set and let $\mathsf{K} : \mathcal{X} \times \mathcal{X} \to \mathcal{L}(\mathcal{Y})$ be an operator-valued kernel on $\mathcal{X}$. Then there exists a unique Hilbert space $\mathcal{H}_\mathsf{K} \subset \mathcal{F}(\mathcal{X}, \mathcal{Y})$ such that*

1. *$\forall (x, y) \in \mathcal{X} \times \mathcal{Y} \quad \mathsf{K}_x y \in \mathcal{H}_\mathsf{K}$, and*
2. *$\forall (h, x) \in \mathcal{H}_\mathsf{K} \times \mathcal{X}, \quad h(x) = \mathsf{K}_x^\# h$.*

Remark 2.28. *This corresponds to Proposition 2.3 in Carmeli et al. (2006). They deal with Hilbert spaces over the field $\mathbb{C}$ whereas we indeed deal with Hilbert spaces over the field $\mathbb{R}$. However, as they highlight at the end of their proof, the additional requirements in Definition 2.25 that for all $(x_1, x_2) \in \mathcal{X}^2$, $\mathsf{K}(x_1, x_2) = \mathsf{K}(x_2, x_1)^\#$ makes the theorem valid for real Hilbert spaces as well.*

Remark 2.29. *The second item of Theorem 2.27 is the equivalent of the reproducing property in the scalar-valued case.*

Alternatively OVKs can be characterized by the existence of an operator-valued feature map in the following sense.

Theorem 2.30. *Let $\mathcal{X}$ be a set. A function $\mathsf{K} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is an OVK if and only if there exists a Hilbert space $\mathcal{V}$ and a mapping $\Psi : \mathcal{X} \to \mathcal{L}(\mathcal{V}, \mathcal{Y})$ such that*

$$\forall (x_1, x_2) \in \mathcal{X}^2, \quad \mathsf{K}(x_1, x_2) = \Psi(x_1) \Psi(x_2)^\#.$$

Remark 2.31. *Similarly to the scalar-valued case, a canonical feature map can be exhibited from the OVK* K*. Indeed, consider* $\Psi : x \mapsto \mathsf{K}_x^\# \in \mathcal{L}(\mathcal{H}_\mathsf{K}, \mathcal{Y})$*, we do have that for all* $(x_1, x_2) \in \mathcal{X}^2$*,* $\mathsf{K}(x_1, x_2) = \mathsf{K}_{x_1}^\# \mathsf{K}_{x_2} = \Psi(x_1)\Psi(x_2)^\#$*. Here again,* $\mathcal{H}_\mathsf{K}$ *plays the role of feature space* $\mathcal{V}$*.*

Among OVKs, a very popular subclass is that of separable ones, for they are simple to interpret and can drastically simplify computations in some cases.

Definition 2.32 (Separable OVK)**.** *We say that an OVK is separable if there exist a kernel* $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ *and a positive self-adjoint bounded operator* $\mathrm{B} \in \mathcal{L}(\mathcal{Y})$ *such that*

$$\forall (x_1, x_2) \in \mathcal{X}^2, \quad \mathsf{K}(x_1, x_2) = k(x_1, x_2)\mathrm{B}.$$

Example 2.33. *As an example, if the output space is* $\mathbb{R}^d$*, then if we consider a kernel* k *on* $\mathcal{X}$ *and* $\mathrm{B} \in \mathbb{R}^{d \times d}$ *a symmetric positive matrix, the following is a valid OVK:*

$$\forall (x_1, x_2) \in \mathcal{X}^2, \quad \mathsf{K}(x_1, x_2) = k(x_1, x_2)\mathrm{B}.$$

Since the output space is $\mathbb{R}^d$*, we represent an operator from* $\mathcal{L}(\mathbb{R}^d)$ *as a matrix in* $\mathbb{R}^{d \times d}$*.*

For function-valued learning it is common to use the product between an input kernel and the integral operator associated with an output kernel.

Example 2.34. *Suppose that* $\mathcal{Y} = \mathsf{L}^2(\Theta, \mu)$ *with* Θ *a compact metric space. Let* $k_\mathcal{X}$ *be a kernel defined on the input space* $\mathcal{X}$*, let* k_Θ *be a kernel defined on* Θ *and let* μ *be a Borel measure on* Θ*. We then define the OVK*

$$\forall (x_1, x_2) \in \mathcal{X}^2, \quad \mathsf{K} = k_\mathcal{X}(x_1, x_2)\mathrm{T}_{k_\Theta}, \tag{2.17}$$

where T_{k_Θ} *is the integral operator from Definition 2.11 associated to the kernel* k_Θ *and the measure* μ*. This is indeed a valid OVK.*

This kernel has been used extensively for FOR using vv-RKHSs (Kadri et al., 2010, 2016; Laforgue et al., 2020). It forces the modeled functions to lie in $\mathcal{H}_{k_\Theta}$ the RKHS associated to the kernel k_Θ. This is the object of the next remark.

Remark 2.35. *The operator* T_{k_Θ} *maps surjectively* $\mathcal{Y} = \mathsf{L}^2(\Theta, \mu)$ *to* $\mathcal{H}_{k_\Theta}$*, the RKHS associated to the kernel* k_Θ*. For FOR using vv-RKHSs, two points of views are somewhat equivalent. On the one hand, we can consider that the outputs are lying in* $\mathsf{L}^2(\Theta, \mu)$ *and use the vv-RKHS associated to the kernel defined in Equation (2.17) as a hypothesis class. On the other hand, we can also say the outputs should lie in* $\mathcal{H}_{k_\Theta}$ *and then use the vv-RKHS associated to* $k_\mathcal{X}\mathrm{I}_{\mathcal{H}_{k_\Theta}}$ *as a hypothesis class. These two approaches are almost equivalent in the sense that* $\mathcal{H}_{k_\mathcal{X}\mathrm{I}_{\mathcal{H}_{k_\Theta}}}$ *and* $\mathcal{H}_{k_\mathcal{X}\mathrm{T}_{k_\Theta}}$ *are unitarily equivalent–see Carmeli et al. (2010, Example 6, Example 7) for details.*

2.2.2 Learning in vv-RKHSs

We place ourselves in the supervised learning setting again, but this time (X, Y) is a couple of random variables taking its values in $\mathcal{X} \times \mathcal{Y}$. To formulate an empirical risk minimization problem, we need to define a loss function which now acts on the Hilbert space $\mathcal{Y}$. Let $L : \mathcal{Y} \times \mathcal{Y} \mapsto \mathbb{R}$ be such a loss. We then consider the following problem with $\mathcal{H}_\mathsf{K}$ a vv-RKHS associated to an OVK $\mathsf{K} : \mathcal{X} \times \mathcal{X} \to \mathcal{L}(\mathcal{Y})$:

$$\min_{h \in \mathcal{H}_K} \frac{1}{n} \sum_{i=1}^{n} L(h(x_i), y_i) + \lambda \|h\|^2_{\mathcal{H}_K}. \tag{2.18}$$

Such problems arise in various fields of application when prediction in a Hilbert space is required. The simplest case is that of multi-output learning where $\mathcal{Y} = \mathbb{R}^d$. We can predict separately the d output coordinates, however in order to leverage the relationship between the output coordinates, it may be better to tap into vector-valued prediction (Micchelli and Pontil, 2005; Álvarez et al., 2012). FOR is the application of particular interest for this thesis. The Hilbert space $\mathcal{Y}$ is then infinite-dimensional, and we can no longer predict separately the multiple outputs. Vv-RKHSs have been applied successfully to overcome this challenge (Lian, 2007; Kadri et al., 2010, 2016). Another field where regression in vv-RKHSs have led to state of the art performances is structured output regression. The output space $\mathcal{Y}$ then consists of a scalar-valued RKHS containing embeddings of structured objects (Brouard et al., 2011; Kadri et al., 2013; Brouard et al., 2016; Laforgue et al., 2020). This is a surrogate approach, therefore to obtain a prediction in the original structured space, a pre-image problem must be solved.

As in the scalar case, a solution to Problem 2.18 can be parametrized by a set of vectors $(\alpha_i)_{i=1}^n \in \mathcal{Y}^n$.

Theorem 2.36 (Micchelli and Pontil 2005, Theorem 4.2)**.** *Let $V : \mathcal{Y}^n \times \mathbb{R} \to \mathbb{R}$ be a function such that for any $\mathbf{y} \in \mathcal{Y}^n$, the partial function $t \mapsto V(\mathbf{y}, t)$ is strictly increasing. Then any solution $\hat{h}$ to the problem*

$$\min_{h \in \mathcal{H}_K} V\Big((h(x_1), \cdots, h(x_n)), \|h\|_{\mathcal{H}_K}\Big), \tag{2.19}$$

can be written in the form

$$\hat{h} = \sum_{i=1}^{n} \mathsf{K}_{x_i} \alpha_i, \tag{2.20}$$

for some $(\alpha_i)_{i=1}^n \in \mathcal{Y}^n$, where for $x \in \mathcal{X}$, the operator $\mathsf{K}_x : \mathcal{Y} \to \mathcal{H}_K$ is defined as in Equation (2.16).

We give here an overview of the proof which is based on the minimal norm interpolant principle.

Proof Suppose that a minimizer $\hat{h}$ exists and set $\hat{\mathbf{y}} = (\hat{y}_i)_{i=1}^n$ with for all $i \in [\![n]\!]$, $\hat{y}_i = \hat{h}(x_i)$. Let $h \in \mathcal{H}_K$ be any function such that for all x_i, $i \in [\![n]\!]$, $h(x_i) = \hat{y}_i$. By definition of $\hat{h}$ we have that

$$V\Big(\hat{\mathbf{y}}, \|\hat{h}\|^2_{\mathcal{H}_K}\Big) \leq V\Big(\hat{\mathbf{y}}, \|h\|^2_{\mathcal{H}_K}\Big).$$

Therefore

$$\|\hat{h}\|_{\mathcal{H}_K} = \min\Big\{\|h\|_{\mathcal{H}_K} \ : \ h(x_i) = \hat{y}_i, \ \ i \in [\![n]\!], \ \ h \in \mathcal{H}_K\Big\}. \tag{2.21}$$

This corresponds to a minimal norm interpolant problem. Since $\hat{h} \in \mathcal{H}_K$, by definition $\hat{\mathbf{y}}$ is in the range of the evaluation operator defined as $h \mapsto (h(x_i))_{i=1}^n$. Consequently, Theorem 3.2 from Micchelli and Pontil (2005) guarantees that the solution $\hat{h}$ to Problem 2.21 indeed has the form given in Equation (2.20). ■

This theorem reduces the domain of optimization to a subspace of $\mathcal{H}_\mathsf{K}$. However, we still have to optimize over n vectors in $\mathcal{Y}$, which remains highly problematic if $\mathcal{Y}$ is infinite dimensional. How to alleviate this problem is a central problematic in this thesis which we will tackle in different manners in Chapter 5 and Chapter 7.

For regularized empirical risk minimization problems (Problem 2.18), an attractive possibility to obtain a parameterization similar to that given by the representer Equation (2.20) is to exploit parametric duality. For minimization in vv-RKHSs, we can get a dual problem that is equivalent to the primal one which can be simpler to solve (especially if the loss is non-differentiable). This has been a cornerstone of many scalar-valued kernel methods such as the SVM or the SVR. The approach can be generalized to vv-RKHSs which is the object of the following theorem. To state it we define the partial losses for $i \in [\![n]\!]$ as $L_i : y \mapsto L(y, y_i)$.

Theorem 2.37 (Dualization, Brouard et al. 2016)**.** *Suppose that L is proper, lower semi-continuous and convex with respect to its first argument, then the solution $\hat{h}$ of Problem 2.18 is unique and is given by*

$$\hat{h} = \frac{1}{2n\lambda} \sum_{i=1}^{n} \mathsf{K}_{x_i} \hat{\alpha}_i, \tag{2.22}$$

where $(\hat{\alpha}_i)_{i=1}^n \in \mathcal{Y}^n$ is the solution of the dual problem

$$\inf_{(\alpha_i)_{i=1}^n \in \mathcal{Y}^n} \sum_{i=1}^{n} L_i^\star(-\alpha_i) + \frac{1}{4n\lambda} \sum_{i=1}^{n} \sum_{j=1}^{n} \langle \alpha_i \mathsf{K}(x_i, x_j) \alpha_j \rangle_\mathcal{Y}. \tag{2.23}$$

Where for $i \in [\![n]\!]$, $L_i^\star$ denotes the Fenchel-Legendre transform of L_i (Definition 2.43). The representation in Equation (2.22) and that from the representer theorem are the same up to a scalar scaling. However, solving Problem 2.23 remains an issue as the dual variables are in $\mathcal{Y}$.

Therefore, vv-RKHSs offer a very rich hypothesis class for learning problems with targets in a Hilbert space. Nevertheless, specific work beyond the representer theorem or the dualization needs to be performed to make the problem amenable for numerical optimization. Indeed, this first layer of simplification is not enough, as $\mathcal{Y}$ is infinite dimensional, optimizing over variables in $\mathcal{Y}^n$ remains highly problematic. To tackle that challenge, the assumption that the OVK is separable (Definition 2.32) is generally key. In Laforgue et al. (2020), they consider a restricted subclass of kernels under which the span of the output functions $(y_i)_{i=1}^n$ is stable; *i.e.* for all $(x_1, x_2) \in \mathcal{X}$ and for all $y \in \mathrm{Span}((y_i)_{i=1}^n)$, $\mathsf{K}(x_1, x_2)y \in \mathrm{Span}((y_i)_{i=1}^n)$. Then under several assumptions on the losses as well, a double-representer theorem is proved. It allows to reduce the number of variables in Problem 2.23 to n^2 real numbers.

Now that we have introduced the advantages and drawbacks of learning in vv-RKHSs, we turn to learning theory. More precisely, we introduce a framework exploiting integral operators. It can be used to obtain theoretical guarantees for learning algorithms in vv-RKHSs using the square loss.

2.2.3 Learning theory with integral operators in vv-RKHSs

On top of their many practical and modeling advantages, scalar-valued RKHSs enjoy favorable properties (Cucker and Smale, 2001). This allowed to back many estimators in RKHSs with strong theoretical guarantees–see *e.g.* Bousquet and Elisseeff (2002);

Steinwart (2005); Bartlett et al. (2005). In the specific case of the square loss, consistency properties of regularized estimators in RKHSs can be proven through a specific technique. A sample-free closed-form solution involving the integral operator associated to the kernel (Definition 2.11) can be exhibited. Then, one can control the deviation between this ideal solution and its empirical approximation (Smale and Zhou, 2007). This technique is of particular interest to us since it has been developed concomitantly for vv-RKHSs (Caponnetto and De Vito, 2007). It constitutes the base of the theoretical results that we derive for approximate FOR later on (Chapter 6). We give here an introduction to the main concepts and key results that make this technique work.

We consider the supervised learning setting with the output random variable Y taking its values in the Hilbert space $\mathcal{Y}$. Let $\mathbf{z} = (x_i, y_i)_{i=1}^n = (\mathbf{x}, \mathbf{y}) \in (\mathcal{X}, \mathcal{Y})^n$ be the sample and let ρ be (X, Y)'s unknown probability measure.

We use the square loss on $\mathcal{Y}$ and recall the definition of the associated expected risk of a regressor $f \in \mathcal{F}(\mathcal{X}, \mathcal{Y})$

$$\mathcal{R}(f) := \mathbb{E}_{(\mathsf{X},\mathsf{Y})\sim\rho}\left[\|\mathsf{Y} - f(\mathsf{X})\|_{\mathcal{Y}}^2\right]. \tag{2.24}$$

The minimizer of this expected risk over the space of all measurable $\mathcal{Y}$-valued functions is the regression function:

$$f_\rho(x) = \int_{\mathcal{Y}} y \mathrm{d}\rho(y|x).$$

Ideally, we want to find a regressor which expected risk is close to that of f_ρ. Nevertheless since we cannot optimize over the space of measurable functions, we look for a regressor in a much smaller hypothesis class $\mathcal{G} \subset \mathcal{F}(\mathcal{X}, \mathcal{Y})$. Consequently, approaching f_ρ may be too ambitious. We can however compare a regressor to the best one in the hypothesis class. In other words, we look for $f_0 \in \mathcal{G}$ such that $\mathcal{R}(f_0)$ is as close as possible to $\inf_{f\in\mathcal{G}} \mathcal{R}(f)$.

The use of a finite sample $\mathbf{z}$ instead of the true measure ρ is another unavoidable source of error. The related quantity of interest is the empirical risk

$$\hat{\mathcal{R}}(f, \mathbf{z}) := \frac{1}{n}\sum_{i=1}^n \|y_i - f(x_i)\|_{\mathcal{Y}}^2. \tag{2.25}$$

Remark 2.38. *To ensure that the expected risk–Equation (2.24)–makes sense, the integral* $\int_{\mathcal{Y}} \|y - f(x)\|_{\mathcal{Y}}^2 \mathrm{d}\rho(x, y)$ *must be finite. To that end, we assume for now that* $((x, y) \mapsto f(x)) \in \mathsf{L}^2(\mathcal{Z}, \rho, \mathcal{Y})$ *and* $((x, y) \mapsto y) \in \mathsf{L}^2(\mathcal{Z}, \rho, \mathcal{Y}))$. *We will make assumptions to ensure this is the case later.*

Note that we have introduced $\mathsf{L}^2(\mathcal{Z}, \rho, \mathcal{Y})$ the space of functions from $\mathcal{Z}$ to $\mathcal{Y}$ which are square integrable with respect to the measure ρ. This space is endowed with the following scalar product

$$\langle \psi_1, \psi_2 \rangle_\rho = \int_{\mathcal{Z}} \langle \psi_1(x, y), \psi_2(x, y) \rangle_{\mathcal{Y}} \, \mathrm{d}\rho(x, y),$$

and the associated norm $\|\cdot\|_\rho$.

Now, consider a general hypothesis class $\mathcal{G} \subset \mathcal{F}(\mathcal{X}, \mathcal{Y})$. We define the canonical inclusion operator of $\mathcal{G}$ into the space $\mathsf{L}^2(\mathcal{Z}, \rho, \mathcal{Y})$ as:

$$\mathrm{A} : f \in \mathcal{G} \mapsto \mathrm{A}f \quad \text{with} \quad \mathrm{A}f : (x, y) \mapsto f(x). \tag{2.26}$$

Then the expected risk is trivially reformulated as

$$\mathcal{R}(f) = \|\mathrm{A}f - Y\|_\rho^2, \tag{2.27}$$

where we have defined the dummy function $Y \in \mathsf{L}^2(\mathcal{Z}, \rho, \mathcal{Y})$ as $Y : (x, y) \mapsto y \in \mathcal{Y}$.

Let $\mathcal{H}_\mathsf{K}$ to a vv-RKHS associated to some OVK K. We now take $\mathcal{G} = \mathcal{H}_\mathsf{K}$. This assumption becomes necessary now because we need the reproducing property to derive the adjoint of A and then obtain a characterization of minimizers of the expected risk in terms of $\mathrm{A}^\#$ and $\mathrm{T} = \mathrm{A}^\#\mathrm{A}$. To make sure that these operators are properly defined and bounded, and that the expected risk is always finite, we need to make further assumptions on the OVK. The assumptions that (i) K is measurable in some sense and that (ii) $x \mapsto \mathrm{Trace}(\mathsf{K}_x^\# \mathsf{K}_x)$ is bounded are for instance sufficient (see Hypothesis 1 in Caponnetto and De Vito 2007).

Remark 2.39. *In practice the assumption that K_x is trace-class for all $x \in \mathcal{X}$ is quite restrictive when $\mathcal{Y}$ is infinite-dimensional. For instance the very simple separable kernel $x_1, x_2 \mapsto k(x_1, x_2) I_\mathcal{Y}$ with k some scalar-valued kernel does not qualify.*

Under those assumptions on the OVK and on the measure, we have the following expressions for the operators (Caponnetto and De Vito, 2007).

Lemma 2.40. *For $\psi \in \mathsf{L}^2(\mathcal{Z}, \rho, \mathcal{Y})$, the adjoint of A acts on ψ as*

$$\mathrm{A}^\# \psi = \int_\mathcal{Z} \mathsf{K}_x \psi(x, y) \mathrm{d}\rho(x, y), \tag{2.28}$$

with the integral converging in $\mathcal{H}_\mathsf{K}$, $\mathrm{A}^\#\mathrm{A}$ is the Hilbert-Schmidt operator on $\mathcal{H}_\mathsf{K}$ given by

$$\mathrm{T} = \mathrm{A}^\#\mathrm{A} = \int_\mathcal{Z} \mathrm{T}_x \mathrm{d}\rho_\mathsf{X}(x), \tag{2.29}$$

with the integral converging in $\mathcal{L}_2(\mathcal{H}_\mathsf{K})$, the set of Hilbert-Schmidt operators from $\mathcal{H}_\mathsf{K}$ to $\mathcal{H}_\mathsf{K}$.

To address the concern that the expected risk must be finite (Remark 2.38) we make the following simple assumption on the measure ρ

$$\int_\mathcal{Z} \|y\|_\mathcal{Y}^2 \mathrm{d}\rho(x, y) < +\infty.$$

Additionally, provided the OVK is bounded (either considering $x \mapsto \mathrm{Trace}(\mathsf{K}_x^\# \mathsf{K}_x)$, or $x \mapsto \|\mathsf{K}(x, x)\|_{\mathcal{L}(\mathcal{Y})}$), we have that

$$\int_\mathcal{Z} \|h(x)\|_\mathcal{Y}^2 \mathrm{d}\rho(x, y) < +\infty.$$

Therefore the expected risk is finite.

We now assume that there exists $h_{\mathcal{H}_\mathsf{K}} \in \mathcal{H}_\mathsf{K}$ such that

$$h_{\mathcal{H}_\mathsf{K}} = \inf_{h \in \mathcal{H}_\mathsf{K}} \mathcal{R}(h). \tag{2.30}$$

Since the function $h \mapsto \|\mathrm{A}h - Y\|_\rho^2$ is convex and differentiable, any minimizer must cancel the gradient which implies that

$$\mathrm{T}h_{\mathcal{H}_\mathsf{K}} = \mathrm{A}^\# Y. \tag{2.31}$$

The quantity that we ultimately want to control is the excess risk of a regressor h over the hypothesis class $\mathcal{H}_\mathsf{K}$:

$$\mathcal{E}(h, \mathcal{H}_\mathsf{K}) := \mathcal{R}(h) - \mathcal{R}(h_{\mathcal{H}_\mathsf{K}}).$$

The characterization of $h \in \mathcal{H}_\mathsf{K}$ in Equation (2.31) combined with a polar decomposition of A allows for reformulating this excess risk for any $h \in \mathcal{H}_\mathsf{K}$ as a distance in $\mathcal{H}_\mathsf{K}$ taken through the operator T.

Lemma 2.41. *For any $h \in \mathcal{H}_\mathsf{K}$,*

$$\mathcal{E}(h, \mathcal{H}_\mathsf{K}) = \|\sqrt{\mathrm{T}}(h - h_{\mathcal{H}_\mathsf{K}})\|_{\mathcal{H}_\mathsf{K}}. \tag{2.32}$$

Then, if we consider the Hilbert-valued KRR estimator solution to the empirical risk minimization problem

$$\min_{h \in \mathcal{H}_\mathsf{K}} \frac{1}{n} \sum_{i=1}^{n} \|h(x_i) - y_i\|_\mathcal{Y}^2 + \lambda \|h\|_{\mathcal{H}_\mathsf{K}}^2, \tag{2.33}$$

it can be expressed using empirical estimates $\mathrm{A}_\mathbf{x}$ and $\mathrm{T}_\mathbf{x}$ of the operators A and T as

$$h_\mathbf{z}^\lambda = (\mathrm{T}_\mathbf{x} + \lambda \mathrm{I})^{-1} \mathrm{A}_\mathbf{x}^\# \mathbf{y}. \tag{2.34}$$

We can inject Equation (2.34) into Equation (2.32) and exploit the characterization of $h_{\mathcal{H}_\mathsf{K}}$ in Equation (2.31). The resulting quantity can them be judiciously divided in several terms–see *e.g.* Caponnetto and De Vito (2007) or Baldassarre et al. (2012) for possible strategies. Then these terms can be bounded using several concentration inequalities in separable Hilbert spaces (Pinelis and Sakhanenko, 1986; Yurinsky, 1995) combined with other non-probabilistic inequalities. A finite sample bounds on the excess risk can then be obtained. The concentration inequalities are applied in two Hilbert spaces: the vv-RKHS $\mathcal{H}_\mathsf{K}$ and the space $\mathcal{L}_2(\mathcal{H}_\mathsf{K})$ of Hilbert-Schmidt operators from $\mathcal{H}_\mathsf{K}$ to $\mathcal{H}_\mathsf{K}$. Consequently, the separability of $\mathcal{H}_\mathsf{K}$ is crucial (it implies the separability of $\mathcal{L}_2(\mathcal{H}_\mathsf{K})$). To make sure we have it, either K can be supposed to be such that $\mathcal{H}_\mathsf{K}$ is separable as in (Caponnetto and De Vito, 2007), or further hypotheses implying the separability of $\mathcal{H}_\mathsf{K}$ can be made on the kernel and the input space. For instance if $\mathcal{X}$ is separable and K is continuous, $\mathcal{H}_\mathsf{K}$ is separable. Then provided the successive inequalities used are tight enough, the obtained bound can imply consistency of the regularized ridge estimator from Equation (2.34) for a certain sequence of regularization parameters (λ_n).

This proof technique works for the square loss as it exploits the kernel ridge estimator's closed-form. The next section is dedicated to basic tools for non-differentiable convex optimization. They are useful in practice in this thesis to extend the scope of possible losses for empirical risk minimization problems in vv-RKHSs through dualization.

2.3 Convex Optimization

As we have exemplified through supervised learning in RKHSs and vv-RKHSs, many machine learning procedures boil down to the resolution of an optimization problem. The most crucial distinction is between convex and non-convex problems. In the former case, any local minimum is also a global one, whereas in the latter case, many local minima may exist and may or may not be global. Deep neural networks are a very well known example of such non-convex problems. In this thesis, the focus on convex losses and vv-RKHSs as hypothesis class ensures the problems we deal with are convex. Convex problems have been the main focus of optimization theory, and we refer the reader to Rockafellar (1970); Boyd et al. (2004); Bauschke and Combettes (2017) for an overview. Thanks to a regularization term, our problems are even strongly convex. This usually makes for faster convergence and additionally, if a minimizer exists, it is unique. The challenges we face come from the space we optimize over: an infinite-dimensional Hilbert space. To tackle the induced challenges, we rely on the framework of (Bauschke and Combettes, 2017) adapted to optimization over Hilbert space of possibly non-differentiable functions. That latter possibility is useful in Chapter 7 and Chapter 8 where we deal with non-differentiable objectives. We denote by $\mathcal{H}$ the generic Hilbert space over which we wish to optimize our objective.

In this framework, a central class of functions is that of proper, convex and lower semi-continuous ones.

Definition 2.42 (Proper, convex, lower semi-continuous functions)**.** *Let $\Gamma_0(\mathcal{H})$ denote the set of functions f from $\mathcal{H}$ to $]-\infty,+\infty]$ that are*

- *proper: $dom(f) := \left\{u \in \mathcal{H} \;:\; f(u) < +\infty\right\} \neq \emptyset$,*
- *convex: for all $u, v \in \mathcal{H}$, for all $t \in [0,1]$, $\quad f(tu + (1-t)v) \leq tf(u) + (1-t)f(v)$, and*
- *lower semi-continuous: for all $u \in \mathcal{H}$, $\quad \underline{\lim}_{u \to v} f(v) \geq f(u)$ where $\underline{\lim}$ denotes limit inferior.*

By construction of the Lagrange multipliers, the Fenchel-Legendre conjugate of a function appears frequently when deriving the dual of an optimization problem. For instance when dualizing the empirical risk minimization problem in vv-RKHSs, the Fenchel-Legendre conjugate of the loss appear in Problem 2.23. We now define this transformation properly.

Definition 2.43 (Fenchel-Legendre conjugate)**.** *The Fenchel-Legendre conjugate of a function $f : \mathcal{H} \to [+\infty, -\infty]$ is defined as*

$$\forall u \in \mathcal{H}, \quad f^\star(u) := \sup_{v \in \mathcal{H}} \langle u, v \rangle_{\mathcal{H}} - f(v). \tag{2.35}$$

This transform has several properties. We highlight the following two basic yet important ones: first the Fenchel-Legendre conjugate of a function is always convex and second, it is involutive. This is the object of the following property.

Proposition 2.44. *Let $f \in \Gamma_0(\mathcal{H})$, then*

$$(f^\star)^\star = f.$$

The Fenchel-Legendre transform of a norm $\|\cdot\|$ on $\mathcal{H}$ is of particular interest later on. A key quantity to derive is its associated *dual norm*.

Definition 2.45 (Dual norm)**.** *Let $\|\cdot\|$ be a norm on the Hilbert space $\mathcal{H}$. Its dual norm $\|\cdot\|^*$ is the norm on $\mathcal{H}$ defined for $v \in \mathcal{H}$ as*

$$\|v\|^* = \sup_{\|u\| \leq 1} \langle u, v \rangle_{\mathcal{H}}.$$

Remark 2.46 (Notation)**.** *We draw the reader's attention to the difference between the Fenchel-Legendre transform of a function f which is denoted by $f^\star$ and the dual norm of a norm $\|\cdot\|$ which we denote by $\|\cdot\|^*$.*

Example 2.47 (Fenchel-Legendre transform of $\|\cdot\|$)**.** *Let $\|\cdot\|$ be a norm on the Hilbert space $\mathcal{H}$ and let $\|\cdot\|^*$ be its associated dual norm. Then the Fenchel-Legendre transform of $\|\cdot\|$ is given by*

$$(\|\cdot\|)^\star = \chi_{\{\mathcal{B}^1_{\|\cdot\|^*}\}}, \tag{2.36}$$

where $\mathcal{B}^1_{\|\cdot\|^*}$ denotes the ball of radius 1 associated to the dual norm $\|\cdot\|^*$ and $\chi_{\{\mathcal{B}^1_{\|\cdot\|^*}\}}(\cdot)$ denotes its indicator function equal to 0 on the ball and $+\infty$ elsewhere. We refer to Boyd et al. (2004, Example 3.26) for a proof (very simple). They do it for $\mathcal{H} = \mathbb{R}^n$ but it remains valid for $\mathcal{H}$ a Hilbert space.

Another important property of the Fenchel-Legendre transform for this thesis is how it acts on the infimal convolution between two functions (Bauschke and Combettes, 2017).

Definition 2.48 (Infimal convolution)**.** *The infimal convolution of two functions $f, g : \mathcal{H} \to]-\infty, +\infty]$ is defined as*

$$f \square g : \begin{pmatrix} \mathcal{H} & \to & [-\infty, +\infty] \\ u & \mapsto & \inf_{v \in \mathcal{H}} f(u - v) + g(v) \end{pmatrix}.$$

Example 2.49 (Moreau envelope)**.** *This operator can be useful when optimizing non-differentiable convex functions. Through the infimal convolution between a function and a small amount of the function $u \mapsto \|u\|^2_{\mathcal{H}}$, a smooth approximation can be obtained. The corresponding approximation is called the Moreau envelope and is defined for a function f and a smoothing parameter $\gamma > 0$ as ${}^{\gamma}f := f \square \left(\frac{1}{2\gamma} \|\cdot\|^2_{\mathcal{H}} \right)$. It is convex, its domain is the whole space $\mathcal{H}$ and it is Fréchet differentiable (Bauschke and Combettes, 2017, Proposition 12.30), Fréchet differentiability being a stronger notion which implies Gâteau differentiability which we introduce later in Definition 2.55.*

Proposition 2.50 (Bauschke and Combettes 2017, Proposition 13.24)**.** *Let $f, g : \mathcal{H} \to]-\infty, +\infty]$ be two functions, then*

$$(f \square g)^\star = f^\star + g^\star. \tag{2.37}$$

Another key tool in non-differentiable optimization is the proximal operator. It can be seen as a proxy for a gradient descent step.

Definition 2.51 (Proximal Operator, Moreau 1965)**.** *The proximal operator is defined as*

$$\forall (f, u) \in \Gamma_0(\mathcal{H}) \times \mathcal{H}, \quad \operatorname{prox}_f(u) := \arg\min_{v \in \mathcal{H}} f(v) + \frac{1}{2}\|u - v\|^2_{\mathcal{H}}. \tag{2.38}$$

The functions considered are restricted to $\Gamma_0(\mathcal{H})$ because this ensures that the proximal operator is well-defined: the minimizer in Equation (2.38) exists and is unique.

Remark 2.52 (Interpretation)**.** *This operator plays a central role in many algorithms for optimizing convex yet non-differentiable functions. It can be interpreted in several ways as a modified gradient step (Parikh and Boyd, 2014, Section 3.3). A possible intuition comes from the fact that*

$$\operatorname{prox}_{\gamma f}(u) = u - \gamma \nabla({}^{\gamma} f)(u).$$

Therefore the proximal operator corresponds to a gradient descent step for minimizing the Moreau envelope ${}^{\gamma} f$ of f with step size γ, in other words, it can be seen as a gradient step for a smooth approximation of f.

Another interesting property gives an explicit link between the proximal operator of a function and that of its Fenchel-Legendre conjugate. This for instance particularly helpful when employing algorithms involving the proximal operator in the dual.

Lemma 2.53 (Moreau decomposition, Moreau 1965)**.** *Let $f \in \Gamma_0(\mathcal{H})$ and $\gamma > 0$. Then*

$$I_{\mathcal{H}} = \operatorname{prox}_{\gamma f}(\cdot) + \gamma \operatorname{prox}_{\frac{1}{\gamma} f^\star}(\cdot/\gamma),$$

where $I_{\mathcal{H}}$ denotes the identity operator on $\mathcal{H}$.

This decomposition can be used to compute the proximal operator associated to a norm $\|\cdot\|$.

Example 2.54 (Proximal operator of a norm $\|\cdot\|$)**.** *For a norm $\|\cdot\|$ on $\mathcal{H}$ and $\gamma > 0$, directly applying the Moreau decomposition combined with Example 2.47 and using the invariance of the indicator function by multiplication with a strictly positive scalar, we get*

$$I_{\mathcal{H}} = \operatorname{prox}_{\gamma\|\cdot\|} + \gamma \operatorname{prox}_{\chi_{\{\mathcal{B}^1_{\|\cdot\|^*}\}}}(\cdot/\gamma).$$

Consequently,

$$\operatorname{prox}_{\gamma\|\cdot\|} = I_{\mathcal{H}} - \gamma \operatorname{prox}_{\chi_{\{\mathcal{B}^\gamma_{\|\cdot\|^*}\}}} = I_{\mathcal{H}} - \gamma \operatorname{Proj}_{\mathcal{B}^\gamma_{\|\cdot\|^*}}, \tag{2.39}$$

where $\operatorname{Proj}_{\mathcal{B}^\gamma_{\|\cdot\|^*}}$ denotes the orthogonal projection on $\mathcal{B}^\gamma_{\|\cdot\|^*}$. Indeed

$$\begin{aligned}
\operatorname{prox}_{\chi_{\{\mathcal{B}^\gamma_{\|\cdot\|^*}\}}}(u) &= \operatorname*{arg\,min}_{v\in\mathcal{H}} \chi_{\{\mathcal{B}^\gamma_{\|\cdot\|^*}\}} + \frac{1}{2}\|u - v\|^2_{\mathcal{H}} \\
&= \operatorname*{arg\,min}_{v\in\mathcal{H}} \chi_{\{\mathcal{B}^\gamma_{\|\cdot\|^*}\}} + \|u - v\|^2_{\mathcal{H}} \\
&= \operatorname{Proj}_{\mathcal{B}^\gamma_{\|\cdot\|^*}}(u).
\end{aligned}$$

Proximal operator are usually used to solve optimization problems involving the sum of a convex and differentiable function f, and a convex and non-differentiable one g. Such a problems are called *composite problems* and take the form

$$\inf_{u\in\mathcal{H}} f(u) + g(u). \tag{2.40}$$

We now precise what is meant by differentiable since when in infinite-dimensional vector spaces, several notions exist. Fréchet differentiability and Gâteau differentiability are two such notions, the former being stronger than the latter. We define next *Gâteau differentiability*, as it is the one we use mostly.

Algorithm 2.1 PROXIMAL GRADIENT DESCENT FOR PROBLEM 2.40

input : Stepsize $\gamma > 0$, number of iterations T
init : $u^{(0)}$
for $t = 1, \dots, T$ **do**
  $u^{(t)} = \operatorname{prox}_{\gamma g}\left(u^{(t-1)} - \gamma \nabla f(u^{(t-1)})\right)$
return $u^{(T)}$

Definition 2.55 (Gâteaux differentiability)**.** *Consider a function $f : \mathcal{H} \to]-\infty, +\infty]$ that is proper and $u \in dom(f)$. For a direction $v \in \mathcal{H}$, the directional derivative of f is*

$$f'(u, v) = \lim_{\alpha \downarrow 0} \frac{f(u + \alpha v) - f(u)}{\alpha},$$

provided that the limit exists. When $f'(u, \cdot)$ is linear in v and continuous, f is said to be Gâteaux differentiable at point u and there exist a unique vector $\nabla f(u) \in \mathcal{H}$ such that

$$\forall v \in \mathcal{H},\ f'(u, v) = \langle \nabla f(u), v \rangle_{\mathcal{H}}.$$

The proximal gradient algorithm presented in Algorithm 2.1 can be used to solve Problem 2.40. Let us assume that $f, g \in \Gamma_0(\mathcal{H})$, that additionally f is Gâteaux differentiable with a $1/\beta$-Lipschitz continuous gradient and that $\gamma \in]0, 2\beta[$. Then if $\operatorname{Argmin}(f + g)$–the set of minimizers of $f+g$–is not empty, the sequence $(u^{(t)})_{t \in \mathbb{N}}$ from Algorithm 2.1 converges weakly to a point in $\operatorname{Argmin}(f + g)$. Moreover, under uniform convexity assumptions on f and g detailed in Bauschke and Combettes (2017, Corollary 28.9), this convergence becomes strong.

We now introduce a more specific form of problems encompassing those we study in Chapter 7. For this form of problems, we also introduce the main concepts of duality. Let $\mathcal{K}$ be a Hilbert space, let $f \in \Gamma_0(\mathcal{H})$ be the function to optimize and let $\mathrm{A} \in \mathcal{L}(\mathcal{H}, \mathcal{K})$ be a linear operator. We consider the problem:

$$\inf_{u \in \mathcal{H}} f(u) \quad \text{subject to} \quad \mathrm{A}u = b. \tag{2.41}$$

This can be rewritten alternatively using an indicator function for the set of constraints $\chi_{\{b\}}(\mathrm{A}u)$ which equals 0 if $\mathrm{A}u - b = 0$ and $+\infty$ otherwise.

$$\inf_{u \in \mathcal{H}} \underbrace{f(u) + \chi_{\{b\}}(\mathrm{A}u)}_{:= \mathcal{P}(u)} \tag{2.42}$$

Example 2.56. *In the context of regularized empirical risk minimization in Problem 2.18, if the loss writes as $L(h(x), y) = L_0(h(x) - y)$, to dualize Problem 2.18 so as to obtain Theorem 2.37, the additional variable $w = (w_i)_{i=1}^n \in \mathcal{Y}^n$ is introduced along with a linear constraint so that the objective remains the same. More precisely, in the notations of Problem 2.41:*

- *$\mathcal{K} = \mathcal{Y}^n$ and $\mathcal{H} = \mathcal{Y}^n \times \mathcal{H}_{\mathsf{K}}$. Then, we set $u = (w, h) \in \mathcal{Y}^n \times \mathcal{H}_{\mathsf{K}}$,*
- *define the linear operator $\mathrm{A} : \mathcal{H} \to \mathcal{Y}^n$ as $\mathrm{A} : u \mapsto (h(x_i) - w_i)_{i=1}^n \in \mathcal{Y}^n$, take $b = (y_i)_{i=1}^n \in \mathcal{Y}^n$,*

- *and we can reformulate the objective as* $f : u \mapsto \frac{1}{n}\sum_{i=1}^{n} L_0(w_i) + \frac{\lambda}{2}\|h\|^2_{\mathcal{H}_K}$.

Definition 2.57 (Lagrangian)**.** *The Lagrangian associated to Problem 2.41 is the following function*

$$\mathfrak{L}\colon \begin{pmatrix} \mathcal{H}\times\mathcal{K} & \to &]-\infty,+\infty] \\ (u,\alpha) & \mapsto & f(u)+\langle \mathrm{A}u-b,\alpha\rangle_{\mathcal{K}} \end{pmatrix}.$$

Minimizing the Lagrangian over the primal variables, we get the dual objective $\mathcal{D}(\alpha)$. The dual problem associated to Problem 2.41 is that of maximizing the dual objective:

$$\sup_{\alpha\in\mathcal{K}} \underbrace{\inf_{u\in\mathcal{H}} \mathfrak{L}(u,\alpha)}_{:=\mathcal{D}(\alpha)}. \tag{2.43}$$

Let $\mathcal{P}^*$ and $\mathcal{D}^*$ be the optimal values respectively for the primal and the dual problems. Since $f \in \Gamma_0(\mathcal{H})$, by Definition 2.42, its domain (set of points where it is not infinite) is non-empty. Consider then a point $u_0 \in \mathrm{dom}(f)$, by definition we have $\mathrm{A}u_0 - b = 0$, therefore, $\mathfrak{L}(u_0,\alpha) = \mathcal{P}(u_0)$, taking the infimum over $u_0 \in \mathrm{dom}(f)$, we get that for all $\alpha \in \mathcal{K}$, $\mathcal{D}(\alpha) \leq \mathcal{P}^*$. Consequently, taking the supremum in α over $\mathcal{K}$, we get the following inequality known as *weak duality*.

$$\mathcal{D}^* \leq \mathcal{P}^*. \tag{2.44}$$

When we do have the equality $\mathcal{D}^* = \mathcal{P}^*$, we say that *strong duality* holds. For optimization over infinite-dimensional space, a sufficient constraint qualification condition to ensuring strong duality holds is stated in Gowda and Teboulle (1990, Theorem 2) for Banach spaces. This condition stems from the refinement proposed in Robinson (1976, Corollary 1) of the initial proposition given in Rockafellar (1974, Theorem 18). To formulate this condition, we need to define the core of a set.

Definition 2.58 (Core)**.** *Let $\mathcal{H}$ be a Hilbert space and let $C \subset \mathcal{K}$ be a subset. The core of C is defined as*

$$\mathrm{core}(C) := \Big\{ u \in C \quad : \quad \forall v \in \mathcal{H},\ \exists \epsilon > 0 : \forall \tau \in [-\epsilon,\epsilon],\ \ u + \tau v \in C \Big\}.$$

Our Problem 2.42 is a particular case of the more general problem for which this condition is formulated in Gowda and Teboulle (1990). In our problem, the second function g in their objective is simply $\chi_{\{b\}}(\cdot)$, and the spaces $\mathcal{H}$ and $\mathcal{K}$ are Hilbert spaces. Stating their Theorem 2 for our problem yields the following constraint qualification condition.

Corollary 2.59. *Let $\mathcal{H}$ and $\mathcal{K}$ be Hilbert spaces, if*

$$0 \in \mathrm{core}\Big(\{b\} - \mathrm{Im}(\mathrm{A})\Big),$$

then strong duality holds for Problem 2.42,

where we have defined

$$\{b\} - \mathrm{Im}(\mathrm{A}) := \{w \in \mathcal{K},\ \exists u \in \mathcal{H} \quad : \quad w = b - \mathrm{A}u\}.$$

Example 2.60. *Continuing Example 2.56, we see that for empirical risk minimization in vv-RKHSs, the constraint qualification from Corollary 2.59 is trivially verified. Indeed, for all $v \in \mathcal{H}$, for any $\epsilon > 0$, we can choose any $h \in \mathcal{H}_K$ and for $\tau \in [-\epsilon, \epsilon]$ set $w = \tau v + (h(x_i))_{i=1}^n - b$, so that setting $u = (h, w)$, $\tau v = b - Au$. Therefore, $\tau v \in \Big(\{b\} - \mathrm{Im}(A)\Big)$ which implies $0 \in \mathrm{core}\Big(\{b\} - \mathrm{Im}(A)\Big)$.*

2.4 Conclusion

In this chapter, we introduced many concepts and tools on which we will rely in the following chapters. The first focus was on kernel methods with a strong focus on machine learning applications. We provided details on RKHSs and their properties and then we introduced vv-RKHSs, an extension of RKHSs to model vector-valued functions. We use as these spaces extensively as a hypothesis class later on. We also gave a brief overview of how integral operators can be used to obtain excess risk bounds for vector-valued kernel ridge estimators. This scheme of proof is central in the derivation of excess risk bounds for some estimators we propose in this thesis. The second focus was on convex optimization. More precisely, we provided some key concepts to optimize objectives which are convex yet non-differentiable based on proximal algorithms and duality. We encounter such problems in upcoming chapters and we rely on these tools to transform these problems and propose algorithms to solve them.

3

Functional data and representation of functions

Contents

In this thesis, we are interested in the problem of predicting functions that lie in a Hilbert space. It may be infinite-dimensional, yet the functions that we actually observe generally exhibit some type of regularity. Therefore, a natural idea is to suppose that they actually lie in a lower dimensional subspace of the original Hilbert space. Supposing we are able to find such relevant subspace, the benefits are twofold. The cost of representation and prediction can be lowered significantly (*compression*) while the relevance of the representation can be more adapted to the problem at hand and help us filter out observational noise (*regularization*).

In this chapter, we forget the input side of the problem to focus only on the outputs $(y_i)_{i=1}^n$. Let us consider the general supervised learning setting with output data lying in a given Hilbert space $\mathcal{Y}$ and input data lying in a space $\mathcal{X}$: we observe an i.i.d. sample $(x_i, y_i)_{i=1}^n$ from a couple of random variables (X, Y) taking its values in $\mathcal{X} \times \mathcal{Y}$. Then the assumption that it is possible to approximate efficiently the output vectors generated by this distribution boils down to the existence of a subspace $\tilde{\mathcal{Y}} \subset \mathcal{Y}$ such that $\mathbb{P}[\mathsf{Y} \in \tilde{\mathcal{Y}}]$ is close to 1, or better, that $\mathsf{Y} \in \tilde{\mathcal{Y}}$ almost surely. Since we do not have access to Y's true distribution ρ_{Y}, we must rely on the sample $(y_i)_{i=1}^n$ we do have access to. Therefore, it is logical to search for a low dimensional space $\tilde{\mathcal{Y}}$ which approximates well the vectors in this sample. A possible way to do so is to look for a dictionary $\boldsymbol{\phi} := (\phi_l)_{l=1}^d \in \mathcal{Y}^d$ such that the $(y_i)_{i=1}^n$ can be represented with small error as a linear combination of elements from $\boldsymbol{\phi}$, or equivalently consider $\tilde{\mathcal{Y}} = \operatorname{Span}\left\{(\phi_l)_{l=1}^d\right\}$.

In Section 3.1, we discuss functional data, their discrete form and the challenges related to them. We also expose some tools and procedures to represent discrete observations as functions. While in Section 3.2 we explore what types of functional spaces can be used for such representation. We also investigate how the choice of such space has a regularizing effect and can express our belief on the smoothness of the functions to represent.

3.1 Functional data and smoothing

This thesis focuses on functional data. Therefore, we set $\mathcal{Y} = \mathrm{L}^2(\Theta)$, the separable Hilbert space of square integrable functions on a compact set $\Theta \subset \mathbb{R}^b$ with respect to the Lebesgue measure for some $b \geq 1$. However, we never observe actual functions but rather noisy evaluations of these. To account for this, our observations have the form $(\boldsymbol{\theta}_i, \tilde{y}_i)_{i=1}^n$; where $\boldsymbol{\theta}_i = (\theta_{is})_{s=1}^{m_i}$ are the locations at which we observe the function y_i. We consider they are generated by the observational model

$$\tilde{y}_{is} = y_i(\theta_{is}) + \epsilon_{is}, \tag{3.1}$$

where the quantities $\left((\epsilon_{is})_{s=1}^{m_i}\right)_{i=1}^n$ correspond to noise components added to the observations. We suppose that all the noises are i.i.d. (both across samples and locations). In this functional setting, we have two additional motivations beyond compression for using an appropriate representation. When the location at which we observe the functions vary, we can no longer use conventional discrete signal representation methods and by choosing an appropriate functional representation, we express a prior that the underlying functions must be smooth in some sense. This has a regularizing effect and can help us to filter out the noise from the discrete evaluations. However, depending on whether the discrete evaluations are sufficient to readily consider the problem as functional or not, a typology must be made.

3.1.1 Dense and sparse functional data

Dense FDA (fully-observed functions). In FDA, the typical assumption is that we are in the so-called *dense FDA* setting (Kokoszka and Reimherr, 2017). This means that the number of observations per function is not an issue: it is supposed to be very high and the evaluation locations are supposed to be scattered in the domain Θ so that we have enough information to make inference on the functions throughout Θ. Generally, in addition the dimension of Θ is very low—most frequently it is equal to 1 and Θ is an interval. Then an usual assumption is that the functions are sampled at a high number of equispaced locations. The locations can also be supposed to consist of a high number of i.i.d. samples from a uniform distribution on Θ.

Sparse FDA (sparsely-observed functions). As opposed to the dense setting, when the number or the diversity of the evaluation locations available per function becomes a problem to treat the problem as functional, we say that we are in the *sparse FDA* setting (Kokoszka and Reimherr, 2017). For instance, the functions can be observed at too few locations and/or these locations may only cover an incomplete part of the domain Θ. Therefore, the notion of sparsely observed functions encompasses several difficulties in one. Consequently, quantifiable assumptions to describe further the different aspects of the notion have been formulated to study the behavior of some functional estimators. Examples include nonparametric estimation of mean and covariance functions (Li and Hsing, 2010; Zhang and Wang, 2016) or the estimation of the mean function with splines (Cai and Yuan, 2011).

Partially-observed functions. In this thesis, we deal mostly with something between the two. This means that we have discrete observations which may not be evaluated at the same locations for all the functions. Yet they cover most of the domain of definition Θ and should be numerous enough to derive a functional representation without

resorting to specific sparse FDA technique. To avoid creating confusion, we therefore use the term *partially-observed functions*.

3.1.2 Smoothing of functional data

Even in the dense setting, observations are necessarily discrete, therefore any procedure dealing with data in a functional way necessarily encompasses a representation step. In FDA, this representation step is generally performed as preprocessing, and is called *smoothing* (Ramsay and Silverman, 2005). An intermediate problem is solved to represent the data in a functional way. To that end, a functional hypothesis space $\mathcal{W} \subset \mathsf{L}^2(\Theta)$ is chosen. It must reflect the properties the underlying functions are likely to exhibit, and be of reasonable dimension (the lower, the better).

A specific regularized empirical risk minimization problem using the available evaluations can for instance be formulated. More precisely, for i in $[\![n]\!]$, the i-th function is smoothed by solving:

$$\min_{w_i \in \mathcal{W}} \sum_{s=1}^{m_i} \ell(w_i(\theta_{is}), \tilde{y}_{is}) + \Omega(w_i). \tag{3.2}$$

Smoothing on a dictionary of functions

In order to provide a description in low dimension, these functional spaces are generally chosen as the span of a given dictionary of functions $\boldsymbol{\phi} = (\phi_l)_{l=1}^d$. In that case, the smoothing problem amounts to finding a set of representation coefficients in $\mathbb{R}^d$. Let us introduce the linear operator:

$$\Phi : \begin{pmatrix} \mathbb{R}^d & \to & \mathsf{L}^2(\Theta) \\ \mathbf{a} & \mapsto & \sum_{l=1}^d \mathbf{a}_l \phi_l \end{pmatrix}. \tag{3.3}$$

The adjoint of this operator in $\mathsf{L}^2(\Theta)$ is given by

$$\Phi^{\#} : \begin{pmatrix} \mathsf{L}^2(\Theta) & \to & \mathbb{R}^d \\ y & \mapsto & (\langle y, \phi_l \rangle_{\mathsf{L}^2(\Theta)})_{l=1}^d \end{pmatrix}. \tag{3.4}$$

When considering a set of locations $\theta_i \in \mathbb{R}^{m_i}$, the empirical approximation $\tilde{\Phi}_i$ of the operator Φ is the following

$$\tilde{\Phi}_i : \begin{pmatrix} \mathbb{R}^d & \to & \mathbb{R}^{m_i} \\ \mathbf{a} & \mapsto & \sum_{l=1}^d \mathbf{a}_l \phi_l(\boldsymbol{\theta}_i) \end{pmatrix}, \tag{3.5}$$

and its adjoint in the Euclidian space $\mathbb{R}^{m_i}$ is given by

$$\tilde{\Phi}_i^{\#} : \begin{pmatrix} \mathbb{R}^{m_i} & \to & \mathbb{R}^d \\ \tilde{y} & \mapsto & (\langle \tilde{y}, \phi_l(\boldsymbol{\theta}_i) \rangle_{\mathbb{R}^{m_i}})_{l=1}^d \end{pmatrix}, \tag{3.6}$$

where we have used the convention that $\phi_l(\boldsymbol{\theta}_i) = (\phi_l(\theta_{is}))_{s=1}^{m_i}$.

A general smoothing problem for the i-th function then takes the form

$$\min_{\mathbf{a}_i \in \mathbb{R}^d} \frac{1}{m_i} \sum_{s=1}^{m_i} \ell(\tilde{\Phi}_i \mathbf{a}_i, \tilde{y}_i) + \Omega(\mathbf{a}_i). \tag{3.7}$$

Smoothing sparsely-observed functions. Fedicated smoothing approaches have been developed for functions that are sparsely-observed. The most well-known ones assume that the sparse evaluations $(\boldsymbol{\theta}_i, \tilde{y}_i)_{i=1}^n$ correspond to several underlying curves drawn from a common distribution ρ_Y. Therefore that information can be exploited to help fill the gaps in a sensible way. In practice, a smoothed covariance function for the underlying functions $(y_i)_{i=1}^n$ is estimated from the discrete evaluations. Then, the eigenfunctions associated to this smooth estimated covariance are used as a basis to represent smoothly the curves (Yao et al., 2005; Xiao et al., 2018); and they could indeed be used as a dictionary in the smoothing problem Problem 3.7 as well. It is also worth noting that this idea has been studied extensively in the context of the functional additive linear model in Petrovich et al. (2018).

Smoothing with square loss and square norm penalty

As a particular instance of Problem 3.7, if we measure the discrepancy through the square loss and add a penalization based on the 2-norm, we obtain the problem

$$\min_{\mathbf{a}_i \in \mathbb{R}^d} \frac{1}{m_i} \|\tilde{\Phi}_i \mathbf{a}_i - \tilde{y}_i\|_{\mathbb{R}^{m_i}}^2 + \frac{\lambda}{m_i} \|\tilde{\Phi}_i \mathbf{a}_i\|_{\mathbb{R}^{m_i}}^2. \tag{3.8}$$

Differentiating the objective in the above with respect to $\mathbf{a}_i$, we get the linear system:

$$\tilde{\Phi}_i^{\#} \tilde{\Phi}_i \mathbf{a}_i = \frac{1}{1-\lambda} \tilde{\Phi}_i^{\#} \tilde{y}_i. \tag{3.9}$$

Therefore, we see that if $(\phi_l)_{l=1}^d$ forms an orthogonal system in $\mathsf{L}^2(\Theta)$ and if we have sufficiently many evaluation locations m_i that are scattered enough so that

$$\frac{1}{m_i} \tilde{\Phi}_i^{\#} \tilde{\Phi}_i \approx \Phi^{\#} \Phi = \mathrm{I}, \tag{3.10}$$

we can approximate the linear system in Equation (3.9) and get a simple expression for a minimizer as

$$\hat{\mathbf{a}}_i = \frac{1}{(1-\lambda) m_i} \tilde{\Phi}_i^{\#} \tilde{y}_i = \frac{1}{(1-\lambda) m_i} \left(\langle \tilde{y}_i, \phi_l(\boldsymbol{\theta}_i) \rangle_{\mathbb{R}^{m_i}} \right)_{l=1}^d. \tag{3.11}$$

This simple calculation shows that provided the evaluation locations $\boldsymbol{\theta}_i \in \mathbb{R}^{m_i}$ allow to correctly estimate the pairwise inner products between the functions $(\phi_l)_{l=1}^d$, the solution to Problem 3.8 can be approximated very efficiently. Indeed, the orthogonal projection of a function $w \in \mathsf{L}^2(\Theta)$ onto $\mathrm{Span}\left\{(\phi_l)_{l=1}^d\right\}$ is given by $\sum_{l=1}^d \langle w, \phi_l \rangle_{\mathsf{L}^2(\Theta)} \phi_l$. Without surprises, Equation (3.11) with $\lambda = 0$ corresponds to this projection estimating the scalar products in $\mathsf{L}^2(\Theta)$.

Smoothing with roughness penalties

As we will see in Section 3.2, a possible hypothesis to traduce our belief on the regularity of the functions is that they have derivatives up to a given order s. To simplify the exposition, we suppose that Θ is an interval in $\mathbb{R}$. Then, without loss of generality, we take $\Theta = [0,1]$. For our purpose the derivative is taken in the weak sense.

Remark 3.1. *The setting $\Theta = [0,1]$ is the typical setting in functional data analysis (FDA), and one must keep in mind any function defined on an interval $[a,b]$ can be rescaled to an equivalent function defined on the interval $[0,1]$.*

Definition 3.2 (Weak derivative)**.** *Let w be a function in $\mathsf{L}^1([0,1])$, w is said to be weak derivable if there exists a function $v \in \mathsf{L}^1([0,1])$ such that for all infinitely differentiable functions φ on $[0,1]$ such that $\varphi(0) = \varphi(1) = 0$,*

$$\int_0^1 w(\theta)\varphi'(\theta)\mathrm{d}\theta = -\int_0^1 v(\theta)\varphi(\theta)\mathrm{d}\theta.$$

The function v is then said to be a weak derivative of w, and in a slight abuse of notation, we refer by w' to any such weak derivative. For and integer $s \geq 1$, higher order weak derivatives $w^{(s)}$ are defined recursively.

Suppose that $\Theta = [0,1]$ and that w admits an order s weak derivative $w^{(s)}$ which is square integrable: $w^{(s)} \in \mathsf{L}^2([0,1])$; then a natural penalty to consider is (Green and Silverman, 1993; Ramsay and Silverman, 2005):

$$\Omega(w) = \lambda \|w^{(s)}\|^2_{\mathsf{L}^2([0,1])} = \lambda \int_0^1 \left(w^{(s)}(\theta)\right)^2 \mathrm{d}\theta, \quad \lambda > 0. \tag{3.12}$$

Example 3.3 (Roughness penalty and splines)**.** *Such roughness penalties are closely linked to spline smoothing. Consider Problem 3.2 with the square loss, the above penalty with derivative of order $s = 2$ and $\mathcal{W}$ the space of functions w for which $w^{(2)} \in \mathsf{L}^2([0,1])$. Then the minimizer is a cubic spline with knots at the locations of observation of the functions to smooth (see* e.g. *de Boor 2001).*

Suppose that the functions in the dictionary $\boldsymbol{\phi}$ all have weak derivatives up to order s and let us denote by $\Phi^{(s)}$ the projection operator (Equation (3.3)) associated to the dictionary of derivatives $(\phi_l^{(s)})_{l=1}^d$. The functions smoothed on this dictionary have the form $\Phi\mathbf{a}$ and then the roughness penalty reads

$$\Omega(\Phi\mathbf{a}) = \lambda \mathbf{a}^{\mathrm{T}} (\Phi^{(s)})^{\#} \Phi^{(s)} \mathbf{a},$$

the matrix $(\Phi^{(s)})^{\#}\Phi^{(s)}$ containing the pairwise inner products in $\mathsf{L}^2([0,1])$ between the order s weak derivatives of the functions $(\phi_l)_{l=1}^d$.

Roughness penalties are a way to express a regularity belief when smoothing. The choice of the dictionary may however remain the most important one, in the sense that as linear combinations of the functions in the dictionary, the smoothed functions inherit its properties.

3.2 Functional spaces, approximation and dictionaries

The object of this section is to introduce several functional spaces that can be used as a hypothesis class for smoothing. We look at how they can be approximated using a dictionary of functions and make links when possible with the smoothness of the functions they contain.

Sobolev spaces

As hinted by roughness penalties, it is intuitive to link regularity to differentiability. A function can be considered to be smooth if it possesses one or more derivatives, the more it possesses, the smoother it is. Therefore if we think a function does possess derivatives up to a certain order, this can help us to estimate it and to filter out noise. In terms of functional spaces, the related notion is that of Sobolev spaces. We rely on the notion of weak derivatives introduced in Definition 3.2 to define such space. To simplify the exposition, we set $\Theta = [0,1]$.

Definition 3.4 (Sobolev space)**.** *Let $s \in \mathbb{N} \geq 1$. The Sobolev space $\mathcal{W}^s([0,1])$ is the subspace of $\mathsf{L}^2([0,1])$ containing functions whose weak derivatives up to order s have a finite $\mathsf{L}^2([0,1])$ norm.*

We relate those spaces to the decay of the approximation error in orthogonal dictionaries in Section 3.2.1 and show some links with RKHSs in Section 3.2.2.

Remark 3.5. *The space $\mathcal{W}^s([0,1])$ endowed with an inner product of the form*

$$\langle v, w \rangle_{\mathcal{W}^s([0,1])} := \sum_{r=1}^{s} a_r \langle v^{(r)}, w^{(r)} \rangle_{\mathsf{L}^2([0,1])}$$

with $(a_r)_{r=1}^s \in (\mathbb{R}_+)^s$ is a Hilbert space. All the corresponding norms being equivalent.

Remark 3.6. *The fact that the weak derivative of order s is square integrable implies that for all $r < s$, the order r derivative is actually continuous.*

Remark 3.7. *Sobolev spaces can indeed be defined for more general Θ. Considering Θ to be an open subset of $\mathbb{R}^b$, the existence of weak derivative up to order s is replaced by the existence of all weak derivatives D^α according to the multi-indices $\alpha \in \mathbb{N}^b$ such that $|\alpha| := \sum_{l=1}^b \alpha_l = s$. More precisely, for such multi-index α, D^α is defined as*

$$\forall \theta = (\theta_1, \cdots, \theta_q) \in \Theta, \quad \mathsf{D}^\alpha w(\theta) = \frac{\partial^{|\alpha|}}{\partial \theta_1^{\alpha_1} \partial \theta_2^{\alpha_2} \cdots \partial \theta_q^{\alpha_q}} w(\theta). \tag{3.13}$$

3.2.1 Usual dictionaries and approximation error

The choice of the dictionary is crucial in making sure our functional representation is meaningful for the problem. If the dictionary is an orthonormal basis, the approximation error can be characterized explicitly in terms of the decay of the representation coefficients. Consider such an orthonormal basis $(\phi_l)_{l \in \mathbb{N}}$ of $\mathsf{L}^2([0,1])$, the linear representation of a function is defined as:

Definition 3.8 (Linear approximation)**.** *The linear approximation of order d of a function $w \in \mathsf{L}^2(\Theta)$ on the orthonormal basis $(\phi_l)_{l\in\mathbb{N}}$ is its orthogonal projection onto* $\mathrm{Span}\left\{(\phi_l)_{l=1}^d\right\}$ *given by*

$$w_{(d)} = \sum_{l=1}^{d} \langle w, \phi_l \rangle_{\mathsf{L}^2(\Theta)} \phi_l.$$

What this approximation leaves behind is therefore the projection on $\mathrm{Span}\left\{(\phi_l)_{l=1}^d\right\}^{\perp}$.

$$w - w_{(d)} = \sum_{l=d+1}^{+\infty} \langle w, \phi_l \rangle_{\mathsf{L}^2(\Theta)} \phi_l.$$

The approximation error is then

$$E_{\boldsymbol{\phi}}(w, d) := \|w - w_{(d)}\|^2_{\mathsf{L}^2(\Theta)}. \tag{3.14}$$

The decay rate of this approximation error can be related to the decay rate of the square of the scalar product between the function and the basis elements (see *e.g.* Theorem 9.1 in Mallat 2008).

Theorem 3.9. *Let $s > \frac{1}{2}$, the approximation error $E_{\boldsymbol{\phi}}(w, d)$ of a function w in the basis $(\phi_l)_{l=1}^{+\infty}$ decays faster than d^{-2s} if w belongs to the space*

$$\mathcal{W}^s_{\boldsymbol{\phi}} := \left\{ w \in \mathsf{L}^2([0,1]) \;\; : \;\; \sum_{l=1}^{+\infty} l^{2s} \langle w, \phi_l \rangle^2_{\mathsf{L}^2(\Theta)} < +\infty \right\}. \tag{3.15}$$

We give a detailed example for Fourier dictionaries for which he space $\mathcal{W}^s_{\boldsymbol{\phi}}$ can be shown to be a Sobolev space under some conditions. This can help us better understand that the representation of a function in a truncated basis can traduce a belief on the function's smoothness.

Fourier dictionaries

For the Fourier basis, the differentiability of a function and the decay rate of its Fourier coefficients can be linked explicitly. This helps us better characterize the approximation error that we make when we use a truncated Fourier basis. We state the results for the coefficients in the Fourier basis in exponential form: $\boldsymbol{\psi} = \left\{\psi_l : \theta \mapsto \mathrm{e}^{i2\pi l\theta}\right\}_{l\in\mathbb{Z}}$. Since the functions we are interested in are real-valued, the coefficients are complex. Let v be a complex-valued function which is square integrable on Θ. We consider the canonical inner product

$$\langle w, v \rangle_{\mathsf{L}^2([0,1])} = \int_0^1 w(\theta)\bar{v}(\theta)\mathrm{d}\theta \in \mathbb{C}.$$

Remark 3.10. *Let $(\phi_l)_{l\in\mathbb{Z}}$ be the Fourier basis in terms of real-valued functions (which is indeed the one used in FDA):*

$$\forall l \in [\![1, +\infty]\!], \quad \phi_l : \theta \mapsto \sqrt{2}\cos(2\pi l\theta), \quad \phi_{-l} : \theta \mapsto \sqrt{2}\sin(2\pi l\theta) \quad \textit{and} \quad \phi_0 : \theta \mapsto 1.$$

There is the following equivalence:

$$w_{(d)} = \sum_{l=-d}^{d} \langle w, \phi_l \rangle_{\mathsf{L}^2([0,1])} \phi_l = \sum_{l=-d}^{d} \langle w, \psi_l \rangle_{\mathsf{L}^2([0,1])} \psi_l$$

The next theorem gives a characterization of Sobolev spaces in terms of decay of the approximation error in the Fourier basis. To avoid boundaries issues the theorem is stated for functions with support strictly included in $[0,1]$ (see *e.g.* Theorem 9.2 in Mallat 2008):

Theorem 3.11. *Let $w \in \mathrm{L}^2([0,1])$ be a function with support strictly included in $[0,1]$. Then $w \in \mathcal{W}^s([0,1])$ if and only if*

$$\sum_{d=1} d^{2s} \frac{E_\psi(w,d)}{d} < \infty. \tag{3.16}$$

This implies that $E_\psi(w,d) = o(d^{-2s})$.

Therefore, for the Fourier basis, choosing to represent a function using a small number of frequencies reflects a prior on weak derivability. In other words, the lowest the number of frequencies we consider, the smoother we believe the function to be. This can have a strong regularizing effect when smoothing discrete functions generated as in Equation (3.1) since the higher frequencies can be left out as noise.

Wavelets dictionaries

Apart from the Fourier basis, wavelets bases are probably the most well known bases for $\mathrm{L}^2(\mathbb{R})$. They stem from the idea of multi-resolution analysis of $\mathrm{L}^2(\mathbb{R})$. A multi-resolution–see *e.g.* Definition 7.1 in Mallat 2008–is a sequence of subspaces $(V_j)_{j\in\mathbb{Z}}$ in which each subspace corresponds to a given resolution level (V_j corresponds to the resolution 2^{-j}). Therefore V_j contains all the subspaces $(V_k)_{k>j}$ corresponding to coarser resolutions. The motivation behind orthogonal wavelet bases is to construct an orthonormal basis $\psi_{(j,n)\in\mathbb{Z}^2}$ of $\mathrm{L}^2(\mathbb{R})$ such that for each resolution level 2^{-j}, $(\psi_{j,n})_{n\in\mathbb{Z}}$ describes the behavior of a signal only at that resolution. This was achieved in Meyer (1985). Formally, since $V_j \subset V_{j-1}$, considering W_j to be the orthogonal complement of V_j in V_{j-1}, $(\psi_{j,n})_{n\in\mathbb{Z}^2}$ is constructed to be an orthonormal basis of W_j. Depending on which *scaling function* is chosen to form an orthogonal basis for the space V_0, several bases can be constructed; we do not detail the construction and refer the reader to Meyer (1993, Section 3.2) or Mallat (2008, Section 7.1). Then, assuming a proper mother wavelet ψ has been obtained, the dilated and translated family

$$\left\{ \psi_{j,n} : \theta \mapsto \frac{1}{\sqrt{2^j}} \psi\left(\frac{\theta - 2^j n}{2^j} \right) \right\}_{(j,n)\in\mathbb{Z}^2},$$

is an orthonormal basis of $\mathrm{L}^2(\mathbb{R})$.

Families of wavelets. Many wavelets bases can be constructed with different properties. For instance, for any s, there exists a wavelet with compact support on $\mathbb{R}$ for which all derivatives up to order s exist and can be constructed (Daubechies, 1996), this gives rise to the well-known family of Daubechies wavelets. We display examples of these in Figure 3.1. However, the corresponding wavelet functions do not have an explicit expression. The Meyer wavelet is another well-known example which is continuously differentiable an infinite number of times (Meyer, 1993) and has an explicit expression in the frequency domain. However it is not compactly supported. For a detailed account of the properties and construction of many well-known wavelets family, we refer to Mehra (2018, Chapter 3). We highlight that different families

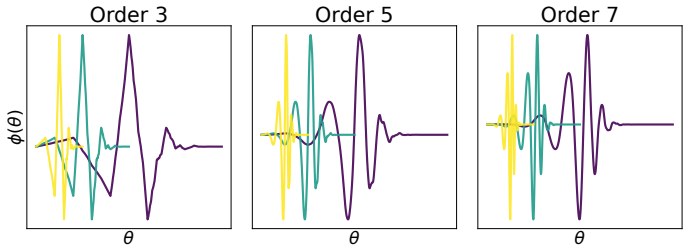


Figure 3.1: Daubechies wavelets with different levels of smoothness

have different characteristics which are desirable, however there is a trade-off between these characteristics. Consequently, each family is adapted to a given application. The main properties are the following:

- *Compact support*: is the mother wavelet compactly supported ?
- *Smoothness*. There is a trade-off between smoothness and compact support, basically the smoother the wavelet, the larger its support.
- *Symmetry*. Is the mother wavelet symmetric ?
- *Orthogonality*. Orthogonality is exploited for faster representation of signals. We only talked about orthogonal wavelets above, but in order to achieve other properties, wavelets which are not orthogonal can be constructed. For instance, biorthogonal wavelets offer the possibility of a symmetric mother wavelet which cannot be achieved for orthogonal wavelets. Also, letting go of orthogonality constraint can enable one to find a closed-form or simplify evaluation, for instance Cohen et al. (1992) propose the family of biorthogonal spline wavelets whose scaling and wavelet function are splines.
- *Vanishing moments*. A wavelet ψ has q vanishing moments if it is orthogonal to any polynomial of degree $q-1$. For approximating $w \in \mathrm{L}^2(\mathbb{R})$, this implies that in practice if w is regular and ψ has enough vanishing moments, the coefficients at finer scales will become smaller faster (see *e.g.* Section 6.1.3 in Mallat 2008).

Orthogonal wavelets dictionaries on an interval. So far we have talked about the real line, but indeed for our applications of representing functional data, we are mostly interested in bases of $\mathrm{L}^2([0,1])$. Several constructions are possible. The simplest ones consist in extending the function on the real line. The extension can be

- *Periodic*: function y is repeated, this introduces discontinuities if $y(0) \neq y(1)$.
- *Symmetric (folding)*: y is extended as to $[-1,1]$ as $y_0(\theta) = y(\theta)$ for $\theta \in [0,1]$ and $y_0(\theta) = y(-\theta)$ for $\theta \in [-1,0]$. Then y_0 is extended periodically to the real line. The obtained signal is continuous.

Then such extended functions can be decomposed on a wavelet basis of $\mathrm{L}^2(\mathbb{R})$. This is equivalent respectively to decomposition of the original signal on so-called *periodic wavelets* or *folded wavelets* (see *e.g.* Section 7.5 in Mallat 2008). However in both constructions, the boundary wavelets (wavelets which support goes beyond the bound-

aries of $[0,1]$) lose their vanishing moments properties (they have respectively 0 vanishing moments for the periodic extension and 1 for the symmetric extension). This results in high coefficients near the boundaries. The vanishing moments can however be kept at the price of slightly more complex construction using *boundary wavelets* (Meyer, 1991; Cohen et al., 1993).

Linear aproximation error for regular functions. In terms of approximation errors, for $s > 0$, let $\mathcal{W}^s([0,1])$ be the Sobolev space of functions that are restrictions over $[0,1]$ of functions in the Sobolev space $\mathcal{W}^s(\mathbb{R})$. Then if the wavelets have q vanishing moments, for $0 < s < q$, belonging to $\mathcal{W}^s([0,1])$ is equivalent to having an error decreasing with an exponent of least $-2s$ with scale (see *e.g.* Theorem 9.5 in Mallat 2008); the scale being the inverse of the resolution. As in the Fourier case, this highlights that representing a function on a truncated wavelet basis can be seen as exploiting a belief on regularity. The more regular we think the function is, the lower the number of scales we consider.

Nonlinear approximation. We talked about linear approximation of very regular signals in wavelets bases. However, generally wavelets start to shine when Fourier representation fails: to represent signals that are not that regular and display local features and singularities. Not surprisingly, to represent such functions with few coefficients, the linear approximation scheme (see Definition 3.8) is not optimal. In that case, selecting the d_0 components from the basis which have the highest inner product (in absolute value) with the function to represent is a better strategy–see *e.g.* Mallat 2008, Section 9.2).

Definition 3.12 (Nonlinear approximation)**.** *Let $(\phi_l)_{l\in\mathbb{N}}$ be an orthonormal basis of $\mathsf{L}^2([0,1])$, the nonlinear approximation of $w \in \mathsf{L}^2(\Theta)$ of order $(d, d_0) \in (\mathbb{N}^*)^2$, $d_0 \leq d$ is the following:*

$$w_{(d,d_0)} = \sum_{l\in\Gamma} \langle w, \phi_l \rangle_{\mathsf{L}^2(\Theta)} \phi_l, \tag{3.17}$$

where $\Gamma := \{l \in [\![d]\!],\ |\langle w, \phi_l \rangle_{\mathsf{L}^2(\Theta)}| \geq t\}$ with $t \in \mathbb{R}_+$ a threshold such that $|\Gamma| = d_0$.

A case where the nonlinear approximation scheme can be shown to highly outperform the linear approximation scheme is that of piecewise regular signals (see *e.g.* Theorem 9.12 in Mallat 2008). Such functions display a finite number of discontinuities and are uniformly Lipschitz between these.

Example 3.13. *If we want to find a common dictionary to represent several functions $(y_i)_{i=1}^n$ drawn i.i.d. from a probability distribution ρ_Y on $\mathsf{L}^2(\Theta)$ that we believe should be similar in some sense (for instance they may share the same singularity points, they may have the same degree of smoothness...), the idea of nonlinear approximation can be used. We can for instance compute all the scalar products $(\langle y_i, \phi_l \rangle_{\mathsf{L}^2(\Theta)})_{i,l=1}^{n,d}$ and select the d_0 atoms for which the quantities $\left(\sum_{i=1}^n |\langle y_i, \phi_l \rangle_{\mathsf{L}^2(\Theta)}|\right)_{l=1}^d$ are the highest.*

In the next section we introduce some basics of spline smoothing, which also enjoy nice properties. The degree of smoothness can be set explicitly and through the use of a dictionary of B-splines, the actual representation of functions can be very efficient. Approximation results similar to those we showed above exist, however we do not expose them to remain concise.

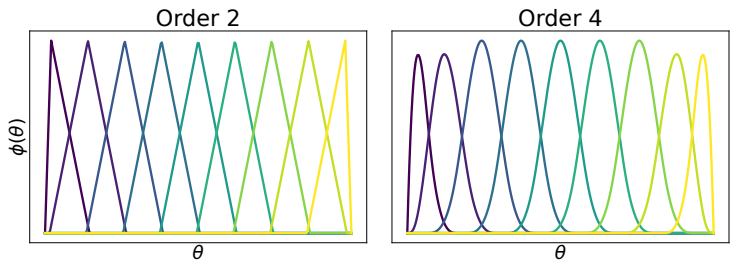


Figure 3.2: B-splines of order 2 and 4

Splines

A highly popular way of representing functional data is through the use of spline functions. These are ubiquitous in the FDA literature (Silverman (1984, 1985); Ramsay and Dalzell (1991) to cite only a few early FDA works). We stick here with the simpler case where $\Theta = [0,1]$, even though splines in higher dimensions can be defined (Wang, 2001). More precisely, a spline scheme is determined by a set $(\tau_t)_{t=0}^m \in \Theta^{m+1}$ of *knots* (located at *breakpoints*, see Remark 3.14), and an order $s \in \mathbb{N}$. A spline is then a function that is piecewise polynomial between the knots, the polynomials being of degree $s-1$ and such that the spline is continuous, and if $s > 2$, such that the derivatives up to order $s-2$ are continuous as well.

Remark 3.14 (Coincident knots)**.** *The knots and breakpoints are not the exact same thing. The latter are distinct, while several knots may be placed at the same breakpoint so as to reduce the smoothness degree at that breakpoint. More precisely, if r knots are placed at the same breakpoint and we are considering splines of order s, then the derivatives up to order $s-r-1$ are continuous at that breakpoint.*

Therefore, to smooth a discretized and possibly noisy function $(\theta_i, \tilde{y}_i)_{i=1}^n$, it is natural to try to find the spline function which fits best the data, for a given order and a set of knots. Depending on these parameters and the data, a perfect interpolation (going through all the observations) may exist or not. In any case, when the data are noisy, exactly interpolating them is not desirable. A natural question is, in practice, can spline fitting be done efficiently ? A possible answer which especially fits our needs in this thesis comes from B-splines.

B-splines. The space of spline functions of order s with knots $(\tau_t)_{t=0}^m \in \Theta^{m+1}$ is a vector space. Several bases of this space can be found, however, the B-splines basis which was first introduced in Schoenberg (1946), is the one that enjoys the best properties and has been preferred in most applications (de Boor, 2001). One of its advantages is that the basis functions verify a simple recursion formula which allows for evaluating them very efficiently (de Boor, 1972; Cox, 1972). Another attractive property is that a B-spline of order s is nonzero over at most s intervals, which are adjacent. In other words, B-splines enjoy a compact support property.

Definition 3.15. *Let $\Theta = [0,1]$ and let $(\tau_t)_{t=0}^m \in \Theta^{m+1}$ be a sequence of knots within that interval. Up to a constant scaling factor, there is a unique spline $B_{t,s}$ of order s satisfying*

$$B_{t,s}(\theta) \begin{cases} \neq 0 & \text{if } \tau_t \leq \theta < \tau_{t+1}, \\ = 0 & \text{otherwise.} \end{cases}$$

Remark 3.16. *The classic normalization is to choose the spline of order 0 as*

$$B_{t,0}(\theta) \begin{cases} = 1 & \text{if } \tau_t \leq \theta < \tau_{t+1}, \\ = 0 & \text{otherwise.} \end{cases}$$

The B-splines of higher order verify the following recursion (de Boor, 1972; Cox, 1972):

Theorem 3.17. *Let $\Theta = [0,1]$ and let $(\tau_t)_{t=0}^m \in \Theta^{m+1}$ be a sequence of distinct knots within that interval. For $s \geq 1$ and $t \leq m - s$, the B-splines verify the recursion*

$$B_{t,s}(\theta) = \frac{\theta - \tau_t}{\tau_{t+s} - \tau_t} B_{t,s-1}(\theta) + \frac{\tau_{t+s} - \theta}{\tau_{t+s} - \tau_t} B_{t+1,s-1}(\theta).$$

Remark 3.18. *When there are coincident knots, the above formula cannot be used for all knots (the recursion implies divisions by zero). We refer the reader to (de Boor, 1972) for details on how to treat coincident knots.*

B-splines of a given order can indeed be used as a dictionary to represent functions. Through the choice of the order, we can set a desired level of smoothness and they benefit from compact support which can be exploited to speed up computations.

Next we turn back to RKHS which we studied extensively in Section 2.1 from the previous chapter. We focus however on the regularity of the functions they permit to model and on how they can be approximated using a dictionary of functions.

3.2.2 Reproducing kernel Hilbert spaces (RKHS)

As we have seen in Section 2.1, RKHSs are functional spaces which enjoy many desirable properties for machine learning. In the light of the present chapter, we show that they also constitute an attractive family of spaces to represent functions. Through the choice of kernel, the corresponding RKHS is a space which contains more or less regular functions. We give first several elements and examples to highlight this. Then in a second part, we study how the RKHSs can be approximated using finite dictionary expansions, which make the representation of functions in these spaces less costly.

RKHS and smoothness

The choice of kernel indeed has an impact on the smoothness of the functions in the RKHS. In fact the functions in the RKHS inherits the smoothness of the kernel in the following sense (this is a corollary of Theorem 10.45 in Wendland 2004, see also Zhou 2008):

Theorem 3.19. *Let Θ be an open subset of $\mathbb{R}^b$. Let k be a kernel such that $k \in \mathcal{C}^{2s}(\Theta \times \Theta)$ (k is $2s$ times continuously differentiable), then $\mathcal{H}_k \subset \mathcal{C}^s(\Theta)$, in other words the RKHS of k contains only s-times continuously differentiable functions.*

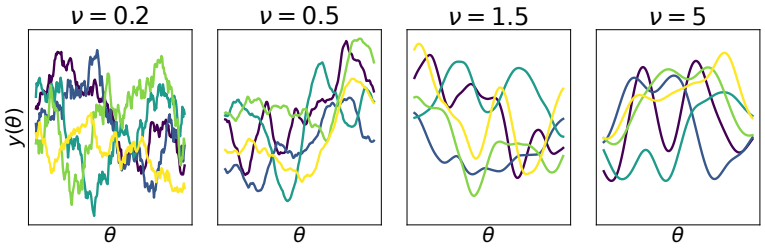


Figure 3.3: Random functions in RKHSs associated to the Matérn kernel for different values of the smoothness parameter ν

A popular class of kernels the effect of the kernel on the smoothness of the functions in the RKHS is that of Matérn kernels. We display examples in Figure 3.3 to show how the parameter ν determines the smoothness of the functions contained in the RKHS.

Example 3.20 (Matérn Kernel). *Let $\Theta \subset \mathbb{R}^b$. For constants $\nu > 0$ and $h > 0$, the Matérn kernel is given by*

$$k_{\nu,h}(\theta_1, \theta_2) = \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}\|\theta_1 - \theta_2\|_{\mathbb{R}^b}}{h} \right)^{\nu} B_\nu \left(\frac{\sqrt{2\nu}\|\theta_1 - \theta_2\|_{\mathbb{R}^b}}{h} \right), \tag{3.18}$$

where Γ is the gamma function and B_ν is the modified Bessel function of the second kind of order ν (see e.g. *Abramowitz and Stegun 1965).*

The parameter h corresponds to the scale parameter (in practice it determines how fast the functions in the RKHS can variate) whereas the parameter ν determines the degree of smoothness of the functions in the RKHS, the higher, the smoother. More precisely, a link can be made with fractional Sobolev spaces. These extend the notion of Sobolev spaces to non-integer orders s. Integer Sobolev spaces can be characterized equivalently in terms of square integrability of their Fourier transform multiplied by the function $\xi \mapsto (1 + |\xi|^2)^{\frac{s}{2}}$, and for this characterization, s need not be an integer–see *e.g.* Demengel and Demengel (2012, Section 4.2). We then have the following result (Tuo and Jeff Wu, 2016, Corollary A.6)

Theorem 3.21. *For $\lfloor \nu + q/2 \rfloor > q/2$, the RKHS generated by the Matérn kernel equals the (fractional) Sobolev space $\mathcal{W}^{\nu+q/2}(\Theta)$ with equivalent norms.*

Thus, purely in terms of functions belonging to the space, the RKHS of the Matérn kernel and the above fractional Sobolev spaces are the same. However, if we endow them with a norm, the resulting spaces may not be the same. For the Matérn kernel, to the best of our knowledge, the Sobolev norm to which the RKHS norm corresponds has not been exhibited. We will see below that for some examples, the correspondence can be exhibited.

Remark 3.22. *When $\nu = r + 1/2$ for some $r \in \mathbb{N}^*$, the expression in Equation (3.18) can be simplified to a product between an exponential function and a polynomial of degree r–see Rasmussen and Williams (2006, Section 4.2, Equation 4.17). For the most common values,*

this yields

$$
\begin{aligned}
k_{1/2,h}(\theta_1,\theta_2) &= \exp\left(-\frac{\|\theta_1-\theta_2\|_{\mathbb{R}^b}}{h}\right),\\
k_{3/2,h}(\theta_1,\theta_2) &= \left(1+\frac{\sqrt{3}\|\theta_1-\theta_2\|_{\mathbb{R}^b}}{h}\right)\exp\left(-\frac{\sqrt{3}\|\theta_1-\theta_2\|_{\mathbb{R}^b}}{h}\right),\\
k_{5/2,h}(\theta_1,\theta_2) &= \left(1+\frac{\sqrt{5}\|\theta_1-\theta_2\|_{\mathbb{R}^b}}{h}+\frac{5\|\theta_1-\theta_2\|^2_{\mathbb{R}^b}}{3h^2}\right)\exp\left(-\frac{\sqrt{3}\|\theta_1-\theta_2\|_{\mathbb{R}^b}}{h}\right).
\end{aligned}
\tag{3.19}
$$

We see that for $\nu = 1/2$ this corresponds to the Laplace (or exponential) kernel.

Remark 3.23. *The Gaussian kernel, defined in Equation (2.1) from the previous chapter, is another well-known example. It actually corresponds to the limit of the Matérn kernel when $\nu \to +\infty$ (Stein, 1999) in the sense that for $h > 0$,*

$$
\lim_{\nu\to+\infty} k_{\nu,h}(\theta_1,\theta_2) = \exp\left(-\frac{\|\theta_1-\theta_2\|^2_{\mathbb{R}^b}}{2h^2}\right) \quad \theta_1,\theta_2 \in \Theta \subset \mathbb{R}^b.
$$

We highlight that in Equation (2.1), we defined it however with another parametrization setting $\gamma = \frac{1}{2h^2}$.

This kernel gives rise to a RKHS of continuously infinitely differentiable functions. With a small stretch, this is a consequence of Theorem 3.19 since the Gaussian kernel is in $\mathcal{C}^\infty(\Theta\times\Theta)$. Therefore a RKHS with Gaussian kernel is a good choice to represent highly regular functions. Some Sobolev spaces endowed with a given scalar product and the corresponding norm, can be shown to be RKHSs and their kernel can be exhibited.

Example 3.24. *Let $\Theta = \mathbb{R}^b$ and consider $\mathcal{W}^s(\mathbb{R}^b)$ the Sobolev space of order s (see Remark 3.7), then for all $s > \frac{q}{2}$, the space $\mathcal{W}^s(\mathbb{R}^b)$ endowed with the scalar product*

$$
\langle v, w\rangle_{\mathcal{W}^s(\mathbb{R}^b)} = \sum_{|\alpha|\le s} \langle \mathrm{D}^\alpha v, \mathrm{D}^\alpha w\rangle_{\mathrm{L}^2(\mathbb{R}^q)},
\tag{3.20}
$$

is a RKHS–see e.g. *Novak et al. (2018)–which kernel can be expressed as an integral on $\mathbb{R}^b$.*

Remark 3.25. *This integral can be computed when $q = 1$. For instance instance if $s = 1$, the reproducing kernel of the Sobolev space is the exponential kernel:*

$$
k_1(\theta_1,\theta_2) = \frac{1}{2}\exp(-|\theta_1-\theta_2|).
$$

For $s = 2$ and $q = 1$, the associated kernel is

$$
k_2(\theta_1,\theta_2) = \frac{\sqrt{3}}{3}\exp\left(-\frac{\sqrt{3}|\theta_1-\theta_2|}{2}\right)\sin\left(\frac{|\theta_1-\theta_2|}{2}+\frac{\pi}{6}\right).
$$

A general formula for all s can be found in Novak et al. (2018, Equation 3).

Another well-known example is the Sobolev space on $[0,1]$ which contains functions w such that $w(0) = w(1)$ (Wahba, 1990, Section 2.1).

Example 3.26. *For $s \geq 1$, the Sobolev space $\mathcal{W}^s([0,1])$ of functions verifying the boundary condition $w(0) = w(1)$ endowed with the scalar product*

$$\langle v, w \rangle_{\mathcal{W}^s([0,1])} = \langle v, w \rangle_{\mathrm{L}^2([0,1])} + \langle v^{(s)}, w^{(s)} \rangle_{\mathrm{L}^2([0,1])}, \tag{3.21}$$

is a RKHS with reproducing kernel

$$k(\theta_1, \theta_2) = 1 + \frac{(-1)^{s-1}}{(2s)!} B_{2s}(\lceil \theta_1 - \theta_2 \rceil), \tag{3.22}$$

where B_{2s} denotes the Bernoulli polynomial of order $2s$ and $\lceil \cdot \rceil$ denotes the fractional part.

Dictionaries to approximate RKHSs

Once we have chosen our kernel k so that the RKHS $\mathcal{H}_k$ is adapted to model the functions of interest, several possibilities are available to actually represent them in $\mathcal{H}_k$. Indeed, $\mathcal{H}_k$ is a functional space, and therefore it is possibly infinite-dimensional.

To overcome this difficulty, one can rely on the representer theorem (see Theorem 2.17). Indeed a smoothing problem of the form Problem 3.2 with $\mathcal{W} = \mathcal{H}_k$ and $\Omega(h) = \lambda \|h\|^2_{\mathcal{H}_k}$ does benefit from it. Therefore, any minimizer has the form

$$\hat{h}_i = \sum_{s=1}^{m_i} \alpha_{is} k(\cdot, \theta_{is}), \quad \alpha_i \in \mathbb{R}^{m_i}.$$

This eludes the issue of infinite dimension. However, since typically in dense FDA, the number of locations per functions is high (conceptually, infinite), this may not be the most efficient way. Indeed solving the problem has a time complexity of the order $\mathcal{O}(m_i^3)$. Therefore, we introduce the following alternatives to approximate functions in RKHSs using dictionaries.

Random Fourier features. We talked about random Fourier features (Rahimi and Recht, 2007) in Section 2.1.4. Let $d \in \mathbb{N}^*$ be the number of random frequencies to consider. Then, provided we know how to sample from ρ, the spectral measure of k, we can approximate functions in the RKHS as linear combinations of the (random) functions:

$$\forall l \in [\![d]\!], \quad \phi_l : \theta \mapsto \frac{1}{\sqrt{d}} \cos(\omega_l^{\mathrm{T}} \theta), \quad \phi_{d+l} : \theta \mapsto \frac{1}{\sqrt{d}} \cos(\omega_l^{\mathrm{T}} \theta), \tag{3.23}$$

where $(\omega_l)_{l=1}^d$ are drawn i.i.d. according to the spectral measure of the kernel ρ. The complexity of solving the smoothing problem in this dictionary is of the order $\mathcal{O}(d^3)$.

Spectral approximation. Another possible way to approximate functions in a RKHS is to express them as a linear combination of a finite number of eigenfunctions of the integral operator associated to the kernel k (see Section 2.1.2). As highlighted in this section, in general we cannot compute the eigenfunctions in closed form, however, we can estimate those from a finite number of observations as illustrated in Equation (2.4) and Equation (2.7). Then the functions in the RKHS can be approximated as a linear combination of the $d \in \mathbb{N}^*$ estimated eigenfunctions associated to the d larger eigenvalues. To better understand why this scheme makes sense, let us consider the true eigendecomposition of the integral operator $\mathrm{T}_{k,\mu}$ (see Definition 2.11) associated to the kernel k and to a Borel measure μ. We have the following equivalent characterization of the RKHS $\mathcal{H}_k$ (see *e.g.* Theorem 4 in Section III of Cucker and Smale 2001):

Theorem 3.27. *Let k be a kernel and let μ be a Borel measure on Θ. Let $(\lambda_l, \phi_l)_{l=1}^{+\infty}$ be the eigenfunction and eigenvalues pairs of the integral operator $T_{k,\mu}$ associated with k and μ. Suppose additionally that for all $l \in \mathbb{N}^*$, $\lambda_l > 0$. Then the space defined as*

$$\left\{ w \in L^2(\Theta, \mu), \quad w = \sum_{l=1}^{+\infty} a_l \phi_l \quad \text{with} \quad \left(\frac{a_l}{\sqrt{\lambda_l}} \right)_{l \in \mathbb{N}} \in \ell^2(\mathbb{N}) \right\}, \tag{3.24}$$

endowed for $w = \sum_{l=1}^{+\infty} a_l \phi_l$ and $v = \sum_{l=1}^{+\infty} b_l \phi_l$ with the scalar product

$$\langle w, v \rangle = \sum_{l=1}^{+\infty} \frac{a_l b_l}{\lambda_l}, \tag{3.25}$$

and the RKHS $\mathcal{H}_k$ are one and the same.

Remark 3.28. *The assumption of strict positivity of the eigenvalues is in fact not restrictive. If some eigenvalues are zero, then we just have to replace $L^2(\Theta, \mu)$ with the span of the eigenfunctions associated to strictly positive eigenvalues and everything remains correct.*

Remark 3.29. *One can also notice that this theorem holds true therefore regardless of the choice of the Borel measure μ.*

Remark 3.30. *In general, there is no reason for the eigenfunctions to belong to the RKHS. An approximation of a function in the RKHS with a finite number of eigenfunctions does not in general belong to the RKHS either.*

The characterization in Theorem 3.27 tells us that the quicker the decay of the eigenvalues, the easier it is to approximate the functions in the RKHS with a low number of eigenfunctions. Since the eigenfunctions form an orthonormal system in $L^2(\Theta, \mu)$, for $w \in \mathcal{H}_k$ we have for all $l \in \mathbb{N}^*$, $a_l = \langle w, \phi_l \rangle_{L^2(\Theta,\mu)}$. Consequently for w to belong to the RKHS, the sequence $(\langle w, \phi_l \rangle^2_{L^2(\Theta,\mu)})_{l \in \mathbb{N}^*}$ must decay significantly faster than the eigenvalues.

Example 3.31. *Examples of decay of eigenvalues for shift-invariant kernels (see Definition 2.21) on the real line with μ the Lebesgue measure include the following (see* e.g. *Williamson et al. 2001 or Section 12.4.6 in Schölkopf and Smola 2002):*

- *For the Laplace kernel $(\theta_1, \theta_2) \longmapsto \exp(-|\theta_1 - \theta_2|)$, the eigenvalues of the integral operator decay polynomially as $(\beta^2 l^{-(\alpha+1)})$ for some $\beta \in \mathbb{R}$ and $\alpha > 0$.*
- *For the Cauchy kernel $(\theta_1, \theta_2) \longmapsto \frac{1}{1+(\theta_1-\theta_2)^2}$ the eigenvalues of the integral operator decay exponentially (or equivalently, geometrically) as $(\beta^2 \exp(-\alpha(l-1))$ for some $\alpha, \beta > 0$.*
- *For the Gaussian kernel $(\theta_1, \theta_2) \longmapsto \exp(-(\theta_1 - \theta_2)^2)$ the eigenvalues of the integral operator decay at a quadratic-exponential rate $(\beta^2 \exp(-\alpha(l-1)^2)$ for some $\alpha, \beta > 0$.*

Example 3.32. *In higher dimension ($\Theta = \mathbb{R}^b$), considering sub-Gaussian measures for μ, we have the following rates of decay (see* e.g. *Bach 2017):*

- *For the Gaussian kernel, the decay of eigenvalues is geometric*

- *For the Matérn kernel leading to a Sobolev space of order s, the rate of decay of the eigenvalues of the integral operator is of the form $(l^{-2s/q})$.*

Therefore, RKHSs constitute attractive spaces to represent functions. Through the choice of kernel we can control the smoothness of the functions they contain and they benefit from many practical properties to actually represent the functions. One can either use the representer theorem or approximation schemes exploiting either random Fourier features or a spectral decomposition.

Nevertheless, the various off-the-shelf dictionaries that we have presented so far may not be adapted for more complex functions. Therefore, in the next section, we introduce some possibilities to learn a dictionary from the observed functions.

3.2.3 Data dependent dictionaries

Functional principal component analysis (FPCA)

Principal component analysis is a key tool in multivariate analysis–see *e.g.* Jolliffe (2002). It consists in decomposing a set of signals along orthogonal directions (vectors) which explain most of their variations. It can be used to reduce the dimension of a set of signals or to filter out the noise (the components which explain a low quantity of the variations can be removed).

It is then a natural idea to extend it to represent functional data. In fact, in the literature of continuous stochastic processes, this idea has been widely studied and covered under the name of Karhunen-Loève expansion (Karhunen, 1947; Loève, 1948). This expansion can be naturally related to integral operators associated with kernels and Mercer's theorem that we approached under two different aspects in Section 2.1.2 and in the previous section. Let Θ be a compact metric space, we consider the probability space endowed with the Borel algebra and a probability measure $\mathbb{P}$. Consider a zero mean stochastic process $(\mathsf{W}(\theta))_{\theta\in\Theta}$ defined over this space with a continuous covariance function $k(\theta_1,\theta_2) = \mathrm{Cov}(\mathsf{W}(\theta_1),\mathsf{W}(\theta_2))$.

By definition of a covariance function, k is a kernel and since it is continuous, it is a Mercer kernel. Therefore there exists an orthonormal basis $(\phi_l)_{l\in\mathbb{N}}$ of $\mathsf{L}^2(\Theta)$ of eigenfunctions of the integral operator associated to k and the Lebesgue measure–Definition 2.11. Let $(\lambda_l)_{l\in\mathbb{N}}$ be the associated eigenvalues. The stochastic process W then admits the following representation

$$\mathsf{W}(\theta) = \sum_{l=1}^{+\infty} \mathsf{A}_l \phi_l(\theta), \quad \theta \in \Theta, \tag{3.26}$$

where the convergence is in $\mathsf{L}^2(\Theta)$ and uniform. The random variables $(\mathsf{A}_l)_{l\in\mathbb{N}}$ have zero mean and for all $l, r \in \mathbb{N}$, $\mathbb{E}[\mathsf{A}_l\mathsf{A}_r] = \delta_{lr}\lambda_l$ where $\delta_{lr} = 1$ if $l = r$ and 0 otherwise. This is the Karhunen-Loève expansion.

However, in practice, we do not have access to the true covariance function k but to sample paths from the stochastic process. The link with FDA is then easy to make. If these sample paths are smooth, considering them as realizations from a function-valued random variable is a somewhat equivalent approach. We therefore want to estimate the eigenfunctions from the data. For a visual intuition, we refer to Figure 3.4. It displays the main functional principal components and the corresponding

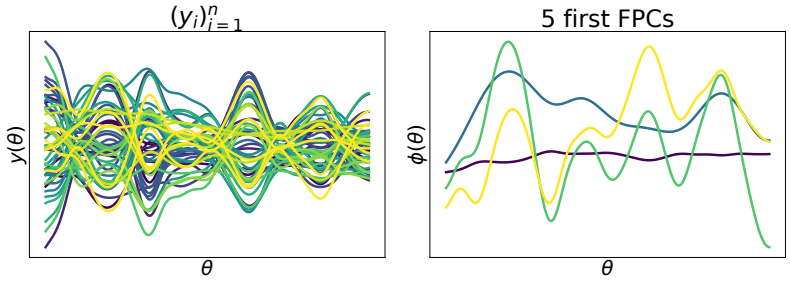


Figure 3.4: Observed synthetic functions and corresponding five most important eigenfunctions

observed functions which are instances from the synthetic dataset we introduce and study in Chapter 7.

From a functional point of view, to ensure that the eigenfunctions display the desired level of smoothness, several approaches are possible (Ramsay and Silverman, 2005; Shang, 2014). The functions can be smoothed beforehand in a dictionary (Ramsay and Dalzell, 1991), and then from these smoothed functions, we can estimate a smooth covariance function. Then, using the same dictionary to represent the eigenfunctions, a matrix eigensystem can be derived. We detail this procedure in Example 3.33. In order to simplify the exposition, we introduce the following vector-valued function associated to a dictionary of functions $(\phi_l)_{l=1}^d$

$$\boldsymbol{\phi}: \begin{pmatrix} \Theta & \to & \mathbb{R}^d \\ \theta & \mapsto & (\phi_1(\theta), \phi_2(\theta), \cdots, \phi_d(\theta))^{\mathrm{T}} \end{pmatrix}. \tag{3.27}$$

Example 3.33 (FPCA with dictionary smoothing)**.** *Suppose then that the observed functions have been smoothed using the dictionary $(\phi_l)_{l=1}^d$, and let $\mathrm{A} \in \mathbb{R}^{d\times n}$ be the matrix containing the estimated representation coefficients. The covariance function can be estimated as*

$$\hat{k}(\theta_1, \theta_2) = \frac{1}{n}\boldsymbol{\phi}(\theta_1)^{\mathrm{T}}\mathrm{A}\mathrm{A}^{\mathrm{T}}\boldsymbol{\phi}(\theta_2).$$

We want to solve the integral equation associated to this estimated covariance

$$\int_\Theta \hat{k}(\theta_1, \theta_2)\xi(\theta_2)\mathrm{d}\theta_2 = \lambda\xi(\theta_1), \quad \theta_1 \in \Theta. \tag{3.28}$$

Then using the same dictionary to represent the eigenfunction $\xi(\theta) = \boldsymbol{\phi}(\theta)^{\mathrm{T}}b$ with $b \in \mathbb{R}^d$, and setting $\mathrm{G} = \Phi^{\#}\Phi \in \mathbb{R}^{d\times d}$, the eigenfunctions and eigenvalues pairs are found by solving the matrix eigensystem

$$\frac{1}{n}\mathrm{G}^{\frac{1}{2}}\mathrm{A}\mathrm{A}^{\mathrm{T}}\mathrm{G}^{\frac{1}{2}}c = \lambda c,$$

and then for each eigenvector, we take $b = \mathrm{G}^{-\frac{1}{2}}c$.

Remark 3.34. *The possibility of the matrix* G *having eigenvalues equal or very close to zero is indeed problematic. However this is not envisioned in FDA as* $\boldsymbol{\phi}$ *is generally supposed to be a linearly independent family.*

Alternatively, if all the functions are observed at a numerous and common set of locations, a classic multivariate principal component analysis can be performed and then the obtained (discrete) eigenvectors can be smoothed in a dictionary.

To ensure the eigenfunctions are very smooth, a popular approach to perform FPCA in FDA is to additionally apply roughness penalties based on derivatives of the form given in Equation (3.12) to the eigenfunctions. Such a procedure and its statistical properties have been studied extensively in Pezzulli and Silverman (1993); Silverman (1996).

Functional dictionary learning

When functions display complex dynamics and are not necessarily smooth or when they are more or less smooth in different regions of their domain of definition, the representations provided by traditional bases will require the use of a large number of dictionary atoms to provide a decent representation. Therefore, general purpose dictionaries will fail to compress the signals efficiently. Indeed FPCA can be used to learn an orthonormal basis capturing the main directions of variations. However, the orthogonality constraint can sometimes be counterproductive because of the rigidity it imposes. We may still need to use many principal components (Mairal et al., 2009).

Another approach is to drop this constraint and allow representation vectors to be redundant. Yet, we can encourage the representation to be efficient in the sense of *sparsity*, meaning that only few representation vectors are used to represent the training signals. The corresponding problem is however NP-hard, and therefore, *dictionary learning* (Elad and Aharon, 2006) was introduced along with efficient approximate algorithms (Aharon et al., 2006; Lee et al., 2007). To expose the main concepts, we stick with discrete signals for now and suppose that all signals are observed at the same locations. For a sparsity level $s \in \mathbb{N}$, the dictionary learning problem reads

$$\begin{aligned} \min_{\mathrm{D}\in\mathbb{R}^{m\times d},\mathrm{A}\in\mathbb{R}^{n\times d}} & \frac{1}{n}\sum_{i=1}^{n} \|\mathrm{D}\mathbf{a}_i - \tilde{y}_i\|^2_{\mathbb{R}^m} \\ \text{subject to} \quad & \|\mathbf{a}_i\|_0 \leq s \quad i \in [\![n]\!], \\ & \|d_l\|_{\mathbb{R}^m} = 1 \quad l \in [\![d]\!], \end{aligned} \tag{3.29}$$

where the 0-norm $\|\cdot\|_0$ corresponds to the number of non-zero coordinates of a vector. The first set of constraints sets the level of sparsity of the representation coefficients and the second one imposes normalization of the atoms. This problems is not solvable in practice (because of the combinatorial nature of the 0-norm constraint). Practical algorithms then rely on alternating between a *sparse coding* step and *dictionary update* step.

Sparse coding. In the sparse coding step, the dictionary is fixed and we want to represent the observed signals sparsely on the dictionary. There are two main approaches.

- *Orthogonal matching pursuit (OMP)* (Pati et al., 1993; Mallat and Zhang, 1993) is a greedy strategy. In the context of dictionary learning, the support of the sparse representation is increased one atom at a time. At each step, the observed signals are represented through least squares on the selected atoms. However, to avoid recomputing the whole solution each time an atom is added, efficient strategies have been designed. The most popular one is to update a Cholesky factorization sequentially–see Sturm and Christensen (2012) for a comparison of many implementations.

- *Basis pursuit* is another possible strategy. It relaxes the $\|\cdot\|_0$ norm constraint to a $\|\cdot\|_1$ norm constraint. Equivalently, a $\|\cdot\|_1$ penalty is added. Then for each observation, a least square problem with $\|\cdot\|_1$ norm penalty is solved. These problems are indeed equivalent to *least absolute shrinkage and selection operator (LASSO)* problems (Tibshirani, 1996). Many efficient algorithms exist to solve these, the most well-know one being the fast iterative shrinkage-thresholding algorithm (FISTA) (Beck and Teboulle, 2009).

Dictionary update. To avoid the cost of optimizing over the full dictionary, block coordinate descent is generally used, in other words, dictionary atoms are updated one at a time. In practice, the most popular algorithm is K-SVD (Aharon et al., 2006), and its approximate version AK-SVD (Rubinstein et al., 2008) is generally used. Dealing with one atom at a time, K-SVD additionally optimizes the corresponding representation coefficients but does so keeping the sparsity structure fixed. This improves the likelihood of finding better atoms at the next atom update step–see *e.g.* Dumitrescu and Irofti (2018, Section 3.5).

Functional dictionary learning. Dictionary learning has been mostly studied for discretized signals, however in this thesis we are interested in representing functional data. It turns out that the same principle of using basis functions to represent the variables in functional PCA demonstrated in Example 3.33 can be applied to dictionary learning as well. Rubinstein et al. (2010) propose the framework of *doubly sparse* dictionary learning as well as an adaptation of the AK-SVD algorithm to solve it efficiently. Essentially, the dictionary atoms themselves are sparse linear combinations of functions from a base dictionary $(\psi_l)_{l=1}^c \in (\mathsf{L}^2(\Theta))^c$. It has been introduced mostly as a way to reduce the bad dependency of dictionary learning with respect to the dimension (number of sampling points). However, as the learnt atoms are represented on a base dictionary of functions, they are themselves functions. More precisely, consider the operator $\Psi : \mathbb{R}^c \to \mathsf{L}^2(\Theta)$ associated to this base dictionary, defined in the same way as Φ in Equation (3.3) for the dictionary $(\phi_l)_{l=1}^d$. We suppose here that the functions $(y_i)_{i=1}^n$ are observed fully. Given a sparsity level $s \in \mathbb{N}$ for the representations in $\boldsymbol{\phi}$ and a sparsity level $r \in \mathbb{N}$ for the representations in $\boldsymbol{\psi}$, the doubly sparse dictionary learning problem reads

$$\begin{aligned} \min_{\mathbf{B}\in\mathbb{R}^{c\times d},\mathbf{A}\in\mathbb{R}^{d\times n}} & \frac{1}{n}\sum_{i=1}^{n} \|\Psi_{(n)}\mathbf{B}\mathbf{a}_i - y_i\|^2_{\mathsf{L}^2(\Theta)} \\ \text{subject to} \quad & \|\mathbf{a}_i\|_0 \le s \quad i \in [\![n]\!], \\ & \|\mathbf{b}_l\|_0 \le r \quad l \in [\![d]\!], \\ & \|\Psi\mathbf{b}_l\|_{\mathsf{L}^2(\Theta)} = 1 \quad l \in [\![d]\!]. \end{aligned} \tag{3.30}$$

Then, the learnt dictionary of functions $(\phi_l)_{l=1}^d$ consists of the following atoms: for all $l \in [\![d]\!]$, $\phi_l = \Psi \mathbf{b}_l$.

The sparse K-SVD algorithm proposed in Rubinstein et al. (2010) for doubly sparse dictionary learning on discrete signals can be extended to the functional case. In the algorithm, the two quantities which involve the output functions are the computation of a weighted sum of these functions and the computation of scalar products between these functions and the atoms from the dictionary. Therefore, the following adaptation to discrete functions are possible.

- If the functions are observed on a regular grid which is dense enough, the algorithm can be run in its original form on the discretized functions. There is no need for smoothing the atoms afterwards since they are themselves linear combinations of functions from the dictionary $(\psi_l)_{l=1}^c$. It is possible to adapt the algorithm to missing data if the base locations of sampling are shared by all the functions yet for some functions, some locations are missing. However, this is possible only as long as the available locations for each observed functions allow for a correct estimation of the scalar products between the atoms of the base dictionary $(\psi_l)_{l=1}^c$ and the observed functions.
- Alternatively, the observed functions can be smoothed beforehand as in Example 3.33 and the algorithm can be adapted to this case as well

Other algorithms have been proposed for the dictionary update stage of doubly sparse dictionary learning (in the discrete case). Sulam et al. (2016) proposes a variation of the normalized iterative hard thresholding algorithm (Blumensath and Davies, 2010), while other more recent algorithms benefit from some theoretical guarantees and a clear computational complexity at the price of a more complex implementation (Nguyen et al., 2019).

3.3 Conclusion

In this chapter, we introduced several procedures to represent functions from discrete and possibly noisy observations. This is the general problem of smoothing which require the choice of a functional hypothesis space to represent the functions in. We particularly focused on which functional spaces could be used so as to exploit the properties of the functions to represent. We emphasized that many of these spaces are naturally the span of a given dictionary of functions or can be approximated through the use of a given dictionary. This is a cornerstone of our approach to functional output regression with projection learning introduced in Chapter 5.

4

Related works on nonlinear functional output regression

Contents

We now start to tackle the problem that is central to this thesis: that of nonlinear functional output regression (FOR). In the previous chapter (Chapter 3), we focused on some of the challenges that functional data represent. We showed how to use finite dimensional representation both to represent and smooth functions. In the next chapters, we illustrate our contribution to this problem with respect to different aspects. Consequently, the present chapter is devoted to the presentation of the main existing works on nonlinear FOR. We present the methods in a self-contained way and try to highlight the advantages and drawbacks of each approach, especially in terms of computational complexity.

4.1 Introduction

Let us first recall the motivation behind the FOR problem. In a large number of fields such as biomedical signal processing, epidemiology monitoring, speech and acoustics, climate science, etc., each data instance consists of a high number of measurements of a common underlying phenomenon. Such high-dimensional data generally enjoys strong smoothness across features. To exploit that structure, it can be interesting to model the underlying functions rather than the vectors of discrete measurements we observe, opening the door to functional data analysis (FDA, see *e.g.* Ramsay and Silverman 2005 or Wang et al. 2016). In practice, as highlighted in Section 3.1.2, FDA generally relies on the assumption that the sampling rate of the observations is high enough to consider them as functions. Of special interest is the general problem of FOR, in which the output variable is a function and the input variable can be of any type, including a function.

While functional linear models have received a great deal of attention–see the additive linear model and its variations (Ramsay and Silverman, 2005; Morris, 2015, and references therein)–, nonlinear ones have been less studied. Reimherr et al. (2018) extend the function-to-function additive linear model by considering a trivariate regression function in a reproducing kernel Hilbert space (RKHS). In non-parametric statistics, Ferraty et al. (2011) introduce and study variations of the Nadaraya-Watson kernel estimator for outputs in a Banach space. Oliva et al. (2015) rather project both input and output functions on orthogonal bases and regress the obtained output coefficients separately on the input ones using approximate kernel ridge regressions (KRR). Finally, extending kernel methods to functional data, Lian (2007); Kadri et al. (2010) introduce a function-valued KRR using a function-valued RKHSs (fv-RKHS) as a hypothesis class. To solve the problem in practice, they discretize the involved functions to obtain an approximation of the different terms in the optimization problem. In that context Kadri et al. (2016) proposes another solution. The closed-form involving an infinite-dimensional linear operator, they rather invert a low-rank approximation of the operator in question. We compare the different characteristics of these methods in Table 4.1 and highlight in red some of their restrictions which we wish to overcome in our contributions.

We develop in this chapter on all of the methods cited above to tackle function-valued regression. More precisely, in Section 4.2, we study the functional kernel ridge regression problem as well as existing techniques to solve it. Section 4.3 is dedicated to function-to-function regression using orthogonal dictionaries. We then shift our focus to the kernelized functional additive model in Section 4.4. Finally, we highlight briefly how kernel regression can be adapted to functional outputs in Section 4.5.

To present the models properly, we first recall the setup. FOR is a supervised learning problem with functional outputs. Given random variables X and Y taking respectively their values in $\mathcal{X}$ and a functional separable Hilbert space $\mathcal{Y}$, we want to infer a prediction function on $\mathcal{X}$ coherent with the unknown joint distribution of (X, Y). We rely on an i.i.d. sample $(x_i, y_i)_{i=1}^n$ to infer a statistical relationship. Depending on the methods, we will need to make further assumptions on $\mathcal{X}$. For instance, when the method is specific to function-to-function regression, $\mathcal{X}$ must be a functional space as well. We may also precise a particular space $\mathcal{Y}$, for instance we may assum that $\mathcal{Y} = \mathsf{L}^2(\Theta)$, the set of square integrable functions on a given compact set $\Theta \subset \mathbb{R}^b$. With this in place, we start with the problem of functional kernel ridge regression.

4.2 Functional kernel ridge regression (FKRR)

The main concepts that are used here, namely operator-valued kernel (OVK) and their associated fv-RKHSs are introduced in Section 2.2. Consequently, we do not redefine these notions here.

Remark 4.1 (fv-RKHSs and vv-RKHSs)**.** *In Section 2.2 we introduced vv-RKHS considering general vectors in a Hilbert space. We did so to keep things unified. However, now that we deal with the FOR problem, we start to distinguish function-valued RKHSs. Consequently, by distinction, from now on we use the term vector-valued RKHS to denote RKHSs of functions with finitely many outputs.*

Method	
FKRR-MC (Lian, 2007; Kadri et al., 2010)	Regression in fv-RKHS
FKRR-EIG (Kadri et al., 2016)	Regression in fv-RKHS
3BE (Oliva et al., 2015)	Triple basis estimator
KAM (Reimherr et al., 2018)	Kernel additive model

Method	Inputs	Representation	Fit complexity
FKRR-MC	Any	Discrete	$\mathcal{O}(n^3 + m^3)$
FKRR-EIG	Any	T_{k_Θ} eigenfunctions	$\mathcal{O}(n^3 + n^2md)$
3BE	Functions	Orthogonal basis	$\mathcal{O}(q^3 + q^2d)$
KAM	Functions	FPCA	$\mathcal{O}(n^2t^2 + d^2m^2 + n^3 + d^3)$

Table 4.1: Nonlinear FOR with square loss: existing methods' characteristics. Time complexity is given assuming separability assumptions on the kernels.

4.2.1 Regression in fv-RKHSs

Let then $\mathsf{K}^{(\mathrm{fun})} \in \mathcal{L}(\mathcal{Y})$ be an OVK and let $\mathcal{H}_{\mathsf{K}^{(\mathrm{fun})}}$ be its associated fv-RKHS. It can be used as a hypothesis class for the problem of Hilbert-valued regression using the square loss associated to the $\|\cdot\|_\mathcal{Y}$ norm:

$$\min_{g \in \mathcal{H}_{\mathsf{K}^{(\mathrm{fun})}}} \frac{1}{n} \sum_{i=1}^{n} \|y_i - g(x_i)\|_\mathcal{Y}^2 + \lambda \|g\|_{\mathcal{H}_{\mathsf{K}^{(\mathrm{fun})}}}^2. \tag{4.1}$$

This problem benefits from the representer theorem (Micchelli and Pontil, 2005)–see also Theorem 2.36–and therefore, any solution to Problem 4.1 has the form

$$g = \sum_{i=1}^{n} \mathsf{K}_{x_i}^{(\mathrm{fun})} \alpha_i, \tag{4.2}$$

with for all $i \in [\![n]\!]$, $\alpha_i \in \mathcal{Y}$. Injecting this representation into Problem 4.1, we obtain:

$$\min_{(\alpha_i)_{i=1}^n \in \mathcal{Y}^n} \frac{1}{n} \sum_{i=1}^{n} \left\| y_i - \sum_{j=1}^{n} \mathsf{K}^{(\mathrm{fun})}(x_i, x_j)\alpha_j \right\|_\mathcal{Y}^2 + \lambda \sum_{i=1}^{n} \sum_{j=1}^{n} \left\langle \mathsf{K}^{(\mathrm{fun})}(x_i, x_j)\alpha_i, \alpha_j \right\rangle_\mathcal{Y}. \tag{4.3}$$

Despite the representer theorem, this problem remains challenging since the coefficients $(\alpha_i)_{i=1}^n$ which are infinite-dimensional. Consequently, an approximation must be used. The assumption that the OVK $\mathsf{K}^{(\mathrm{fun})}$ is separable is then a cornerstone of all the existing resolution strategies. Moreover, the output functions are generally modeled using a scalar-valued RKHS. As highlighted in Remark 2.35, there are two ways of achieving this. One can consider that the output functions lie in $\mathrm{L}^2(\Theta, \mu)$ and use the OVK

$$\mathsf{K}^{(\mathrm{fun})} = k_\mathcal{X}(x_1, x_2)\mathrm{T}_{k_\Theta}, \tag{4.4}$$

with $k_\mathcal{X}$ a kernel on $\mathcal{X}$ and T_{k_Θ} the integral operator associated to a kernel k_Θ and a Borel measure μ on Θ. We recall the action of this operator for $y \in \mathrm{L}^2(\Theta, \mu)$ and $\theta_1 \in \Theta$,

$$(\mathrm{T}_{k_\Theta} y)(\theta_1) = \int_\Theta y(\theta_2) k_\Theta(\theta_1, \theta_2) \mathrm{d}\mu(\theta_2),$$

Or, another possible approach is to consider we are dealing with a $\mathcal{H}_{k_\Theta}$-valued problem and use the OVK

$$\mathrm{K}^{(\mathrm{fun})} = k_{\mathcal{X}}(x_1, x_2)\mathrm{I}_{\mathcal{H}_{k_\Theta}}.$$

Remark 4.2. *However, we the find the first option more relevant. Indeed, modeling the output functions in a RKHS $\mathcal{H}_{k_\Theta}$ is more of a choice of a hypothesis class and the output functions generally do not belong to $\mathcal{H}_{k_\Theta}$.*

Two main possibilities have been proposed to solve Problem 4.3

1. The problem can be discretized using a finite number of locations $(\theta_s)_{s=1}^m \in \Theta^m$, this is what is done in Lian (2007); Kadri et al. (2010). We detail this possibility in Section 4.2.2.

2. A finite rank approximation of the integral operator T_{k_Θ} can be used to obtain a finite dimensional parametrization of the representation coefficients, this is the solution proposed in Kadri et al. (2016). This approach is the object of Section 4.2.3.

4.2.2 Discretization approach

We now present the first approach using the OVK from Equation (4.4) and the locations $(\theta_s)_{s=1}^m$. The representation coefficients $(\alpha_i)_{i=1}^n$ are replaced by their evaluations at the points $(\theta_s)_{s=1}^m$. We denote by $(\mathbf{a}_i)_{i=1}^n$ these discretized functions, where for all $i \in [\![n]\!]$, $\mathbf{a}_i \in \mathbb{R}^m$. Let $\mathrm{A} \in \mathbb{R}^{m\times n}$ be the matrix whose columns are the vectors $(\mathbf{a}_i)_{i=1}^n$. Note that we deliberately choose to present the problem in matrix form, as we will use the same approach in Chapter 7. Therefore, this subsection can serve as an introduction to the procedure.

Let $\mathrm{K}_\Theta \in \mathbb{R}^{m\times m}$ be the kernel matrix associated to the kernel k_Θ and the locations $(\theta_s)_{s=1}^m$, the action of the integral operator can be approximated as

$$\forall i \in [\![n]\!], \quad \mathrm{T}_{k_\Theta}\alpha_i \approx \frac{1}{m}\mathrm{K}_\Theta \mathbf{a}_i.$$

Let now $\mathrm{K}_{\mathcal{X}} \in \mathbb{R}^{n\times n}$ be the kernel matrix associated to a kernel on $\mathcal{X}$ and the observations $(x_i)_{i=1}^n$. The regularization term can be approximated as

$$\|g\|^2_{\mathcal{H}_{\mathrm{K}^{(\mathrm{fun})}}} \approx \frac{1}{m^2}\,\mathrm{Trace}\left(\mathrm{A}^{\mathrm{T}}\mathrm{K}_\Theta \mathrm{A}\mathrm{K}_{\mathcal{X}}\right), \tag{4.5}$$

the $\frac{1}{m^2}$ term arising from the successive discrete approximations of the integral operator and that of the scalar product in $\mathrm{L}^2(\Theta, \mu)$.

Let $\mathrm{Y} \in \mathbb{R}^{m\times n}$ be the matrix whose i-th column corresponds to the evaluations of the function y_i at the locations $(\theta_s)_{s=1}^m$, we can approximate the first term in the optimization problem as

$$\frac{1}{nm}\left\|\mathrm{Y} - \frac{1}{m}\mathrm{K}_\Theta \mathrm{A}\mathrm{K}_{\mathcal{X}}\right\|^2_{\mathbb{R}^{m\times n}},$$

where the norm $\|\cdot\|_{\mathbb{R}^{m\times n}}$ stands for the Frobenius norm.

Using those approximations, we get the following discrete version of Problem 4.3

$$\min_{A\in\mathbb{R}^{n\times m}} \|Y - K_\Theta A K_\mathcal{X}\|^2_{\mathbb{R}^{m\times n}} + mn\lambda \operatorname{Trace}\left(A^T K_\Theta A K_\mathcal{X}\right). \tag{4.6}$$

Canceling the gradient with respect to the matrix A yields the matrix equation

$$K_\Theta^2 A K_\mathcal{X}^2 + \lambda mn K_\Theta A K_\mathcal{X} = K_\Theta Y K_\mathcal{X}.$$

Assuming that $K_\mathcal{X}$ and K_Θ are full rank, we can multiply left by K_Θ^{-1} and right by $K_\mathcal{X}^{-1}$, yielding

$$K_\Theta A K_\mathcal{X} + \lambda mn A = Y. \tag{4.7}$$

This is a discrete time Sylvester equation, and efficient techniques exist to solve these (Sima, 1996). More precisely, Equation (4.7) can be solved in $\mathcal{O}(n^3+m^3+n^2m+nm^2) \asymp \mathcal{O}(n^3+m^3)$ time. It is also possible to exploit the structure of the problem to compute an eigendecomposition. More precisely, Equation (4.7) is equivalent to the linear system (Dinuzzo et al., 2011)

$$(K_\mathcal{X} \otimes K_\Theta + mn\lambda \mathbf{I}) \operatorname{vec}(A) = \operatorname{vec}(Y). \tag{4.8}$$

It can therefore be solved efficiently with time complexity of essentially $\mathcal{O}(n^3+m^3)$ performing two eigendecompositions, one of $K_\mathcal{X}$ and on of K_Θ. An eigendecomposition of $K_\mathcal{X} \otimes K_\Theta$ can then be deduced. Indeed, let $(\nu_i, \mathbf{v}_i)_{i=1}^n$ and $(\lambda_s, \mathbf{u}_s)_{s=1}^m$ be the eigenvalue/eigenvector pairs respectively of $K_\mathcal{X}$ and K_Θ. Then the eigenvalue/eigenvector pairs of $K_\mathcal{X} \otimes K_\Theta$ are $(\nu_i\lambda_s, \mathbf{v}_i \otimes \mathbf{u}_s)_{i,s=1}^{n,m}$.

Time complexity. The overall time complexity of fitting FKRR with the discretization approach is then dominated by $\mathcal{O}(n^3+m^3)$.

4.2.3 Eigendecomposition approach

The other possible approach consists in deriving a closed-form for the functional representation coefficients $\boldsymbol{\alpha} \in \mathcal{Y}^n$ and ultimately represent these using a truncated basis of eigenfunctions of the integral operator T_{k_Θ}. This yields a finite dimensional system whose solution approximates that of the original infinite dimensional one.

To derive the closed-form for the functional coefficient, we must first introduce some tools of algebra using operator matrices. The block operator matrix resulting from the Kronecker product between $K_\mathcal{X}$ and T_{k_Θ}: $K_\mathcal{X} \otimes T_{k_\Theta} \in \mathcal{Y}^{n\times n}$ is a linear operator in $\mathcal{L}(\mathcal{Y}^n)$ represented as

$$K_\mathcal{X} \otimes T_{k_\Theta} = \begin{pmatrix} k_\mathcal{X}(x_1,x_1)T_{k_\Theta} & \cdots & k_\mathcal{X}(x_1,x_n)T_{k_\Theta} \\ \vdots & \cdots & \vdots \\ k_\mathcal{X}(x_n,x_1)T_{k_\Theta} & \cdots & k_\mathcal{X}(x_n,x_n)T_{k_\Theta} \end{pmatrix}.$$

Its action is given for $\boldsymbol{\alpha} \in \mathcal{Y}^n$ by

$$(K_\mathcal{X} \otimes T_{k_\Theta})\boldsymbol{\alpha} = \begin{pmatrix} \sum_{j=1}^n k_\mathcal{X}(x_1,x_j)T_{k_\Theta}\alpha_j \\ \vdots \\ \sum_{j=1}^n k_\mathcal{X}(x_n,x_j)T_{k_\Theta}\alpha_j \end{pmatrix} \in \mathcal{Y}^n.$$

Using this representation, and letting $\boldsymbol{\alpha} \in \mathcal{Y}^n$ be the functional coefficients for the estimator in Equation (4.2), Problem 4.3 can be rewritten compactly as

$$\frac{1}{n}\|\mathbf{y} - \mathrm{K}_{\mathcal{X}} \otimes \mathrm{T}_{k_\Theta} \boldsymbol{\alpha}\|^2_{\mathcal{Y}^n} + \lambda \langle \mathrm{K}_{\mathcal{X}} \otimes \mathrm{T}_{k_\Theta} \boldsymbol{\alpha}, \boldsymbol{\alpha} \rangle_{\mathcal{Y}^n}. \tag{4.9}$$

Setting the gradient with respect to $\boldsymbol{\alpha}$ to zero yields a ridge-type closed-form

$$\hat{\boldsymbol{\alpha}} = (\mathrm{K}_{\mathcal{X}} \otimes \mathrm{T}_{k_\Theta} + \lambda n \mathbf{I}_{\mathcal{Y}^n})^{-1} \mathbf{y}. \tag{4.10}$$

To solve this infinite dimensional linear system, Kadri et al. (2016) compute a finite rank approximation of $(\mathrm{K}_{\mathcal{X}} \otimes \mathrm{T}_{k_\Theta} + \lambda n \mathbf{I}_{\mathcal{Y}^n})^{-1}$. More precisely, let $(\nu_i, \mathbf{v}_i)_{i=1}^n$ and $(\lambda_l, \phi_l)_{l \in J}$ be the eigendecompositions respectively of $\mathrm{K}_{\mathcal{X}}$ and T_{k_Θ}. We draw the reader's attention the fact that $(\phi_l)_{l=1}^d$ are eigenfunctions and not eigenvectors. Nevertheless, to ensure that the eigendecomposition of T_{k_Θ} is actually useful, we must further suppose that k_Θ is continuous. Then since Θ is compact, $(\lambda_l, \phi_l)_{l \in J}$ is at most countable and forms an orthonormal system in $\mathrm{L}^2(\Theta, \mu)$ (see Section 2.1.2).

We can leverage the Kronecker structure. Thanks to this, the eigendecomposition of $\mathrm{K}_{\mathcal{X}} \otimes \mathrm{T}_{k_\Theta}$ is given by

$$(\nu_i \lambda_l, \mathbf{v}_i \otimes \phi_l)_{i=1,l=1}^{n,+\infty}.$$

Using the d eigenfunctions of T_{k_Θ} associated to its largest eigenvalues, the solution to Equation (4.10) is approximated as

$$(\mathrm{K}_{\mathcal{X}} \otimes \mathrm{T}_{k_\Theta} + \lambda n \mathbf{I}_{\mathcal{Y}^n})^{-1} \mathbf{y} \approx \sum_{i=1}^{n} \sum_{l=1}^{d} \frac{1}{\nu_i \lambda_l + \lambda} \langle \mathbf{y}, \mathbf{v}_i \otimes \phi_l \rangle_{\mathcal{Y}^n} \mathbf{v}_i \otimes \phi_l.$$

This can be developed as

$$(\mathrm{K}_{\mathcal{X}} \otimes \mathrm{T}_{k_\Theta} + \lambda n \mathbf{I}_{\mathcal{Y}^n})^{-1} \mathbf{y} \approx \sum_{i=1}^{n} \sum_{l=1}^{d} \frac{1}{\nu_i \lambda_l + \lambda} \left(\sum_{j=1}^{n} v_{ij} \langle y_j, \phi_l \rangle_{\mathcal{Y}} \right) \mathbf{v}_i \otimes \phi_l, \tag{4.11}$$

where for $i, j \in [\![n]\!]$, v_{ij} denotes the j-th coordinate of the eigenvector $\mathbf{v}_i$. Consequently, the scalar products $(\langle y_j, \phi_l \rangle_{\mathcal{Y}})_{i=1,l=1}^{n,d}$ can be computed (approximately) only once.

However, this requires the knowledge of the eigendecomposition of T_{k_Θ} in closed-form which is restrictive as highlighted in Section 2.1.2, for that reason, in practice Kadri et al. (2016) use a Laplace output kernel (see Example 2.15).

Time complexity. Since the eigendecomposition of T_{k_Θ} is supposed to be known in closed-form, we only have to compute that of $\mathrm{K}_{\mathcal{X}}$ which has time complexity $\mathcal{O}(n^3)$. However, the computation of the approximate dual coefficients in Equation (4.11) represent as well a significant overhead. Let us consider that $m \in \mathbb{N}$ locations in Θ are used to discretize the functions. Then the computation of the eigenfunctions $(\mathbf{v}_i \otimes \phi_l)_{i,l=1}^{n,d}$ dominate the complexity of solving Equation (4.11). Indeed, the corresponding time complexity is $\mathcal{O}(n^2 d m)$. Consequently, the overall time complexity to fit the method is dominated by the terms $\mathcal{O}(n^3 + n^2 d m)$.

Remark 4.3 (Adaptation for unknown eigendecomposition)**.** *If the eigendecomposition of T_{k_Θ} is not known, this method of resolution can be adapted using an empirical eigendecomposition of the kernel–see Equation (2.5).*

4.3 Triple basis estimator (3BE)

Another proposition which is closer to kernel projection learning that we introduce in Chapter 5 is that of Oliva et al. (2015). They propose to represent separately the input and output functions on truncated orthonormal bases obtaining a set of input and output decomposition coefficients. Therefore, this approach is dedicated to function-to-function regression. Let us suppose that $\mathcal{X} = \mathrm{L}^2(\Xi)$ with $\Xi \subset \mathbb{R}^p$ a compact set for some $p \in \mathbb{N}$. Let also $(\psi_t)_{t=1}^c$ and $(\phi_l)_{l=1}^d$ be truncated orthonormal bases respectively of $\mathcal{X}$ and $\mathcal{Y}$. For all $i \in [\![n]\!]$, let $\mathrm{A} \in \mathbb{R}^{n \times c}$ be the matrix which (i,t)-th entry is $\langle \psi_t, x_i \rangle_{\mathcal{X}}$ for $i \in [\![n]\!]$ and $t \in [\![c]\!]$. On the output functions' side, for all $l \in [\![d]\!]$, let $\mathbf{b}_l = (\langle \phi_l, y_i \rangle_{\mathcal{Y}})_{i=1}^n$.

Consider as well that we have drawn $q \in \mathbb{N}$ random Fourier features (RFF, Rahimi and Recht 2007, see also Section 2.1.4 for details) associated to a shift invariant kernel k on $\mathbb{R}^c$ (the space of coefficients of representation of the input functions on $(\psi_t)_{t=1}^c$). Let $\mathrm{Z} \in \mathbb{R}^{n \times 2q}$ be the matrix containing the evaluations of these RFFs for the input coefficients (the rows of the matrix A). Then, each output coefficient is predicted separately from the RFFs of the input coefficients. In other words, for all $l \in [\![d]\!]$, we solve the problem

$$\min_{\mathbf{w}_l \in \mathbb{R}^{2q}} \frac{1}{n} \|\mathbf{b}_l - \mathrm{Z}\mathbf{w}_l\|_{\mathbb{R}^n}^2 + \lambda \|\mathbf{w}_l\|_{\mathbb{R}^{2q}}^2. \tag{4.12}$$

The solution is given by

$$\hat{\mathbf{w}}_l = (\mathrm{Z}^\mathrm{T}\mathrm{Z} + \lambda \mathrm{I})^{-1} \mathrm{Z}^\mathrm{T} \mathbf{b}_l.$$

Time complexity (3BE). The matrix $(\mathrm{Z}^\mathrm{T}\mathrm{Z} + \lambda \mathrm{I})$, can be inverted only once to solve Problem 4.12 for all $l \in [\![d]\!]$. But we must still compute the coefficients $(\mathrm{Z}^\mathrm{T}\mathrm{Z} + \lambda \mathrm{I})^{-1} \mathrm{Z}^\mathrm{T} \mathbf{b}_l$ for all $l \in [\![d]\!]$, therefore the overall time complexity is $\mathcal{O}(q^3 + q^2 d)$.

This idea can be extended to general FOR. Suppose $\mathcal{X}$ is no longer a functional space, but rather any space on which a kernel $k_{\mathcal{X}}$ can be defined. We can pose Problem 4.12 with the input data intervening through a kernel instead of through RFFs on representation coefficients for the input functions.

$$\min_{h \in \mathcal{H}_{k_{\mathcal{X}}}} \frac{1}{n} \sum_{i=1}^n \|b_{li} - h(x_i)\|_{\mathcal{Y}}^2 + \lambda \|h\|_{\mathcal{H}_{k_{\mathcal{X}}}}^2. \tag{4.13}$$

This problem is a classic kernel ridge regression (KRR) problem with scalar outputs. It benefits from a representer theorem (see Theorem 2.17), and the optimal representer coefficients are given by

$$\alpha = (\mathrm{K}_{\mathcal{X}} + \lambda \mathrm{I})^{-1} \mathbf{b}_l.$$

Time complexity (1BE). Thanks to the particular form of the KRR solution, we can compute only once the inverse $(\mathrm{K}_{\mathcal{X}} + \lambda \mathrm{I})^{-1}$ to solve Problem 4.13 for all $l \in [\![d]\!]$. Therefore, the complexity here is the same as for a classic KRR problem, except that the matrix products $(\mathrm{K}_{\mathcal{X}} + \lambda \mathrm{I})^{-1} \mathbf{b}_l$ must be computed for $l \in [\![d]\!]$ resulting in an overall time complexity dominated by $\mathcal{O}(n^3 + dn^2)$.

We refer to the obtained estimator as single basis estimator (**1BE**).

Remark 4.4 (Link with the kernel projection learning ridge estimator)**.** *In the next chapter (Chapter 5), we propose an estimator which leverages vv-RKHSs and representation on a dictionary of functions. We propose a closed-form for this estimator when the square*

*loss is used. We highlight here that the **1BE** estimator corresponds to a particular case of our ridge estimator (Proposition 5.13) when the separable kernel $k_{\mathcal{X}}\mathrm{I}$ is used and the dictionary $(\phi_l)_{l=1}^d$ is orthonormal.*

4.4 Kernel additive model

In this section, we deal only with function-to-function regression. The most well-known model in functional data analysis is without doubts the additive linear model (Ramsay and Silverman, 2005; Morris, 2015).

4.4.1 Additive linear model

For each $\theta \in \Theta$, it models the evaluation of the output function $y_i(\theta)$ as an integral of the corresponding input function x_i over the input domain Ξ against a weighting function $b(\cdot, \theta)$ plus a constant term $a(\theta)$. Formally, the following empirical risk is minimized over the functions $a : \Theta \to \mathbb{R}$ and $b : \Xi \times \Theta \to \mathbb{R}$:

$$\frac{1}{n}\sum_{i=1}^{n}\left\| y_i - a - \int_{\Xi} b(\xi, \cdot)x_i(\xi)\mathrm{d}\xi \right\|^2_{\mathsf{L}^2(\Theta)}. \tag{4.14}$$

The common way to proceed is to represent the functions to learn (a and b) using truncated bases of $\mathsf{L}^2(\Xi)$ and $\mathsf{L}^2(\Theta)$. Let then $(\psi_t)_{t=1}^c$ and $(\phi_l)_{l=1}^d$ be dictionaries of functions pertaining respectively to $\mathsf{L}^2(\Xi)$ and $\mathsf{L}^2(\Theta)$. Using the convention that for all $(\xi, \theta) \in \Xi \times \Theta$

$$\boldsymbol{\psi}(\xi) = (\psi_t(\xi))_{t=1}^c \in \mathbb{R}^c \quad \text{and} \quad \boldsymbol{\phi}(\theta) = (\phi_l(\theta))_{l=1}^d \in \mathbb{R}^d.$$

Then we represent the functional regression coefficients as

$$a(\theta) = \mathbf{a}\boldsymbol{\phi}(\theta), \quad \mathbf{a} \in \mathbb{R}^d,$$
$$b(\xi, \theta) = \boldsymbol{\psi}(\xi)^{\mathrm{T}}\mathrm{B}\boldsymbol{\phi}(\theta), \quad \mathrm{B} \in \mathbb{R}^{c\times d},$$

and use these expressions for a and b in Equation (4.14). Consequently, we minimize the objective from Equation (4.14) with respect to the variables $\mathbf{a} \in \mathbb{R}^d$ and $\mathrm{B} \in \mathbb{R}^{c\times d}$.

Remark 4.5. *Importantly, there is no explicit regularization penalty in the problem, however some regularization is achieved implicitly through the choice of the dictionaries and their size (parameters c and d). We have highlighted in Chapter 3 how such truncation had an explainable regularizing effect, especially when the used dictionaries encode a notion of frequency. This is for instance the case for Fourier bases, wavelets bases or functional principal components.*

We have introduced this well-known model because it helps to understand the kernelized version proposed by Reimherr et al. (2018).

4.4.2 Kernel additive model

Reimherr et al. (2018) revisit the additive model keeping the integral form, yet introducing much more flexibility in the types of dependencies that can be modeled. To that end they assume the function under the integral is in a RKHS. This function

takes three inputs, the first is a location $\xi \in \Xi$ of the input domain, the second is the location $\theta \in \Theta$ of the output domain, and the third is the input function evaluated at ξ. They also add a regularization term using the RKHS norm, which on top of its smoothing effects, allows for finite parameterization through a particular instance of the representer theorem. The problem is the following

$$\min_{h \in \mathcal{H}_{k^{(\text{add})}}} \frac{1}{n} \sum_{i=1}^{n} \left\| y_i - \int_{\Xi} h(\xi, \cdot, x_i(\xi)) \mathrm{d}\xi \right\|^2_{\mathsf{L}^2(\Theta)} + \lambda \|h\|^2_{\mathcal{H}_{k^{(\text{add})}}}, \tag{4.15}$$

where $\lambda > 0$ is a regularization parameter, $k^{(\text{add})} : (\Xi, \Theta, \mathbb{R})^2 \to \mathbb{R}$ is a scalar-valued kernel and $\mathcal{H}_{k^{(\text{add})}}$ is its RKHS.

The authors solve Problem 4.15 representing the functions in $\mathsf{L}^2(\Theta)$ on the orthonormal family of empirical functional principal components associated to the output functions $(y_i)_{i=1}^n$. Using this representation they derive a specific representer theorem to reduce the number of variables of the problem to n^2. This bears similarity with the idea of the *double representer theorem* formalized in Laforgue et al. (2020).

More precisely, the first key element is to reformulate the problem in terms of coordinates on the orthonormal system formed by the empirical functional principal components $(\phi)_{l=1}^n$. This system can indeed be completed to constitute an orthonormal basis of $\mathsf{L}^2(\Theta)$, let $(\phi_l)_{l=1}^{+\infty}$ be the resulting basis. Then the $\mathsf{L}^2(\Theta)$ distance between the functions in the data-fitting term of Problem 4.15 equals the $\ell^2(\mathbb{N})$ distance between their respective scalar products with the elements of $(\phi_l)_{l=1}^{+\infty}$. In this new norm, using the reproducing property of $\mathcal{H}_{k^{(\text{add})}}$ and the linearity of both the integral and the scalar product, the data fitting term from Problem 4.15 can be rewritten as

$$\begin{aligned}
&\frac{1}{n} \sum_{i=1}^{n} \int_{\Theta} \left(y_i(\theta) - \int_{\Xi} h(\theta, \xi, x_i(\xi)) \mathrm{d}\xi \right)^2 \mathrm{d}\theta \\
&= \frac{1}{n} \sum_{i=1}^{n} \sum_{l=1}^{+\infty} \left(\langle y_i, \phi_l \rangle_{\mathsf{L}^2(\Theta)} - \left\langle h, \int_{\Xi} \int_{\Theta} k^{(\text{add})}_{(\theta, \xi, x_i(\xi))} \phi(\theta)_l \mathrm{d}\theta \mathrm{d}\xi \right\rangle_{\mathcal{H}_{k^{(\text{add})}}} \right)^2 .
\end{aligned}$$

From this expression, we conclude that the solution to Problem 4.15 must belong to the space

$$\mathcal{S} := \left\{ \int_{\Xi} \int_{\Theta} k^{(\text{add})}_{(\theta, \xi, x_i(\xi))} \phi_l(\theta) \mathrm{d}\theta \mathrm{d}\xi, \quad i \in [\![n]\!], \ l \in [\![d]\!] \right\}. \tag{4.16}$$

Indeed, any $h \in \mathcal{H}_{k^{(\text{add})}}$ can be written as $h = h_{\mathcal{S}} + h^{\perp}$ where $h \in \mathcal{S}$ and $h^{\perp} \in \mathcal{S}^{\perp}$. Yet, the scalar product between h and the generating vectors of $\mathcal{S}$ in Equation (4.16) shows that $h^{\perp}$ leaves the data-fitting term unchanged. Therefore, from Pythagoras' theorem, it strictly increases the regularization term. We conclude that any solution to Problem 4.15 belongs to $\mathcal{S}$.

We can then write any such solution as

$$\hat{h}(\theta, \xi, x) = \sum_{i=1}^{n} \sum_{l=1}^{n} \alpha_{id} \int_{\Theta} \int_{\Xi} k^{(\text{add})} \left((\theta, \xi, x(\xi)), (\theta', \xi', x_i(\xi')) \right) \phi_l(\theta') \mathrm{d}\theta' \mathrm{d}\xi',$$

where $\boldsymbol{\alpha} = (\alpha_{id})_{i,l=1}^{n,d} \in \mathbb{R}^{n\times d}$ are representation coefficients. We can take either $d = n$ if we wish to use all the principal components, or $d < n$ if we wish to approximate the problem using only these associated to the largest eigenvalues.

In order to rewrite Problem 4.15 compactly with $\boldsymbol{\alpha}$ as variable, the following set of quadri-indexed quantities is introduced. For $i, j \in [\![n]\!]$ and $l, r \in [\![d]\!]$, define

$$A_{iljr} = \int_{\Xi}\int_{\Theta}\int_{\Xi}\int_{\Theta} k^{(\mathrm{add})}\Big((\theta,\xi,x_j(\xi)),(\theta',\xi',x_i(\xi'))\Big)\phi_r(\theta)\phi_l(\theta')\mathrm{d}\theta\,\mathrm{d}\xi\,\mathrm{d}\theta'\mathrm{d}\xi'. \quad (4.17)$$

Then, Problem 4.15 can be rewritten as

$$\sum_{i=1}^{n}\sum_{l=1}^{d}\left(\langle y_i,\phi_l\rangle_{\mathrm{L}^2(\Theta)} - \sum_{i,r=1}^{n,d} A_{iljr}\alpha_{jr}\right)^2 + \lambda \sum_{i,l,j,r=1}^{n,d,n,d} \alpha_{il}A_{iljr}\alpha_{jr}.$$

To put the above problem in a familiar ridge regression form, let us consider the matrix $\mathsf{A} \in \mathbb{R}^{dn\times dn}$ obtained by collapsing the indices (i,l) and (j,r) together using the entries from Equation (4.17). Let us also define by $\mathrm{R} \in \mathbb{R}^{d\times n}$ the matrix which i-th column is $(\langle y_i,\phi_l\rangle_{\mathrm{L}^2(\Theta)})_{l=1}^{d}$. We can then rewrite the problem compactly as

$$\min_{\boldsymbol{\alpha}\in\mathbb{R}^{d\times n}} \|\mathrm{vec}(\mathrm{R}) - \mathsf{A}\,\mathrm{vec}(\boldsymbol{\alpha})\|^2_{\mathbb{R}^{dn}} + \lambda\,\mathrm{vec}(\boldsymbol{\alpha})^{\mathrm{T}}\mathsf{A}\,\mathrm{vec}(\boldsymbol{\alpha}). \quad (4.18)$$

Canceling the gradient with respect to $\mathrm{vec}(\boldsymbol{\alpha})$, we obtain

$$\mathrm{vec}(\hat{\boldsymbol{\alpha}}) = (\mathsf{A}^{\mathrm{T}}\mathsf{A} + \lambda\mathsf{A})^{-1}\mathsf{A}\,\mathrm{vec}(\mathrm{R}).$$

Supposing further that A is invertible, and observing that A is symmetric by the definition of its entries, we can simplify by A and obtain the more compact form

$$\mathrm{vec}(\hat{\boldsymbol{\alpha}}) = (\mathsf{A} + \lambda\mathbf{I})^{-1}\,\mathrm{vec}(\mathrm{R}). \quad (4.19)$$

We now highlight a possibility to drastically improve computational complexity of the estimator, which was not given in Reimherr et al. (2018). To that end, notice that if the kernel is separable in the following sense:

$$k^{(\mathrm{add})}\Big((\theta,\xi,u),(\theta',\xi',u')\Big) = k_{\Xi\times\mathbb{R}}\Big((\xi,u),(\xi',u')\Big)k_{\Theta}(\theta,\theta'), \quad (4.20)$$

the matrix A can be written as a Kronecker product between the following matrices. First let us define the input kernel matrix as

$$\mathrm{K}_{\mathcal{X}} := \left(\int_{\Xi}\int_{\Xi} k_{\Xi\times\mathbb{R}}\Big((\xi,x_j(\xi)),(\xi',x_i(\xi'))\Big)\mathrm{d}\xi\,\mathrm{d}\xi'\right)_{i,j=1}^{n} \in \mathbb{R}^{n\times n}, \quad (4.21)$$

and the output kernel matrix as

$$\mathrm{K}_{\Theta} := \left(\int_{\Theta}\int_{\Theta} k_{\Theta}(\theta,\theta')\phi_r(\theta)\phi_l(\theta')\mathrm{d}\theta\,\mathrm{d}\theta'\right)_{l,r=1}^{d} \in \mathbb{R}^{d\times d}. \quad (4.22)$$

We then have that

$$\mathsf{A} = \mathrm{K}_{\mathcal{X}} \otimes \mathrm{K}_{\Theta}. \tag{4.23}$$

Time complexity. Consequently, we can solve Equation (4.19) essentially with time complexity $\mathcal{O}(n^3 + d^3)$ using either a Sylvester solver or by performing an eigendecomposition of $\mathrm{K}_{\mathcal{X}}$ and one of K_{Θ} as we do to solve Equation (4.8). However, the real computational challenge for this method is rather to compute the matrices $\mathrm{K}_{\mathcal{X}}$ and K_{Θ}. Each of their entries involve double integrals: if we discretize these double integrals over $t \in \mathbb{N}$ input locations from Ξ and $m \in \mathbb{N}$ output locations from Θ, the cost of computing these matrices are respectively $\mathcal{O}(n^2 t^2)$ and $\mathcal{O}(d^2 m^2)$. Generally, the number of locations t and m is relatively high (they correspond to discretization points to represent *functions*). Consequently, these terms will completely dominate the computations. This also implies that if the kernel $k^{(\mathrm{add})}$ is not separable in the sense of Equation (4.20), computing the matrix A is out of reach numerically: it has time complexity $\mathcal{O}(n^2 t^2 d^2 m^2)$ which cannot be envisioned even for small values of the different quantities.

4.5 Kernel estimator

Kernel smoothing is a classic method to interpolate a function. In the context of regression in statistics it has been introduced simultaneously in Nadaraya (1964); Watson (1964). Based on a kernel which traduces a notion of similarity on the input space $\mathcal{X}$, it predicts the value at a new point in $\mathcal{X}$ as a local average of the outcomes on the training data weighted by the kernel. The average is then local in the sense that the kernels used generally have maximum mass at zeros and start to vanish as the distance between the points increases. Since the kernel enables flexibility in the inputs, using kernel regression for functional inputs is quite natural (Ferraty and Vieu, 2002).

To deal with more complex outputs however, a Nadaraya-Watson kernel estimator has been studied in Ferraty et al. (2011) in the general setting of Banach spaces. Considering a kernel function $k_0 : \mathbb{R} \longmapsto \mathbb{R}$ combined with a given semi-metric S on $\mathcal{X}$, for all $x \in \mathcal{X}$, they use the following estimator:

$$\frac{\sum_{i=1}^{n} k_0 \circ S(x, x_i) y_i}{\sum_{i=1}^{n} k_0 \circ S(x, x_i)}. \tag{4.24}$$

Time complexity. This method is very fast as fitting it boils down to memorizing the training data, however it can lack precision.

4.6 Conclusion

In this chapter, we have described the main existing approaches to nonlinear FOR. We gave details on the corresponding estimators and the numerical procedures to compute them, highlighting their computational complexity. In the next chapter, we introduce an approach to FOR combining vv-RKHSs and representation on a dictionary of functions. In order to assess the efficiency of our proposition, we will benchmark it on several problems against the FOR methods presented in this chapter.

Part II

Functional output regression

5

Kernel projection learning

Contents

In this chapter, we introduce a general approach to Hilbert-valued regression which exploits the representation of infinite-dimensional vectors in a finite dictionary. It learns to predict representation coefficients in this dictionary, solving however the original problem of Hilbert-valued regression. A key application of this method is functional output regression (FOR). In that context, our method can exploit the rich possibilities that exist to represent functions using dictionaries. We have highlighted some of these possibilities in Chapter 3, where we have also related the dictionaries to functional spaces and the properties of the functions they contain. To be able to tackle a wide class of problems, we focus on nonlinear regression, to that end we use a vector-valued reproducing kernel Hilbert space (vv-RKHS) as a hypothesis class to predict the coefficients in the dictionary. They are particularly relevant in that context as they can model complex nonlinear relationships yet remain efficient when we have few observations, which is typically the case in FOR problems.

After a brief introduction to Hilbert-valued regression, we introduce the framework of projection learning in Section 5.1. Then in Section 5.2 we study it extensively using vector-valued reproducing kernel Hilbert spaces (vv-RKHS) as hypothesis class. We propose estimators and computational strategies and apply the framework to FOR with partially observed functions. Ultimately, Section 5.3 validates the proposed estimators numerically, drawing comparisons with other nonlinear FOR methods. We highlight that this chapter corresponds to the contributions of

- **D. Bouche**, M. Clausel, F. Roueff and F. d'Alché-Buc. Nonlinear Functional Output Regression: A Dictionary Approach. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 235–243, 2021,

except for the theoretical analysis of the estimators which is deferred to the next chapter.

5.1 Projection learning

In this section, we introduce how the classical empirical risk minimization paradigm can be extended to the case of Hilbert-valued outputs in Section 5.1.1 and in this context, we propose our general framework for Hilbert-valued regression using dictionaries in Section 5.1.2.

5.1.1 Hilbert-valued regression

Let us assume that the output space $\mathcal{Y}$ is a separable Hilbert space and that $\mathcal{X}$ is a measurable space. We want to infer a dependency between two random variables X and Y taking their values respectively in $\mathcal{X}$ and $\mathcal{Y}$. Suppose that $\mathsf{Z} := (\mathsf{X}, \mathsf{Y})$ is distributed according to an unknown probability distribution ρ on $\mathcal{Z} := \mathcal{X} \times \mathcal{Y}$. Indeed we have only access to an i.i.d. sample $(x_i, y_i)_{i=1}^n \in \mathcal{Z}^n$. From this sample, we want to fit a statistical model which exploits the relationship between X and Y so that it can predict an element of $\mathcal{Y}$ for any $x \in \mathcal{X}$. However, since $\mathcal{Y}$ is a Hilbert space, the model must be Hilbert-valued. Let then $L : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ be a loss function measuring the discrepancy between two elements in $\mathcal{Y}$, and let $\mathcal{G} \subset \mathcal{F}(\mathcal{X}, \mathcal{Y})$ be a hypothesis class of $\mathcal{Y}$-valued functions. The choice of this set constitutes a first level of approximation and it can have a regularizing effect. Ideally, we would want to minimize the expected risk over it.

$$\min_{f \in \mathcal{G}} \mathcal{R}(f) := \mathbb{E}_{(\mathsf{X},\mathsf{Y})\sim\rho}\Big[L(\mathsf{Y}, f(\mathsf{X}))\Big]. \tag{5.1}$$

Example 5.1 (Square loss)**.** *The most natural loss is the square loss associated to the norm* $\|\cdot\|_{\mathcal{Y}}$*:*

$$(y_1, y_2) \mapsto \|y_1 - y_2\|_{\mathcal{Y}}^2. \tag{5.2}$$

In the case where $\mathcal{Y} = \mathsf{L}^2(\Theta)$ *the space of square integrable functions defined on some compact set* $\Theta \subset \mathbb{R}^b$*, this loss corresponds to the following integral*

$$(y_1, y_2) \mapsto \|y_1 - y_2\|_{\mathsf{L}^2(\Theta)}^2 = \int_\Theta (y_1(\theta) - y_2(\theta))^2 \mathrm{d}\theta. \tag{5.3}$$

However as we do not have access to the distribution ρ, we rather minimize an estimator of this expected risk based on the finite sample we do have access to, leading to the well known empirical risk minimization problem. A regularization term $\Omega : \mathcal{G} \to \mathbb{R}$ is generally added to further control the model's complexity

$$\min_{f \in \mathcal{G}} \frac{1}{n} \sum_{i=1}^n L(y_i, f(x_i)) + \Omega(f). \tag{5.4}$$

5.1.2 Approximated Hilbert-valued regression

The fact that $\mathcal{Y}$ is infinite-dimensional is indeed challenging. However, if we have reasons to think that with high probability, the random variable Y takes its values in a subspace of $\mathcal{Y}$ whose dimension is relatively low, we can drastically reduce the complexity of solving Problem 5.8. A possible way to exploit this assumption, is to suppose that this subspace can be spanned by a dictionary of vectors $\boldsymbol{\phi} := (\phi_l)_{l=1}^d \in \mathcal{Y}^d$. We first recall the definition of the projection operator associated with this dictionary which we introduced informally in Chapter 3.

Definition 5.2 (Projection operator)**.** *Let $(\phi_l)_{l=1}^d \in \mathcal{Y}^d$ be a family of vectors in $\mathcal{Y}$, we define their associated projection operator as*

$$\Phi : \begin{pmatrix} \mathbb{R}^d & \to & \mathcal{Y} \\ \mathbf{a} & \mapsto & \sum_{l=1}^d a_l \phi_l \end{pmatrix}. \tag{5.5}$$

The adjoint of this operator in $\mathcal{Y}$ is given by

$$\Phi^\# : \begin{pmatrix} \mathcal{Y} & \to & \mathbb{R}^d \\ y & \mapsto & (\langle y, \phi_l \rangle_{\mathcal{Y}})_{l=1}^d \end{pmatrix}, \tag{5.6}$$

therefore $\Phi^\# \Phi \in \mathbb{R}^{d \times d}$ is the matrix whose entries are the pairwise scalar products between the elements of the dictionary. In other words, its (l, r)-th entry is $\langle \phi_l, \phi_r \rangle_{\mathcal{Y}}$.

Remark 5.3. *Φ is bounded. Indeed, its $\mathcal{L}(\mathbb{R}^d, \mathcal{Y})$-norm is equal to the largest eigenvalue of the Gram matrix associated to the dictionary $\Phi^\# \Phi$.*

Then, supposing that we have chosen a dictionary of vectors which is adapted to the problem at hand, we propose to consider a hypothesis class which incorporates this approximation directly in Problem 5.1. More precisely, let $\mathcal{H} \subset \mathcal{F}(\mathcal{X}, \mathbb{R}^d)$ be a hypothesis class of $\mathbb{R}^d$-valued functions. We then propose to tackle the $\mathcal{Y}$-valued problem

$$\min_{h \in \mathcal{H}} \mathcal{R}(\Phi \circ h). \tag{5.7}$$

In terms of regularized empirical risk minimization, this yields

$$\min_{h \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^n L(y_i, \Phi h(x_i)) + \Omega_{\mathcal{H}}(h), \tag{5.8}$$

where $\Omega_{\mathcal{H}} : \mathcal{H} \to \mathbb{R}$ is a regularization function.

Remark 5.4 (Indirect regularization)**.** *In Problem 5.8, the regularization function works on h which is not ultimately the function that we use for prediction. Therefore this regularization is somewhat indirect in the sense that we could regularize with respect to the function $\Phi \circ h$ directly. For this chapter, as we use vector-valued RKHSs as function class for $\mathcal{H}$, regularizing through the RKHS norm of h enables us to have a representer theorem. However we will propose an approach in which we can regularize directly on $\Phi \circ h$ in Chapter 8 (Example 8.8).*

We highlight that this framework is quite general in the sense that we search a solution in the hypothesis space $\mathcal{G} = \{f : x \longmapsto \Phi h(x),\ h \in \mathcal{H}\}$ and solve a Hilbert-valued problem at the price of solving a multi-output one ($\mathbb{R}^d$-valued) in $\mathcal{H}$. The loss remains however functional. Moreover, any regression model capable of handling multiple outputs (*e.g.* neural networks, random forests, kernel methods...) is eligible. All this works of course assuming we have a dictionary $(\phi_l)_{l=1}^d$ which represents well the observations distributed according to ρ_Y. In practice, for functions, there are many possibilities to find such representation, we refer the reader to Chapter 3 for more details. To choose among these possibilities, we can look for a dictionary in which the observed vectors $(y_i)_{i=1}^n$ are represented with a low error.

Next, we focus on solving Problem 5.8 using vv-RKHSs as a hypothesis class $\mathcal{H}$.

5.2 Kernel projection learning

We have introduced RKHSs in Section 2.1 as well as their extension to model vector-valued functions in Section 2.2 and highlighted their advantageous properties which made them popular in machine learning. We therefore propose to study the projection learning problem using vv-RKHSs as a hypothesis class and show how those properties can be exploited in this case in Section 5.2.1. We focus mostly on the square loss and propose an estimator in closed-form. We demonstrate as well how this estimator can be computed efficiently if we additionally suppose that the operator-valued kernel is separable in Section 5.2.2. Finally, based on this estimator, we propose an estimator specific to functional output regression with missing observations in Section 5.2.3.

5.2.1 Vv-RKHSs and representer theorem

Let $\mathsf{K} : \mathcal{X} \times \mathcal{X} \longmapsto \mathcal{L}(\mathbb{R}^d)$ be an OVK and $\mathcal{H}_\mathsf{K} \subset \mathcal{F}(\mathcal{X}, \mathbb{R}^d)$ its associated vv-RKHS. We recall that for all $x_1 \in \mathcal{X}$ the operator $\mathsf{K}_{x_1} \in \mathcal{L}(\mathbb{R}^d, \mathcal{H}_\mathsf{K})$ is defined as

$$\mathsf{K}_{x_1} : u \mapsto \mathsf{K}_{x_1} u, \quad \text{with } \mathsf{K}_{x_1} u : x_2 \mapsto \mathsf{K}(x_2, x_1)u \in \mathbb{R}^d. \tag{5.9}$$

Then, we consider Problem 5.8 choosing $\mathcal{H} = \mathcal{H}_\mathsf{K}$ as a vector-valued hypothesis class. Setting the regularization as $\Omega_{\mathcal{H}_\mathsf{K}}(h) := \|h\|^2_{\mathcal{H}_\mathsf{K}}$ yields the following:

$$\min_{h \in \mathcal{H}_\mathsf{K}} \frac{1}{n} \sum_{i=1}^n L(y_i, \Phi h(x_i)) + \lambda \|h\|^2_{\mathcal{H}_\mathsf{K}}. \tag{5.10}$$

To solve Problem 5.10, we highlight in Corollary 5.5 that it benefits from a representer theorem. Therefore, it can then be restated as a problem with nd variables.

Corollary 5.5 (Representer theorem)**.** *For any solution $h^\lambda_\mathbf{z}$ to Problem 5.10, there exists $\boldsymbol{\alpha} \in (\mathbb{R}^d)^n$ such that*

$$h^\lambda_\mathbf{z} = \sum_{j=1}^n \mathsf{K}_{x_j} \alpha_j. \tag{5.11}$$

Proof This is a direct application of the representer theorem stated in Theorem 2.36 setting for $\mathbf{u} \in (\mathbb{R}^d)^n$, $V(\mathbf{u}, t) = \frac{1}{n}\sum_{i=1}^n L(y_i, \Phi u_i)) + \lambda t^2$, which is indeed always strictly increasing with respect to t regardless of $\mathbf{u}$. ■

Proposition 5.6 (Existence and uniqueness of minimizer)**.** *Suppose that*

1. *for all $y \in \mathcal{Y}$, $L(y,.) \in \Gamma_0(\mathcal{Y})$ (see Definition 2.42), and*
2. K *is bounded: there exists $\kappa \geq 0$ such that for all $x \in \mathcal{X}$, $\|\mathsf{K}(x,x)\|_{\mathcal{L}(\mathbb{R}^d)} \leq \kappa$.*

Then a minimizer $h^\lambda_{\mathbf{z}}$ of Problem 5.10 exists and it is unique.

Proof Let us first prove that all terms in the objective are in $\Gamma_0(\mathcal{H}_\mathsf{K})$. Since K is bounded, for all $x \in \mathcal{X}$, $\mathsf{K}^\#_x$ is bounded as a consequence of Lemma 6.2 (we introduce this Lemma further down in the theoretical part because we use it extensively there, therefore as it is not central, we do not re-expose it here). Φ is also bounded as highlighted in Remark 5.3. Consequently by stability of the set $\Gamma_0(\mathcal{H}_\mathsf{K})$ with respect to the composition with bounded linear operators (Bauschke and Combettes, 2017, Proposition 9.5), for all $y \in \mathcal{Y}$ and $x \in \mathcal{X}$, $h \mapsto L(y, \Phi\mathsf{K}^\#_x h)$ is in $\Gamma_0(\mathcal{H}_\mathsf{K})$. Thus the objective in Problem 5.10 is in $\Gamma_0(\mathcal{H}_\mathsf{K})$.

Now, remark that $h \mapsto \lambda\|h\|^2_{\mathcal{H}_\mathsf{K}}$ is supercoercive. Indeed $\frac{\|h\|^2_{\mathcal{H}_\mathsf{K}}}{\|h\|_{\mathcal{H}_\mathsf{K}}} = \|h\|_{\mathcal{H}_\mathsf{K}}$ which limit is obviously $+\infty$ when $\|h\|_{\mathcal{H}_\mathsf{K}} \to +\infty$.

Consequently, from Bauschke and Combettes (2017, Corollary 11.16), the objective in Problem 5.10 is coercive and therefore it has a minimizer over $\mathcal{H}_\mathsf{K}$ Moreover, since $h \mapsto \lambda\|h\|^2_{\mathcal{H}_\mathsf{K}}$ is strictly convex, this minimizer is unique. ■

Remark 5.7 (Difference with composition of an OVK with the map Φ)**.** *It is possible to combine an OVK with a linear map (Carmeli et al., 2010) to define a new OVK . Therefore, the $\mathcal{L}(\mathcal{Y})$-valued OVK defined as*

$$\mathsf{K}_\Phi : (x_1, x_2) \mapsto \Phi\mathsf{K}(x_1, x_2)\Phi^\#, \tag{5.12}$$

is valid. Yet it is not the same thing to solve

$$\min_{g \in \mathcal{H}_{\mathsf{K}_\Phi}} \frac{1}{n}\sum_{i=1}^n L(y_i, g(x_i)) + \lambda\|g\|^2_{\mathcal{H}_{\mathsf{K}_\Phi}}, \tag{5.13}$$

and Problem 5.10. Indeed, the representer theorem applies, but it yields a solution of the form

$$g^\lambda_{\mathbf{z}} = \sum_{j=1}^n \Phi\mathsf{K}_{x_j}\Phi^\#\xi_j,$$

where for all $j \in [\![n]\!]$, $\xi_j \in \mathcal{Y}$. To avoid optimizing over $\mathcal{Y}$, a change of variable $\zeta_j = \Phi^\#\xi_j$ can be done, nevertheless, the search must be restricted to $\mathrm{Im}(\Phi^\#)$ if we want to recover the real solution to Problem 5.13. Yet, we have no way to numerically optimize over this space. In any case, Problem 5.13 and Problem 5.10 are not equivalent.

Remark 5.8 (Low-rank function-valued learning in structured output prediction)**.** *Another approach exists to perform function-valued regression in a low-dimensional subspace of a Hilbert space $\mathcal{Y}$. Given a fitted estimator $\hat{g}$, Brogat-Motte et al. (2023) search for the best subspace (of a given dimension d) to project this estimator. More precisely, they seek the orthogonal projection P minimizing $\frac{1}{n}\sum_{i=1}^n \|\mathrm{P}\hat{g} - \hat{g}\|_\mathcal{Y}$. In practice this amounts*

to projecting onto the span of the d eigenvectors of the empirical covariance operator of $\hat{g}$ associated to its d largest eigenvalues. This approach is different from ours as it is a post hoc one. First a $\mathcal{Y}$-valued estimator is fitted and second it is projected. Therefore this works only if a theoretical closed-form is known for the estimator, consequently the authors focus on the $\mathcal{Y}$-valued kernel ridge regression for which they propose a very efficient procedure. In KPL, we first choose an approximation dictionary that should represent well the outputs and second we solve the problem.

5.2.2 Ridge estimator with outputs in a separable Hilbert space

We now derive a closed-form solution for Problem 5.10 leveraging the representation given in Equation (5.11). To that end, let us endow the space $\mathcal{Y}^n$ with the natural scalar product defined for $\mathbf{y}, \mathbf{y}' \in \mathcal{Y}^n$ as

$$\langle \mathbf{y}, \mathbf{y}' \rangle_{\mathcal{Y}^n} = \sum_{i=1}^{n} \langle y_i, y_i' \rangle_{\mathcal{Y}},$$

so that it is a Hilbert space.

We now highlight some basic facts and introduce (or recall) some notations so as to derive this closed-form smoothly. Let us first define the linear operator K associated with the OVK K and the input observations $(x_i)_{i=1}^n$.

$$\mathsf{K} \colon \begin{pmatrix} \mathbb{R}^{d\times n} & \to \mathbb{R}^{d\times n} \\ \boldsymbol{\alpha} & \mapsto \left[\sum_{j=1}^{n} \mathsf{K}(x_i, x_j)\alpha_j\right]_{i=1}^{n} \end{pmatrix}, \tag{5.14}$$

where we recall that $\left[\sum_{j=1}^{n} \mathsf{K}(x_i, x_j)\alpha_j\right]_{i=1}^{n}$ is the matrix whose i-th column is $\sum_{j=1}^{n} \mathsf{K}(x_i, x_j)\alpha_j$.

Remark 5.9. *This operator is one of the possible counterparts of the kernel matrix for OVKs. Another way to proceed is to use a kernel block matrix K which can be represented as*

$$\begin{pmatrix} \mathsf{K}(x_1, x_1) & \cdots & \mathsf{K}(x_1, x_n) \\ \vdots & & \vdots \\ \mathsf{K}(x_n, x_1) & \cdots & \mathsf{K}(x_n, x_n) \end{pmatrix} \in \mathbb{R}^{nd\times nd}, \tag{5.15}$$

and which therefore acts on a large vector in $\mathbb{R}^{dn}$ consisting of the concatenation of the columns of the input matrix. However, in our case, it is more practical to remain in the space of matrices ($\mathbb{R}^{d\times n}$) because of the projection step occurring afterwards.

Remark 5.10. *We recall that from Definition 2.25, for all $i, j \in [\![n]\!]$, we have $\mathsf{K}(x_i, x_j) = \mathsf{K}(x_j, x_i)^{\mathrm{T}}$. From there it is easy to see that $\mathsf{K}^{\#} = \mathsf{K}$ with respect to the scalar product $\langle \cdot, \cdot \rangle_{\mathbb{R}^{d\times n}}$. Indeed for $\boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathbb{R}^{d\times n}$*

$$\begin{aligned} \langle \boldsymbol{\beta}, \mathsf{K}\boldsymbol{\alpha} \rangle_{\mathbb{R}^{d\times n}} &= \sum_{i=1}^{n}\sum_{j=1}^{n} \langle \beta_i, \mathsf{K}(x_i, x_j)\alpha_j \rangle_{\mathbb{R}^d} \\ &= \sum_{i=1}^{n}\sum_{j=1}^{n} \langle \mathsf{K}(x_j, x_i)\beta_i, \alpha_j \rangle_{\mathbb{R}^d} \\ &= \langle \mathsf{K}\boldsymbol{\beta}, \boldsymbol{\alpha} \rangle_{\mathbb{R}^{d\times n}}. \end{aligned}$$

Moreover, K *is positive as a consequence of the positive definiteness of* K *(Definition 2.25): for* $\boldsymbol{\alpha} \in \mathbb{R}^{d\times n}$,

$$\langle \boldsymbol{\alpha}, \mathsf{K}\boldsymbol{\alpha}\rangle_{\mathbb{R}^{d\times n}} = \sum_{i=1}^{n}\sum_{j=1}^{n}\langle \alpha_i, \mathsf{K}(x_i,x_j)\alpha_j\rangle_{\mathbb{R}^d} \geq 0.$$

Finally for $\boldsymbol{\beta} = (\beta_i)_{i=1}^n \in \mathcal{H}^n$ with $\mathcal{H}$ a Hilbert space and an operator $\mathrm{A} \in \mathcal{L}(\mathcal{H},\mathcal{K})$ where $\mathcal{K}$ is some Hilbert space, we recall the notation of the factorized operator $\mathrm{A}_{(n)}$

$$\mathrm{A}_{(n)} : \begin{pmatrix} \mathcal{H}^n & \to \mathcal{K}^n \\ \boldsymbol{\beta} & \mapsto (\mathrm{A}\beta_i)_{i=1}^n \end{pmatrix}. \tag{5.16}$$

Remark 5.11. *In the following, we have* $\mathcal{H} = \mathbb{R}^d$ *and* $\mathcal{K} = \mathcal{Y}$*. We set the convention that in this case, we assimilate the space* $(\mathbb{R}^d)^n$ *to the space of matrices* $\mathbb{R}^{d\times n}$ *and therefore the factorized operator* $\mathrm{A}_{(n)}$ *applies* A *to the columns of the input matrix* $\boldsymbol{\beta}$*.*

Remark 5.12. *It is also worth noting that the action of* $(\mathrm{A}^\#\mathrm{A})_{(n)}$ *is actually the same as that of the matrix* $(\mathrm{A}^\#\mathrm{A})$*. Indeed, by definition of the matrix product for* $\boldsymbol{\alpha} \in \mathbb{R}^{d\times n}$,

$$(\mathrm{A}^\#\mathrm{A})\boldsymbol{\alpha} = \left[(\mathrm{A}^\#\mathrm{A})\alpha_i\right]_{i=1}^n.$$

However, for instance for notions such as the rank, there are differences between $(\mathrm{A}^\#\mathrm{A})$ *taken as an operator in* $\mathcal{L}(\mathbb{R}^d)$ *and* $\mathrm{A}^\#\mathrm{A}$ *as an operator in* $\mathcal{L}(\mathbb{R}^{d\times n})$*. Consequently, we keep the notation* $(\mathrm{A}^\#\mathrm{A})_{(n)}$ *to avoid confusions.*

Putting it all together, when considering the square loss, Problem 5.8 can be rewritten as

$$\min_{\boldsymbol{\alpha}\in\mathbb{R}^{d\times n}} \frac{1}{n}\|\mathbf{y} - \Phi_{(n)}\mathsf{K}\boldsymbol{\alpha}\|_{\mathcal{Y}^n}^2 + \lambda\langle \boldsymbol{\alpha}, \mathsf{K}\boldsymbol{\alpha}\rangle_{\mathbb{R}^{d\times n}}. \tag{5.17}$$

We are now ready to derive a closed-form solution for Problem 5.17.

Proposition 5.13 (Ridge estimator)**.** *The minimum in Problem 5.17 is achieved by any* $\hat{\boldsymbol{\alpha}} \in \mathbb{R}^{d\times n}$ *verifying*

$$(\mathsf{K}(\Phi^\#\Phi)_{(n)}\mathsf{K} + n\lambda\mathsf{K})\hat{\boldsymbol{\alpha}} = \mathsf{K}\Phi_{(n)}^\#\mathbf{y}. \tag{5.18}$$

Moreover if K *is full rank then* $((\Phi^\#\Phi)_{(n)}\mathsf{K} + n\lambda\mathbf{I})$ *is invertible and* $\hat{\boldsymbol{\alpha}}$ *is such that*

$$\hat{\boldsymbol{\alpha}} = ((\Phi^\#\Phi)_{(n)}\mathsf{K} + n\lambda\mathbf{I})^{-1}\Phi_{(n)}^\#\mathbf{y}. \tag{5.19}$$

We define the ridge estimator as

$$h_{\mathbf{z}}^\lambda := \sum_{j=1}^{n}\mathsf{K}_{x_j}\hat{\alpha}_j. \tag{5.20}$$

Proof Let V be the objective in Problem 5.17, since it is convex the minimizer $\boldsymbol{\alpha}$ must cancel the gradient (we recall that this minimizer exists and is unique, the square loss is indeed proper, convex and continuous and therefore provided K is bounded, Proposition 5.6 applies). The gradient is given by

$$\nabla V(\boldsymbol{\alpha}) = -\frac{2}{n}\mathsf{K}^{\#}(\Phi_{(n)})^{\#}\mathbf{y} + \frac{2}{n}\mathsf{K}^{\#}\Phi_{(n)}^{\#}\Phi_{(n)}\mathsf{K}\boldsymbol{\alpha} + 2\lambda\mathsf{K}\boldsymbol{\alpha}.$$

Using that $(\Phi_{(n)})^{\#}\Phi_{(n)} = \Phi_{(n)}^{\#}\Phi_{(n)} = (\Phi^{\#}\Phi)_{(n)}$ and that K is self-adjoint, canceling the gradient yields

$$(\mathsf{K}(\Phi^{\#}\Phi)_{(n)}\mathsf{K} + n\lambda\mathsf{K})\hat{\boldsymbol{\alpha}} = \mathsf{K}\Phi_{(n)}^{\#}\mathbf{y},$$

which corresponds to Equation (5.18).

Now let us assume that K is full-rank. We can then simplify by K^{-1} on the left to obtain the equivalent system

$$((\Phi^{\#}\Phi)_{(n)}\mathsf{K} + n\lambda\mathbf{I})\hat{\boldsymbol{\alpha}} = \Phi_{(n)}^{\#}\mathbf{y}.$$

Let us now show that $(\Phi^{\#}\Phi)_{(n)}\mathsf{K} + n\lambda\mathbf{I}$ is invertible. We have that K is a strictly positive and self-adjoint operator on a finite dimensional space. Therefore $\mathsf{K}(\Phi^{\#}\Phi)_{(n)}\mathsf{K}$ is a positive self-adjoint operator, which implies that $\mathsf{K}(\Phi^{\#}\Phi)_{(n)}\mathsf{K} + n\lambda\mathsf{K}$ is strictly positive and self-adjoint and it is defined on a finite-dimensional space which implies its invertibility.

To finish with, $(\Phi^{\#}\Phi)_{(n)}\mathsf{K} + n\lambda\mathbf{I} = \mathsf{K}^{-1}(\mathsf{K}(\Phi^{\#}\Phi)_{(n)}\mathsf{K} + n\lambda\mathsf{K})$, therefore it is a composition of invertible operators, and consequently, it is itself invertible. ■

Remark 5.14. *$\Phi^{\#}\Phi$ is the Gram matrix the dictionary, therefore if $(\phi_l)_{l=1}^d$ is orthonormal, Equation (5.19) simplifies to*

$$\hat{\boldsymbol{\alpha}} = (\mathsf{K} + n\lambda\mathbf{I})^{-1}\Phi_{(n)}^{\#}\mathbf{y}.$$

Remark 5.15. *We highlight that the quantity $\Phi_{(n)}^{\#}\mathbf{y}$ actually corresponds to the matrix of pairwise scalar products between the observed functions and the elements of the dictionary. Indeed,*

$$\Phi_{(n)}^{\#}\mathbf{y} = \left[\Phi^{\#}y_i\right]_{i=1}^n \in \mathbb{R}^{d\times n},$$

where $[\Phi^{\#}y_i]_{i=1}^n$ is the matrix whose i-th column is $\Phi^{\#}y_i$. Additionally, we have from Equation (5.6) that for $i \in [\![n]\!]$,

$$\Phi^{\#}y_i = \left(\langle y_i, \phi_l\rangle_{\mathcal{Y}}\right)_{l=1}^d.$$

In practice, solving the linear system in Equation (5.18) has time complexity $\mathcal{O}(n^3d^3)$ which is far too high, even to tackle learning problems of medium size. Among OVKs, separable ones are a very popular subclass, for they are simple to interpret and can drastically simplify computations in some cases. Let us then suppose that K is separable (Definition 2.32), therefore there exists a positive symmetric matrix $\mathrm{B} \in \mathbb{R}^{d\times d}$ and a scalar-valued kernel $k_{\mathcal{X}} : \mathcal{X}\times\mathcal{X} \to \mathbb{R}$ such that

$$\forall x_1, x_2 \in \mathcal{X}, \quad \mathsf{K}(x_1, x_2) = k_{\mathcal{X}}(x_1, x_2)\mathrm{B}.$$

In that case Equation (5.19) is equivalent to the linear matrix equation

$$(\Phi^{\#}\Phi)\mathrm{B}\boldsymbol{\alpha}\mathrm{K}_{\mathcal{X}} + n\lambda\boldsymbol{\alpha} = \Phi^{\#}_{(n)}\mathbf{y}. \tag{5.21}$$

Two classic resolution strategies can then be used

- Equation (5.21) is a discrete time Sylvester equation (Dinuzzo et al., 2011) for which efficient solvers exist (Sima, 1996), more precisely, such equation can be solved in $\mathcal{O}(n^3 + d^3 + n^2 d + nd^2) \asymp \mathcal{O}(n^3 + d^3)$ time.

- Or if we wish to test many values of λ, another strategy is to vectorize the equation. It then exhibits a Kronecker product structure which can be exploited to compute an eigendecomposition of the matrix of interest. More precisely, we can use the fact that $\mathrm{vec}((\Phi^{\#}\Phi)\mathrm{B}\boldsymbol{\alpha}\mathrm{K}_{\mathcal{X}}) = \mathrm{K}_{\mathcal{X}}^{\mathrm{T}} \otimes ((\Phi^{\#}\Phi)\mathrm{B})\,\mathrm{vec}(\boldsymbol{\alpha}) = \mathrm{K}_{\mathcal{X}} \otimes ((\Phi^{\#}\Phi)\mathrm{B})\,\mathrm{vec}(\boldsymbol{\alpha})$. Therefore, Equation (5.21) is equivalent to

  $$(\mathrm{K}_{\mathcal{X}} \otimes ((\Phi^{\#}\Phi)\mathrm{B}) + n\lambda \mathrm{I}_{nd})\,\mathrm{vec}(\boldsymbol{\alpha}) = \mathrm{vec}(\Phi^{\#}_{(n)}\mathbf{y}). \tag{5.22}$$

  Then an eigendecomposition of $\mathrm{K}_{\mathcal{X}} \otimes ((\Phi^{\#}\Phi)\mathrm{B})$ can be deduced from one of $\mathrm{K}_{\mathcal{X}}$ and one of $(\Phi^{\#}\Phi)\mathrm{B}$ (Horn and Johnson, 1991, Theorem 4.2.12) in $\mathcal{O}(n^3 + d^3)$ time. Indeed, let $(\nu_i, \mathbf{v}_i)_{i=1}^n$ and $(\eta_l, \mathbf{w}_l)_{l=1}^d$ be the eigenvalue/eigenvector pairs respectively of $\mathrm{K}_{\mathcal{X}}$ and $(\Phi^{\#}\Phi)\mathrm{B}$. Then the eigenvalue/eigenvector pairs of $\mathrm{K}_{\mathcal{X}} \otimes ((\Phi^{\#}\Phi)\mathrm{B})$ are $(\nu_i\eta_l, \mathbf{v}_i \otimes \mathbf{w}_l)_{i,l=1}^{n,d}$.

Once we have found the representer coefficients $\hat{\boldsymbol{\alpha}}$ solving the system in Equation (5.22), the predicted function at a new input point $x \in \mathcal{X}$ is given by

$$\Phi\mathrm{B}\hat{\boldsymbol{\alpha}}\mathbf{k}_{\mathcal{X}}(x), \quad \text{with } \mathbf{k}_{\mathcal{X}}(x) := \left(k_{\mathcal{X}}(x, x_i)\right)_{i=1}^n. \tag{5.23}$$

Remark 5.16 (Other losses)**.** *For other losses, even if the kernel is separable, it is no longer possible to find a closed-form, however we can resort to iterative algorithms for optimization. The problem does benefit from the representer theorem and, in the case of a separable kernel, Problem 5.10 can be rewritten as*

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^{d \times n}} \frac{1}{n} \sum_{i=1}^n L_{y_i}\left(\Phi\mathrm{B}\boldsymbol{\alpha}\mathbf{k}_{\mathcal{X}}(x_i)\right) + \lambda \langle \mathrm{K}_{\mathcal{X}}, \boldsymbol{\alpha}^{\mathrm{T}}\mathrm{B}\boldsymbol{\alpha}\rangle_{\mathbb{R}^{n \times n}}.$$

We have exploited the separability of the kernel at least to avoid forming the block operator matrix associated to K*. It lies in* $\mathbb{R}^{dn \times dn}$ *and therefore performing matrix products with such big matrices would be prohibitive. The gradient with respect to the coefficients* $\boldsymbol{\alpha}$ *is given by*

$$\frac{1}{n}\mathrm{B}\mathrm{G}(\boldsymbol{\alpha})\mathrm{K}_{\mathcal{X}} + \lambda \mathrm{B}\boldsymbol{\alpha}\mathrm{K}_{\mathcal{X}}, \tag{5.24}$$

where $\mathrm{G}(\boldsymbol{\alpha})$ *is defined column-wise as*

$$\mathrm{G}(\boldsymbol{\alpha}) := \left[\Phi^{\#}\nabla L_{y_i}(\Phi\mathrm{B}\boldsymbol{\alpha}\mathbf{k}_{\mathcal{X}}(x_i))\right]_{i=1}^n \in \mathbb{R}^{d \times n}. \tag{5.25}$$

5.2.3 Functional case with partially-observed functions

In the particular case where $\mathcal{Y} = \mathsf{L}^2(\Theta)$ for some $\Theta \subset \mathbb{R}^b$ a compact subset, this corresponds to the functional data setting introduced in Section 3.1. We highlighted that typically, the output functions $(y_i)_{i=1}^n$ are not observed as functions but rather through discrete and possibly noisy evaluations of those functions on the domain Θ.

Therefore, we suppose that we only observe each y_i at a set of locations $\boldsymbol{\theta}_i := (\theta_{is})_{s=1}^{m_i} \in \Theta^{m_i}$. The learning problem depicted in Problem 5.17 has now to be solved using a partially observed functional output sample:

$$\tilde{\mathbf{z}} := (x_i, (\boldsymbol{\theta}_i, \tilde{y}_i))_{i=1}^n, \tag{5.26}$$

where for all $i \in [\![n]\!]$, $\boldsymbol{\theta}_i \in \Theta^{m_i}$, $\tilde{y}_i \in \mathbb{R}^{m_i}$ with $m_i \in \mathbb{N}^*$ the number of observations available for the i-th function, and for all $s \in [\![m_i]\!]$, $\theta_{is} \in \Theta$ and $\tilde{y}_{is} \in \mathbb{R}$.

Then if the observations from Equation (5.26) enable us to compute reasonably good estimators of the pairwise scalar products $\left(\langle y_i, \phi_l \rangle_{\mathcal{Y}}\right)_{l=1,i=1}^{d,n}$, we can plug-in these estimators into the closed-form solution Equation (5.19). Let $\tilde{\boldsymbol{\nu}} \in \mathbb{R}^{d \times n}$ be a matrix stacking estimators for these scalar products. For instance, an empirical mean can be used

$$\tilde{\nu}_{li} = \frac{1}{m_i} \sum_{s=1}^{m_i} \tilde{y}_{is} \phi_l(\theta_{is}). \tag{5.27}$$

Remark 5.17. *This problem of estimating scalar products naturally arises when using a dictionary to represent functions while only partial observations are available. Notably, it did appear in the problem of smoothing using a dictionary in Chapter 3.*

So as to highlight the nature of the two approximations that we make in defining our estimator, we recall the definition of the approximated projection operator for the set of locations $\boldsymbol{\theta}_i \in \Theta^{m_i}$ (introduced in Chapter 3)

$$\tilde{\Phi}_i : \begin{pmatrix} \mathbb{R}^d & \to & \mathbb{R}^{m_i} \\ \mathbf{a} & \mapsto & \sum_{l=1}^d a_l \phi_l(\boldsymbol{\theta}_i) \end{pmatrix}, \tag{5.28}$$

and that of its adjoint in the Euclidian space $\mathbb{R}^{m_i}$:

$$\tilde{\Phi}_i^{\#} : \begin{pmatrix} \mathbb{R}^{m_i} & \to & \mathbb{R}^d \\ \tilde{y} & \mapsto & (\langle \tilde{y}, \phi_l(\boldsymbol{\theta}_i) \rangle_{\mathbb{R}^{m_i}})_{l=1}^d \end{pmatrix},$$

where we have used the convention that $\phi_l(\boldsymbol{\theta}_i) = \left(\phi_l(\theta_{is})\right)_{s=1}^{m_i}$.

Next we define the plug-in ridge estimator using the empirical mean to estimate the scalar products. To be valid, this estimator requires that enough observations are available so that the two following approximations hold for all $i \in [\![n]\!]$,

$$\begin{aligned} \frac{1}{m_i} \tilde{\Phi}_i^{\#} \tilde{y}_i &\approx \Phi^{\#} y_i \\ \frac{1}{m_i} \tilde{\Phi}_i^{\#} \tilde{\Phi}_i &\approx \Phi^{\#} \Phi. \end{aligned} \tag{5.29}$$

When these are reasonable, we can use the following estimator.

Definition 5.18 (Plug-in ridge estimator)**.** *Let $\tilde{\boldsymbol{\nu}}$ be the matrix whose entries are the empirical mean estimates of the scalar products–Equation (5.27). Suppose that K is full rank and let $\tilde{\boldsymbol{\alpha}} \in \mathbb{R}^{n\times d}$ be such that*

$$\tilde{\boldsymbol{\alpha}} = ((\Phi^{\#}\Phi)_{(n)}\mathsf{K} + n\lambda\mathbf{I})^{-1}\tilde{\boldsymbol{\nu}}. \tag{5.30}$$

We then define the plug-in ridge estimator as

$$h^{\lambda}_{\tilde{\mathbf{z}}} := \sum_{j=1}^{n} \mathsf{K}_{x_j}\tilde{\alpha}_j. \tag{5.31}$$

Exploiting the separability of the kernel as in Equation (5.22), the coefficients of the plug-in ridge estimator are found solving

$$(\mathrm{K}_{\mathcal{X}} \otimes ((\Phi^{\#}\Phi)\mathrm{B}) + n\lambda\mathbf{I})\,\mathrm{vec}(\boldsymbol{\alpha}) = \mathrm{vec}(\tilde{\boldsymbol{\nu}}). \tag{5.32}$$

As highlighted earlier, the time complexity of solving such a system can be reduced to $\mathcal{O}(n^3 + d^3)$.

When those approximations do not hold, we can no longer exploit the separability. To see this, let us formulate an analogous to Problem 5.10 with the square loss approximated using the available observations of the functions to approximate the square loss. This problem reads

$$\min_{h\in\mathcal{H}_{\mathsf{K}}} \frac{1}{n}\sum_{i=1}^{n} \left\| \frac{\tilde{y}_i}{\sqrt{m_i}} - \frac{\tilde{\Phi}_i}{\sqrt{m_i}} h(x_i) \right\|^2_{\mathbb{R}^{m_i}} + \lambda\|h\|^2_{\mathcal{H}_{\mathsf{K}}}. \tag{5.33}$$

Let us define the approximation $\tilde{\boldsymbol{\Phi}}$ of $\Phi_{(n)}$ using the discrete observations. It acts on the columns of a matrix $\boldsymbol{\beta} \in \mathbb{R}^{d\times n}$ in the following way

$$\tilde{\boldsymbol{\Phi}} : \begin{pmatrix} \mathbb{R}^{d\times n} & \to & \prod_{i=1}^{n}\mathbb{R}^{m_i} \\ \boldsymbol{\beta} & \mapsto & \left(\frac{\tilde{\Phi}_i}{\sqrt{m_i}}\beta_i\right)_{i=1}^{n} \end{pmatrix}, \tag{5.34}$$

Problem 5.33 also benefits from a representer theorem in the same way as Problem 5.10, it is a direct Corollary of Theorem 2.36. We can therefore reformulate it as

$$\min_{\boldsymbol{\alpha}\in\mathbb{R}^{d\times n}} \frac{1}{n}\|\tilde{\mathbf{y}} - \tilde{\boldsymbol{\Phi}}\mathsf{K}\boldsymbol{\alpha}\|^2_{\prod_{i=1}^{n}\mathbb{R}^{m_i}} + \lambda\langle\boldsymbol{\alpha}, \mathsf{K}\boldsymbol{\alpha}\rangle_{\mathbb{R}^{d\times n}}.$$

Carrying the same steps as in the proof of Proposition 5.13, we obtain that $\boldsymbol{\alpha}$ is solution to the linear system

$$((\tilde{\boldsymbol{\Phi}}^{\#}\tilde{\boldsymbol{\Phi}})\mathsf{K} + n\lambda\mathbf{I})\boldsymbol{\alpha} = \tilde{\boldsymbol{\Phi}}^{\#}\tilde{\mathbf{y}}. \tag{5.35}$$

We highlight that $\tilde{\mathbf{\Phi}}^{\#}\tilde{\mathbf{y}}$ is the matrix of scalar products estimated through the empirical mean at the available locations (it is equal to $\tilde{\boldsymbol{\nu}}$). Note also that the operator $\tilde{\mathbf{\Phi}}^{\#}\tilde{\mathbf{\Phi}} \in \mathcal{L}(\mathbb{R}^{d\times n})$ is given by

$$\tilde{\mathbf{\Phi}}^{\#}\tilde{\mathbf{\Phi}} : \begin{pmatrix} \mathbb{R}^{d\times n} & \to & \mathbb{R}^{d\times n} \\ \boldsymbol{\beta} & \mapsto & \left[\frac{\tilde{\Phi}_i^{\#}\tilde{\Phi}_i}{m_i}\beta_i\right]_{i=1}^{n} \end{pmatrix},$$

Therefore, even when the kernel is separable, the linear system in Equation (5.35) cannot be solved efficiently because we can no longer exploit the separability as we do to solve Equation (5.22). This is because instead of using the matrix $\Phi^{\#}\Phi$ for all $i \in [\![n]\!]$, we use a different estimate $\frac{\tilde{\Phi}_i^{\#}\tilde{\Phi}_i}{m_i}$ for each $i \in [\![n]\!]$.

Remark 5.19 (Same locations)**.** *A possible case however where we can have very sparse observations for the functions and still retain the computational efficiency is when we observe all the functions at the same locations. In that case the estimation of the Gram matrix is common to all the output functions and we can exploit the separability.*

Example 5.20 (Integral losses and gradient estimation)**.** *Following up on Remark 5.16, we highlight that, when considering another loss than the square loss, it is possible to estimate the gradient in Equation (5.24) from partial observations as well. For instance considering an integral loss of the form*

$$L : (y_1, y_2) \mapsto \int_{\Theta} \ell(y_1(\theta), y_2(\theta))\mathrm{d}\theta, \tag{5.36}$$

where ℓ is a real-valued convex loss on $\mathbb{R}$. Indeed, the gradient of such loss is given by

$$\nabla L_{y_i} : y \longmapsto \Big(\theta \longmapsto \ell(y_i(\theta), y(\theta))\Big).$$

Therefore for $i \in [\![n]\!]$, the i-th column of $\mathrm{G}(\boldsymbol{\alpha})$ which is $\Phi^{\#}\nabla L_{y_i}(\Phi \mathrm{B}\boldsymbol{\alpha}\boldsymbol{k}_{\mathcal{X}}(x_i))$ can be estimated, for instance through the empirical mean as

$$\frac{1}{m_i}\sum_{s=1}^{m_i} \ell\Big(y_i(\theta_{is}), \boldsymbol{\phi}(\theta_{is})^{\mathrm{T}}\mathrm{B}\boldsymbol{\alpha}\boldsymbol{k}_{\mathcal{X}}(x_i)\Big)\boldsymbol{\phi}(\theta_{is}), \tag{5.37}$$

where we have used the convention that for $\theta \in \Theta$, $\boldsymbol{\phi}(\theta) = \left((\phi_l(\theta))_{l=1}^{d}\right)^{\mathrm{T}} \in \mathbb{R}^d$.

Note that, as opposed to the strategy to compute the plug-in ridge estimator in Equation (5.32), when L is not the square loss, we cannot exploit the separability as we must optimize over all the rows of $\boldsymbol{\alpha}$ jointly.

Now that we have introduced the kernel projection learning framework along with some corresponding estimators, we propose to study those empirically, and compare them to existing nonlinear FOR methods.

5.3 Numerical experiments

In this section, we study empirically our proposed estimators as well as the main existing nonlinear FOR methods presented in Chapter 4. In Section 5.3.1 we give some details on the different estimators and we introduce the metrics used in the experiments. Then we benchmark all the methods on three function-to-function problems. In Section 5.3.2, using a synthetic dataset, we investigate how the estimators react to several kinds of corruption of the output functions. In order to compare the performances of the estimators on real datasets, in Section 5.3.3 we study a problem linked to medical imaging while in Section 5.3.4 is dedicated to comparisons on a synthetic speech inversion dataset.

5.3.1 Preliminary elements

So far we have introduced the kernel projection learning framework, and we have proposed several corresponding estimators. We recall what these are and give computation details. We also briefly recall the other nonlinear FOR methods against which we benchmark our estimators.

Estimators

We proposed two estimators in closed-form for the square loss based on a ridge type closed-form. The ridge estimator from Proposition 5.13 is the most general one, it can deal with outputs in a separable Hilbert space. Nevertheless, in the experiments, we rather use the plug-in ridge estimator introduced in Definition 5.18 as it works for FOR with partially observed functions. We refer to it as **ridge-plug-KPL** in the following sections. We have also shown how the separability of the OVK could be exploited to reduce the fitting time complexity to $\mathcal{O}(n^3 + d^3)$. We therefore use separable OVKs in the following and exploit this separability using a Sylvester equation solver.

We have also shown how gradient-based optimization could be implemented in the same setting of FOR with partial observations in Example 5.20 when using integral losses. In that case, we indeed can no longer benefit from the separability trick. As an example of integral loss, we propose to study KPL with the integral *logcosh loss* for $\nu > 0$

$$L^{(\nu)}_{\mathrm{lch}}(y_1, y_2) = \frac{1}{\nu} \int_\Theta \log(\cosh(\nu(y_1(\theta) - y_2(\theta))\mathrm{d}\theta. \tag{5.38}$$

The ground loss behaves similarly to the Huber loss (Huber, 1964), it is roughly quadratic around 0 and roughly linear elsewhere, except that it is $\mathcal{C}^\infty(\Theta^2)$ which is advantageous as we use second order method for optimization. The parameter ν gives us control on its behavior around 0, as it grows bigger, $\ell^{(\nu)}_{\mathrm{lch}}$ tends to the absolute loss. We illustrate this loss defined on $\mathbb{R}$ and $\mathbb{R}^2$ respectively in Figure 5.1 and Figure 5.2. We refer to the estimator resulting from solving Problem 5.10 using this estimated gradient from Example 5.20 and the integral logcosh loss from Equation (5.38) as **logcosh-KPL**.

It is also of interest to compare the plug-in ridge estimator which works well when the approximations in Equation (5.29) are reasonable, and the estimator which results from posing the KPL problem directly using the available observations as in Problem 5.33. We refer to that last estimator with the abbreviation **ridge-iter-KPL**.

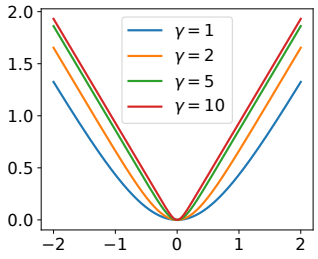


Figure 5.1: Logcosh loss on $\mathbb{R}$.

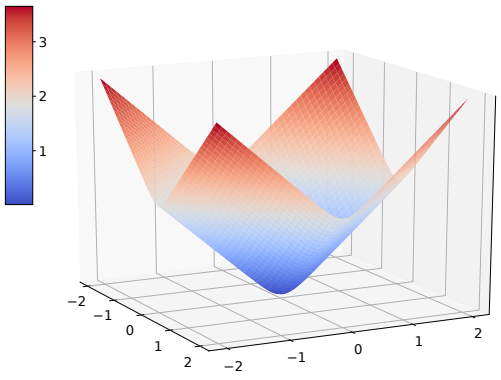


Figure 5.2: Logcosh loss on $\mathbb{R}^2$ ($\nu = 5$).

For both logcosh-KPL and ridge-iter-KPL, we use L-BFGS-B (Zhu et al., 1997a) as solver. We have found that its use of approximate second order information greatly improved convergence speed.

Finally, we benchmark the estimators introduced above against the main existing nonlinear FOR methods presented in Chapter 4.

- Functional kernel ridge regression (**FKRR**) corresponds to the kernel ridge regression using function-valued RKHSs (fv-RKHS) as hypothesis class (Lian, 2007; Kadri et al., 2010, 2016). We present this method in more detail in Section 4.2. For those experiments, we solve FKRR using the discretization approach presented in Section 4.2.2. We tested both approaches, (this one) and the eigendecomposition one (see Section 4.2.3) and found that the former was overall faster, because the latter require to compute Kronecker products large vectors (see the *time complexity* paragraph in Section 4.2.3) which constitutes a speed bottleneck. We use a Sylvester solver to solve the corresponding problem.

- Triple basis estimator (**3BE**) stands for the method which consists in projecting the input and the output functions on orthogonal bases and using an approximate kernel ridge regression with random Fourier features (RFF) to regress separately each output coefficient on the input ones. This is the method proposed in Oliva et al. (2015), we describe it in more detail in Section 4.3.

- Kernel additive model (**KAM**) corresponds to the kernelized version of the functional linear additive model proposed in Reimherr et al. (2018). We give more details in Section 4.4. We use a separable kernel in the sense of Equation (4.20) so that the model is computationally tractable, and exploit the Kronecker structure we pointed out in Equation (4.23) (a possibility which we recall was not highlighted by the authors).

- Kernel estimator (**KE**) corresponds to an extension of the Nadaraya-Watson estimator for functional output regression (Ferraty et al., 2011). We do not include it in all benchmarks as its performances are overall quite far from those of the other estimators.

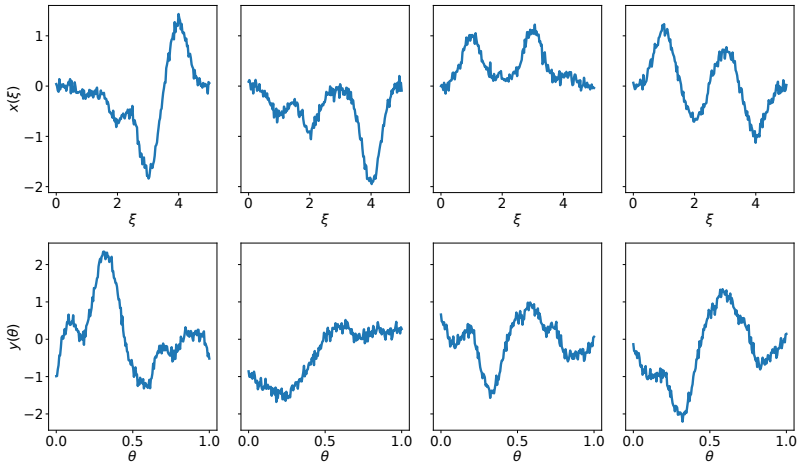


Figure 5.3: Examples from the synthetic dataset.

Metrics

We now introduce metrics we use to measure the performances of the estimators or to quantify the level of corruption. Given observed functions $(\boldsymbol{\theta}_i, \tilde{y}_i)_{i=1}^n$ and predicted ones $(\hat{y}_i)_{i=1}^n \in \mathsf{L}^2(\Theta)$, we define the mean square error (MSE) as

$$\mathrm{MSE} := \frac{1}{n}\sum_{i=1}^{n}\sum_{s=1}^{m_i}(\hat{y}_i(\theta_{is}) - \tilde{y}_{is})^2.$$

On the synthetic dataset, in one experiment, we add noise to the output functions. In order to measure the level of corruption we use the signal to noise ratio (SNR); for a noise level σ we define it as

$$\mathrm{SNR} := \frac{1}{\sigma n}\sum_{i=1}^{n}\frac{1}{m_i}\sum_{s=1}^{m_i}\left|\tilde{y}_{is}\right|.$$

5.3.2 Synthetic data

We propose to validate the efficiency of the KPL estimators on a function-to-function synthetic dataset. More precisely, we look at how different kinds of contamination of the output functions affect the estimators. We now describe the generation process.

Experimental setting

Generation process. We draw $r \in \mathbb{N}$ independent zero mean Gaussian processes (GP) with Gaussian covariance functions. More precisely, for $t \in [\![r]\!]$ the GP V_t has covariance $(\theta_1, \theta_2) \mapsto \exp\Big(-\frac{(\theta_2-\theta_1)^2}{b_t^2}\Big)$. We then keep these GPs fixed. In practice, we take $r = 4$ and $b_1 = 0.1$, $b_2 = 0.25$, $b_3 = 0.1$ and $b_4 = 0.25$. To generate an input/output pair, we draw r coefficients $\mathbf{a} \in \mathbb{R}^r$ i.i.d. according to a uniform distribution $\mathcal{U}\big([-1,1]\big)$. Let B_4 denote the cardinal cubic B-spline (de Boor, 2001); it is symmetric around $\xi = 2$ and of width 4. Let then $\bar{B}_4 : \xi \mapsto B_4(4\xi + 2)$ (a centered version of B_4 rescaled to have width 1). We consider the input function $x(\xi) := \sum_{t=1}^{r} a_t \bar{B}_4(\xi - t)$ with $\xi \in [0, 5]$. To

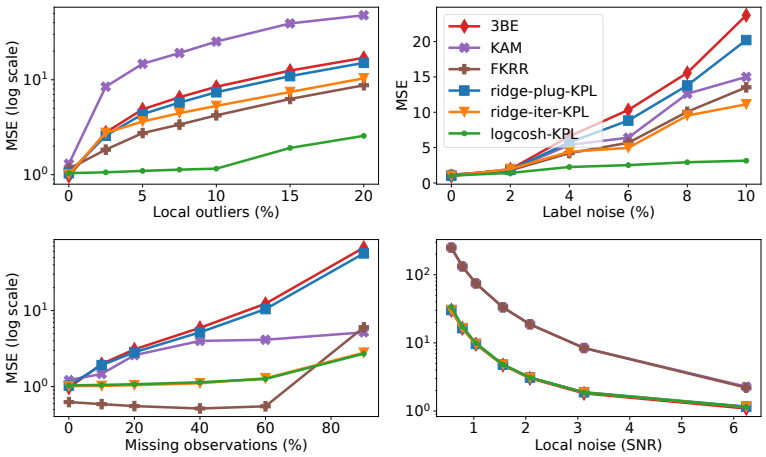


Figure 5.4: Comparison of KPL estimators with other nonlinear FOR methods for different types of corruption.

it we associate the output function $y(\theta) = \sum_{t=1}^{r} a_t V_t(\theta)$ with $\theta \in [0,1]$. In practice, we observe x and y on regular grids of size 200. For the experiments with missing data, we remove sampling points from these grids. Finally we add Gaussian noise on the input observations with standard deviation $\sigma_x = 0.07$ in all experiments. Examples of data generated that way with a Gaussian noise of standard deviation $\sigma_y = 0.1$ are shown in Figure 5.3.

Corruption modalities. We study the effect of four types of corruptions of the training data: local outliers, label noise, missing observations and local noise. In the first case, observations from the output functions are replaced with random draws in their range. To precise that notion, let $a = \min_{i\in[\![n]\!],s\in[\![m_i]\!]} \tilde{y}_{is}$ and let $b = \max_{i\in[\![n]\!],s\in[\![m_i]\!]} \tilde{y}_{is}$, then we draw the local outliers uniformly in the interval $[a,b]$. In the second case, some output functions are replaced with erroneous ones. More precisely, consider a portion $\tau \in [0,1]$ of contaminated output functions. Then, we draw uniformly at random $\lfloor \tau n \rfloor$ indices from $[\![n]\!]$. Among that set of indices, we randomly swap the output functions. In the third case we remove observations from the output functions uniformly at random. Finally, in the last one we add Gaussian noise to these observations.

Procedure and parameters. In this set of experiments, we compute the means over 10 runs with different train/test split. For all methods, we chose through cross-validation the regularization parameter, the parameter(s) of the kernel and the dictionary if relevant. For KPL estimators, we take $\mathsf{K} = k\mathrm{I}$ with k a scalar-valued Gaussian kernel and use a truncated Fourier basis. For 3BE, we use truncated Fourier bases as both input and output dictionaries and a Gaussian kernel. For KAM, we use a separable product of three Gaussian kernels (see Section 4.4.2 for more information on the particular kernel used for this method). Finally for FKRR, we use a Gaussian kernel as input kernel and Laplacian kernel as output kernel. To go further, note that we detail further all the parameters considered as well as the experimental procedure in Appendix A.1.

Comments on the results

The evolution of the MSEs for several levels of corruption are displayed in Figure 5.4. Looking at the two top panels, we observe that logcosh-KPL is robust to both types of outliers. This is not surprising given that the logcosh loss mimics the absolute value

loss for large deviations whereas all the other benchmarked methods are based on the quadratic loss which is known to be sensitive to outliers. This robustness comes however at the price of a higher computational complexity–see Example 5.20. Another interesting point is that the ridge estimator stemming the problem posed using the available observations(ridge-iter-KPL, solution to Problem 5.33) is significantly more robust than the plug-in ridge estimator.

On the bottom left panel of Figure 5.4, we observe that FKRR is the most robust method when observations are missing. When a reasonable number of observations is missing, it actually improves its performances slightly, even though one must keep in mind that the MSE is in logarithmic scale. Both KPL estimators obtained from solving a problem readily posed with missing observations (ridge-iter-KPL, logcosh-KPL) are also much more robust than the dictionary-based methods which use the true Gram matrix of the dictionary (ridge-plug-KPL, 3BE). This is not surprising since as highlighted in Equation (5.29), for these methods to work, the empirical approximation of the Gram matrix of the dictionary must be close to the true one. If the locations of observation of the functions are two sparse, such assumption is not reasonable anymore, and we do observe a radical degradation of the performance as more and more observations are missing.

Finally, regarding local Gaussian noise added to the output functions, dictionary based methods (ridge-plug-KPL, ridge-iter-KPL, logcosh-KPL and 3BE) are much more efficient. This is probably due to the fact that the noise's distribution is centered. In these methods, the output functions appear through scalar products with elements of the dictionary, thus the noise can be partially evened-out.

5.3.3 Diffusion tensor imaging dataset (DTI)

Dataset. We now consider the DTI dataset.[1]. It consists of 382 Fractional anisotropy (FA) profiles inferred from DTI scans along two tracts–corpus callosum (CCA) and right corticospinal (RCS). The scans were performed on 142 subjects; 100 multiple sclerosis (MS) patients and 42 healthy controls. MS is an auto-immune disease which causes the immune system to gradually destroy myelin (the substance which isolates and protects the axons of nerve cells). It gradually results in brain lesions and severe disability. FA profiles are frequently used as an indicator for demyelification which causes a degradation of the diffusivity of the nerve tissues. The latter process is however not well understood and does not occur uniformly in all regions of the brain. We thus propose here to use our method to try to predict FA profiles along the RCS tract from FA profiles along the CCA tract. So as to remain in an i.i.d. framework, we consider only the first scans of MS patients resulting in $n = 100$ pairs of functions. The functions are observed on regular grids of sizes 93 and 54 respectively for the CCA and RCS tracts. However, significant parts of the FA profiles along the RCS tract are missing, we are thus dealing with sparsely sampled functions. Examples of instances from this dataset are shown in Figure 5.5.

Experimental setting. The reported means and standard deviations are computed over 20 runs with different train/test split. For all methods (except KE) we center the output functions using the training examples and add back the corresponding mean to the predictions. We perform linear smoothing if necessary–for FKRR and KAM.

[1] This dataset was collected at Johns Hopkins University and the Kennedy-Krieger Institute and is freely available as a part of the *Refund* R package

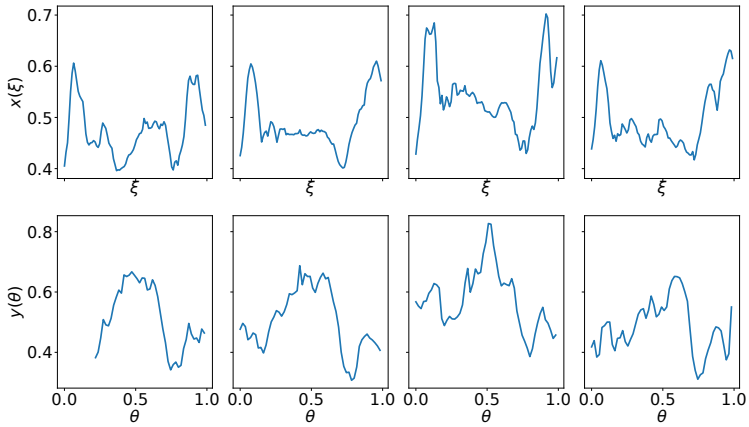


Figure 5.5: Examples from the DTI dataset.

Table 5.1: MSEs on the DTI dataset.

KE	0.231 ± 0.025
3BE	0.227 ± 0.017
KAM	0.222 ± 0.021
FKRR	0.215 ± 0.020
Ridge-KPL	0.211 ± 0.022
Logcosh-KPL	**0.209 ± 0.020**

We split the data as $n_{\text{train}} = 70$ and $n_{\text{test}} = 30$. For the dictionary-based methods (KPL and 3BE), we use wavelet dictionaries. More precisely, we consider several families of Daubechies wavelets (Daubechies, 1996)–see also Section 3.2.1. For KPL, we take a kernel of the form $\mathsf{K} = k_{\mathcal{X}}\mathsf{D}$ with $k_{\mathcal{X}}$ a Gaussian kernel and D a diagonal matrix with diagonal decreasing with the corresponding wavelet scale. When using wavelets, we extend the signal symmetrically to avoid boundary effects. For all methods we select through cross-validation the regularization parameter, the dictionary when relevant and the parameters of the output kernel for FKRR as well as all the parameters of the product of Gaussian kernels used for KAM. To go further, note that we detail all the parameters considered as well as the experimental procedure in Appendix A.2.

Comments on the results. The studied methods perform almost equally well, with a slight advantage for KPL estimators. The combination of an efficient use of wavelets (well suited to non-smooth data) with the scale-dependent regularization induced by the kernel $\mathsf{K} = k\mathsf{D}$ may explain this.

5.3.4 Synthetic speech inversion dataset

Experimental setting

Dataset. We consider a speech inversion problem: from an acoustic speech signal, we estimate the underlying vocal tract (VT) configuration that produced it (Richmond, 2002). Such information can improve performance in speech recognition systems or in speech synthesis. The dataset was introduced by Mitra et al. (2009); it is generated by a

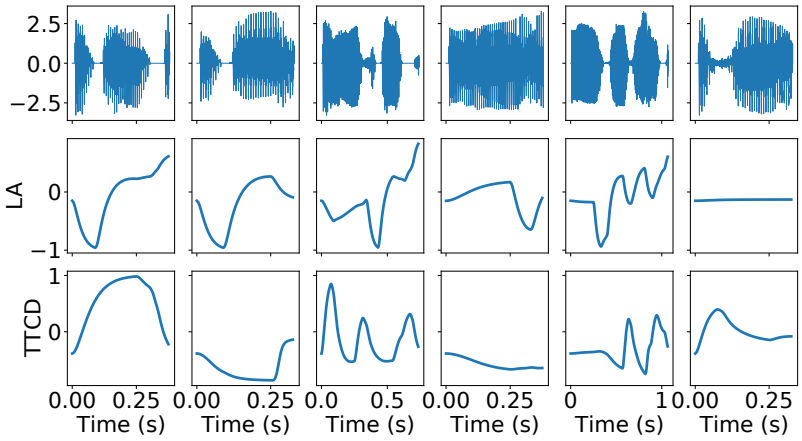


Figure 5.6: Examples from the speech dataset.

software synthesizing words from an articulatory model. It consists of a corpus of $n = 413$ pronounced words with 8 distinct VT functions: lip aperture (LA), lip protrusion (LP), tongue tip constriction degree (TTCD), tongue tip constriction location (TTCL), tongue body constriction degree (TBCD), tongue body constriction location (TBCL), Velum (VEL) and Glottis (GLO). We give some examples from this dataset in Figure 5.6 displaying only two of the VT functions.

Procedure and parameters. To match words of varying lengths, we extend symmetrically both the input sounds and the VT functions matching the longest word. We represent the sounds using 13 mel-frequency cepstral coefficients (MFCC), the input data thus consist of vector-valued functions. We split the data as $n_{\text{train}} = 300$ and $n_{\text{test}} = 113$. We normalize the output functions so that they take their values in $[-1, 1]$. To deal with the vector-valued functional inputs, we use an integral of Gaussian kernels on the standardized MFCCs (KPL, FKRR, 1BE/KPL)–see Equation (5.39). For KAM we take Laplace kernels for both input and output locations, and use a Gaussian kernel defined on $\mathbb{R}^{13}$ to compare the evaluations of the standardized MFCCs. More precisely, the input data consist of matrices in $\mathbb{R}^{m\times 13}$ (here the number of discretization points is the same for the input and for the output functions, so we have $t = m$ discretization points for the MFCCs). These correspond to discrete observations from $\mathbb{R}^{13}$-valued functions. Let $(\tilde{x}_i)_{i=1}^n$ be these observations of the discrete MFCCs, where for all $i \in [\![n]\!]$, $\tilde{x}_i \in \mathbb{R}^{m\times 13}$.

We now give the formula for the kernel that we use for ridge-DL-KPL, 1BE/ridge-Four-KPL and FKRR. Its integral expression is:

$$(x_1, x_2) \longmapsto \int_{[0,1]} \exp\left(\frac{-\|x_2(\xi) - x_1(\xi)\|^2_{\mathbb{R}^{13}}}{\sigma^2}\right) \mathrm{d}\xi.$$

However, in practice, we approximate it using the discrete MFCCs.

$$(\tilde{x}_1, \tilde{x}_2) \longmapsto \frac{1}{m}\sum_{s=1}^{m} \exp\left(\frac{-\|\tilde{x}_{2s} - \tilde{x}_{1s}\|^2_{\mathbb{R}^{13}}}{\sigma^2}\right). \tag{5.39}$$

For KAM, we use the kernel defined on $\left([0,1]\times[0,1]\times\mathbb{R}^{13}\right)^2$ by:

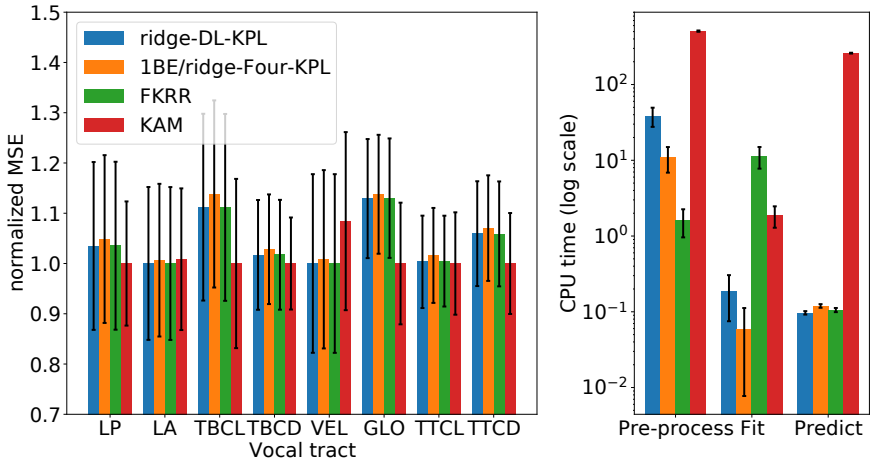


Figure 5.7: MSEs and CPU times on the speech dataset.

$$((\theta_1,\xi_1,u_1),(\theta_2,\xi_2,u_2)) \longmapsto \exp\left(\frac{-|\xi_1-\xi_2|}{\sigma_1}\right)\exp\left(\frac{-|\theta_1-\theta_2|}{\sigma_2}\right)\exp\left(\frac{-\|u_1-u_2\|^2_{\mathbb{R}^{13}}}{\sigma_3^2}\right). \tag{5.40}$$

Let us also give more details on the normalization of MFCCs.They are of different magnitudes, therefore we want to avoid biasing the norms to be over-representative of the larger ones. Thus before we apply the kernels described above, we standardize the MFCCs using the training data. For the r-th MFCC, we set $\text{avg}^{(r)} := \frac{1}{n_{\text{train}}m}\sum_{i=1}^{n_{\text{train}}}\sum_{s=1}^{m}\tilde{x}_{is}^{(r)}$ and $\text{std}^{(r)} := \sqrt{\frac{1}{n_{\text{train}}m-1}\sum_{i=1}^{n_{\text{train}}}\sum_{s=1}^{m}(\tilde{x}_{is}^{(r)}-\text{avg}^{(r)})^2}$, and use as input data $\left(\left(\frac{x_i^{(r)}}{\text{std}^{(r)}}\right)_{r=1}^{13}\right)_{i=1}^{n_{\text{train}}}$.

The MSEs for the 8 VTs (left panel) as well as an analysis of the computation times (right panel) are displayed in Figure 5.7. *Pre-process* entails all pre-processing operations (e. g. computing the kernel matrices, learning the dictionary, computing the Gram matrix of ϕ), *fit* measures the fitting time per se (solving the relevant linear system) and *predict* measures the prediction time on the test set (for all methods, it entails computing new kernel matrices). **ridge-DL-KPL** is the KPL ridge estimator with ϕ learnt by solving a discretized vanilla dictionary learning with sparsity inducing penalty–see the second part of Section 3.2.3. **1BE/ridge-Four-KPL** corresponds to 1BE (or equivalently KPL with $\mathsf{K} = k\mathrm{I}$) with ϕ a Fourier family. To give an order of idea, we use 30 atoms for the learnt dictionaries while the numbers of atoms selected by cross-validation for the Fourier ones are around 100. We do not include KE in the figure as it performed poorly on this dataset. To go further, note that we detail further all the parameters considered as well as the experimental procedure in Appendix A.3.

Comments on the results

For 4 out of 8 VTs (LP, LA, TBCD, TTCL), the performances of the methods are comparable, with KAM being slightly more precise. On the remaining 4 VTs, ridge-DL-KPL, 1BE/ridge-Four-KPL and FKRR beat KAM on one (VEL) and are beaten by KAM on the 3 others (TBCL, GLO, TTCD). A possible explanation is that KAM predicts locally the functions while the other three methods have more of a global approach. De-

pending on the properties of the functions and the nature of the dependency between input and output functions, one or the other could be more favorable. However KAM's main weakness is its computational cost for pre-processing and prediction, which makes it impractical to use on medium-sized datasets and impossible to use on larger ones. The particularly time-consuming operation in question is the computation of the kernel matrices in Equation (4.21) and Equation (4.22) for each of their entries involve double integrals. The three other methods display very close MSEs, with 1BE/ridge-Four-KPL being a bit less precise than the two others. Ridge-DL-KPL and FKRR perform equally well. However for the former the main computational burden comes from a pre-processing operation (learning the dictionary), which is performed only once per dataset (or once per fold in a cross-validation); whereas for the latter it comes from fitting the method, which must be done many times so as to tune its parameters. Moreover for Ridge-DL-KPL, once a number of atoms yielding a good approximation has been found and the dictionary has been learnt, no further tuning must be performed for the outputs, whereas for FKRR an output kernel must be chosen.

5.4 Conclusion

We have introduced the framework of projection learning to solve regression problems with outputs in a separable Hilbert space. It exploits the representation power of dictionaries directly in the empirical risk minimization problem and therefore circumvent the issues linked to infinite-dimensional outputs. We then focused on the use of vv-RKHSs as a hypothesis class and introduced kernel projection learning, for which we proposed several estimators. Among those, we derived two in closed-form. A ridge estimator for Hilbert-valued regression, and the plug-in ridge estimator for functional output regression with partially observed functions. We proposed efficient computation strategies for these estimators. We showed as well that our proposed estimators work well experimentally, in comparison with other nonlinear functional output regression methods. However, we have not studied those estimators theoretically. This is therefore the object of the next chapter, in which we provide an excess risk analysis of the ridge and plug-in ridge estimators.

6

Excess risk guarantees for kernel projection learning

Contents

In this chapter, we focus on a theoretical analysis of two estimators that we proposed in the previous chapter. More precisely, we work on bounding their excess risk. This quantity measures in terms of expected ("true") risk, how far away an estimator is from the infimum in the chosen hypothesis class. In other words, it quantifies the efficiency of an estimation procedure in terms of risk. It is then desirable to exhibit the dependency of the excess risk bound with respect to quantities of interest such as the number of samples or the number of observations per function in the case of FOR. First, in Section 6.1 we study the ridge estimator introduced in Proposition 5.13 considering outputs lying in a separable Hilbert space $\mathcal{Y}$. To improve overall clarity, the corresponding proof is deferred to Section 6.2. Second, in Section 6.3 we study the plug-in ridge estimator introduced in Definition 5.18 using the empirical mean to estimate the scalar products in $L^2(\Theta)$ with Θ a compact set. We make the assumption that the functions are observed at locations drawn uniformly at random on Θ. The proof of the bound being once again deferred to the dedicated Section 6.4. This chapter corresponds to the theoretical contributions of

- **D. Bouche**, M. Clausel, F. Roueff and F. d'Alché-Buc. Nonlinear Functional Output Regression: A Dictionary Approach. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 235–243, 2021,

which were not included in Chapter 5.

6.1 Excess risk bound for the ridge estimator

In order to derive theoretical results, we need to make stronger assumptions on the input space $\mathcal{X}$ than in previous chapters. Up to now it was any space on which a kernel can be defined. Here we make the following assumption which also circumvent any concerns regarding measurability.

Assumption 6.1. *$\mathcal{X}$ is a separable metric space.*

Other than that, the first assumption we make regards the kernel. More precisely, we suppose that it is bounded and continuous. The assumption of boundedness is used extensively in the proofs, especially in conjunction with the following lemma (see *e.g.* Micchelli and Pontil 2005).

Lemma 6.2. *Let $\mathcal{H}_\mathsf{K}$ be a vv-RKHS on $\mathbb{R}^d$ associated to a positive matrix-valued kernel K. Then we have for all $x \in \mathcal{X}$:*

$$\|h(x)\|_{\mathbb{R}^d} \leq \|h\|_{\mathcal{H}_\mathsf{K}} \, \|\mathsf{K}(x,x)\|^{1/2}_{\mathcal{L}(\mathbb{R}^d)}.$$

Additionally, since for all $x \in \mathcal{X}$, $h(x) = \mathsf{K}_x^\# h$, this implies that

$$\|\mathsf{K}_x\|_{\mathcal{L}(\mathbb{R}^d,\mathcal{H}_\mathsf{K})} = \|\mathsf{K}_x^\#\|_{\mathcal{L}(\mathcal{H}_\mathsf{K},\mathbb{R}^d)} \leq \|\mathsf{K}(x,x)\|^{1/2}_{\mathcal{L}(\mathbb{R}^d)}. \tag{6.1}$$

Remark 6.3. *The boundedness assumption is not very restrictive. Indeed, if we were considering the case of Hilbert-valued regression using OVK on $\mathcal{Y}$, it would be. However, thanks to the projection on a dictionary, K is an OVK on $\mathbb{R}^d$, which is therefore finite dimensional.*

The continuity assumption is a way to transfer the separability of the input space $\mathcal{X}$ to the vv-RKHS $\mathcal{H}_\mathsf{K}$. It is crucial because the concentration inequalities in Hilbert spaces that we use require separability.

Assumption 6.4. *K is a vector-valued continuous kernel and there exists $\kappa > 0$ such that for $x \in \mathcal{X}$, $\|\mathsf{K}(x,x)\|_{\mathcal{L}(\mathbb{R}^d)} \leq \kappa$.*

Remark 6.5. *We suppose that κ is independent from d. This is for instance the case if for $x \in \mathcal{X}$, $\mathsf{K}(x,x)$ is diagonal or block diagonal with bounded coefficients. More generally, we can rely on the fact that κ is bounded by the maximal $\|\cdot\|_1$-norm of the columns of $\mathsf{K}(x,x)$, which can easily be imposed to be independent of d.*

Next, we highlight useful bounds on the operator norm of the projection operator as well as on that of its adjoint and that of its associated Gram matrix.

Lemma 6.6. *Let $\boldsymbol{\phi} := (\phi_1, ..., \phi_d) \in \mathcal{Y}^d$, and let Φ be its associated projection operator–see Equation (5.5). Then*

$$\|\Phi\|_{\mathcal{L}(\mathbb{R}^d,\mathcal{Y})} = \sqrt{v_1}, \tag{6.2}$$

$$\|\Phi^\#\|_{\mathcal{L}(\mathcal{Y},\mathbb{R}^d)} = \sqrt{v_1} \text{ and} \tag{6.3}$$

$$\|\Phi^\#\Phi\|_{\mathcal{L}(\mathbb{R}^d)} = v_1, \tag{6.4}$$

where v_1 corresponds to the largest eigenvalue of the Gram matrix of the dictionary $\Phi^\#\Phi$.

Proof

$$\sup_{\mathbf{a}\in\mathbb{R}^d} \frac{\|\Phi\mathbf{a}\|_{\mathcal{Y}}^2}{\|\mathbf{a}\|_{\mathbb{R}^d}^2} = \frac{\mathbf{a}^{\mathrm{T}}\Phi^{\#}\Phi\mathbf{a}}{\mathbf{a}^{\mathrm{T}}\mathbf{a}}.$$

This is the Rayleigh quotient associated to $\Phi^{\#}\Phi$ therefore it is maximized by the largest eigenvalue of $\Phi^{\#}\Phi$ which we call v_1. Consequently,

$$\|\Phi\|_{\mathcal{L}(\mathbb{R}^d,\mathcal{Y})} = \sqrt{v_1}.$$

Since the operator Φ is bounded, $\|\Phi^{\#}\|_{\mathcal{L}(\mathcal{Y},\mathbb{R}^d)} = \|\Phi\|_{\mathcal{L}(\mathbb{R}^d,\mathcal{Y})}$ implying Equation (6.6). Finally a similar reasoning with the Rayleigh quotient of $(\Phi^{\#}\Phi)^2$ leads to Equation (6.7). ■

We see that this bound is not very informative, and consequently, to better understand it, we will need to make further assumptions on the dictionary. An interesting notion is that of *Riesz families* (Casazza, 2000).

Definition 6.7 (Riesz family)**.** *$\boldsymbol{\phi} \in \mathcal{Y}^d$ is a Riesz family of $\mathcal{Y}$ with constants $(c_{\boldsymbol{\phi}}, C_{\boldsymbol{\phi}})$ if it is linearly independent and for any $\mathbf{u} \in \mathbb{R}^d$,*

$$c_{\boldsymbol{\phi}} \|\mathbf{u}\|_{\mathbb{R}^d} \leq \left\| \sum_{l=1}^{d} u_l \phi_l \right\|_{\mathcal{Y}} \leq C_{\boldsymbol{\phi}} \|\mathbf{u}\|_{\mathbb{R}^d}.$$

If in addition for all $l \in [\![d]\!]$, $\|\phi_l\|_{\mathcal{Y}} = 1$, it is a normed Riesz family.

Remark 6.8. *Riesz families provide a natural generalization of orthonormal families as a normed Riesz family with $c_{\boldsymbol{\phi}} = C_{\boldsymbol{\phi}} = 1$ is orthonormal.*

Remark 6.9 (Dependence of v_1 in d)**.** *Intuitively, the more redundant the dictionary, the worse the dependency of v_1 in the number of atoms. Two extreme cases can help us understand this.*

1. *If $\boldsymbol{\phi}$ is an orthonormal family in $\mathcal{Y}$, then $\Phi^{\#}\Phi = \mathrm{I}$ and therefore $C_{\boldsymbol{\phi}} = 1$.*
2. *If $\boldsymbol{\phi}$ is a Riesz family, we have that $v_1 \leq C_{\boldsymbol{\phi}}^2$ and therefore it is not dependent on d either.*
3. *By opposition, if $\boldsymbol{\phi}$ consists of d-times the same vector, supposing this vector's norm is equal to 1, $\Phi^{\#}\Phi$'s largest eigenvalue is d.*

In the following, we make the proof assuming that the dictionary is a Riesz basis, but one must keep in mind that all the results remain valid if we replace the corresponding constant with the largest eigenvalue of the Gram matrix of the dictionary. However, as highlighted above it may hide a dependency in d.

Assumption 6.10. *The dictionary $\boldsymbol{\phi}$ is a normed Riesz family in $\mathcal{Y}$ with upper constant $C_{\boldsymbol{\phi}}$.*

Then combining Lemma 6.6 with Definition 6.7 readily yields the following lemma.

Lemma 6.11. *Let $\boldsymbol{\phi} = (\phi_1, ..., \phi_d) \in \mathcal{Y}^d$, and let Φ be its associated projection operator–see Equation (5.5). Then*

$$\|\Phi\|_{\mathcal{L}(\mathbb{R}^d, \mathcal{Y})} \leq C_{\boldsymbol{\phi}}, \tag{6.5}$$

$$\|\Phi^{\#}\|_{\mathcal{L}(\mathcal{Y}, \mathbb{R}^d)} \leq C_{\boldsymbol{\phi}}, \text{ and} \tag{6.6}$$

$$\|\Phi^{\#}\Phi\|_{\mathcal{L}(\mathbb{R}^d)} \leq C_{\boldsymbol{\phi}}^2, \tag{6.7}$$

where $C_{\boldsymbol{\phi}}$ corresponds to the largest eigenvalue of the Gram matrix of the dictionary $\Phi^{\#}\Phi$.

We also assume that the infimum of the expected risk over the hypothesis class is attained. Such a hypothesis is used in many works on learning theory for kernel methods (Caponnetto and De Vito, 2007; Baldassarre et al., 2012; Li et al., 2021). It is worth noting also that our hypothesis class consists of compositions of a function in a vv-RKHS with the projection on a dictionary, therefore the choice of the dictionary determines the possible models as well.

Assumption 6.12. *There exists $h_{\mathcal{H}_K} \in \mathcal{H}_K$ such that $\mathcal{R}(\Phi \circ h_{\mathcal{H}_K}) = \inf_{h \in \mathcal{H}_K} \mathcal{R}(\Phi \circ h)$. This implies the existence of a ball of radius $R > 0$ in $\mathcal{H}_K$ containing $h_{\mathcal{H}_K}$, as a consequence*

$$\|h_{\mathcal{H}_K}\|_{\mathcal{H}_K} \leq R. \tag{6.8}$$

Finally, we suppose that the distribution ρ of the couple of random variables (X, Y) generates almost surely elements of $\mathcal{Y}$ with finite norm.

Assumption 6.13. *Let (X, Y) be distributed according to ρ. We suppose that there exists $L \geq 0$ such that almost surely*

$$\|Y\|_{\mathcal{Y}} \leq L.$$

Under those assumptions, the following finite sample excess risk bound for the ridge estimator defined in Proposition 5.13 holds.

Proposition 6.14. *Let $0 < \eta < 1$, then taking $\lambda = \lambda_n^*(\eta/2) := 6\kappa C_{\boldsymbol{\phi}}^2 \frac{\log(4/\eta)\sqrt{d}}{\sqrt{n}}$, we have that with probability at least $1 - \eta$,*

$$\mathcal{R}(\Phi \circ h_{\mathbf{z}}^{\lambda}) - \mathcal{R}(\Phi \circ h_{\mathcal{H}_K}) \leq 27\left(\frac{B_0}{\sqrt{d}} + B_1\sqrt{d}\right)\frac{\log\left(4/\eta\right)}{\sqrt{n}},$$

where we have defined the constants $B_0 := (L + \sqrt{\kappa} C_{\boldsymbol{\phi}} R)^2$ and $B_1 := \kappa C_{\boldsymbol{\phi}}^2 R^2$.

This bound implies the consistency of the ridge estimator in the number of samples n.

Sketch of proof. The scheme of proof is essentially the one we presented in Section 2.2.3. It mostly bounds the difference between specific integral operators related to the ridge regression and their empirical counterparts (Caponnetto and De Vito,

2007). In practice, the proof will proceed as follows. The ridge estimator is reformulated in terms of empirical operators and the excess risk $\mathcal{R}(\Phi \circ h) - \mathcal{R}(\Phi \circ h_{\mathcal{H}_K})$ as a distance $\|\sqrt{T_\Phi}(h - h_{\mathcal{H}_K})\|^2_{\mathcal{H}_K}$ in the vv-RKHS $\mathcal{H}_K$ for an operator $T_\Phi \in \mathcal{L}(\mathcal{H}_K)$. The expression of the ridge estimator is then injected into this distance. The technical part of the proof consists in separating judiciously the resulting quantity in different terms which involve distances between operators and their empirical approximations. These terms can then be bounded with high probability using concentration inequalities in separable Hilbert spaces.

6.2 Proof of the bound for the ridge estimator

This section is dedicated to proving Proposition 6.14. In Section 6.2.1 we introduce operators of interest and reformulate the excess risk using these. We then formulate empirical approximations of these operators using the available data and derive closed-form solution for the ridge estimators in terms of these empirical operators in Section 6.2.2. Section 6.2.3 is dedicated to the introduction and proof of several concentration results and lemmas on which we rely. Section 6.2.4 puts all the elements together to prove Proposition 6.14.

6.2.1 Integral operators and excess risk reformulation

Let us recall the expression of the expected and empirical risks for kernel projection learning. For a function $h \in \mathcal{H}_K$, the expected risk of the function $\Phi \circ h$ is

$$\mathcal{R}(\Phi \circ h) = \int_{\mathcal{Y}} \|y - \Phi h(x)\|^2_{\mathcal{Y}} d\rho(x, y). \tag{6.9}$$

Since we do not have access to the generating measure ρ, but rather to a finite sample $\mathbf{z}$, we minimize the empirical risk as a proxy

$$\hat{\mathcal{R}}(\Phi \circ h, \mathbf{z}) := \frac{1}{n} \sum_{i=1}^{n} \|y_i - \Phi \circ h(x_i)\|^2_{\mathcal{Y}}. \tag{6.10}$$

Remark 6.15. *Indeed, for the definition of the expected risk–Equation (6.9)–to make sense, the corresponding integral must be finite. It is the case here because* $((x, y) \mapsto \Phi h(x)) \in L^2(\mathcal{Z}, \rho, \mathcal{Y})$ *as a consequence of Assumption 6.4 and Equation (6.6);* $((x, y) \mapsto y) \in L^2(\mathcal{Z}, \rho, \mathcal{Y}))$ *as well as a consequence of Assumption 6.13.*

We have used the notation $L^2(\mathcal{Z}, \rho, \mathcal{Y})$, we recall that it denotes the space of functions from $\mathcal{Z}$ to $\mathcal{Y}$ which are square integrable with respect to the measure ρ, this space being endowed with the scalar product

$$\langle \psi_1, \psi_2 \rangle_\rho = \int_{\mathcal{Z}} \langle \psi_1(x, y), \psi_2(x, y) \rangle_{\mathcal{Y}} \, d\rho(x, y),$$

and the associated norm $\|\cdot\|_\rho$.

We now consider the following operator. It plays the same role as the canonical inclusion operator defined in Equation (2.26), yet it encompasses additionally a mapping through Φ. Essentially, we will follow the same process as in Section 2.2.3 using this modified operator. More precisely, we define $A_\Phi : \mathcal{H}_K \longrightarrow L^2(\mathcal{Z}, \rho, \mathcal{Y})$ as

$$A_\Phi : h \mapsto A_\Phi h \quad \text{with} \quad (A_\Phi h) : (x, y) \mapsto \Phi K_x^\# h. \tag{6.11}$$

Indeed, we can reformulate the expected risk in terms of A_Φ for any $h \in \mathcal{H}_K$,

$$\begin{aligned} \|A_\Phi h - Y\|_\rho^2 &= \int_{\mathcal{Z}} \|\Phi K_x^\# h - y\|_{L^2(\Theta)}^2 \, d\rho(x, y) \\ &= \int_{\mathcal{Z}} \|\Phi h(x) - y\|_{L^2(\Theta)}^2 \, d\rho(x, y) \\ &= \mathcal{R}(\Phi \circ h). \end{aligned} \tag{6.12}$$

From this operator, we define T_Φ as

$$T_\Phi := A_\Phi^\# A_\Phi. \tag{6.13}$$

In order to reformulate the excess risk later, we derive the following necessary condition to be a minimizer of the expected risk involving the operators T_Φ and A_Φ.

Lemma 6.16. *Assume that there exists $h_{\mathcal{H}_K} \in \mathcal{H}_K$ such that*

$$h_{\mathcal{H}_K} = \inf_{h \in \mathcal{H}_K} \mathcal{R}(\Phi \circ h).$$

Then, for all $h \in \mathcal{H}_K$,

$$\langle h, T_\Phi h_{\mathcal{H}_K} - A_\Phi^\# Y \rangle_{\mathcal{H}_K} = 0; \tag{6.14}$$

or equivalently:

$$T_\Phi h_{\mathcal{H}_K} = A_\Phi^\# Y, \tag{6.15}$$

with $Y \in L^2(\mathcal{Z}, \rho, \mathcal{Y})$ denoting the function $Y : (x, y) \longmapsto y$.

Proof We use the formulation of the expected risk from Equation (6.12). The function $h \longmapsto \mathcal{R}(\Phi \circ h) = \|A_\Phi h - Y\|_\rho^2$ is convex. Its differential is given by

$$D\mathcal{R}(\Phi \circ h_{\mathcal{H}_K})(h) = 2\langle A_\Phi h, A_\Phi h_{\mathcal{H}_K} - Y \rangle_\rho = 2\langle h, A_\Phi^\# A_\Phi h_{\mathcal{H}_K} - A_\Phi^\# Y \rangle_{\mathcal{H}_K} = 2\langle h, T_\Phi h_{\mathcal{H}_K} - A_\Phi^\# Y \rangle_{\mathcal{H}_K}.$$

We then must have for all $h \in \mathcal{H}_K$,

$$\langle h, T_\Phi h_{\mathcal{H}_K} - A_\Phi^\# Y \rangle_{\mathcal{H}_K} = 0.$$

■

Note that Assumption 6.12 corresponds to the main hypothesis for this lemma. We can use the formulation of the expected risk from Equation (6.12) in combination with the characterization of $h_{\mathcal{H}_K}$ in this lemma from Equation (6.14). It implies that for any $h \in \mathcal{H}_K$, we can reformulate the excess risk of h as a distance in $\mathcal{H}_K$ between h and $h_{\mathcal{H}_K}$ taken through the operator T_Φ.

Lemma 6.17. *We have that for any $h \in \mathcal{H}_K$,*

$$\mathcal{R}(\Phi \circ h) - \mathcal{R}(\Phi \circ h_{\mathcal{H}_K}) = \|\sqrt{T_\Phi}(h - h_{\mathcal{H}_K})\|^2_{\mathcal{H}_K}. \tag{6.16}$$

Proof

$$\begin{aligned}\mathcal{R}(\Phi \circ h) - \mathcal{R}(\Phi \circ h_{\mathcal{H}_K}) &= \|A_\Phi h - Y\|^2_\rho - \|A_\Phi h_{\mathcal{H}_K} - Y\|^2_\rho \\ &= \|A_\Phi(h - h_{\mathcal{H}_K})\|^2_\rho + 2\langle A_\Phi(h - h_{\mathcal{H}_K}), A_\Phi h_{\mathcal{H}_K} - Y\rangle_\rho \\ &= \|A_\Phi(h - h_{\mathcal{H}_K})\|^2_\rho,\end{aligned}$$

where we have used Equation (6.14). Since we have the following polar decomposition $A_\Phi = U\sqrt{A^\#_\Phi A_\Phi} = U\sqrt{T_\Phi}$ with U a partial isometry from the closure of $\mathrm{Im}(\sqrt{T_\Phi})$ onto the closure of $\mathrm{Im}(A_\Phi)$,

$$\|A_\Phi(h - h_{\mathcal{H}_K})\|_\rho = \|U\sqrt{T_\Phi}(h - h_{\mathcal{H}_K})\|_\rho = \|\sqrt{T_\Phi}(h - h_{\mathcal{H}_K})\|_{\mathcal{H}_K}.$$

■

Such reformulation enables us to decompose the excess risk in terms that we can easily control using concentration inequalities in Hilbert spaces.

6.2.2 Empirical approximations and closed form solutions

We now define empirical approximations of the operators A_Φ and T_Φ. Using these approximations, we can derive a closed-form for the minimizer of the regularized empirical risk. We utilize this closed-form to bound the excess risk in the subsequent proof.

To define these approximations, we need to precise the integral expressions of $A^\#_\Phi$ and T_Φ. This is the object of the following lemma, which is almost a restatement of Proposition 1 from Caponnetto and De Vito (2005). All the arguments in their proof are readily verified in our case as well, consequently, we do not rewrite the proof here.

Let us define for all $x \in \mathcal{X}$ the operators $K_{x,\Phi} := K_x \Phi^\#$ and $T_{x,\Phi} := K_{x,\Phi} K^\#_{x,\Phi}$.

Lemma 6.18. *For $\psi \in L^2(\mathcal{Z}, \rho, \mathcal{Y})$, the adjoint of A_Φ applied to ψ is given by*

$$A^\#_\Phi \psi = \int_{\mathcal{Z}} K_{x,\Phi}\psi(x,y)\, d\rho(x,y), \tag{6.17}$$

with the integral converging in $\mathcal{H}_K$;

$A^\#_\Phi A_\Phi$ is the Hilbert Schmidt operator on $\mathcal{H}_K$ given by

$$A^\#_\Phi A_\Phi = T_\Phi = \int_{\mathcal{X}} T_{x,\Phi}\, d\rho_X(x), \tag{6.18}$$

with the integral converging in $\mathcal{L}_2(\mathcal{H}_K)$.

Empirical approximations of the operators A_Φ and T_Φ can then straightforwardly be set as

$$A^{\#}_{\mathbf{x},\Phi}\mathbf{w} = \frac{1}{n}\sum_{i=1}^{n} K_{x_i,\Phi} w_i, \quad \mathbf{w} = (w_i)_{i=1}^n \in \mathcal{Y}^n.$$
$$(A_{\mathbf{x},\Phi} h)_i = K^{\#}_{x_i,\Phi} h = \Phi h(x_i), \quad h \in \mathcal{H}_K, \quad \forall i \in [\![n]\!].$$
$$T_{\mathbf{x},\Phi} = A^{\#}_{\mathbf{x},\Phi} A_{\mathbf{x},\Phi} = \frac{1}{n}\sum_{i=1}^{n} T_{x_i,\Phi}.$$

Defining the regularized empirical risk of $\Phi \circ h$ for any $h \in \mathcal{H}_K$ as

$$\begin{aligned}\hat{\mathcal{R}}^{\lambda}(\Phi \circ h, \mathbf{z}) &:= \hat{\mathcal{R}}(\Phi \circ h, \mathbf{z}) + \lambda \|h\|^2_{\mathcal{H}_K} \\ &= \frac{1}{n}\sum_{i=1}^{n} \|K^{\#}_{x_i,\Phi} h - y_i\|^2_{\mathcal{Y}} + \lambda \|h\|^2_{\mathcal{H}_K},\end{aligned}$$

the following closed-form for its minimizer can be derived using the operators introduced above.

Lemma 6.19. *There exists a unique minimizer $h^{\lambda}_{\mathbf{z}}$ of $h \in \mathcal{H}_K \longmapsto \hat{\mathcal{R}}^{\lambda}(\Phi \circ h, \mathbf{z})$ which is given by*

$$h^{\lambda}_{\mathbf{z}} := (T_{\mathbf{x},\Phi} + \lambda \mathrm{I})^{-1} A^{\#}_{\mathbf{x},\Phi}\mathbf{y}. \tag{6.19}$$

Proof Proposition 5.6 guarantees the existence of a minimizer. Since $\lambda > 0$, $h \longmapsto \hat{\mathcal{R}}^{\lambda}(\Phi \circ h, \mathbf{z})$ is strictly convex, this minimizer is unique and is obtained by setting the gradient to zero. The differential of the objective is given by

$$\begin{aligned}\mathrm{D}\hat{\mathcal{R}}^{\lambda}(\Phi \circ h_0, \mathbf{z})(h_1) &= \frac{2}{n}\sum_{i=1}^{n} \langle K^{\#}_{x_i,\Phi} h_0 - y_i, K^{\#}_{x_i,\Phi} h_1 \rangle_{\mathcal{Y}} + 2\lambda \langle h_0, h_1 \rangle_{\mathcal{H}_K} \\ &= 2\Big\langle \Big(\frac{1}{n}\sum_{i=1}^{n} T_{x_i,\Phi} + \lambda\Big) h_0 - \frac{1}{n}\sum_{i=1}^{n} K_{x_i,\Phi} y_i, h_1 \Big\rangle_{\mathcal{H}_K} \\ &= 2\langle (T_{\mathbf{x},\Phi} + \lambda \mathrm{I}) h_0 - A^{\#}_{\mathbf{x},\Phi}\mathbf{y}, h_1 \rangle_{\mathcal{H}_K}.\end{aligned}$$

As a consequence, $h^{\lambda}_{\mathbf{z}}$ is characterized by

$$(T_{\mathbf{x},\Phi} + \lambda \mathrm{I}) h^{\lambda}_{\mathbf{z}} - A^{\#}_{\mathbf{x},\Phi}\mathbf{y} = 0.$$

Since $T_{\mathbf{x},\Phi}$ is positive and $\lambda > 0$, $(T_{\mathbf{x},\Phi} + \lambda \mathrm{I})$ is invertible and thus

$$h^{\lambda}_{\mathbf{z}} = (T_{\mathbf{x},\Phi} + \lambda \mathrm{I})^{-1} A^{\#}_{\mathbf{x},\Phi}\mathbf{y}.$$

■

We conclude therefore that $h^{\lambda}_{\mathbf{z}}$ is the same object as the ridge estimator from Equation (5.19), only it is represented in terms of the operators introduced above. This is needed to carry out an excess risk analysis.

6.2.3 Concentration results

In this section, we derive concentration results we rely on to prove Proposition 6.14. First, we provide a bound on the Hilbert-Schmidt norm of $\mathrm{T}_{x,\Phi}$ which will be useful to derive these results. Then, we introduce a Bernstein concentration inequality for random variables in a separable Hilbert space as well as some other useful results. Finally, we use these tools to derive the desired concentration lemma.

Bound on Hilbert-Schmidt norm of $\mathrm{T}_{x,\Phi}$

The closed-form in Lemma 6.19 brings out the key role of the operator $\mathrm{T}_{\mathbf{x},\Phi}$. Unsurprisingly, we will need concentration results on $\mathrm{T}_{\mathbf{x},\Phi}$ in the final proof. An intermediate result to achieve this goal is to bound the Hilbert-Schmidt norm of $\mathrm{T}_{x,\Phi}$ which is the object of the next lemma.

For all $x \in \mathcal{X}$, we recall the definition of the following operators

- $\mathsf{K}_{x,\Phi} : \mathcal{Y} \longrightarrow \mathcal{H}_{\mathsf{K}}$ is defined by $\mathsf{K}_{x,\Phi} := \mathsf{K}_x \Phi^{\#}$ with K_x as defined in Equation (5.9).
- $\mathrm{T}_{x,\Phi} : \mathcal{H}_{\mathsf{K}} \longrightarrow \mathcal{H}_{\mathsf{K}}$ is defined as $\mathrm{T}_{x,\Phi} := \mathsf{K}_{x,\Phi} \mathsf{K}^{\#}_{x,\Phi}$.

Observe that $\mathrm{T}_{x,\Phi}$ is of finite rank and positive. We can then deduce the following bound on its Hilbert-Schmidt norm.

Lemma 6.20. *Assume that there exists $\kappa \geq 0$ such that for all $x \in \mathcal{X}$,*

$$\|\mathsf{K}(x,x)\|_{\mathcal{L}(\mathbb{R}^d)} \leq \kappa, \tag{6.20}$$

then for all $x \in \mathcal{X}$,

$$\|\mathrm{T}_{x,\Phi}\|_{\mathcal{L}_2(\mathcal{H}_{\mathsf{K}})} \leq \sqrt{d}\kappa C_{\phi}^2. \tag{6.21}$$

Proof For all $x \in \mathcal{X}$, $\mathrm{Rank}(\mathrm{T}_{x,\Phi}) \leq d$. Let $(e_l)_{l=1}^{\mathrm{Rank}(\mathrm{T}_{x,\Phi})}$ be an orthonormal basis of $\mathrm{Im}(\mathrm{T}_{x,\Phi})$. We complete it to $(e_l)_{l\in\mathbb{N}^*}$ an orthonormal basis of $\mathcal{H}_{\mathsf{K}}$. Since $\mathrm{Im}(\mathrm{T}_{x,\Phi})$ is a finite dimensional subspace of $\mathcal{H}_{\mathsf{K}}$ and $\mathrm{T}_{x,\Phi}$ is self adjoint, we have that $\mathrm{Im}(\mathrm{T}_{x,\Phi}) = \mathrm{Ker}(\mathrm{T}_{x,\Phi})^{\perp}$. As a consequence, for all $l > \mathrm{Rank}(\mathrm{T}_{x,\Phi})$, $\mathrm{T}_{x,\Phi} e_l = 0$, which implies

$$\|\mathrm{T}_{x,\Phi}\|^2_{\mathcal{L}_2(\mathcal{H}_{\mathsf{K}})} = \sum_{l=1}^{\mathrm{Rank}(\mathrm{T}_{x,\Phi})} \langle \mathrm{T}_{x,\Phi} e_l, \mathrm{T}_{x,\Phi} e_l \rangle_{\mathcal{H}_{\mathsf{K}}} = \sum_{l=1}^{\mathrm{Rank}(\mathrm{T}_{x,\Phi})} \langle \mathsf{K}^{\#}_x e_l, \Phi^{\#} \Phi \mathsf{K}(x,x) \Phi^{\#} \Phi \mathsf{K}^{\#}_x e_l \rangle_{\mathbb{R}^d}.$$

Using Cauchy-Schwarz in the previous expression along with Equation (6.7), Equation (6.20) and Equation (6.1) we obtain

$$\|\mathrm{T}_{x,\Phi}\|^2_{\mathcal{L}_2(\mathcal{H}_{\mathsf{K}})} \leq C_{\phi}^4 \kappa \sum_{l=1}^{\mathrm{Rank}(\mathrm{T}_{x,\Phi})} \|\mathsf{K}^{\#}_x e_l\|^2_{\mathbb{R}^d} \leq C_{\phi}^4 \kappa^2 \,\mathrm{Rank}(\mathrm{T}_{x,\Phi}) \leq d C_{\phi}^4 \kappa^2,$$

which achieves the proof. ∎

Concentration tools

We now state a concentration inequality that we use to control the different terms in our decomposition of the excess risk later on.

The following is a direct consequence of a Bernstein inequality for independent random variables in a separable Hilbert space–see Proposition 3.3.1 in Yurinsky 1995 or Theorem 3 in Pinelis and Sakhanenko 1986. It also corresponds to Proposition 2 in Caponnetto and De Vito (2007, Proposition 2).

Lemma 6.21. *Let ξ be a random variable taking its values in a real separable Hilbert space $\mathcal{K}$ such that there exist $H \geq 0$ and $\sigma \geq 0$ such that*

$$\|\xi\|_{\mathcal{K}} \leq \frac{H}{2} \text{ almost surely, and}$$
$$\mathbb{E}[\|\xi\|_{\mathcal{K}}^2] \leq \sigma^2.$$

Let $n \in \mathbb{N}$ and $(\xi_1, ..., \xi_n)$ be i.i.d. realizations of ξ. Let $0 < \eta < 1$, then

$$\mathbb{P}\left[\left\|\frac{1}{n}\sum_{i=1}^{n}\xi_i - \mathbb{E}[\xi]\right\|_{\mathcal{K}} \leq 2\Big(\frac{H}{n} + \frac{\sigma}{\sqrt{n}}\Big)\log\frac{2}{\eta}\right] \geq 1 - \eta.$$

We will need in the final proof to deduce concentration properties of $\sqrt{\mathrm{T}_{\mathbf{x},\Phi}}$ from concentration properties of $\mathrm{T}_{\mathbf{x},\Phi}$. To that end, the upcoming lemma is central. It corresponds to Theorem X.1.1 in Bhatia (1997) where it is stated for positive symmetric matrices. However, their proof remains fully valid for positive bounded operators defined on real separable Hilbert spaces.

Lemma 6.22. *Let $\mathcal{K}$ be a real separable Hilbert space, let $\mathrm{A}, \mathrm{B} \in \mathcal{L}(\mathcal{K})$ be two positive operators. Then, we have*

$$\|\sqrt{\mathrm{A}} - \sqrt{\mathrm{B}}\|_{\mathcal{L}(\mathcal{K})} \leq \sqrt{\|\mathrm{A} - \mathrm{B}\|_{\mathcal{L}(\mathcal{K})}}.$$

Intermediate concentration results

We now use the different assumptions and intermediate results to derive intermediate concentration results on different operators of interest which will be the cornerstone of the proof of Proposition 6.14.

Lemma 6.23. *Let $0 < \eta < 1$, then with probability at least $1 - \eta$*

$$\|\mathrm{A}^{\#}_{\mathbf{x},\Phi}\mathbf{y} - \mathrm{T}_{\mathbf{x},\Phi}h_{\mathcal{H}_\mathsf{K}}\|_{\mathcal{H}_\mathsf{K}} \leq \delta_1(n, \eta),$$

with δ_1 defined as

$$\delta_1(n, \eta) := 6(\sqrt{\kappa}C_\phi L + \kappa C_\phi^2 R)\frac{\log\left(2/\eta\right)}{\sqrt{n}}. \tag{6.22}$$

Proof Let us define the function $\xi_1 : \mathcal{Z} \longrightarrow \mathcal{H}_\mathsf{K}$ as $\xi_1 : (x, y) \longmapsto \mathsf{K}_{x,\Phi}(y - \Phi h_{\mathcal{H}_\mathsf{K}}(x)) = \mathsf{K}_{x,\Phi}(y - \mathsf{K}^{\#}_{x,\Phi}h_{\mathcal{H}_\mathsf{K}})$.

Observe that

$$\frac{1}{n}\sum_{i=1}^{n}\xi_1(x_i, y_i) = \mathrm{A}^{\#}_{\mathbf{x},\Phi}\mathbf{y} - \mathrm{T}_{\mathbf{x},\Phi}h_{\mathcal{H}_\mathsf{K}},$$

and using Equation (6.15), that

$$\mathbb{E}_{X,Y\sim\rho}\Big[\xi_1(X,Y)\Big] = \int_{\mathcal{Z}} K_{x,\Phi} y \, d\rho(x,y) - \left(\int_{\mathcal{Z}} K_{x,\Phi} K_{x,\Phi}^{\#} \, d\rho(x,y)\right) h_{\mathcal{H}_{\mathcal{H}_K}} = A_{\Phi}^{\#} Y - T_{\Phi} h_{\mathcal{H}_K} = 0.$$

The aim is now to apply the Bernstein inequality of Lemma 6.21 to the random variable (RV) $\xi_1(X,Y)$. First, we have almost surely

$$\begin{aligned}\|\xi_1(X,Y)\|_{\mathcal{H}_K} = \|K_{X,\Phi}(Y - \Phi h_{\mathcal{H}_K}(X))\|_{\mathcal{H}_K} &\le \|K_{X,\Phi}\|_{\mathcal{L}(L^2(\Theta),\mathcal{H}_K)} \|Y - \Phi h_{\mathcal{H}_K}(X))\|_{L^2(\Theta)} \\ &\le \sqrt{\kappa} C_\phi (\|Y\|_{L^2(\Theta)} + \|K_{X,\Phi}^{\#} h\|_{L^2(\Theta)}) \\ &\le \sqrt{\kappa} C_\phi (L + \sqrt{\kappa} C_\phi R), \end{aligned} \tag{6.23}$$

where we have used the inequality $\|K_{x,\Phi}\|_{\mathcal{L}(L^2(\Theta),\mathcal{H}_K)} = \|K_{x,\Phi}^{\#}\|_{\mathcal{L}(L^2(\Theta),\mathcal{H}_K)} \le \sqrt{\kappa} C_\phi$. This is an immediate consequence of Equation (6.5) and Equation (6.1), as well as Assumption 6.13 and Assumption 6.12.

Equation (6.23) also implies

$$\mathbb{E}_{X,Y\sim\rho}[\|\xi_1(X,Y)\|_{\mathcal{H}_K}^2] \le \kappa C_\phi (L + \sqrt{\kappa} C_\phi R)^2.$$

Hence we can apply Lemma 6.21, yielding that with probability at least $1-\eta$,

$$\begin{aligned}\|A_{\mathbf{x},\Phi}^{\#} \mathbf{y} - T_{\mathbf{x},\Phi} h_{\mathcal{H}_K}\|_{\mathcal{H}_K} &\le (\sqrt{\kappa} C_\phi L + \kappa C_\phi^2 R) \log\left(2/\eta\right) \left(\frac{4}{n} + \frac{2}{\sqrt{n}}\right) \\ &\le 6(\sqrt{\kappa} C_\phi L + \kappa C_\phi^2 R) \frac{\log\left(2/\eta\right)}{\sqrt{n}}. \end{aligned}$$

■

Lemma 6.24. *Let $0 < \eta < 1$, then with probability at least $1-\eta$*

$$\|T_{\mathbf{x},\Phi} - T_\Phi\|_{\mathcal{L}_2(\mathcal{H}_K)} \le \delta_2(n,d,\eta),$$

with δ_2 defined as

$$\delta_2(n,d,\eta) := 6\kappa C_\phi^2 \frac{\log\left(2/\eta\right)\sqrt{d}}{\sqrt{n}}. \tag{6.24}$$

Proof We introduce the function $\xi_2 : \mathcal{Z} \longrightarrow \mathcal{L}_2(\mathcal{H}_K)$ as $\xi_2 : x, y \longmapsto T_{x,\Phi}$.

We have that

$$\mathbb{E}_{X,Y\sim\rho}[\xi_2(X,Y)] = \int_{\mathcal{X}} T_{x,\Phi} \, d\rho_X(x) = T_\Phi.$$

And from Lemma 6.20, we have almost surely

$$\|\xi_2(X,Y)\|_{\mathcal{L}_2(\mathcal{H}_K)} \leq \kappa C_\phi^2 \sqrt{d},$$

which implies as well

$$\mathbb{E}_{X,Y\sim\rho}[\|\xi_2(X,Y)\|^2_{\mathcal{L}_2(\mathcal{H}_K)}] \leq \kappa^2 C_\phi^4 d.$$

Since K is continuous and $\mathcal{X}$ is separable, $\mathcal{H}_K$ is separable. As a consequence the space $\mathcal{L}_2(\mathcal{H}_K)$ is also separable, we can thus apply Lemma 6.21, yielding that with probability at least $1-\eta$,

$$\begin{aligned}\|T_{\mathbf{x},\Phi} - T_\Phi\|_{\mathcal{L}_2(\mathcal{H}_K)} &\leq \kappa C_\phi^2 \sqrt{d} \log\left(4/\eta\right)\left(\frac{4}{n} + \frac{2}{\sqrt{n}}\right) \\ &\leq 6\kappa C_\phi^2 \sqrt{d}\frac{\log\left(2/\eta\right)}{\sqrt{n}}.\end{aligned}$$

■

Lemma 6.25. *Let $0 < \eta < 1$, then with probability at least $1-\eta$ the two following inequalities hold:*

$$\begin{aligned}\|A^{\#}_{\mathbf{x},\Phi}\mathbf{y} - T_{\mathbf{x},\Phi} h_{\mathcal{H}_K}\|_{\mathcal{H}_K} &\leq \delta_1(n, \eta/2) \\ \|T_{\mathbf{x},\Phi} - T_\Phi\|_{\mathcal{L}_2(\mathcal{H}_K)} &\leq \delta_2(n, d, \eta/2),\end{aligned}$$

with δ_1 and δ_2 defined respectively in Equation (6.22) and Equation (6.24).

Proof This is a union bound using Lemma 6.23 and Lemma 6.24. ■

6.2.4 Final proof

We are now ready to prove Proposition 6.14. We follow the same proof strategy as (Baldassarre et al., 2012). To that end, we first prove the following intermediate proposition of which Proposition 6.14 is a direct consequence.

Proposition 6.26. *Let $0 < \eta < 1$, provided λ is taken such that*

$$\lambda \geq 6\kappa C_\phi^2 \frac{\log\left(4/\eta\right)\sqrt{d}}{\sqrt{n}} = \delta_2(n, d, \eta/2), \tag{6.25}$$

we have with probability at least $1-\eta$ that

$$\mathcal{R}(\Phi \circ h^\lambda_{\mathbf{z}}) - \mathcal{R}(\Phi \circ h_{\mathcal{H}_K}) \leq \frac{9}{2}\left(\frac{36(\sqrt{\kappa}C_\phi L + \kappa C_\phi^2 R)^2 \log\left(4/\eta\right)^2}{\lambda n} + \lambda R^2\right). \tag{6.26}$$

Proof

We introduce h^λ as

$$h^\lambda := (\mathrm{T}_{\mathbf{x},\Phi} + \lambda \mathrm{I})^{-1} \mathrm{T}_{\mathbf{x},\Phi} h_{\mathcal{H}_\mathrm{K}}. \tag{6.27}$$

We consider the following decomposition of the risk using Equation (6.16),

$$\begin{aligned}\mathcal{R}(\Phi \circ h_\mathbf{z}^\lambda) - \mathcal{R}(\Phi \circ h_{\mathcal{H}_\mathrm{K}}) &= \|\sqrt{\mathrm{T}_\Phi}(h_\mathbf{z}^\lambda - h_{\mathcal{H}_\mathrm{K}})\|_{\mathcal{H}_\mathrm{K}}^2 \\ &\leq 2\|\sqrt{\mathrm{T}_\Phi}(h_\mathbf{z}^\lambda - h^\lambda)\|_{\mathcal{H}_\mathrm{K}}^2 + 2\|\sqrt{\mathrm{T}_\Phi}(h^\lambda - h_{\mathcal{H}_\mathrm{K}})\|_{\mathcal{H}_\mathrm{K}}^2. \end{aligned} \tag{6.28}$$

We first bound the term $\|\sqrt{\mathrm{T}_\Phi}(h_\mathbf{z}^\lambda - h^\lambda)\|_{\mathcal{H}_\mathrm{K}}$. Using the expression of $h_\mathbf{z}^\lambda$ from Lemma 6.19, we have that

$$\begin{aligned}\sqrt{\mathrm{T}_\Phi}(h_\mathbf{z}^\lambda - h^\lambda) = \sqrt{\mathrm{T}_{\mathbf{x},\Phi}}(\mathrm{T}_{\mathbf{x},\Phi} + \lambda \mathrm{I})^{-1}(\mathrm{A}_{\mathbf{x},\Phi}^\# \mathbf{y} - \mathrm{T}_{\mathbf{x},\Phi} h_{\mathcal{H}_\mathrm{K}}) \\ + (\sqrt{\mathrm{T}_\Phi} - \sqrt{\mathrm{T}_{\mathbf{x},\Phi}})(\mathrm{T}_{\mathbf{x},\Phi} + \lambda \mathrm{I})^{-1}(\mathrm{A}_{\mathbf{x},\Phi}^\# \mathbf{y} - \mathrm{T}_{\mathbf{x},\Phi} h_{\mathcal{H}_\mathrm{K}}).\end{aligned} \tag{6.29}$$

For all $a \geq 0$, $\frac{\sqrt{a}}{a+\lambda} \leq \frac{1}{2\sqrt{\lambda}}$, since $\mathrm{T}_{\mathbf{x},\Phi}$ is positive. Hence, by spectral theorem we obtain

$$\|\sqrt{\mathrm{T}_{\mathbf{x},\Phi}}(\mathrm{T}_{\mathbf{x},\Phi} + \lambda \mathrm{I})^{-1}\|_{\mathcal{L}(\mathcal{H}_\mathrm{K})} \leq \max_{a \in \mathrm{Sp}(\mathrm{T}_{\mathbf{x},\Phi})} \frac{\sqrt{a}}{a+\lambda} \leq \max_{a \in \mathbb{R}_+} \frac{\sqrt{a}}{a+\lambda} \leq \frac{1}{2\sqrt{\lambda}}, \tag{6.30}$$

where $\mathrm{Sp}(\mathrm{T}_{\mathbf{x},\Phi})$ denotes the spectrum of $\mathrm{T}_{\mathbf{x},\Phi}$.

Similarly, since for all $a \geq 0$, $\frac{1}{a+\lambda} \leq \frac{1}{\lambda}$, we have as well

$$\|(\mathrm{T}_{\mathbf{x},\Phi} + \lambda \mathrm{I})^{-1}\|_{\mathcal{L}(\mathcal{H}_\mathrm{K})} \leq \frac{1}{\lambda}.$$

Taking the norm in Equation (6.29), applying Minkowski's inequality and using Lemma 6.22 as well as the last two displays yields

$$\|\sqrt{\mathrm{T}_\Phi}(h_\mathbf{z}^\lambda - h^\lambda)\|_{\mathcal{H}_\mathrm{K}} \leq \|\mathrm{A}_{\mathbf{x},\Phi}^\# \mathbf{y} - \mathrm{T}_{\mathbf{x},\Phi} h_{\mathcal{H}_\mathrm{K}}\|_{\mathcal{H}_\mathrm{K}} \left(\frac{1}{2\sqrt{\lambda}} + \frac{\sqrt{\|\mathrm{T}_\Phi - \mathrm{T}_{\mathbf{x},\Phi}\|_{\mathcal{L}(\mathcal{H}_\mathrm{K})}}}{\lambda} \right). \tag{6.31}$$

Now dealing with the term on the right-hand side in Equation (6.28), using the definition of h^λ in Equation (6.27):

$$\begin{aligned}\sqrt{\mathrm{T}_\Phi}(h_{\mathcal{H}_\mathrm{K}} - h^\lambda) &= \sqrt{\mathrm{T}_\Phi}(\mathrm{I} - (\mathrm{T}_{\mathbf{x},\Phi} + \lambda \mathrm{I})^{-1} \mathrm{T}_{\mathbf{x},\Phi}) h_{\mathcal{H}_\mathrm{K}} \\ &= (\sqrt{\mathrm{T}_\Phi} - \sqrt{\mathrm{T}_{\mathbf{x},\Phi}})(\mathrm{I} - (\mathrm{T}_{\mathbf{x},\Phi} + \lambda \mathrm{I})^{-1} \mathrm{T}_{\mathbf{x},\Phi}) h_{\mathcal{H}_\mathrm{K}} \\ &+ \sqrt{\mathrm{T}_{\mathbf{x},\Phi}}(\mathrm{I} - (\mathrm{T}_{\mathbf{x},\Phi} + \lambda \mathrm{I})^{-1} \mathrm{T}_{\mathbf{x},\Phi}) h_{\mathcal{H}_\mathrm{K}}.\end{aligned} \tag{6.32}$$

Since for all $a \geq 0$, $\sqrt{a}\left(1 - \frac{a}{a+\lambda}\right) = \frac{\sqrt{a}\lambda}{a+\lambda} \leq \frac{1}{2}\sqrt{\lambda}$, using the same arguments as in Equation (6.30) yields

$$\|\sqrt{\mathrm{T}_{\mathbf{x},\Phi}}(\mathrm{I} - (\mathrm{T}_{\mathbf{x},\Phi} + \lambda \mathrm{I})^{-1} \mathrm{T}_{\mathbf{x},\Phi})\|_{\mathcal{L}(\mathcal{H}_\mathrm{K})} \leq \frac{1}{2}\sqrt{\lambda}.$$

Moreover, since for all $a \geq 0$, $1 - \frac{a}{a+\lambda} = \frac{\lambda}{a+\lambda} \leq 1$, similarly we obtain

$$\|\mathrm{I} - (\mathrm{T}_{\mathbf{x},\Phi} + \lambda \mathrm{I})^{-1} \mathrm{T}_{\mathbf{x},\Phi}\|_{\mathcal{L}(\mathcal{H}_\mathrm{K})} \leq 1.$$

Thus, taking the norm in Equation (6.32), using Minkowski's inequality, Lemma 6.22 and Equation (6.8) yields

$$\|\sqrt{\mathrm{T}_\Phi}(h_{\mathcal{H}_\mathrm{K}} - h^\lambda)\|_{\mathcal{H}_\mathrm{K}} \leq R\sqrt{\|\mathrm{T}_\Phi - \mathrm{T}_{\mathbf{x},\Phi}\|_{\mathcal{L}(\mathcal{H}_\mathrm{K})}} + \frac{R}{2}\sqrt{\lambda}. \tag{6.33}$$

Combining Equation (6.31) and Equation (6.33) with Lemma 6.25, for $0 < \eta < 1$, we have with probability at least $1 - \eta$

$$\|\sqrt{\mathrm{T}_\Phi}(h^\lambda_{\mathbf{z}} - h^\lambda)\|_{\mathcal{H}_\mathrm{K}} \leq \delta_1(n, \eta/2)\left(\frac{1}{2\sqrt{\lambda}} + \frac{\sqrt{\delta_2(n,d,\eta/2)}}{\lambda}\right)$$
$$\|\sqrt{\mathrm{T}_\Phi}(h_{\mathcal{H}_\mathrm{K}} - h^\lambda)\|_{\mathcal{H}_\mathrm{K}} \leq R\sqrt{\delta_2(n,d,\eta/2)} + \frac{R}{2}\sqrt{\lambda}.$$

Using the condition on λ given by Equation (6.25), still with probability at least $1 - \eta$ it holds that

$$\|\sqrt{\mathrm{T}_\Phi}(h^\lambda_{\mathbf{z}} - h^\lambda)\|_{\mathcal{H}_\mathrm{K}} \leq \frac{3}{2\sqrt{\lambda}}\delta_1(n, \eta/2), \tag{6.34}$$
$$\|\sqrt{\mathrm{T}_\Phi}(h_{\mathcal{H}_\mathrm{K}} - h^\lambda)\|_{\mathcal{H}_\mathrm{K}} \leq \frac{3R}{2}\sqrt{\lambda}. \tag{6.35}$$

Combining Equation (6.34) and Equation (6.35) into Equation (6.28) yields that with probability at least $1 - \eta$,

$$\mathcal{R}(\Phi \circ h^\lambda_{\mathbf{z}}) - \mathcal{R}(\Phi \circ h_{\mathcal{H}_\mathrm{K}}) \leq \frac{9}{2}\left(\frac{\delta_1(n, \eta/2)^2}{\lambda} + R^2\lambda\right).$$

■

In Proposition 6.26, we have a compromise in λ in the two terms. Taking $\lambda = \mathcal{O}\left(\frac{1}{\sqrt{n}}\right)$ yields the best compromise. So as to satisfy the condition from Equation (6.25), we take $\lambda = 6\kappa C_\phi^2 \frac{\log(4/\eta)\sqrt{d}}{\sqrt{n}}$, which after simplifications in the constants yields Proposition 6.14.

6.3 Excess risk bound for the plug-in ridge estimator

In this section, we focus on the functional output regression case. More precisely, we set $\mathcal{Y} = \mathrm{L}^2(\Theta)$ with $\Theta \subset \mathbb{R}^b$ some compact set.

Remark 6.27. *For fully-observed functions, the excess risk bound in Proposition 6.14 applies provided the proper assumptions are verified, since indeed the space $\mathsf{L}^2(\Theta)$ is a separable Hilbert space.*

Suppose now we are in the partially-observed setting: we observe the output functions at random locations distributed on Θ according to a probability measure μ. More precisely, let us suppose that for all $i \in [\![n]\!]$, we observe the function y_i at the locations $\boldsymbol{\theta}_i \in \Theta^m$ resulting in the observations $\tilde{y}_i \in \mathbb{R}^m$. The observed sample has the form

$$\tilde{\mathbf{z}} := (x_i, (\boldsymbol{\theta}_i, \tilde{y}_i))_{i=1}^n.$$

We introduce the notation $\tilde{\mathbf{y}} := (\tilde{y}_i)_{i=1}^n$ and highlight that since there is no added noise, we have for all $i \in [\![n]\!]$

$$\tilde{y}_i = (y_i(\theta_{is}))_{s=1}^m.$$

To simplify the exposition, we suppose that we are given the same number of observations m per function. We also assume that Θ is a normalized domain such that

$$\int_\Theta 1 \mathrm{d}\theta = 1. \tag{6.36}$$

We treat here the simplest eventuality for the distribution of the locations to benefit from classic Monte Carlo convergence results: we suppose that μ is a uniform distribution over Θ.

Assumption 6.28. *μ is a uniform probability measure over the compact domain Θ.*

Moreover, to derive the bounds, we need to make sure that the functions involved are uniformly bounded on the domain Θ (in the almost sure sense for the output functions). We therefore make the following additional assumptions.

Assumption 6.29. *There exists $M(d) \geq 0$ such that for all $\theta \in \Theta$ and for all $l \in [\![d]\!]$, $|\phi_l(\theta)| \leq M(d)$.*

We also make the same type of assumption on the observed functions in the almost sure sense.

Assumption 6.30. *There exists $L \geq 0$ such that for all $\theta \in \Theta$, almost surely*

$$|\mathsf{Y}(\theta)| \leq L.$$

Remark 6.31. *This assumption is a bit stronger than Assumption 6.13 which we made to derive the excess risk bound for outputs in a separable Hilbert space in Proposition 6.14.*

Remark 6.32. *The dependence in d is specific to the family to which $\boldsymbol{\phi}$ belongs; for wavelets we have $M(d) = 2^{r(\Theta,d)/2} \max_{\theta\in\Theta} |\psi(\theta)|$ with ψ the mother wavelet and $r(\Theta, d) \in \mathbb{N}$ the number of dilatations included in $\boldsymbol{\phi}$, whereas for a Fourier dictionary we have $M(d) = 1$.*

We then have the following excess risk bound for the plug-in ridge estimator from Definition 5.18.

Proposition 6.33. *Let $0 < \eta < 1$ and take $\lambda = \lambda_n^*(\eta/3) := 6\kappa C_\phi^2 \frac{\log(6/\eta)\sqrt{d}}{\sqrt{n}}$, then with probability at least $1-\eta$, we have that*

$$\mathcal{R}(\Phi \circ h_{\mathbf{z}}^{\lambda}) - \mathcal{R}(\Phi \circ h_{\mathcal{H}_K}) \leq \left(\frac{B_2(d)\sqrt{n}}{m^2} + \frac{B_3(d)}{m^{3/2}} + \frac{9C(d)^2}{2\sqrt{n}m} + \frac{B_4(d)}{\sqrt{n}} \right) \log(6/\eta),$$

with $C(d) := \frac{LM(d)}{C_\phi}$, $B_2(d) := 18\sqrt{d}\left(C(d) + \frac{R}{\sqrt{d}}\right)^2$, $B_3(d) := B_2(d) - 18\frac{R^2}{\sqrt{d}}$, $B_4(d) := \frac{81}{2}\left(\frac{B_0}{\sqrt{d}} + B_1\sqrt{d}\right)$ and B_0 and B_1 are defined as Proposition 6.14.

We highlight that if $m \asymp \sqrt{n}$, then this bounds yields consistency for the plug-in ridge estimator.

Sketch of proof. We use the same strategy as for proving Proposition 6.14, except that we define an additional empirical approximation of the operator A_Φ linked to the fact that the output functions are partially-observed. Then when separating the excess risk in different terms, this results in a third term which we control with high probability leveraging the concentration of the empirical operator using the partial observations to the one using the full output functions.

6.4 Proof of the bound for the plug-in ridge estimator

In Section 6.4.1, we reformulate the plug-in ridge estimator in terms of empirical approximations of the operators A_Φ and T_Φ using the partially observed functions. Then in Section 6.4.2 we introduce and prove a concentration result for the term involving the partial observations in our division of the excess risk. Finally Section 6.4.3 put these different elements together along with results from Section 6.2 to prove Proposition 6.33.

6.4.1 Reformulation of the estimator in terms of operators

For $i \in [\![n]\!]$, we recall the definition of the approximated projection operator $\tilde{\Phi}_i$ at locations $\boldsymbol{\theta}_i$ given at the beginning of the chapter in Equation (5.28).

$$\tilde{\Phi}_i : \begin{pmatrix} \mathbb{R}^d & \to & \mathbb{R}^{m_i} \\ a & \mapsto & \sum_{l=1}^d \mathbf{a}_l \phi_l(\boldsymbol{\theta}_i) \end{pmatrix},$$

Let us recall also that the solution when the output functions are fully observed reads (Lemma 6.19):

$$h_{\mathbf{z}}^{\lambda} = (T_{\mathbf{x},\Phi} + \lambda I)^{-1} A_{\mathbf{x},\Phi}^{\#} \mathbf{y},$$

with

$$A_{\mathbf{x},\Phi}^{\#} \mathbf{w} = \frac{1}{n} \sum_{i=1}^n K_{x_i} \Phi^{\#} w_i \quad \text{for} \quad \mathbf{w} \in L^2(\Theta)^n.$$

We now consider that for $i \in [\![n]\!]$, the i-th output function is partially observed at locations $(\boldsymbol{\theta}_i)_{i=1}^n$ and define an estimator in this setting. To that end let us introduce the following operator

$$\mathrm{A}^{\#}_{\mathbf{x},\tilde{\Phi}} \tilde{\mathbf{w}} = \frac{1}{n} \sum_{i=1}^{n} \mathsf{K}_{x_i} \frac{\tilde{\Phi}_i^{\#}}{m} \tilde{w}_i \quad \text{with} \quad \tilde{\mathbf{w}} \in \mathbb{R}^{n \times m},$$

The solution we consider using partially observed functions is then

$$h^{\lambda}_{\tilde{\mathbf{z}}} := (\mathrm{T}_{\mathbf{x},\Phi} + \lambda \mathrm{I})^{-1} \mathrm{A}^{\#}_{\mathbf{x},\tilde{\Phi}} \tilde{\mathbf{y}}.$$

It is another equivalent expression in terms of operators for the plug-in ridge estimator from Definition 5.18.

6.4.2 Concentration results

In this section, we prove an intermediate concentration result which bounds with high probability the deviation between the ridge estimator (fully-observed functions) and the plug-in ridge estimator (partially-observed functions). Then, using a union bound this result can be combined with the two intermediate probabilistic bounds already derived in Lemma 6.23 and Lemma 6.24. We can then prove the excess risk bound for the case of partially-observed functions presented in Proposition 6.33.

Lemma 6.34. *Let $0 < \eta < 1$, then with probability at least $1 - \eta$*

$$\|\mathrm{A}^{\#}_{\mathbf{x},\tilde{\Phi}} \tilde{\mathbf{y}} - \mathrm{A}^{\#}_{\mathbf{x},\Phi} \mathbf{y}\|_{\mathcal{H}_\mathsf{K}} \leq \delta_3(n, m, d, \eta),$$

with δ_3 defined as

$$\delta_3(n, m, d, \eta) := \left(\frac{4(L\sqrt{\kappa}\sqrt{d}M(d) + \sqrt{\kappa}C_{\boldsymbol{\phi}}R)}{m} + \frac{2L\sqrt{\kappa}\sqrt{d}M(d)}{\sqrt{n}\sqrt{m}} \right) \log(2/\eta). \tag{6.37}$$

Proof Let us define the function $\xi_3 : \mathcal{X} \times \mathsf{L}^2(\Theta) \times \Theta \longrightarrow \mathcal{H}_\mathsf{K}$ as

$$\xi_3 : (x, y, \theta) \longmapsto y(\theta)\mathsf{K}_x \boldsymbol{\phi}(\theta) - \mathsf{K}_x \Phi^{\#} y.$$

The proof relies on the fact that

$$\begin{aligned} \frac{1}{n} \sum_{i=1}^{n} \frac{1}{m} \sum_{s=1}^{m} \xi_3(x_i, y_i, \theta_{is}) &= \frac{1}{n} \sum_{i=1}^{n} \mathsf{K}_{x_i} \frac{\tilde{\Phi}_i^{\#}}{m} \tilde{y}_i - \mathsf{K}_{x_i} \Phi^{\#} y_i \\ &= \mathrm{A}^{\#}_{\mathbf{x},\tilde{\Phi}} \tilde{\mathbf{y}} - \mathrm{A}^{\#}_{\mathbf{x},\Phi} \mathbf{y}. \end{aligned}$$

We recall the definition of the vector-valued function associated to the dictionary which we introduced in Chapter 3:

$$\boldsymbol{\phi} : \begin{pmatrix} \Theta & \to & \mathbb{R}^d \\ \theta & \mapsto & (\phi_1(\theta), \phi_2(\theta), \cdots, \phi_d(\theta))^{\mathrm{T}} \end{pmatrix}.$$

Let $(\mathsf{X}_i, \mathsf{Y}_i)_{i=1}^n$ be n i.i.d. RVs distributed according to the distribution ρ. Let $(\vartheta_{is})_{i=1,s=1}^{n,m}$ be nm i.i.d. RVs distributed according to the distribution μ. For all $i \in [\![n]\!]$ and for all $s \in [\![m]\!]$ we then define the RVs W_{is} as

$$\begin{aligned} \mathsf{W}_{is} &:= \xi_3(\mathsf{X}_i, \mathsf{Y}_i, \vartheta_{is}) \\ &= \mathsf{Y}_i(\vartheta_{is})\mathsf{K}_{\mathsf{X}_i}\boldsymbol{\phi}(\vartheta_{is}) - \mathsf{K}_{\mathsf{X}_i}\Phi^{\#}\mathsf{Y}_i \\ &= \mathsf{Y}_i(\vartheta_{is})\mathsf{K}_{\mathsf{X}_i}\boldsymbol{\phi}(\vartheta_{is}) - \mathbb{E}[\mathsf{Y}_i(\vartheta)\mathsf{K}_{\mathsf{X}_i}\boldsymbol{\phi}(\vartheta)|\mathsf{X}_i, \mathsf{Y}_i], \end{aligned} \tag{6.38}$$

where the last line holds because μ is the uniform distribution and because we have assumed that $|\Theta| = \int_\Theta 1 \mathrm{d}\theta = 1$ in Equation (6.36).

We denote by $\mathbb{P}[.|\mathbf{z}]$ the probability conditional on the realization of the sample $\mathbf{z}$, thus

$$\mathbb{P}[.|\mathbf{z}] = \mathbb{P}[.|\mathsf{X}_i = x_i, \mathsf{Y}_i = y_i, \;\; i \in [\![n]\!]]$$

Then, Equation (6.38) implies that $\mathbb{E}[\mathsf{W}_{is}|\mathbf{z}] = 0$.

We define as well for all $s \in [\![m]\!]$,

$$\overline{\mathsf{W}}_s := \frac{1}{n}\sum_{i=1}^{n} \mathsf{W}_{is}.$$

We have almost surely that

$$\begin{aligned} \|\overline{\mathsf{W}}_s\|_{\mathcal{H}_\mathsf{K}} \leq \frac{1}{n}\sum_{i=1}^{n} \|\mathsf{W}_{is}\|_{\mathcal{H}_\mathsf{K}} &\leq \frac{1}{n}\sum_{i=1}^{n} ((|\mathsf{Y}_i(\vartheta_{is})|)\|\mathsf{K}_{\mathsf{X}_i}\boldsymbol{\phi}(\vartheta_{is})\|_{\mathcal{H}_\mathsf{K}} + \|\mathsf{K}_{\mathsf{X}_i}\Phi^{\#}\mathsf{Y}_i\|_{\mathcal{H}_\mathsf{K}}) \\ &\leq L\sqrt{\kappa}\sqrt{d}M(d) + \sqrt{\kappa}C_{\boldsymbol{\phi}}R. \end{aligned}$$

We have used Assumption 6.30 and Assumption 6.29 as well as Equation (6.6).

Since for all $s \in [\![m]\!]$, the RVs $(\mathsf{W}_{is})_{i=1}^n$ are independent conditionally on $\mathbf{z}$:

$$\mathbb{E}[\|\overline{\mathsf{W}}_s\|_{\mathcal{H}_\mathsf{K}}^2|\mathbf{z}] = \frac{1}{n^2}\sum_{i=1}^{n} \mathbb{E}[\|\mathsf{W}_{is}\|_{\mathcal{H}_\mathsf{K}}^2|\mathbf{z}]. \tag{6.39}$$

Using the fact that $\mathbb{E}[\mathsf{Y}_i(\vartheta_{is})\mathsf{K}_{\mathsf{X}_i}\boldsymbol{\phi}(\vartheta_{is})|\mathbf{z}] = \mathsf{K}_{x_i}\Phi^{\#}y_i$, the identity

$$\mathbb{E}[\|\mathsf{U} - \mathbb{E}[\mathsf{U}]\|_{\mathcal{H}_\mathsf{K}}^2] = \mathbb{E}[\|\mathsf{U}\|_{\mathcal{H}_\mathsf{K}}^2]$$

gives us

$$\mathbb{E}[\|\mathsf{W}_{is}\|_{\mathcal{H}_\mathsf{K}}^2|\mathbf{z}] = \mathbb{E}[\|\mathsf{Y}_i(\vartheta_{is})\mathsf{K}_{\mathsf{X}_i}\boldsymbol{\phi}(\vartheta_{is})\|_{\mathcal{H}_\mathsf{K}}^2|\mathbf{z}]. \tag{6.40}$$

Then using Equation (6.40) into Equation (6.39) along with Assumption 6.30 and Assumption 6.29

$$\mathbb{E}[\|\overline{\mathsf{W}}_s\|_{\mathcal{H}_\mathsf{K}}^2|\mathbf{z}] \leq \frac{1}{n}L^2\kappa d M(d)^2.$$

We can apply Lemma 6.21 to obtain

$$\mathbb{P}\left[\left\|\frac{1}{m}\sum_{s=1}^{m}\overline{\mathsf{W}}_s\right\|_{\mathcal{H}_\mathsf{K}} \leq \left(\frac{4(L\sqrt{\kappa}\sqrt{d}M(d)+\sqrt{\kappa}C_\phi R)}{m}+\frac{2L\sqrt{\kappa}\sqrt{d}M(d)}{\sqrt{n}\sqrt{m}}\right)\log(2/\eta)\middle|\mathbf{z}\right]\geq 1-\eta.$$

Multiplying the above inequality by $\mathbb{P}[\mathbf{z}]$ and integrating over $\mathbf{z}\in\mathcal{Z}^n$, yields

$$\mathbb{P}\left[\left\|\mathsf{A}^{\#}_{\mathbf{x},\tilde{\Phi}}\tilde{\mathbf{y}}-\mathsf{A}^{\#}_{\mathbf{x},\Phi}\mathbf{y}\right\|_{\mathcal{H}_\mathsf{K}} \leq \left(\frac{4(L\sqrt{\kappa}\sqrt{d}M(d)+\sqrt{\kappa}C_\phi R)}{m}+\frac{2L\sqrt{\kappa}\sqrt{d}M(d)}{\sqrt{n}\sqrt{m}}\right)\log(2/\eta)\right]\geq 1-\eta.$$

■

Lemma 6.35. *Let $0<\eta<1$, then with probability at least $1-\eta$ the three following inequalities hold:*

$$\|\mathsf{A}^{\#}_{\mathbf{x},\Phi}\mathbf{y}-\mathsf{T}_{\mathbf{x},\Phi}h_{\mathcal{H}_\mathsf{K}}\|_{\mathcal{H}_\mathsf{K}}\leq\delta_1(n,\eta/3) \tag{6.41}$$

$$\|\mathsf{T}_{\mathbf{x},\Phi}-\mathsf{T}_\Phi\|_{\mathcal{L}_2(\mathcal{H}_\mathsf{K})}\leq\delta_2(n,d,\eta/3) \tag{6.42}$$

$$\|\mathsf{A}^{\#}_{\mathbf{x},\tilde{\Phi}}\tilde{\mathbf{y}}-\mathsf{A}^{\#}_{\mathbf{x},\Phi}\mathbf{y}\|_{\mathcal{H}_\mathsf{K}}\leq\delta_3(n,m,d,\eta/3), \tag{6.43}$$

with δ_1, δ_2 and δ_3 respectively defined as in Equation (6.22), Equation (6.24) and Equation (6.37).

Proof This Lemma is an union bound using Lemma 6.23, Lemma 6.24 and Lemma 6.34. ■

6.4.3 Final proof

We are now ready to prove Proposition 6.33. To do so we prove the following intermediate result of which the proposition of interest is a direct consequence.

Proposition 6.36. *Let $0<\eta<1$, provided λ is taken such that*

$$\lambda\geq 6\kappa C_\phi^2\frac{\log\left(6/\eta\right)\sqrt{d}}{\sqrt{n}}=\delta_2(n,d,\eta/3), \tag{6.44}$$

we have with probability at least $1-\eta$ that

$$\mathcal{R}(\Phi\circ h^\lambda_{\tilde{\mathbf{z}}})-\mathcal{R}(\Phi\circ h_{\mathcal{H}_\mathsf{K}})\leq\frac{27}{4}\left(\left(\frac{A_0(d)^2}{\lambda m^2}+\frac{2A_0(d)A_1(d)}{\lambda\sqrt{n}m^{3/2}}+\frac{A_1(d)^2}{\lambda nm}+\frac{A_2^2}{\lambda n}\right)\log(6/\eta)^2+\lambda R^2\right), \tag{6.45}$$

with

$$A_0(d):=4(L\sqrt{\kappa}\sqrt{d}M(d)+\sqrt{\kappa}C_\phi R)$$
$$A_1(d):=2L\sqrt{\kappa}\sqrt{d}M(d)$$
$$A_2:=6(\sqrt{\kappa}C_\phi L+\kappa C_\phi^2 R).$$

Proof Taking h^λ as in Equation (6.27), we consider the following decomposition of the risk using Equation (6.16)

$$\begin{aligned}\mathcal{R}(\Phi \circ h^\lambda_{\tilde{\mathbf{z}}}) - \mathcal{R}(\Phi \circ h_{\mathcal{H}_\mathsf{K}}) &= \|\sqrt{T_\Phi}(h^\lambda_{\tilde{\mathbf{z}}} - h_{\mathcal{H}_\mathsf{K}})\|^2_{\mathcal{H}_\mathsf{K}} \\ &\leq 3\|\sqrt{T_\Phi}(h^\lambda_{\tilde{\mathbf{z}}} - h^\lambda_{\mathbf{z}})\|^2_{\mathcal{H}_\mathsf{K}} + 3\|\sqrt{T_\Phi}(h^\lambda_{\mathbf{z}} - h^\lambda)\|^2_{\mathcal{H}_\mathsf{K}} + 3\|\sqrt{T_\Phi}(h^\lambda - h_{\mathcal{H}_\mathsf{K}})\|^2_{\mathcal{H}_\mathsf{K}}.\end{aligned} \tag{6.46}$$

We focus on the term on the left as we have already controlled the two others in the proof of Proposition 6.26. Using the same strategy as for proving Equation (6.31), we get that

$$\|\sqrt{T_\Phi}(h^\lambda_{\tilde{\mathbf{z}}} - h^\lambda_{\mathbf{z}})\|_{\mathcal{H}_\mathsf{K}} \leq \|\mathrm{A}^{\#}_{\mathbf{x},\tilde{\Phi}}\tilde{\mathbf{y}} - \mathrm{A}^{\#}_{\mathbf{x},\Phi}\mathbf{y}\|_{\mathcal{H}_\mathsf{K}}\left(\frac{1}{2\sqrt{\lambda}} + \frac{\sqrt{\|\mathrm{T}_\Phi - \mathrm{T}_{\mathbf{x},\Phi}\|_{\mathcal{L}(\mathcal{H}_\mathsf{K})}}}{\lambda}\right). \tag{6.47}$$

Combining Equation (6.31), Equation (6.33) and Equation (6.47) with Lemma 6.35, for $0 < \eta < 1$, the three following inequalities are verified with probability at least $1-\eta$

$$\begin{aligned}\|\sqrt{T_\Phi}(h^\lambda_{\tilde{\mathbf{z}}} - h^\lambda_{\mathbf{z}})\|_{\mathcal{H}_\mathsf{K}} &\leq \delta_3(n,m,d,\eta/3)\left(\frac{1}{2\sqrt{\lambda}} + \frac{\sqrt{\delta_2(n,d,\eta/3)}}{\lambda}\right) \\ \|\sqrt{\mathrm{T}_\Phi}(h^\lambda_{\mathbf{z}} - h^\lambda)\|_{\mathcal{H}_\mathsf{K}} &\leq \delta_1(n,\eta/3)\left(\frac{1}{2\sqrt{\lambda}} + \frac{\sqrt{\delta_2(n,d,\eta/3)}}{\lambda}\right) \\ \|\sqrt{\mathrm{T}_\Phi}(h_{\mathcal{H}_\mathsf{K}} - h^\lambda)\|_{\mathcal{H}_\mathsf{K}} &\leq R\sqrt{\delta_2(n,d,\eta/3)} + \frac{R}{2}\sqrt{\lambda}.\end{aligned}$$

Using the condition on λ given by Equation (6.44), still with probability at least $1-\eta$:

$$\|\sqrt{T_\Phi}(h^\lambda_{\tilde{\mathbf{z}}} - h^\lambda_{\mathbf{z}})\|_{\mathcal{H}_\mathsf{K}} \leq \frac{3}{2\sqrt{\lambda}}\delta_3(n,m,d,\eta/3) \tag{6.48}$$

$$\|\sqrt{\mathrm{T}_\Phi}(h^\lambda_{\mathbf{z}} - h^\lambda)\|_{\mathcal{H}_\mathsf{K}} \leq \frac{3}{2\sqrt{\lambda}}\delta_1(n,\eta/3) \tag{6.49}$$

$$\|\sqrt{\mathrm{T}_\Phi}(h_{\mathcal{H}_\mathsf{K}} - h^\lambda)\|_{\mathcal{H}_\mathsf{K}} \leq \frac{3R}{2}\sqrt{\lambda}. \tag{6.50}$$

Combining Equation (6.48), Equation (6.49) and Equation (6.50) into Equation (6.46) yields that with probability at least $1-\eta$:

$$\mathcal{R}(\Phi \circ h^\lambda_{\tilde{\mathbf{z}}}) - \mathcal{R}(\Phi \circ h_{\mathcal{H}_\mathsf{K}}) \leq \frac{27}{4}\left(\frac{\delta_3(n,m,d,\eta/3)^2}{\lambda} + \frac{\delta_1(n,\eta/3)^2}{\lambda} + R^2\lambda\right).$$

In Proposition 6.36, we have a compromise in λ. Taking $\lambda = \mathcal{O}\left(\frac{1}{\sqrt{n}}\right)$ yields the best one. So as to satisfy the condition on λ–Equation (6.44)–we take $\lambda = 6\kappa C_\phi^2 \frac{\log(6/\eta)\sqrt{d}}{\sqrt{n}}$. After simplifications in the constants we get Proposition 6.33. ■

6.5 Conclusion

In this chapter, we have derived finite sample excess risk bounds for two estimators that we have proposed in Chapter 5: the ridge estimator for outputs in a separable Hilbert space, and the plug-in ridge estimator for functional output regression with partially observed output functions. These bounds give us an idea of the estimation error that is made compared to the best estimator in the hypothesis class predicting coordinates in a dictionary using a vv-RKHS. Therefore, the approximation aspects induced by the use of a dictionary are not tackled and are interesting extensions to consider for future work. Another interesting aspect is that we just proved the consistency and did not work to obtain optimal rates. Nevertheless, the proof technique should remain valid overall. For future work, we could then make further assumptions on the generating distribution as in Caponnetto and De Vito (2007) and work to adapt the proof.

7

A dual approach to functional output regression for sparsity or robustness

Contents

In this chapter, we introduce another approach to functional output regression (FOR). We have hinted in Chapter 5 that extending the scope of this problem to other losses is indeed desirable, for instance in the presence of outliers, to which the square loss is known to be very sensitive. In the scalar-valued case, the Huber loss (Huber, 1964) can be used to perform robust regression. Another desirable property is sparsity, which the ϵ-insensitive loss is well known to induce (Drucker et al., 1996). We propose extensions of those losses to measure discrepancy between functions as *convoluted losses*. These are expressed as *infimal convolution* between the $\|\cdot\|_{\mathcal{Y}}^2$ and either a p-norm term (Huber loss) or an indicator function of the p-norm ball (ϵ-insensitive loss). Through the parameter p, robustness to different kind of outliers or different kind of sparsity can be achieved. Focusing on function-valued reproducing kernel Hilbert spaces (fv-RKHSs) as hypothesis class, we investigate dual optimization of the associated empirical risk minimization problem. However, the resulting dual variables are infinite dimensional. This raises the question of how to represent the dual variables efficiently, yet in a way that is compatible with the terms appearing in the dual problem. This chapter corresponds to the contribution of

- A. Lambert, **D. Bouche**, Z. Szabó, and F. d'Alché-Buc. Functional Output Regression with Infimal Convolution: Exploring the Huber and ϵ-insensitive Losses In *International Conference on Machine Learning (ICML)*, 2022.

We solve the challenges gradually. In Section 7.1, we introduce the global framework of FOR with fv-RKHSs and infimal convoluted losses. Then, we focus on this problem with the Huber losses in Section 7.2 and propose accelerated proximal gradient descent algorithms based on finite dimensional representations of the dual variables. Section 7.3 builds on the representations and results from the previous section to propose similar algorithms for the ϵ-insensitive losses. Finally, in Section 7.4, we investigate numerically several properties of the proposed losses and estimators on three function-to-function regression benchmarks, while Section 7.5 provides some concluding remarks.

7.1 Functional output regression with infimal convoluted losses

First, let us briefly recall the problem of Hilbert-valued regression with a special focus on FOR.

7.1.1 FOR with function-valued RKHSs

Let X and Y be random variables with values respectively in $\mathcal{X}$ and $\mathcal{Y} = \mathsf{L}^2(\Theta, \mu)$; the space of square integrable functions on a given compact set $\Theta \subset \mathbb{R}^b$ with respect to μ a Borel probability measure. We want to estimate a prediction function on $\mathcal{X}$ that is statistically coherent with the unknown distribution ρ of $\mathsf{Z} = (\mathsf{X}, \mathsf{Y})$. We therefore rely on an i.i.d. sample $(x_i, y_i)_{i=1}^n$ for inference. We then choose a hypothesis class of function-valued functions $\mathcal{G} \subset \mathcal{F}(\mathcal{X}, \mathsf{L}^2(\Theta, \mu))$, a loss function on $\mathcal{Y}$ and minimize the associated empirical risk. It is sometimes regularized to avoid overfitting. We refer to Section 5.1.1 for a more detailed presentation regarding statistical learning with empirical risk minimization.

We now choose an fv-RKHS $\mathcal{H}_\mathsf{K}$ associated to an operator-valued kernel (OVK) K on $\mathcal{Y}$ as a hypothesis class. Let $L : \mathcal{Y} \to \mathbb{R}$ be a loss function (we consider here losses taking as input the difference between two elements). If we take $L(\cdot) = \|\cdot\|_\mathcal{Y}^2$, then we get the FOR problem in fv-RKHS $\mathcal{H}_\mathsf{K}$ proposed and studied in Lian (2007); Kadri et al. (2010, 2016), we refer also to Section 4.2. The associated empirical risk minimization problem is

$$\inf_{h \in \mathcal{H}_\mathsf{K}} \frac{1}{n} \sum_{i=1}^n L(y_i - h(x_i)) + \frac{\lambda}{2} \|h\|_{\mathcal{H}_\mathsf{K}}^2. \tag{7.1}$$

We will readily make the assumption that the kernel is *separable* of the form

$$\mathsf{K} = k_\mathcal{X} \mathrm{T}_{k_\Theta},$$

where $\mathrm{T}_{k_\Theta} \in \mathcal{L}(\mathcal{Y})$ is the integral operator associated to a kernel k_Θ on Θ and to the measure μ. We recall briefly its action on $y \in \mathcal{Y}$:

$$(\mathrm{T}_{k_\Theta} y)(\theta_1) = \int_\Theta y(\theta_2) k_\Theta(\theta_1, \theta_2) \mathrm{d}\mu(\theta_2),$$

for any $\theta_1 \in \Theta$. We refer the reader to Section 2.1.2 for more details on this operator. Note also, that it forces the predicted output functions to lie in $\mathcal{H}_{k_\Theta}$ the RKHS of k_Θ. We invite the reader to go back to Example 2.34 and Remark 2.35 for precisions.

Problem 7.1 indeed benefits from the representer theorem (see Micchelli and Pontil 2005 or Theorem 2.36). Therefore, for any solution $\hat{h}$ to Problem 7.1, there exists $\hat{\boldsymbol{\alpha}} \in \mathcal{Y}^n$ such that

$$\hat{h} = \frac{1}{\lambda n} \sum_{i=1}^{n} k_{\mathcal{X}}(\cdot, x_i) \mathrm{T}_{k_\Theta} \hat{\alpha}_i.$$

However, this exhibits the core issue when dealing with fv-RKHSs: the variables $\boldsymbol{\alpha} \in \mathcal{Y}^n$ are infinite-dimensional. Therefore another layer of representation must be added to make the problem solvable. As highlighted in Section 4.2.3, when L is the square loss, a closed-form exists for the optimal $\hat{\boldsymbol{\alpha}}$, however it involves the inversion of an infinite dimensional operator. One must rely on approximation. On the one hand in Lian (2007); Kadri et al. (2010), the integral operator is discretized upstream which leads to a closed-form involving the inverse of a $m \times m$ matrix, m being the number of discretization points. On the other hand, Kadri et al. (2016) relies on a finite rank approximation of the integral operator using its eigendecomposition to obtain a computable expression for the approximate coefficients. We also refer to Section 4.2 for more details.

7.1.2 Convoluted losses

Our aim in this chapter is to tackle Problem 7.1 for a wider family of losses. More precisely, we focus on two aspects: robustness to outliers and sparsity in the model's coefficients, which are two desirable properties a loss function can enforce.

Robust functional data analysis. The square loss leads to an estimate of the conditional expectation of the functional outputs given the input data. This is known to be very sensitive to outliers, which can stem for instance from malicious attacks or defective sensors. Several works in FDA have focused on various ways of dealing with functional outliers.To robustly estimate the mean of a set of functions, Cadre (2001) studies the estimation of the median for data lying in a Banach space. For functional linear regression, Zhu et al. (2011) propose a robust and fully Bayesian functional mixed-model, while Maronna and Yohai (2013) rely on a bounded loss function. Another alternative is introduced in Kalogridis and Van Aelst (2019) who combine a preprocessing robust functional principal component analysis with multivariate robust linear regression. It is also worth noting that the Huber loss has been used for functional inputs and scalar outputs for nonparametric estimation (Crambes et al., 2008), linear regression in Shin and Lee (2016) with an emphasis on theoretical properties, and in the wider context of M-estimation in Qingguo (2017); Boente et al. (2020). Another approach to robustness is to detect the outliers downstream, remove them from the dataset and then perform inference. We do not proceed this way, yet it is worth mentioning that non-supervised algorithms exist to detect functional anomalies. For instance, Nagy et al. (2017) leverage functional data depths to detect shape anomalies in functions, while Staerman et al. (2019) introduce an extension of the isolation forest algorithm (Liu et al., 2008) for functional data.

Local and global outliers. However, when dealing with functions, outliers can take many forms (Hubert et al., 2015). Among those varieties, the focus of this chapter is on the distinction between local and global outliers. Local ones can for instance be caused by defective sensors introducing irrelevant and/or extreme measurements, but only at a few locations. Local outliers can stem from registration issues which shift some functions, causing these to make no sense in their entirety compared to normal functions. Malicious attacks could also deliberately introduce outliers of any type. A possible scenario we experiment with is the addition of a minus sign to some functions to corrupt the model greatly. This would lead to global outliers displaying the same functional characteristics as normal functions.

Functional sparsity. Sparsity in the coefficients of the model is another desirable property, especially in the context of kernel methods. If only a relatively low number of coefficients is used, prediction is more efficient. Nevertheless for functions, sparsity could mean many things. As for robustness, the notion of locality is also interesting to consider. On the one hand, among the functional coefficients, few could be used, resulting in *global sparsity*. On the other hand, the functional coefficients could themselves be null at a high number of locations of the domain Θ yielding *local sparsity*.

Convoluted losses. To design losses which can enforce either robustness or sparsity in these ways for functional outputs, we investigate infimal convolutions (Definition 2.48) between the square norm $\|\cdot\|_{\mathcal{Y}}^2$ and a regularizing term depending on the $\|\cdot\|_p$ norm which will encourage the estimator to have the desired property. This enables us to extend the Huber loss as well as the smooth ϵ-insensitive loss (Lee et al., 2005) with the parameter $p \in \mathbb{N}$ giving us control over the local/global aspect of either robustness or sparsity. Several works have paved the way for the study of these losses. In the OVK literature, ϵ-insensitive losses for vector-valued regression have been introduced by Sangnier et al. (2017) for finite-dimensional outputs. They exploit parametric duality, which thanks to the form of the losses results in a tractable dual problem. From there they propose an efficient solver leading to data sparse estimators. For infinite dimensional outputs, Laforgue et al. (2020) explore a generalization of this approach encompassing both the Huber and the ϵ-insensitive losses. Our approach extends the families of losses that they propose through the use of specific p-norms in functional spaces, p acting as hinted earlier as a locality parameter. We also propose a new numerical approach to the problem which is more flexible. Notably, the choice of OVK in Laforgue et al. (2020) is restricted to separable ones for which an eigendecomposition of the integral operator associated to the output kernel is known in closed-form (see Section 2.1.2).

Now, let us define formally what a convoluted loss is.

Definition 7.1 (Convoluted loss)**.** *A convoluted loss has the form*

$$L = \frac{1}{2}\|\cdot\|_{\mathcal{Y}}^2 \square g, \tag{7.2}$$

for $g : \mathcal{Y} \to]-\infty, +\infty]$ some function.

Problem 7.1 does enjoy a representer theorem, however we will rely here rather on the framework of dualization (see Theorem 2.37) which additionally yields an equivalent dual problem. In our case, this problem is easier to solve properly thanks to a property of infimal convolution under the Fenchel-Legendre transform (see Proposition 2.50). More precisely, we have that

$$L^\star = \frac{1}{2}\|\cdot\|_{\mathcal{Y}}^2 + g^\star. \tag{7.3}$$

7.1.3 Learning with convoluted losses

From there, we can dualize Problem 7.1, this is a consequence of Theorem 2.37 which was introduced and proved in Brouard et al. (2016).

Lemma 7.2. *Let $L = \frac{1}{2}\|\cdot\|_{\mathcal{Y}}^2 \square g$ be a convoluted loss function for some $g : \mathcal{Y} \to]-\infty, +\infty]$. The solution to Problem 7.1 is given by*

$$\hat{h} = \frac{1}{\lambda n}\sum_{i=1}^{n} k_{\mathcal{X}}(\cdot, x_i)\mathrm{T}_{k_\Theta}\hat{\alpha}_i, \tag{7.4}$$

with $\hat{\boldsymbol{\alpha}} \in \mathcal{Y}^n$ being the solution to the dual problem

$$\inf_{\boldsymbol{\alpha}\in\mathcal{Y}} \sum_{i=1}^{n}\Big(\frac{1}{2}\|\alpha_i\|_{\mathcal{Y}}^2 - \langle\alpha_i, y_i\rangle_{\mathcal{Y}} + g^\star(\alpha_i)\Big) + \frac{1}{2\lambda n}\sum_{i=1}^{n}\sum_{j=1}^{n} k_{\mathcal{X}}(x_i, x_j)\langle\alpha_i, \mathrm{T}_{k_\Theta}\alpha_j\rangle_{\mathcal{Y}}. \tag{7.5}$$

Remark 7.3. *Indeed, in our case, for all $i \in [\![n]\!]$, L_{y_i} from Theorem 2.37 is*

$$L_{y_i} : y \mapsto L(y_i - y).$$

And since we have that for any function $f : \mathcal{Y} \to \mathbb{R}$

$$\Big(y \mapsto f(y_i - y)\Big)^\star = \Big(y \mapsto \langle y, y_i\rangle_{\mathcal{Y}} + f^\star(-y)\Big),$$

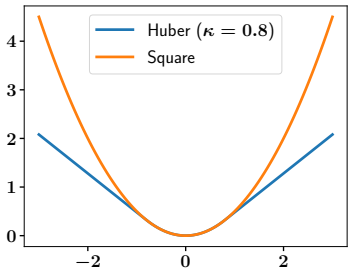
using Equation (7.3), we do find the objective in Problem 7.5.

Problem 7.5 (dual) is more manageable than Problem 7.1 (primal). The search space is reduced from $\mathcal{H}_K$ to $\mathcal{Y}^n$ with an explicit primal-dual link given by Equation (7.4). It remains challenging though in several aspects. It is a *composite problem*, meaning that the objective consists of the sum of a quadratic part (differentiable) and a non-differentiable term, here $g^\star$. Many algorithms have been proposed to solve these efficiently, the main family being that of *proximal algorithms* (Parikh and Boyd, 2014). They are applicable if we can compute the *proximity operator* of $g^\star$ fast enough (most of the time it implies that it should be known in closed-form). For a brief introduction, we refer the reader to Section 2.3. This first difficulty will restrict the p-norms we can consider. Another set of challenges comes from the functional dual variables $(\alpha_i)_{i=1}^n \in \mathcal{Y}^n$. We must approximate these in finite dimension so as to solve the problem. A key issue being that the proximal operator of $g^\star$ may not be computable using the finite dimensional representation.

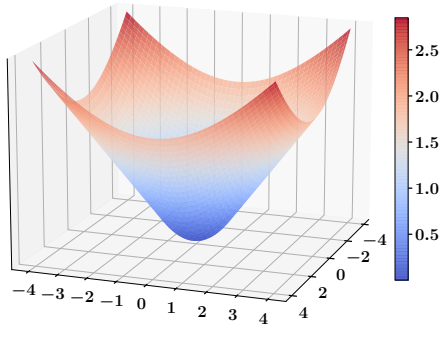
Now with the general framework in place, the next section focuses on robust functional regression with the Huber loss.

7.2 Robust FOR: learning with the functional Huber loss

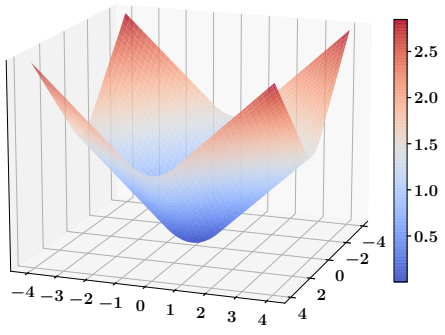
We now introduce the generalized Huber loss on $\mathcal{Y}$. The definition relies on functional p-norms, which we extend so that they can be equal to $+\infty$. Indeed, $\mathcal{Y} = \mathrm{L}^2(\Theta, \mu)$, therefore, for $p > 2$, in general we cannot state that for $y \in \mathcal{Y}$, $\|y\|_p$ is finite.



(a) Huber loss ($\kappa = 0.8$) and square loss



(b) H^2_κ ($\kappa = 0.8$)



(c) H^1_κ ($\kappa = 0.8$)

Figure 7.1: Huber losses on $\mathbb{R}$ and $\mathbb{R}^2$.

$$\forall y \in \mathcal{Y}, \quad \|y\|_p \begin{cases} = \left(\int_\Theta y(\theta)^p \mathrm{d}\mu(\theta) \right)^{\frac{1}{p}} & \text{if } y \in \mathrm{L}^p(\Theta, \mu), \\ = +\infty & \text{otherwise.} \end{cases} \tag{7.6}$$

Remark 7.4. *The possibility of the p-norm taking infinite values is not an issue since we rely on a framework of convex optimization which allows for this; we just need to ensure that the function is proper (*i.e. *not infinite over the whole set to optimize over). For more precision we refer the reader to Section 2.3 and especially to Definition 2.42.*

Our definition also relies on the classic notion of conjugate exponent which we recall.

Definition 7.5 (Conjugate exponent)**.** *Let $p \in [1, +\infty]$, the conjugate exponent of p is the exponent q such that*

$$\frac{1}{p} + \frac{1}{q} = 1,$$

with the convention that $\frac{1}{+\infty} = 0$.

We can then define the Huber loss on $\mathcal{Y}$.

Definition 7.6 ($\mathcal{Y}$-Huber loss)**.** *Let* $\kappa > 0$, $p \in [1, +\infty]$ *and let* q *be the conjugate exponent of* p*. We define the Huber loss with parameters* (κ, p) *as*

$$H_\kappa^p := \frac{1}{2}\|\cdot\|_{\mathcal{Y}}^2 \square \kappa\|\cdot\|_p. \tag{7.7}$$

We display this loss for $\mathcal{Y} = \mathbb{R}$ in Figure 7.1a and for $\mathcal{Y} = \mathbb{R}^2$ with $p = 2$ in Figure 7.1b and $p = 1$ in Figure 7.1c. Similar figures for other values of p can be seen in Appendix B.7.

Remark 7.7. *Setting* $\mathcal{Y} = \mathbb{R}$*, for any* p*,* H_κ^p *reduces to the Huber loss on the real line.*

The following result gives a more explicit expression of H_κ^p, which can help us understand its effect.

Proposition 7.8. *Let* $\kappa > 0$, $p \in [1, +\infty]$*, and* q *the conjugate exponent of* p*. Then for all* $y \in \mathcal{Y}$*,*

$$H_\kappa^p(y) = \frac{1}{2}\|\mathrm{Proj}_{\mathcal{B}_\kappa^q}(y)\|_{\mathcal{Y}}^2 + \kappa\|y - \mathrm{Proj}_{\mathcal{B}_\kappa^q}(y)\|_p.$$

Remark 7.9. *For general* p*, the value of* $H_\kappa^p(y)$ *can not be computed straightforwardly due to the complexity of the projection on* $\mathcal{B}_\kappa^q$*. Note also that for* $p = 2$*, one gets back the loss investigated by Laforgue et al. (2020).*

The upcoming proposition is central to formulate the dual Problem 7.5 using the loss H_κ^p. Indeed, the latter corresponds to the convoluted loss in Equation (7.2) setting $g = \|\cdot\|_p$, and therefore to formulate the dual problem, we must be able to compute $\|\cdot\|_p^\star$. For finite dimensional $\mathcal{Y}$, it is well known that $\|\cdot\|_p^\star = \chi_{\{\mathcal{B}_1^q\}}$ with q the conjugate exponent of p–see *e.g.* (Bauschke and Combettes, 2017). However, to the best of our knowledge, it has never been proved for infinite dimensional $\mathcal{Y}$.

Proposition 7.10. *Let* $p \in [1, +\infty]$ *and let* q *be its conjugate exponent. Then the Fenchel-Legendre conjugate of* $\|\cdot\|_p$ *is given by*

$$\|\cdot\|_p^\star = \chi_{\{\mathcal{B}_1^q\}},$$

where $\chi_{\{\mathcal{B}_1^q\}}$ *denotes the indicator function of the ball of radius 1 in* $\mathcal{Y}$ *taken with respect to the* q*-norm.*

The proofs for those two results (Proposition 7.8 and Proposition 7.10) are a bit technical, therefore we defer those respectively to Appendix B.2 and Appendix B.1.

Since we deal rather with balls of radius κ, we highlight the following direct implication of Proposition 7.10.

Corollary 7.11. *Let* $\kappa > 0$*, let* $p \in [1, +\infty]$ *and let* q *be its conjugate exponent. Then the Fenchel-Legendre conjugate of* $\kappa\|\cdot\|_p$ *is given by*

$$(\kappa\|\cdot\|_p)^\star = \chi_{\{\mathcal{B}_\kappa^q\}}. \tag{7.8}$$

Table 7.1: Fenchel-Legendre properties, for any $f, g\colon \mathcal{Y} \to [-\infty, +\infty]$ and $p, q \in [1, +\infty]$ such that $\frac{1}{p} + \frac{1}{q} = 1$.

Function	Fenchel-Legendre conjugate
$\frac{1}{2}\|\cdot\|_{\mathcal{Y}}^2$	$\frac{1}{2}\|\cdot\|_{\mathcal{Y}}^2$
$\|\cdot\|_p$	$\chi_{\{\mathcal{B}_1^q\}}$
ϵf	$\epsilon f^\star(\frac{\cdot}{\epsilon})$ for all $\epsilon > 0$
$f(\cdot - y)$	$f^\star + \langle \cdot, y\rangle_{\mathcal{Y}}$ for all $y \in \mathcal{Y}$
$f \square g$	$f^\star + g^\star$

Proof Using the third line of Table 7.1 (see *e.g.* Bauschke and Combettes 2017, Proposition 13.23 (i)), we have that

$$(\kappa\|\cdot\|_p)^\star = \kappa\|\cdot\|_p^\star(\cdot/\kappa).$$

Then applying Proposition 7.10, removing the κ scaling (the indicator function either equals 0 or $+\infty$ and therefore is unchanged by multiplication by a strictly positive scalar) and using the fact that $\chi_{\{\mathcal{B}_\kappa^q\}} = \chi_{\{\mathcal{B}_1^q\}}(\cdot/\kappa)$ conclude the proof. ■

We can now state the dual problem when the Huber loss is used.

Proposition 7.12 (Dual Huber). *Let $\kappa > 0$, $p \in [1, +\infty]$ and let q be the conjugate exponent to p. The dual of Problem 7.5 when using the Huber loss H_κ^p reads*

$$\inf_{\alpha \in \mathcal{Y}^n} \sum_{i=1}^n \Big(\frac{1}{2}\|\alpha_i\|_{\mathcal{Y}}^2 - \langle \alpha_i, y_i\rangle_{\mathcal{Y}}\Big) + \frac{1}{2\lambda n}\sum_{i=1}^n\sum_{j=1}^n k_{\mathcal{X}}(x_i, x_j)\langle \alpha_i, \mathrm{T}_{k_\Theta}\alpha_j\rangle_{\mathcal{Y}} \qquad (7.9)$$
$$\text{subject to } \|\alpha_i\|_q \le \kappa, \quad i \in [\![n]\!].$$

Proof This is a direct combination of Lemma 7.2 and Corollary 7.11. ■

Influence of κ and p. In the dual problem, the use of H_κ^p imposes inequality constraints on the functional q-norm of the dual variables. We can understand why this brings robustness as in the prediction formula–Equation (7.4)–, each dual variable is associated with an observation. Therefore bounding in some sense the influence of each on the final estimator should bring robustness. In this case, it is bounded in the sense of the q-norm of the dual variables.

The parameter κ controls the strictness of the constraint, the lower, the stricter. Consequently, as κ grows bigger, the constraints eventually becomes irrelevant and we recover the solution to the classical function-valued ridge regression problem.

The parameter p controls the norm used to define the constraints, q being p's conjugate exponent. To get an intuition of how this may affect the estimator, the following example takes a closer look at the case $p = 1$ ($q = +\infty$).

Example 7.13 (Robustness to local outliers)**.** *Let us take a look at Equation (7.4), which we recall here:*

$$\hat{h} = \frac{1}{\lambda n} \sum_{i=1}^{n} k_{\mathcal{X}}(\cdot, x_i) \mathrm{T}_{k_\Theta} \hat{\alpha}_i.$$

In the extreme case of $p = 1$, we have $q = +\infty$, meaning that the dual variables cannot be greater than κ in absolute value for all $\theta \in \Theta$. For $x \in \mathcal{X}$, the prediction at the location $\theta \in \Theta$ is given by:

$$\hat{h}(x)(\theta) = \frac{1}{\lambda n} \sum_{i=1}^{n} k_{\mathcal{X}}(\cdot, x_i) \int_\Theta k_\Theta(\theta, \theta_1) \hat{\alpha}_i(\theta_1) \mathrm{d}(\theta_1).$$

*Even though the dual variables act through the integral operator T_{k_Θ}, if the kernel is translation invariant the associated base kernel reaches its maximum at 0 (*e.g. *Laplace or Gaussian kernel). Consequently, for the i-th observation, the value of $\alpha_i(\theta)$ will have the greater impact on the prediction $\hat{h}(x)(\theta)$. It will also have an effect on the locations neighboring θ.*

Hence, in the presence of local outliers, it is desirable to threshold the contribution of each observation at each location. The constraint on the ∞-norm of the dual variables does just that. If the threshold concerns the 2-norm however, the global contribution of an observation is targeted through its associated dual variable. Consequently, the latter should be much less efficient for local outliers. We validate this intuition with an empirical study of the sensitivity of the loss H_κ^p in Section 7.4.2.

We now wish to tackle Problem 7.9 numerically. We see that in a proximal gradient algorithm, the projection on the ball $\mathcal{B}_\kappa^q$ will be performed n times per step. This de facto reduces the values of p that we can consider. Indeed, the projection could be computed iteratively for any q but given the number of times it will be performed, this is unrealistic. The next proposition highlights that for $q = 2$ and $q = +\infty$ we are able to compute the projection on $\mathcal{B}_\kappa^q$ in closed-form. Therefore the associated values of p for which Problem 7.9 is tractable are $p = 2$ and $p = 1$.

Proposition 7.14 (Projection on $\mathcal{B}_\kappa^q$)**.** *Let $\kappa > 0$. The projection on $\mathcal{B}_\kappa^q$ is tractable for $q = 2$ and $q = +\infty$ and can be expressed for all $(\alpha, \theta) \in \mathcal{Y} \times \Theta$ as*

$$\mathrm{Proj}_{\mathcal{B}_\kappa^2}(\alpha) = \min\left(1, \frac{\kappa}{\|\alpha\|_\mathcal{Y}}\right)\alpha, \tag{7.10}$$

$$\left(\mathrm{Proj}_{\mathcal{B}_\kappa^\infty}(\alpha)\right)(\theta) = \mathrm{sign}\left(\alpha(\theta)\right)\min\left(\kappa, |\alpha(\theta)|\right). \tag{7.11}$$

Proof

- Equation (7.10) is well known for $\mathcal{Y}$ a Hilbert space. The problem of projection is

$$\arg\min_{y \in \mathcal{B}_\kappa^2} \|\alpha - y\|_\mathcal{Y}^2.$$

We can write y as $y = \nu\alpha + w$ with $w \perp \alpha$ which implies that if $y \in \mathcal{B}_\kappa^2$, $\nu\alpha \in \mathcal{B}_\kappa^2$. Because $w \perp \alpha$, we can therefore always reduce the objective by setting $w = 0$. Consequently, we must choose y collinear to α. The problem is then one dimensional: find $\nu \geq 0$ such that $\nu\alpha \in \mathcal{B}_\kappa^2$. Then basic computations lead to $\nu = \min\left(1, \frac{\kappa}{\|\alpha\|_\mathcal{Y}}\right)$.

- Let us first highlight that for $\alpha \in \mathcal{Y}$, the function $\left(\theta \mapsto \operatorname{sign}(\alpha(\theta)) \min(\kappa, |\alpha(\theta)|)\right)$ is measurable and it is in $\mathcal{Y}$. Formally, we can prove Equation (7.11) using the Moreau decomposition (Lemma 2.53):

$$\operatorname{prox}_{\chi_{\{\mathcal{B}_\kappa^\infty\}}}(\alpha) + \operatorname{prox}_{\chi^\star_{\{\mathcal{B}_\kappa^\infty\}}}(\alpha) = \alpha \tag{7.12}$$

And indeed by definition of the proximal operator:

$$\begin{aligned}\operatorname{prox}_{\chi_{\{\mathcal{B}_\kappa^\infty\}}}(\alpha) &= \operatorname*{arg\,min}_{y \in \mathcal{Y}} \frac{1}{2}\|\alpha - y\|_{\mathcal{Y}}^2 + \chi_{\{\mathcal{B}_\kappa^\infty\}}(y) \\ &= \operatorname*{arg\,min}_{y \in \mathcal{Y}} \|\alpha - y\|_{\mathcal{Y}}^2 + \chi_{\{\mathcal{B}_\kappa^\infty\}}(y) \\ &= \operatorname{Proj}_{\mathcal{B}_\kappa^\infty}(\alpha).\end{aligned} \tag{7.13}$$

And applying the Fenchel-Legengre transform to Equation (7.8),

$$\kappa\|\cdot\|_1 = \chi^\star_{\{\mathcal{B}_\kappa^\infty\}}. \tag{7.14}$$

We get that

$$\operatorname{prox}_{\chi^\star_{\{\mathcal{B}_\kappa^\infty\}}}(\alpha) = \operatorname{prox}_{\kappa\|\cdot\|_1}(\alpha) \tag{7.15}$$

Then combining Equation (7.13) with Equation (7.12) and injecting Equation (7.15) into the resulting equation, we get

$$\operatorname{Proj}_{\mathcal{B}_\kappa^\infty}(\alpha) = \alpha - \operatorname{prox}_{\kappa\|\cdot\|_1}(\alpha).$$

It is well known that $\operatorname{prox}_{\kappa\|\cdot\|_1}$ is the pointwise application of the *soft-thresholding operator*, therefore for all $\theta \in \Theta$

$$\begin{aligned}\operatorname{Proj}_{\mathcal{B}_\kappa^\infty}(\alpha)(\theta) &= \alpha(\theta) - \left||\alpha(\theta)| - \kappa\right|_+ \operatorname{sign}(\alpha(\theta)) \\ &= \operatorname{sign}(\alpha(\theta)) \min(\kappa, |\alpha(\theta)|).\end{aligned}$$

■

For $p = q = 2$, the projection simply consists of a rescaling by a scalar involving the $\|\cdot\|_{\mathcal{Y}}$ norm of the dual variable. The case $p = 1$ ($q = +\infty$) is more challenging as it involves a *pointwise* projection. Therefore, for a representation to be valid, it must allow us to perform this projection, and consequently it should give us pointwise control over the dual variables.

In order to solve Problem 7.9, we propose to use two different representations. In Section 7.2.1, we advocate representing the dual variables by linear splines and approximating the action of T_{k_Θ} by Monte-Carlo (MC) sampling. An alternative approach (elaborated in Section 7.2.2) relies on a finite-rank approximation of T_{k_Θ} using its eigendecomposition. It is applicable only for $p = 2$ but it performs dimensionality reduction.

7.2.1 Linear splines approach

We now describe the linear splines approach. It gives full control over the dual variables (including pointwisely) and therefore allows for solving approximately Problem 7.9 for $p = 1$ and $p = 2$.

Table 7.2: Correspondence between the quantities involved in Problem 7.5 depending on the representation.

	Linear splines	Eigenvectors of T_{k_Θ}
$\sum_{i=1}^n \frac{1}{2}\|\alpha_i\|_{\mathcal{Y}}^2$	$\frac{1}{2m}\operatorname{Trace}\left(A^T A\right)$	$\operatorname{Trace}\left(\frac{1}{2}A^T A\right)$
$\sum_{i=1}^n \langle \alpha_i, y_i\rangle_{\mathcal{Y}}$	$\frac{1}{m}\operatorname{Trace}\left(A^T Y\right)$	$\operatorname{Trace}\left(A^T R\right)$
$\sum_{i=1}^n\sum_{j=1}^n k_{\mathcal{X}}(x_i,x_j)\left\langle \alpha_i, T_{k_\Theta}\alpha_j\right\rangle_{\mathcal{Y}}$	$\frac{1}{m^2}\operatorname{Trace}\left(A^T K_\Theta A K_{\mathcal{X}}\right)$	$\operatorname{Trace}\left(A^T \tilde{\Lambda} A K_{\mathcal{X}}\right)$

A linear spline is a piecewise linear curve. It can be encoded by a set of ordered locations or anchor points $(\theta_s)_{s=1}^m \in \Theta^m$, and by a vector of size m corresponding to the evaluation of the spline at these points. In practice, we often take the locations of sampling of the functions $(y_i)_{i=1}^n \in \mathcal{Y}^n$. We now consider those anchors fixed. The n dual variables can be represented using the matrix of their evaluations at $(\theta_s)_{s=1}^m$. Let $A := [\mathbf{a}_i]_{i=1}^n = (\alpha_i(\theta_s))_{s,i=1}^{m,m} \in \mathbb{R}^{m\times n}$, with $\mathbf{a}_i$ being the i-th column of A.

Then, the action of the integral operator can be approximated by Monte Carlo (MC) as

$$T_{k_\Theta}\alpha \approx \frac{1}{m}\sum_{s=1}^m k_\Theta(\cdot,\theta_s)\alpha(\theta_s). \tag{7.16}$$

Injecting this in Equation (7.4), we explore estimators of the form

$$h(x)(\theta) = \frac{1}{\lambda nm}\sum_{i=1}^n k_{\mathcal{X}}(x,x_i)\sum_{s=1}^m a_{si}k_\Theta(\theta,\theta_s). \tag{7.17}$$

Remark 7.15 (Linear splines or discretization?)**.** *In practice, setting the problem in terms of linear splines rather than simple MC averages as in Section 4.2.2 is interesting if the locations at which we observe the output functions vary. Linear splines then allow for evaluating the functions at a common (drawn or chosen) set of locations. In the case we study, all the functions are evaluated at the same places, thus the two approaches are equivalent.*

Using the spline representation in Problem 7.9, we do the following approximations, which are equivalent to those presented for the functional kernel ridge regression in Section 4.2.2.

- **Squared norm of the dual variables**: an MC average using the locations $(\theta_s)_{s=1}^m$ yields:
$$\frac{1}{2}\sum_{i=1}^n \|\alpha_i\|_{\mathcal{Y}}^2 \approx \frac{1}{2m}\operatorname{Trace}(A^T A).$$
- **Scalar products with the output functions**: Let $Y \in \mathbb{R}^{m\times n} = (y_i(\theta_s))_{s,i=1}^{m,n}$ regroup the observations of the output functions at the locations $(\theta_s)_{s=1}^m$, then
$$\sum_{i=1}^n \langle \alpha_i, y_i\rangle_{\mathcal{Y}} \approx \frac{1}{m}\operatorname{Trace}(A^T Y).$$

- **Regularization term**: let $\mathrm{K}_{\mathcal{X}} \in \mathbb{R}^{n\times n}$ and $\mathrm{K}_{\Theta} \in \mathbb{R}^{m\times m}$ be the kernel matrices respectively associated to the kernels $k_{\mathcal{X}}$ with input data $(x_i)_{i=1}^n$ and k_{Θ} with input data $(\theta_s)_{s=1}^m$. Then,
$$\|h\|_{\mathcal{H}_{\mathrm{K}}}^2 \approx \frac{1}{\lambda n m^2}\mathrm{Trace}(\mathrm{A}^{\mathrm{T}}\mathrm{K}_{\Theta}\mathrm{A}\mathrm{K}_{\mathcal{X}}).$$
This term involves two successive approximations at the locations $(\theta_s)_{s=1}^m$, the first for the integral operator–Equation (7.16)–and the second for the functional inner products.

- **Constraints**: we can also approximate the integral defining $\|\alpha\|_q$–Equation (7.6)–through a MC average. For all $i \in [\![n]\!]$, we use
$$\|\alpha_i\|_q \approx \frac{1}{m^{\frac{1}{q}}}\|\mathbf{a}_i\|_q,$$
where the q-norm for a finite dimensional vector $\mathbf{a} \in \mathbb{R}^m$ is given by
$$\|\mathbf{a}\|_q := \left(\sum_{s=1}^m a_s^q\right)^{\frac{1}{q}}.$$
We can write more compactly the set of constraints from Problem 7.9 using the composite (q,∞)-norm on the columns of the matrix A:
$$\|\mathrm{A}\|_{q,\infty} \leq m^{\frac{1}{q}}\kappa.$$

Gathering the different terms (summarized in Table 7.2) yields the following relaxation of Problem 7.9:

$$\inf_{\mathrm{A}\in\mathbb{R}^{m\times n}} \mathrm{Trace}\left(\frac{1}{2}\mathrm{A}^{\mathrm{T}}\mathrm{A} - \mathrm{A}^{\mathrm{T}}\mathrm{Y} + \frac{1}{2\lambda n m}\mathrm{A}^{\mathrm{T}}\mathrm{K}_{\Theta}\mathrm{A}\mathrm{K}_{\mathcal{X}}\right) \quad \text{subject to} \quad \|\mathrm{A}\|_{q,\infty} \leq m^{\frac{1}{q}}\kappa. \tag{7.18}$$

Remark 7.16. *The decomposable assumption on the kernel* K *plays a role in the regularization. It has the effect of disentangling the action of both Gram matrices* $\mathrm{K}_{\mathcal{X}}$ *and* K_{Θ}.

We propose to tackle Problem 7.18 using accelerated proximal gradient descent (APGD). The proximal step amounts to a projection of the coefficients on the q-ball of radius $m^{1/q}\kappa$. The technique is summarized in Algorithm 7.1.

The gradient stepsize γ can be computed to ensure convergence: one must set $\gamma < \frac{2}{C}$ where C is the Lipschitz constant associated to the gradient of the objective function; here $C \leq 1 + \frac{1}{\lambda n}\|\mathrm{K}_{\mathcal{X}}\|_{\mathrm{op}}\|\mathrm{K}_{\Theta}\|_{\mathrm{op}}$. The initialization can either be the null matrix $\mathrm{A}^{(0)} = \mathbf{0}_{\mathbb{R}^{m\times n}}$ or the solution of the unconstrained optimization problem obtained in closed-form with time complexity $\mathcal{O}(n^3 + m^3)$–see Section 4.2.2. Moreover, since the objective function in Problem 7.18 is the sum of two functions, one convex and differentiable with Lipschitz continuous gradient (the quadratic form) and one convex and lower semi-continuous (the indicator function of the constraint set), the optimal worst case complexity is $\mathcal{O}\left(\frac{1}{T^2}\right)$ (Beck and Teboulle, 2009). In practice the time complexity per iteration in Algorithm 7.1 is dominated by the computation of the matrix $\mathrm{K}_{\Theta}\mathrm{V}\mathrm{K}_{\mathcal{X}}$ which is $\mathcal{O}(n^2 m + m^2 n)$.

Algorithm 7.1 APGD with linear splines

input : Gram matrices $\mathrm{K}_{\mathcal{X}}, \mathrm{K}_{\Theta}$, data matrix Y, regularization parameter λ, loss parameters (κ, p) or (ϵ, p), gradient step γ

init : $\mathrm{A}^{(0)}, \mathrm{A}^{(-1)} = \mathbf{0} \in \mathbb{R}^{m \times n}$

for *epoch* t *from* 0 *to* $T-1$ **do**

// gradient step

$\mathrm{V} = \mathrm{A}^{(t)} + \frac{t-2}{t+1}\left(\mathrm{A}^{(t)} - \mathrm{A}^{(t-1)}\right)$

$\mathrm{U} = \mathrm{V} - \gamma\left(\mathrm{V} + \frac{1}{\lambda n m}\mathrm{K}_{\Theta}\mathrm{V}\mathrm{K}_{\mathcal{X}} - \mathrm{Y}\right)$

// projection step

if $p = 2$ **then**

for *column* $i \in [\![n]\!]$ **do**

$\mathbf{a}_i^{(t+1)} = \min\left(\frac{\sqrt{m}\kappa}{\|\mathbf{u}_i\|_2}, 1\right)\mathbf{u}_i$ // if H^2_κ

$\mathbf{a}_i^{(t+1)} = \left|1 - \frac{\gamma\epsilon}{\sqrt{m}\|\mathbf{u}_i\|_2}\right|_+ \mathbf{u}_i$ // if ℓ^2_ϵ

else

for *column* $i \in [\![n]\!]$ **do**

for *row* $s \in [\![m]\!]$ **do**

$a_{is}^{(t+1)} = \mathrm{sign}(u_{is})\min\left(\kappa, |u_{is}|\right)$ // if H^1_κ

$a_{is}^{(t+1)} = \mathrm{sign}(u_{is})\left||u_{is}| - \frac{\gamma\epsilon}{m}\right|_+$ // if ℓ^∞_ϵ

7.2.2 Eigendecomposition approach

In this section, we propose an alternative dual representation based on an approximate eigendecomposition of T_{k_Θ}. The precedent solution incurs a computational cost which is dependent on the number of anchors m, and thus we seek a more compressed representation. However there is a trade-off between flexibility and efficiency: it will work only for $p = 2$. Indeed, if we represent the dual variables in a dictionary of functions, we can approximate all the terms in Problem 7.9, but not the constraints; when $p = 1$ ($q = +\infty$), we cannot project onto the ball $\mathcal{B}^\infty_\kappa$ having access only to representation coefficients in the dictionary.

Let us now focus on the case $p = 2$. Any dictionary of function could be used, yet an orthonormal family makes for simpler computations. It allows for straightforward approximation of all terms except the one involving T_{k_Θ}. To circumvent this problem, it is natural to select directions well-suited to T_{k_Θ}. Therefore we use the d eigenfunctions of T_{k_Θ} associated to its largest eigenvalues.

Since computing the eigendecomposition of T_{k_Θ} in closed-form is most of the times not possible (see Section 2.1.2), we perform an approximate eigendecomposition as in Example 2.16. We recall here briefly the elements for completeness. Suppose that we use the locations $(\theta_s)_{s=1}^m$. Let then $\mathrm{U} \in \mathbb{R}^{m \times m}$ be the orthonormal matrix which columns $(\mathbf{u}_s)_{s=1}^m$ are the eigenvectors of K_Θ and let $(\hat{\lambda}_s)_{s=1}^m \in \mathbb{R}^m$ be the corresponding eigenvalues. We use the approximate eigenvalue/eigenvector pairs $(\tilde{\lambda}_s, \tilde{\phi}_s)_{s=1}^m$ with

$$\begin{aligned} \forall s \in [\![m]\!], \quad \tilde{\phi}_s : \theta \mapsto \frac{1}{\sqrt{m}\tilde{\lambda}_s}\mathbf{k}_{\mathcal{X}}(\theta)^{\mathrm{T}}\mathbf{u}_s, \\ \tilde{\lambda}_s = \frac{1}{m}\hat{\lambda}_s. \end{aligned} \tag{7.19}$$

Using the first d of these eigenfunctions for some $d < m$, we compress the representation. We store the largest d eigenvalues defined in Equation (7.19) in a diagonal matrix $\tilde{\Lambda} \in \mathbb{R}^{d\times d}$.

The problem is now parameterized by a matrix $\mathrm{A} = [\mathbf{a}_i]_{i=1}^n \in \mathbb{R}^{d\times n}$ with each column $\mathbf{a}_i \in \mathbb{R}^d$ encoding the coefficients of the dual variable α_i on the $(\phi_l)_{l=1}^d$ orthogonal family. Using the true eigenfunctions $(\phi_l)_{l=1}^d$, the action of the integral operator can be simplified before discretization, and the estimator then reads

$$h(x)(\theta) = \frac{1}{\lambda n}\sum_{i=1}^{n}\sum_{l=1}^{d} a_{li}\lambda_l k_{\mathcal{X}}(x, x_i)\phi_l(\theta). \tag{7.20}$$

We then inject the estimated eigenfunctions and eigenvalues, yielding the estimator

$$h(x)(\theta) = \frac{1}{\lambda n}\sum_{i=1}^{n}\sum_{l=1}^{d} a_{li}\tilde{\lambda}_l k_{\mathcal{X}}(x, x_i)\tilde{\phi}_l(\theta). \tag{7.21}$$

We store in $\mathrm{R} \in \mathbb{R}^{d\times n}$ the scalar products between the observed data and the eigenbasis: its (l, i)-th entry is $\langle y_i, \phi_l\rangle_{\mathcal{Y}}$. In practice, we approximate those scalar products by a MC average:

$$\langle y_i, \phi_l\rangle_{\mathcal{Y}} \approx \frac{1}{m}\langle \tilde{\phi}_s(\boldsymbol{\theta}), \tilde{y}_i\rangle_{\mathbb{R}^m}. \tag{7.22}$$

More precisely, we use the following approximations for the different terms:

- **Squared norm of the dual variables**: since the eigenfunctions are orthonormal, $\left\|\sum_{l=1}^d a_{li}\phi_l\right\|_{\mathcal{Y}}^2 \approx \|\mathbf{a}_i\|_2^2$, therefore

$$\frac{1}{2}\sum_{i=1}^{n}\|\alpha_i\|_{\mathcal{Y}}^2 \approx \mathrm{Trace}\Big(\frac{1}{2}\mathrm{A}^{\mathrm{T}}\mathrm{A}\Big).$$

- **Scalar products with the output functions:**

$$\langle \alpha_i, y_i\rangle_{\mathcal{Y}} \approx \sum_{l=1}^{d} a_{li}\langle \phi_l, y_i\rangle_{\mathcal{Y}} = \mathbf{a}_i^{\mathrm{T}}\mathbf{r}_i.$$

  Consequently, we use the approximation

$$\sum_{i=1}^{n}\langle \alpha_i, y_i\rangle_{\mathcal{Y}} \approx \mathrm{Trace}\Big(\mathrm{A}^{\mathrm{T}}\mathrm{R}\Big).$$

- **Regularization term**: we have that

$$\langle \alpha_i, \alpha_j\rangle_{\mathcal{Y}} \approx \Big\langle \sum_{l=1}^{d} a_{li}\phi_l, \mathrm{T}_{k_\Theta}\sum_{r=1}^{d} a_{rj}\phi_r\Big\rangle_{\mathcal{Y}}.$$

  Since $(\phi_l)_{l=1}^d$ is an orthonormal family of eigenfunctions of T_{k_Θ}:

$$\Big\langle \sum_{l=1}^{d} a_{li}\phi_l, \mathrm{T}_{k_\Theta}\sum_{r=1}^{d} a_{rj}\phi_r\Big\rangle_{\mathcal{Y}} = \mathbf{a}_i^{\mathrm{T}}\Lambda\mathbf{a}_j,$$

Algorithm 7.2 APGD with eigenbasis representation

input : Gram matrix $\mathrm{K}_{\mathcal{X}}$, matrix of estimated eigenvalues $\tilde{\Lambda}$, data scalar product matrix R, regularization parameter λ, loss parameters $(\kappa, 2)$ or $(\epsilon, 2)$, gradient step γ

init : $\mathrm{A}^{(0)}, \mathrm{A}^{(-1)} = \mathbf{0} \in \mathbb{R}^{d \times n}$

for *epoch* t *from* 0 *to* $T-1$ **do**

// gradient step

$\mathrm{V} = \mathrm{A}^{(t)} + \frac{t-2}{t+1}\left(\mathrm{A}^{(t)} - \mathrm{A}^{(t-1)}\right)$

$\mathrm{U} = \mathrm{V} - \gamma\left(\mathrm{V} + \frac{1}{\lambda n}\tilde{\Lambda}\mathrm{V}\mathrm{K}_{\mathcal{X}} - \mathrm{R}\right)$

// proximal step

for *column* $i \in [\![n]\!]$ **do**

$\mathbf{a}_i^{(t+1)} = \min\left(\frac{\kappa}{\|\mathbf{u}_i\|_2}, 1\right)\mathbf{u}_i$ // if H_κ^2

$\mathbf{a}_i^{(t+1)} = \left|1 - \frac{\gamma\epsilon}{\|\mathbf{u}_i\|_2}\right|_+ \mathbf{u}_i$ // if ℓ_ϵ^2

return $\mathrm{A}^{(T)}$

therefore,

$$\sum_{i=1}^{n}\sum_{j=1}^{n} k_{\mathcal{X}}(x_i, x_j)\left\langle \alpha_i, \mathrm{T}_{k_\Theta}\alpha_j\right\rangle_{\mathcal{Y}} \approx \mathrm{Trace}\left(\mathrm{A}^{\mathrm{T}}\tilde{\Lambda}\mathrm{A}\mathrm{K}_{\mathcal{X}}\right).$$

- **Contraints**: Since $\|\alpha_i\|_{\mathcal{Y}}^2 \approx \|\mathbf{a}_i\|_2^2$, we can approximate the constraints, and write them compactly using the $(2,\infty)$-matrix norm as

$$\|\mathrm{A}\|_{2,\infty} \leq \kappa.$$

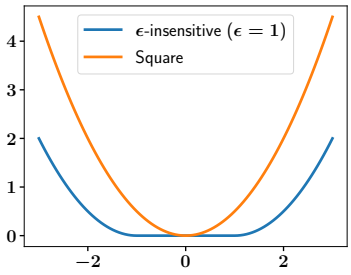
The correspondence between the different terms are summarized in Table 7.2. The optimization then reduces to

$$\inf_{\mathrm{A}\in\mathbb{R}^{d\times n}} \ \mathrm{Trace}\left(\frac{1}{2}\mathrm{A}^{\mathrm{T}}\mathrm{A} - \mathrm{A}^{\mathrm{T}}\mathrm{R} + \frac{1}{2\lambda n}\mathrm{A}^{\mathrm{T}}\tilde{\Lambda}\mathrm{A}\mathrm{K}_{\mathcal{X}}\right) \quad \text{subject to} \quad \|\mathrm{A}\|_{2,\infty} \leq \kappa. \tag{7.23}$$

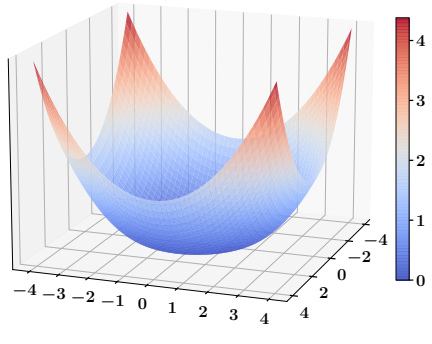
Again, one can use APGD to solve this task, we give the corresponding procedure in Algorithm 7.2.

7.3 Sparse FOR: learning with the functional ϵ-insensitive loss

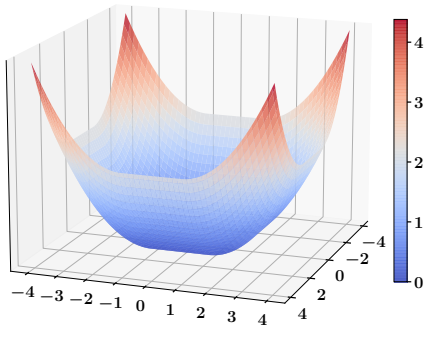
We now investigate losses arising from infimal convolution of $1/2\|\cdot\|_{\mathcal{Y}}^2$ with the indicator function of a ball associated to the $\|\cdot\|_p$ norm. This results in losses which are insensitive to smaller discrepancy in the sense of that norm. An interesting property is that they promote sparsity on the dual coefficients in different ways depending on the value of p. This section is dedicated to the FOR problem using these losses. It will however be much shorter than the previous one on Huber losses since many of the introduced elements are common.



(a) ϵ-insensitive loss ($\epsilon = 1$) and square loss



(b) ℓ^2_ϵ ($\epsilon = 1$)



(c) ℓ^∞_ϵ ($\epsilon = 1$)

Figure 7.2: Huber losses on $\mathbb{R}$ and $\mathbb{R}^2$.

Definition 7.17 (ϵ-insensitive loss)**.** *Let $\epsilon > 0$ and $p \in [1, +\infty]$. We define the ϵ-insensitive version of the square loss with parameters (ϵ, p) as*

$$\ell^p_\epsilon := \frac{1}{2}\|\cdot\|^2_{\mathcal{Y}} \square \chi_{\{\mathcal{B}^p_\epsilon\}}.$$

We display this loss for $\mathcal{Y} = \mathbb{R}$ in Figure 7.2a and for $\mathcal{Y} = \mathbb{R}^2$ with $p = 2$ in Figure 7.2b and $p = +\infty$ in Figure 7.2c. Similar figures for other values of p can be seen in Appendix B.7.

When $\mathcal{Y} = \mathbb{R}$, it reduces to the classical ϵ-insensitive square loss regardless of p. We highlight as well that setting $\epsilon = 0$ recovers the square loss. The following proposition (counterpart of Proposition 7.8) sheds light on the effect of the infimal convolution on the square loss.

Proposition 7.18. *Let $\epsilon > 0$ and $p \in [1, +\infty]$. Then for all $y \in \mathcal{Y}$,*

$$\ell^p_\epsilon(y) = \frac{1}{2}\|y - \mathrm{Proj}_{\mathcal{B}^p_\epsilon}(y)\|^2_{\mathcal{Y}}. \tag{7.24}$$

As it is not central in the exposition, we defer the proof of this proposition to Appendix B.3.

Remark 7.19. *Proposition 7.18 means that $\ell_\epsilon^p(y) = 0$ when $\|y\|_p \leq \epsilon$, which is the desired effect: small residuals are ignored.*

Nevertheless, for general p, $\ell_\epsilon^p(y)$ is not straightforward to compute due to the projection $\mathrm{Proj}_{\mathcal{B}_\epsilon^p}(y)$. Yet through a dual approach, Problem 7.1 can still be tackled computationally.

Proposition 7.20 (Dual ϵ-insensitive)**.** *Let $\epsilon \geq 0, p \in [1, +\infty]$, and let q be the conjugate exponent of p ($\frac{1}{p} + \frac{1}{q} = 1$). The dual of Problem 7.1 reads*

$$\inf_{(\alpha_i)_{i=1}^n \in \mathcal{Y}^n} \sum_{i=1}^n \left[\frac{1}{2}\|\alpha_i\|_{\mathcal{Y}}^2 - \langle \alpha_i, y_i \rangle_{\mathcal{Y}} + \epsilon \|\alpha_i\|_q \right] + \frac{1}{2\lambda n} \sum_{i=1}^n \sum_{j=1}^n k_{\mathcal{X}}(x_i, x_j) \left\langle \alpha_i, \mathrm{T}_{k_\Theta} \alpha_j \right\rangle_{\mathcal{Y}}. \tag{7.25}$$

Proof This is a direct combination of Lemma 7.2, Corollary 7.11 (replacing κ with ϵ), and the involutive property of the Fenchel-Legendre transform (Proposition 2.44). ■

Influence of ϵ and p. Compared to the square loss, ℓ_ϵ^p induces an additional term $\epsilon \sum_{i=1}^n \|\alpha_i\|_q$ in the dual. This can be seen as a mixed norm on $\mathcal{Y}^n$. For $\boldsymbol{\alpha} \in \mathcal{Y}$, set

$$\|\boldsymbol{\alpha}\|_{q,1} = \sum_{i=1}^n \|\alpha_i\|_q, \tag{7.26}$$

then the additional regularization induced by the loss is indeed the 1-norm of the vector of q-norms of the dual variables. Such norms are known to induce structured sparsity to the solution (Bach et al., 2012).

Example 7.21 (Cases $p = +\infty$ and $p = 2$)**.** *On the one hand, for $p = +\infty$ ($q = 1$), the penalty reduces to the 1-norm on $\mathcal{Y}^n$ of $\boldsymbol{\alpha}$, and therefore, the sparsity has no structure, which leaves maximum flexibility to obtain local sparsity. On the other hand, for $p = 2$ ($q = 2$), the norm is the $(2,1)$-norm which results in global sparsity: whole functions are set to zero.*

The challenges involving the representation of the dual variables, and the computability of the different terms composing Problem 7.25 are similar to those evoked in Section 7.2. We have however traded the constraints on the q-norms of the dual variables against an additional non-smooth term. We also address this convex non-smooth optimization problem through the APGD algorithm, and the proximal step now involves $\mathrm{prox}_{\gamma\epsilon\|\cdot\|_q}$ for a suitable gradient stepsize $\gamma > 0$.

Proposition 7.22 (Proximal q-norm)**.** *Let $\epsilon > 0$. The proximal operator of $\epsilon\|\cdot\|_q$ is computable for $q = 1$ and $q = 2$, and given for all $(\alpha, \theta) \in \mathcal{Y} \times \Theta$ by*

$$\left(\mathrm{prox}_{\epsilon\|\cdot\|_1}(\alpha)\right)(\theta) = \mathrm{sign}\left(\alpha(\theta)\right) \left| |\alpha(\theta)| - \epsilon \right|_+ \tag{7.27}$$

$$\mathrm{prox}_{\epsilon\|\cdot\|_2}(\alpha) = \alpha \left| 1 - \frac{\epsilon}{\|\alpha\|_{\mathcal{Y}}} \right|_+ . \tag{7.28}$$

Proof Using the Moreau decomposition (Lemma 2.53), we have that

$$\operatorname{prox}_{\chi_{\{\mathcal{B}_\epsilon^p\}}}(\alpha) + \operatorname{prox}_{\chi^\star_{\{\mathcal{B}_\epsilon^p\}}}(\alpha) = \alpha.$$

Then using Corollary 7.11 and the involutive property of the Fenchel-Legendre transform (Proposition 2.44), we have that

$$\begin{aligned}\operatorname{prox}_{\epsilon\|\cdot\|_q}(\alpha) &= \alpha - \operatorname{prox}_{\chi_{\{\mathcal{B}_\epsilon^p\}}}(\alpha) \\ &= \alpha - \operatorname{Proj}_{\mathcal{B}_\epsilon^p}(\alpha).\end{aligned} \tag{7.29}$$

We know from Proposition 7.14 that such projection is tractable for $p = 2$ ($q = 2$) and $p = +\infty$ ($q = 1$), and we have their expression, which combined with Equation (7.29) gives the claimed results. ■

7.3.1 Linear splines approach

Similarly to what was presented in Section 7.2, we use linear splines to represent the dual variables in order to have pointwise control over them to be able to compute the proximal operators. Keeping the notations, the optimization boils down to

$$\inf_{\mathrm{A}\in\mathbb{R}^{n\times m}} \operatorname{Trace}\left(\frac{1}{2}\mathrm{A}^{\mathrm{T}}\mathrm{A} - \mathrm{A}^{\mathrm{T}}\mathrm{Y} + \frac{1}{2\lambda nm}\mathrm{A}^{\mathrm{T}}\mathrm{K}_\Theta\mathrm{A}\mathrm{K}_{\mathcal{X}}\right) + \frac{\epsilon}{m^{\frac{1}{q}}}\sum_{i=1}^{n}\|\mathbf{a}_i\|_q. \tag{7.30}$$

We use APGD to solve it with steps detailed in Algorithm 7.1. When $q = 1$, the proximal operator is the soft-thresholding operator, akin to promote sparsity in the dual coefficients, which is an equivalent aspect of Example 7.21.

7.3.2 Eigendecomposition approach

As in Section 7.2.2, we can also represent the dual variables in a truncated basis of eigenfunctions of T_{k_Θ} when $p = 2$. Then, using the same notation as in Equation (7.23), we get

$$\inf_{\mathrm{A}\in\mathbb{R}^{n\times d}} \operatorname{Trace}\left(\frac{1}{2}\mathrm{A}^{\mathrm{T}}\mathrm{A} - \mathrm{A}^{\mathrm{T}}\mathrm{R} + \frac{1}{2\lambda n}\mathrm{A}^{\mathrm{T}}\tilde{\Lambda}\mathrm{A}\mathrm{K}_{\mathcal{X}}\right) + \epsilon\sum_{i=1}^{n}\|\mathbf{a}_i\|_2. \tag{7.31}$$

APGD is applied to tackle this problem in Algorithm 7.2. Notice that the proximal operator in this case is the block soft-thresholding operator, known to promote structured column-wise sparsity, which is another way of looking at Example 7.21.

7.4 Numerical experiments

In this section, we study empirically the proposed Huber and ϵ-insensitive losses as well as the corresponding proposed algorithms. The experiments are centered around two key directions:

1. The first goal is to understand the accuracy/sparsity trade-off of the ϵ-insensitive loss as a function of the regularization λ and insensitivity parameter ϵ. As we have seen in the dual Problem 7.25, the loss ℓ_ϵ^p induces an additional regularization term $\epsilon\|\boldsymbol{\alpha}\|_{q,1} = \epsilon\sum_{i=1}^n \|\alpha_i\|_q$ which promotes sparsity on top of the regularization in the fv-RKHS norm $\lambda\|h\|^2_{\mathcal{H}_K}$. Therefore, in order to obtain a fair amount of sparsity, we must decrease λ and increase ϵ. Nevertheless, the two terms having different effects on the solution, it is crucial to understand if there is a trade-off between accuracy and sparsity.

2. Our second aim is to quantify the robustness of the Huber losses to different forms of outliers, focusing on global versus local ones. Indeed, as highlighted in Section 7.2, the choice of the parameter p should determine the degree of locality of the outliers that the Huber loss can filter out. Unfortunately, we have also seen that Problem 7.9 is tractable only for two values: $p = 1$ and $p = 2$. For $p = 1$, we have given an explanation for the robustness to local outliers in Example 7.13. We propose to study first the robustness of the loss itself for various $p \in [1, 2]$ in Section 7.4.2. Then we study the proposed estimators for $p = 1$ and $p = 2$. To gain further insight into the different types of robustness, we designed 3 types of functional outliers with distinct characteristics which we use to test the estimators.

We investigate three benchmarks: a synthetic dataset based on Gaussian processes, followed by two real-world ones, one arising in the context of neuroimaging and the other in speech analysis. We investigate both questions on the synthetic data, and provide further insights for the first and the second question on the neuroimaging and the speech dataset, respectively.

Next, we detail how we generate the different outliers as well as the synthetic dataset.

7.4.1 Preliminaries

Corruption

We now introduce the three outlier types used in our experiments. Local outliers affect the functions only on small portions of Θ whereas global ones contaminate them in their entirety. To corrupt the functions $(y_i)_{i\in[\![n]\!]}$, we first draw a set $I \subset \{1,\dots,n\}$ of size $\lfloor \tau n \rfloor$ corresponding to the indices to contaminate; with $\tau \in [0, 1]$ the proportion of contaminated functions. Then, we perform different kinds of corruption:

- **Type 1**: Denote by $|I|$ the cardinal of the ordered set I and for $j \in [\![|I|]\!]$ denote by I_j the j-th element of I. Let ω be the permutation defined for $j \in [\![|I|]\!]$ as $\omega(I_j) = I_{j+1}$ if $j < |I|$ and $\omega(I_{|I|}) = I_1$, then for $i \in I$, the data point (x_i, y_i) is replaced by $(x_i, -y_{\omega(i)})$.

- **Type 2**: Given covariance parameters $\boldsymbol{\sigma} \in \mathbb{R}^r$ and an intensity parameter $\zeta > 0$, we draw a Gaussian process $g_c \sim \mathcal{GP}(0, k_{\sigma_c})$ for $c \in [\![r]\!]$ where k_{σ_c} is the Gaussian covariance function with standard deviation σ_c. Then, for $i \in I$, we replace (x_i, y_i) with $\left(x_i, \sum_{c\in[\![r]\!]} a_{ic} g_c\right)$ where the coefficients $(a_{ic})_{i,c=1}^{n,r}$ are drawn i.i.d. from a uniform distribution $\mathcal{U}([-0.5\zeta, 0.5\zeta])$.

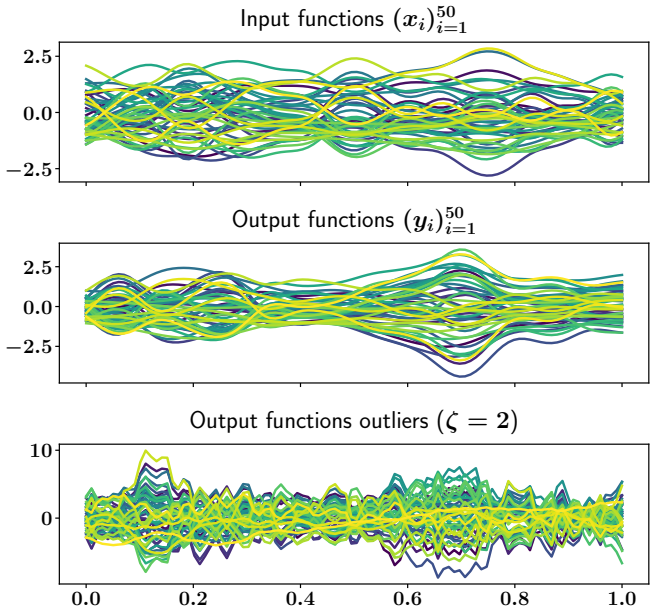


Figure 7.3: Examples from the toy dataset and corresponding type 2 outliers.

- **Type 3**: For each $i \in I$, a randomly chosen fraction $\xi \in [0,1]$ of the discrete observations for y_i is replaced by random draws from a uniform distribution $\mathcal{U}([-b_{\max}, b_{\max}])$, where $b_{\max} := \max_{i,s} |y_i(\theta_{is})|$.

The corruptions of Type 1 and 2 are global whereas that of Type 3 is local. In terms of characteristics, for Type 1, the outliers have similar functional properties as the non-outliers, whereas with Type 2, the outliers become completely different, as illustrated in the bottom panel of Figure 7.3. Finally, when selecting parameters through cross-validation on corrupted datasets, we replace the mean with the median to mitigate the effect of outliers on the selection.

Generation of the toy data

Given covariance parameters $(\boldsymbol{\sigma}^{\text{in}}, \boldsymbol{\sigma}^{\text{out}}) \in \mathbb{R}^r \times \mathbb{R}^r$ for $c \in [\![r]\!]$ we draw and fix Gaussian processes $g_c^{\text{in}} \sim \mathcal{GP}(0, k_{\sigma_c^{\text{in}}})$ and $g_c^{\text{out}} \sim \mathcal{GP}(0, k_{\sigma_c^{\text{out}}})$ where k_σ denotes the Gaussian kernel of standard deviation σ. We then generate n samples as

$$(x_i, y_i)_{i=1}^n = \left(\sum_{c \in [\![r]\!]} u_{ic} g_c^{\text{in}}, \sum_{c \in [\![r]\!]} u_{ic} g_c^{\text{out}} \right)_{i \in [\![n]\!]},$$

where the coefficients $(u_{ic})_{i,c=1}^{n,r}$ are drawn i.i.d. according to a uniform distribution $\mathcal{U}([-0.5, 0.5])$. In the experiments, we take $r = 4$ and set $\boldsymbol{\sigma}^{\text{in}} = \boldsymbol{\sigma}^{\text{out}} = (0.05, 0.1, 0.5, 0.7)$. We show input and output functions drawn in this manner in the first and second row of Figure 7.3. In the bottom row we display outliers of Type 2 with $\boldsymbol{\sigma} = (0.01, 0.05, 1, 4)$ and intensity $\zeta = 2$. For the contaminated indices $i \in I$ we add the corresponding outlier to the function y_i.

Optimization, evaluation and algorithms

We give here some details on optimization and introduce some evaluation metrics.

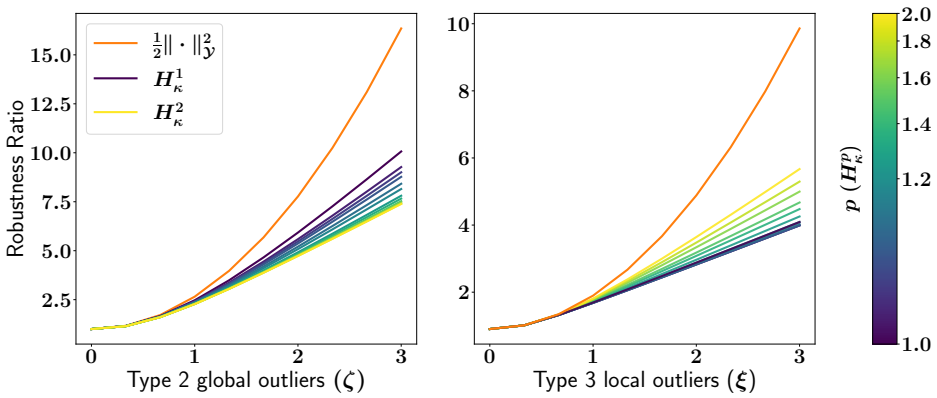


Figure 7.4: Sensitivity of H_κ^p to outliers for various $p \in [1, 2]$

Let $\left((y_i(\theta_{is}))_{s\in[\![m_i]\!]}\right)_{i\in[\![n]\!]}$ be the set of observed discretized functions and let $\left((\hat{y}_i(\theta_{is}))_{s\in[\![m_i]\!]}\right)_{i\in[\![n]\!]}$ be an estimated set of discretized functions, where $(\theta_{is})_{s\in[\![m_i]\!]}$ denotes the observation locations for y_i. We use the mean squared error defined as

$$\text{MSE} := \frac{1}{n}\sum_{i=1}^{n}\sum_{s=1}^{m}(y_i(\theta_{is}) - \hat{y}_i(\theta_{is}))^2.$$

When $m_i = m$ for all i, we normalize it by m and define NMSE:= $\frac{1}{m}$ MSE.

For the estimators related to the losses H_κ^1 and ℓ_ϵ^∞ we solve the problem based on the representation with linear splines (see Section 7.2.1 and Section 7.3.1 respectively); this is the only possible approach. However for the losses H_κ^2 and ℓ_ϵ^2 we exploit the representation using a truncated basis of approximate eigenfunctions (see Section 7.3.1 and Section 7.3.2 respectively), in doing so we reduce the computational cost. Concerning optimization, we deploy the APGD method (Beck and Teboulle, 2009) with backtracking line search, and adaptive restart (O'Donoghue and Candès, 2015). The initialization in APGD is carried out with the closed-form solution available for the square loss using a Sylvester solver.

However, before evaluating the estimators, we first investigate general robustness properties of the Huber loss.

7.4.2 Influence of p for H_κ^p

In this section we study empirically how the choice of p affects the sensitivity of the Huber loss H_κ^p to different kinds of outliers.

We recall that solving Problem 7.18 involves the computation of n projections on a q-ball at each APGD iteration. For $p \notin \{1, 2\}$, such projection must be computed in an iterative fashion and running APGD with these inner iterations is too time consuming. However to compute the loss H_κ^p using Proposition 7.8 only one projection is necessary per loss computation. We exploit this to study empirically the sensitivity of the Huber losses H_κ^p to global and local outliers for different values of p.

The impact of the outliers on the solution of a regularized empirical risk minimization problem is partly determined by their contribution to the data-fitting term compared to that of the normal observations. In order to investigate this aspect, we next study

and define a quantity which we call robustness ratio.

Robustness ratio. Let $(e_i)_{i=1}^n \in \mathcal{Y}^n$ be a set of functional residuals and let $(\tilde{e}_i)_{i=1}^n$ be the same functional residuals but contaminated with outliers. In practice, we have to choose a probability distribution to draw the $(e_i)_{i=1}^n$ from, and an outlier distribution to corrupt those. For the residuals, we use the synthetic data generation process, and for the outliers, we consider type 2 for global outliers and type 3 for local ones. We then define the *robustness ratio* as

$$\text{Robustness Ratio} := \inf_{\kappa \geq 0} \frac{1}{n} \sum_{i=1}^{n} \frac{H_\kappa^p(\tilde{e}_i)}{H_\kappa^p(e_i)}.$$

The best value is 1: the loss is not affected at all by the outliers, but it is indeed unachievable. In practice, we restrain to $p \in [1, 2]$, and for each value, we reduce the search for κ to empirical quantiles of the q-norms of the uncorrupted functions $(e_i)_{i=1}^n$, with q being p's conjugate exponent. It makes sense to do so since κ corresponds to a q-norm threshold which should separate suspected outliers from observations deemed normal (see Proposition 7.8). In practice, for each level of corruption and each p, we compute the robustness ratio for κ equal to the $\{0.5, 0.6, 0.7, 0.8, 0.9, 0.95, 0.99\}$-th empirical quantiles and select the value which minimizes the ratio. This indeed corresponds to an ideal setting: in practice we never have access to the uncorrupted data and we can never optimize κ in this way. Thus this analysis reflects a general robustness property of the loss in an optimal setting.

Empirical study of the influence of p. In accordance with one's expectation, when the data is contaminated with global outliers (left panel of Figure 7.4), it is better to choose $p = 2$ whereas when the contamination is local (right panel of Figure 7.4), $p = 1$ is almost the best choice; even though it seems that choosing p slightly larger could be a tad better. Nevertheless this analysis has its limits; indeed we do not take into account the interplay between the data-fitting term and the regularization term which takes place during optimization. This could explain why the losses H_κ^1 and H_κ^2 perform equally well for global outliers later in this section, whereas based only on the robustness ratio analysis (left panel of Figure 7.4) we would have said otherwise. However, the findings in presence of local outliers (right panel of Figure 7.4) are coherent with what we observe later for the losses H_κ^1 and H_κ^2.

Now that we have investigated the robustness as a general property of the Huber loss, we study the behavior of our proposed estimators associated with the Huber loss for $p = 1$ and $p = 2$, and to the ϵ-insensitive loss for $p = 2$ and $p = +\infty$.

7.4.3 Robustness and sparsity on synthetic data

The impact of the different losses are investigated in detail on the function-to-function synthetic dataset we introduced in Section 7.4.1. The kernels $k_\mathcal{X}$ and k_Θ are chosen to be Gaussian and the experiments are averaged over 20 draws with training and testing samples of size 100. We give further details of the tuning procedure in Appendix B.4.

Sparsity-accuracy trade-off for the ϵ-insensitive loss

To study the interaction between λ and ϵ and the resulting sparsity-accuracy trade-offs, we add i.i.d. Gaussian noise with standard deviation 0.5 to the observations of the output functions. The resulting NMSE values are summarized in Figure 7.5. For both

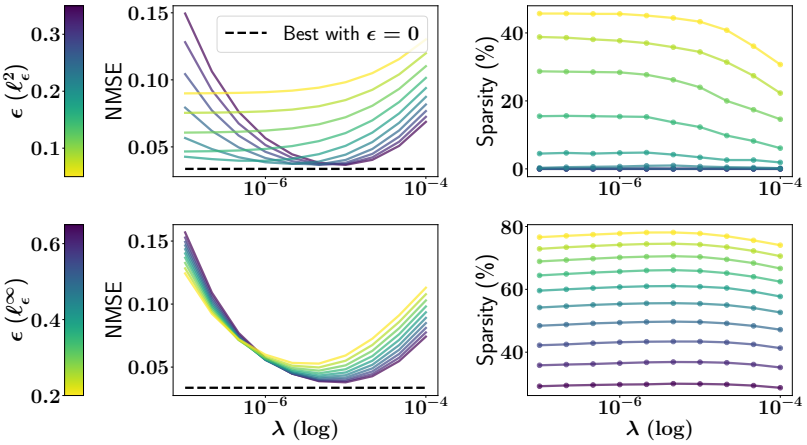


Figure 7.5: Interaction between regularization λ and insensitivity ϵ for the loss ℓ^2_ϵ (1st row) and ℓ^∞_ϵ (2nd row).

the ℓ^2_ϵ and the ℓ^∞_ϵ loss, one can reduce λ and increase ϵ so as to obtain a fair amount of sparsity while making a small compromise in terms of accuracy. We highlight that the type of sparsity is not the same for the loss ℓ^2_ϵ and for the loss ℓ^∞_ϵ. In the former case, we use a truncated basis of eigenfunctions to represent the dual variables, and therefore we have much less coefficients than in the latter case where each dual variable has the same length as the observed functions. Then, in the case of $p = 2$, we have a lesser sparsity percentage, but it is row-wise and the number of overall coefficients (nd) is much lower than for $p = +\infty$ (nm coefficients).

Robustness with Huber loss

We now investigate the robustness of the Huber loss to different types of outliers while selecting both λ and κ through robust cross-validation. The resulting NMSE values are summarized in Figure 7.6. As it can be seen in the first row, the losses H^1_κ and H^2_κ are significantly more robust to global outliers than the square loss, both when their intensity ζ and the proportion τ of contaminated samples increase. When the contamination is local, the second row of the figure shows the closer one gets to the whole sample being contaminated ($\tau = 1$), the less robust H^2_κ becomes. On the other hand, looking at the bottom right panel, we see that H^1_κ is remarkably robust (when the whole sample is contaminated $\tau = 1$). One can interpret this phenomenon by noticing that the loss H^2_κ can be less sensitive to big discrepancies between functions in the $\|.\|_\mathcal{Y}$ norm sense, but if all samples are contaminated locally a little, the outliers are meddled in the norm and so H^2_κ becomes inefficient.

Interaction between λ and κ parameter for the Huber loss

To highlight the interaction between the regularization parameter λ and the parameter κ of the Huber loss, we plot the NMSE values for various values of λ and κ using the toy dataset corrupted with Type 2 (global) and Type 3 (local) outliers. The results are displayed in Figure 7.7b and confirm that by making κ and λ vary, when the data are corrupted, we can always find a configuration that is significantly more robust than the square loss. In accordance with one's expectation, when dealing with

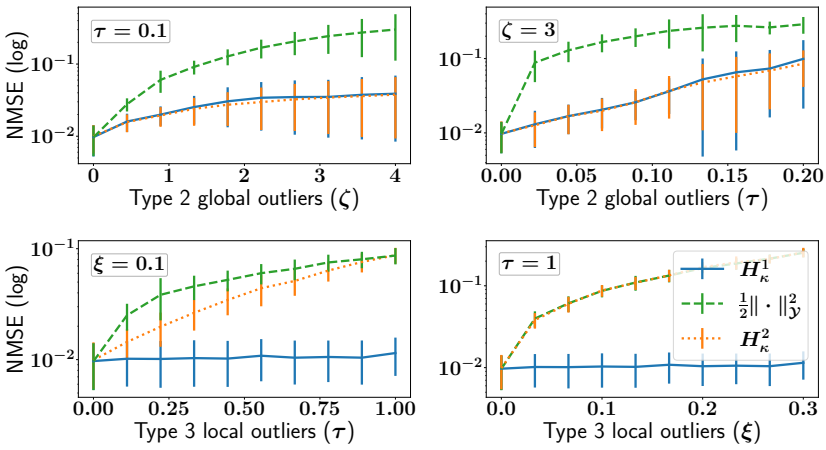
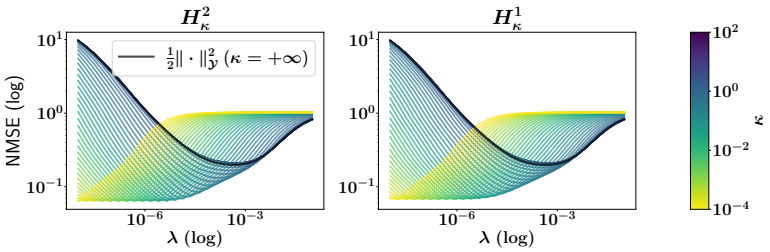
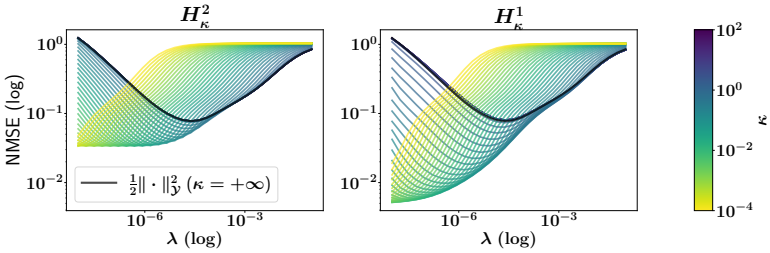


Figure 7.6: Robustness to different types of outliers.



(a) Type 2 outliers with $\tau = 0.2$ and $\zeta = 2$



(b) Type 3 outliers with $\tau = 0.3$ and $\xi = 0.3$

Figure 7.7: NMSE as a function of λ and Huber losses' κ with two types of outliers.

local outliers (Figure 7.7a), the loss H_κ^1 is much more efficient than the loss H_κ^2. However, when the outliers are global (Figure 7.7a), the two losses perform equally well.

7.4.4 Experiments on the DTI dataset

In our next experiment we considered the DTI benchmark[1]. We have already used this dataset in Section 5.3.3, we refer to this section for a detailed description. As a brief reminder, the dataset contains a collection of fractional anisotropy profiles deduced from diffusion tensor imaging scans, and we take the first scans of the $n = 100$ multiple sclerosis patients. The profiles are given along two tracts, the corpus callosum and the right corticospinal. The goal is to predict the latter function from the former, which can be framed as a function-to-function regression problem. When some functions admit missing observations, we fill in the gaps by linear interpolation, and later use

[1] This dataset was collected at the Johns Hopkins University and the Kennedy-Krieger Institute.

Table 7.3: MSEs and sparsity on the DTI dataset

λ	Metric	$1/2\|\|\cdot\|\|^2_{\mathcal{Y}}$	H^2_κ	H^1_κ	ℓ^2_ϵ	ℓ^∞_ϵ
10^{-5}	MSE (10^{-1})	2.5±0.19	**2.21**±0.31	**2.21**±0.31	2.41±0.26	2.5±0.23
	Sparsity	-	-	-	27.4±17.2%	85.9±10.7%
10^{-3}	MSE (10^{-1})	**2.18**±0.27	2.23±0.32	2.21±0.32	2.2±0.29	**2.18**±0.28
	Sparsity	-	-	-	3.4±6.9%	12.7±10.5%

Table 7.4: MSEs on speech data

VT	$1/2\|\|\cdot\|\|^2_{\mathcal{Y}}$	H^2_κ	H^1_κ	ℓ^2_ϵ	ℓ^∞_ϵ
LP	**6.58**±0.62	6.59±0.62	6.59±0.64	**6.58**±0.62	**6.58**±0.62
LA	4.65±0.55	4.65±0.55	4.66±0.55	**4.64**±0.55	**4.64**±0.55
TBCL	**4.26**±0.46	**4.26**±0.46	4.27±0.46	**4.26**±0.46	**4.26**±0.46
TBCD	4.67±0.37	4.68±0.38	4.7±0.38	**4.67**±0.38	**4.67**±0.38
VEL	**2.94**±0.5	**2.94**±0.5	2.95±0.5	**2.94**±0.5	**2.94**±0.5
GLO	**7.25**±0.65	7.26±0.65	**7.25**±0.64	**7.25**±0.65	**7.25**±0.65
TTCL	3.76±0.21	3.76±0.21	3.74±0.2	**3.73**±0.21	**3.73**±0.21
TTCD	5.93±0.34	5.94±0.34	5.93±0.35	**5.92**±0.34	**5.92**±0.34

Table 7.5: MSEs on contaminated speech data

VT	Type 1 Outliers ($\tau = 0.1$)			Type 3 outliers ($\tau = 0.1$, $\xi = 0.1$)		
	$1/2\|\|\cdot\|\|^2_{\mathcal{Y}}$	H^2_κ	H^1_κ	$1/2\|\|\cdot\|\|^2_{\mathcal{Y}}$	H^2_κ	H^1_κ
LP	9.4±0.75	9.4±0.66	**9.19**±0.79	7.53±0.58	7.62±0.59	**7.0**± 0.59
LA	5.72±0.76	5.63±0.71	**5.52**±0.69	**5.06**±0.6	5.11±0.6	5.09±0.55
TBCL	6.71±0.96	6.14±0.97	**5.98**±0.93	5.06±0.51	5.16±0.48	**4.72**± 0.54
TBCD	**5.8**±0.41	5.86±0.44	5.83±0.44	5.18±0.4	5.26±0.41	**5.08**± 0.4
VEL	4.37±0.56	3.76±0.62	**3.76**±0.59	3.52±0.57	3.54±0.58	**3.41**± 0.57
GLO	9.61±0.87	**9.51**±0.86	9.53±0.84	7.94±0.61	8.02±0.61	**7.76**± 0.61
TTCL	15.06±2.22	9.51±0.63	**9.48**±0.6	**5.89**±0.43	5.91±0.45	6.62±0.66
TTCD	8.15±0.48	**7.96**±0.49	8.02±0.51	6.63±0.44	6.74±0.42	**6.36**± 0.39

the MSE as a metric. We use a Gaussian kernel for $k_{\mathcal{X}}$, a Laplacian one for k_Θ and average over 10 runs with $n_{\text{train}} = 70$ and $n_{\text{test}} = 30$. Note that we give further details on the procedure and parameters used in Appendix B.5.

The results are coherent with those obtained on the synthetic data: a compromise can be made between the two parameters λ and ϵ to get increased sparsity, as can be observed in Table 7.3. Moreover, we highlight that even for optimal regularization with respect to the square loss $\lambda = 10^{-3}$ (in the sense that it results in the best in the best average score on the test set), one gets a fair amount of sparsity while getting the same score with ℓ^∞_ϵ and a very small difference with ℓ^2_ϵ.

7.4.5 Speech data

In this section, we focus on a speech inversion problem (Mitra et al., 2009). We have already studied this problem with the same dataset in Section 5.3.4, we refer the reader to this section for more details. To describe the problem briefly, our goal is to predict a vocal tract (VT) configuration that likely produced a speech signal (Richmond, 2002). This benchmark encompasses $n = 413$ synthetically pronounced words

to which 8 VT functions are associated: LA, LP, TTCD, TTCL, TBCD, TBCL, VEL, GLO. We predict the VT functions separately in eight subproblems.

Since the words are of varying length, we use the MSE as metric and extend symmetrically the signals to match the longest word for in training. We encode the input sounds through 13 mel-frequency cepstral coefficients (MFCC) and normalize the VT functions' values to the range [-1, 1]. We average over 10 train-test splits taking $n_{\text{train}} = 250$ and $n_{\text{test}} = 163$. Finally we take an integral Gaussian kernel on the standardized MFCCs (see Equation (5.39) for details) as $k_{\mathcal{X}}$ and a Laplace kernel as k_{Θ}. Note that we give further details on the procedure and parameters used in Appendix B.6.

We first compare all the losses on untainted data in Table 7.4. Then to evaluate the robustness of the Huber losses, we ran experiments on contaminated data with two configurations. In the first case, we add Type 1 (global) outliers with $\tau = 0.1$ and in the second one, we add Type 3 (local) outliers with $\tau = 0.1$ and $\xi = 0.1$. The results are displayed in Table 7.5. In the contaminated setting, one gets results similar to ones obtained on the synthetic dataset. The loss H_{κ}^{1} works especially well for local outliers whereas the loss H_{κ}^{2} is robust only to global outliers.

7.5 Conclusion

We proposed extensions of the Huber and ϵ-insensitive losses for functional data. Compared to existing formulation, we framed these as infimal convolution between the square loss and the functional p-norm or the indicator function of a p-norm ball respectively. The parameter p introduces the possibility to enforce several type of robustness or sparsity, local or global. We then used function-valued RKHS as hypothesis class, and tackled the corresponding empirical risk minimization problem through dualization. To overcome the challenges stemming from both the dual terms associated to the losses and the functional nature of the dual variables, we proposed appropriate representations. Then we solved the resulting approximate problems using accelerated proximal gradient descent for the values of p allowing for computation of the proximity operators in closed-form.

8

Kernel projection learning: extensions and improvements

Contents

This short chapter is dedicated to our ongoing work to improve the projection learning framework and especially kernel projection learning. We covered those subjects in Chapter 5. However, studying our estimators led us to raise several questions. The first one is particularly relevant when using redundant dictionaries. As we have seen, the Gram matrix $\Phi^{\#}\Phi$ of the dictionary plays a central role in the computation of the estimators, however the more redundant the dictionary is, the more rank-deficient $\Phi^{\#}\Phi$ becomes. This degrades the conditioning of the problem and means we are predicting too much representation coefficients. We address this issue in Section 8.1. So far we have focused solely on reducing the complexity linked to the functional outputs, however as a kernel method, the computational time complexity of KPL is cubic with respect to the number of observations. To address this issue, we propose in Section 8.2 a linear version of projection learning amenable to the use of input features so that random Fourier features or Nyström features can be used. Finally, this linear version allows for an explicit control on the coefficients of the model associated to each atom of the dictionary. We propose to exploit this to automatically select the relevant atoms using a regularization in the $(1,2)$-norm in Section 8.3.

8.1 Effective rank approach

We recall briefly the problem of kernel projection learning but refer back to Section 5.1.1 for a more detailed presentation. After that we motivate and introduce a modified estimation procedure.

8.1.1 Motivation

Given a space $\mathcal{X}$, a separable functional Hilbert space $\mathcal{Y}$, let X and Y be random variables taking their values respectively in $\mathcal{X}$ and $\mathcal{Y}$. From any observation of X, we want to associate a relevant value in $\mathcal{Y}$ in accordance with the joint distribution of (X,Y). As we have access to a finite i.i.d. sample $(x_i, y_i)_{i=1}^n$, we rely on statistical inference. To do so we minimize a regularized empirical risk over a hypothesis class $\mathcal{G} \subset \mathcal{F}(\mathcal{X},\mathcal{Y})$ of possible models. To quantify the discrepancies in $\mathcal{Y}$, a loss $L : \mathcal{Y} \to \mathbb{R}$ is used. The main challenge is that $\mathcal{Y}$ is potentially infinite dimensional. A possible way to circumvent this is to use a model representing the functions in $\mathcal{Y}$ using a dictionary of functions $\boldsymbol{\phi} = (\phi_l)_{l=1}^d \in \mathcal{Y}^d$. To that end, we recall the definition (Definition 5.2) of the projection operator associated to the dictionary $\Phi : \mathbf{a} \in \mathbb{R}^d \mapsto \sum_{l=1}^d a_l \phi_l \in \mathcal{Y}$. We then use an intermediary hypothesis class $\mathcal{H} \subset \mathcal{F}(\mathcal{X},\mathbb{R}^d)$ to predict coefficients on the dictionary through this operator. We studied in details in Section 5.2 the specific case where $\mathcal{H} = \mathcal{H}_\mathsf{K}$ is the vector-valued reproducing kernel Hilbert space (vv-RKHS) associated to an operator-valued kernel (OVK) $\mathsf{K} : \mathcal{X} \times \mathcal{X} \to \mathcal{L}(\mathbb{R}^d)$, and the vv-RKHS norm is used as regularization. The following problem results:

$$\min_{h \in \mathcal{H}_\mathsf{K}} \frac{1}{n} \sum_{i=1}^n L(y_i - \Phi h(x_i)) + \lambda \|h\|^2_{\mathcal{H}_\mathsf{K}}. \tag{8.1}$$

We have highlighted in Corollary 5.5 that Problem 8.1 benefits from a finite parametrization through a representer theorem. More precisely, any solution $h^\lambda_\mathbf{z}$ to Problem 8.1 has the form

$$h^\lambda_\mathbf{z} = \sum_{j=1}^n \mathsf{K}_{x_j} \alpha_j, \tag{8.2}$$

for some $\boldsymbol{\alpha} \in \mathbb{R}^{d \times n}$.

For the square loss, injecting this representation into Problem 8.1 leads to

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^{d \times n}} \frac{1}{n} \|\mathbf{y} - \Phi_{(n)} \mathbf{K} \boldsymbol{\alpha}\|^2_{\mathcal{Y}^n} + \lambda \langle \boldsymbol{\alpha}, \mathbf{K} \boldsymbol{\alpha} \rangle_{\mathbb{R}^{d \times n}}. \tag{8.3}$$

where the operator $\mathbf{K}$ associated to the input observations is defined from the OVK K as in Equation (5.14). We showed in Proposition 5.13 that provided $\mathbf{K}$ is invertible, the optimal representer coefficients can be computed in closed-form as

$$\hat{\boldsymbol{\alpha}} = ((\Phi^\# \Phi)_{(n)} \mathbf{K} + n\lambda \mathbf{I})^{-1} \Phi^\#_{(n)} \mathbf{y}. \tag{8.4}$$

However, since $\operatorname{Rank}\left((\Phi^\# \Phi)_{(n)}\right) = n \operatorname{Rank}(\Phi^\# \Phi)$, we have that

$$\operatorname{Rank}\left((\Phi^\# \Phi)_{(n)} \mathbf{K}\right) \leq \min\left(n \operatorname{Rank}(\Phi^\# \Phi), \operatorname{Rank}(\mathbf{K})\right).$$

This implies that if the operator $\Phi^\# \Phi$ is not full rank, neither is the operator $(\Phi^\# \Phi)_{(n)} \mathbf{K}$. However, since $\operatorname{Rank}(\Phi^\# \Phi) = \operatorname{Dim}(\operatorname{Span}(\boldsymbol{\phi}))$, when the dictionary is redundant, $\Phi^\# \Phi$ is indeed rank-deficient. This degrades the conditioning of the problem, even though the regularization in Problem 8.3 partially addresses the issue by ensuring $(\Phi^\# \Phi)_{(n)} \mathbf{K} + n\lambda \mathbf{I}$

is invertible. Moreover, if the dictionary is too redundant, predicting d coefficients (one per atom in ϕ) is too much compared to the actual dimension of $\mathrm{Span}(\phi)$. These facts led us to consider a modified estimation procedure.

8.1.2 Effective-rank KPL

A natural idea is to perform an eigendecomposition of $\Phi^{\#}\Phi$ and parameterize our vector-valued model using the eigenvectors associated to significant eigenvalues (non-zero or above a small threshold). This idea is a classic one. It has been investigated early to address the issues posed by highly correlated predictors in linear regression Massy (1965). It also has been applied to reduce the dimension of the outputs in the multivariate linear model (Izenman, 1975; Reinsel and Velu, 1998).

We propose to use this approach to increase the efficiency of projection learning when the dictionary is redundant. Let $(\mathbf{w}_l, \upsilon_l)_{l=1}^d$ be an eigendecomposition of the positive and symmetric matrix $\Phi^{\#}\Phi$. Let $\mathrm{W} \in \mathbb{R}^{d\times d}$ be the orthogonal matrix which columns are the eigenvectors $(\mathbf{w}_l)_{l=1}^d$ and let Υ be the diagonal matrix containing the eigenvalues in decreasing order, we have that

$$\Phi^{\#}\Phi = \mathrm{W}\Upsilon\mathrm{W}^{\mathrm{T}}.$$

Let $\tau > 0$ be a small threshold at which we wish to truncate the eigenvalues and let $r \in \mathbb{N}^*$ be the number of eigenvalues greater than τ. We then consider the eigenvectors $\check{\mathrm{W}} \in \mathbb{R}^{d\times r}$ associated to those eigenvalues and the diagonal matrix containing them $\check{\Upsilon} \in \mathbb{R}^{r\times r}$. We propose to study the problem

$$\min_{\check{h}\in\mathcal{H}_{\check{\mathrm{K}}}} \frac{1}{n}\sum_{i=1}^{n} L(y_i - \Phi\check{\mathrm{W}}\check{h}(x_i)) + \lambda\|\check{h}\|^2_{\mathcal{H}_{\check{\mathrm{K}}}}, \tag{8.5}$$

where now we consider an OVK $\check{\mathrm{K}} : \mathcal{X}\times\mathcal{X} \to \mathcal{L}(\mathbb{R}^r)$ and $\mathcal{H}_{\check{\mathrm{K}}}$ its associated vv-RKHS; $L : \mathcal{Y} \to \mathbb{R}$ being a functional loss. In the very same fashion as Problem 8.1, Problem 8.5 benefits from a representer theorem (see Theorem 2.36). Consequently, any solution to this problem $\check{h}^\lambda$ has the form

$$\check{h}^\lambda = \sum_{j=1}^{n} \check{\mathrm{K}}_{x_j}\alpha_j, \tag{8.6}$$

for some $\boldsymbol{\alpha} \in \mathbb{R}^{r\times n}$.

Dimensionality reduction for $\|\cdot\|_{\mathcal{Y}}$-composed losses

We have motivated this new problem when the square loss is used through two aspects: reduction of the dimension and improvement of the conditioning of the matrix to invert. However, even for other losses the former aspect can remain relevant. However, we need however to first formalize a specific form of loss.

Definition 8.1 ($\|\cdot\|_{\mathcal{Y}}$-composed loss)**.** *We say that a loss function $L : \mathcal{Y} \to \mathbb{R}$ is a $\|\cdot\|_{\mathcal{Y}}$-composed loss if it can be written as*

$$\forall y \in \mathcal{Y}, \quad L(y) = \ell\Big(\|y\|_{\mathcal{Y}}\Big), \tag{8.7}$$

with $\ell : \mathbb{R} \to \mathbb{R}$ a loss on the real line.

Example 8.2 (Huber and ϵ-insensitive losses)**.** *Examples of such losses include the Huber loss H_κ^2 and the ϵ-insensitive loss ℓ_ϵ^2 that we studied in Chapter 7. Indeed, we have*

$$H_\kappa^2 = h_\kappa\left(\|\cdot\|_{\mathcal{Y}}\right) \quad \textit{and} \quad \ell_\epsilon^2 = l_\epsilon\left(\|\cdot\|_{\mathcal{Y}}\right),$$

where h_κ and l_ϵ are respectively the Huber loss of parameter κ and the quadratic epsilon loss with parameter ϵ on the real line.

The following lemma justifies posing Problem 8.5 in order to reduce the dimension when L is $\|\cdot\|_{\mathcal{Y}}$-composed.

Lemma 8.3. *Consider an eigendecomposition of the Gram matrix of the dictionary in an orthonormal basis $\Phi^\#\Phi = \mathrm{W}\Upsilon\mathrm{W}^\mathrm{T}$. Suppose the eigenvalues are sorted in decreasing order. Let $\check{\mathrm{W}} \in \mathbb{R}^{d\times r}$ be the matrix which columns are the eigenvectors associated to non-zero eigenvalues. Let us denote for any vector $\mathbf{a} \in \mathbb{R}^d$, $\check{\mathbf{a}} := (a_l)_{l=1}^r \in \mathbb{R}^r$. Then for any $\mathbf{a} \in \mathbb{R}^d$ and any $y \in \mathcal{Y}$,*

$$\|\Phi\mathrm{W}\mathbf{a} - y\|_{\mathcal{Y}} = \|\Phi\check{\mathrm{W}}\check{\mathbf{a}} - y\|_{\mathcal{Y}}. \tag{8.8}$$

Proof Let $\mathrm{Im}(\Phi)$ be the range of the operator Φ (it corresponds to the span of the dictionary), since Φ is of rank at most d, $\mathrm{Im}(\Phi)$ is a finite dimensional subspace of $\mathcal{Y}$. Consequently, any vector can be decomposed as $y = y_0 + y^\perp$ with $y_0 \in \mathrm{Im}(\Phi)$ and $y^\perp \in (\mathrm{Im}(\Phi))^\perp$. Then, by definition, there exists $\mathbf{a}_0 \in \mathbb{R}^d$ such that $y_0 = \Phi\mathbf{a}_0$. Using Pythagoras' theorem, we have that

$$\begin{aligned}\|\Phi\mathrm{W}\mathbf{a} - y\|_{\mathcal{Y}}^2 &= \|\Phi\mathrm{W}\mathbf{a} - \Phi\mathbf{a}_0 - y^\perp\|_{\mathcal{Y}}^2 \\ &= \|\Phi\mathrm{W}\mathbf{a} - \Phi\mathbf{a}_0\|_{\mathcal{Y}}^2 + \|y^\perp\|_{\mathcal{Y}}^2.\end{aligned} \tag{8.9}$$

Developing the norm in the first term of the right hand side of the previous equation, we get that

$$\begin{aligned}\|\Phi\mathrm{W}\mathbf{a} - \Phi\mathbf{a}_0\|_{\mathcal{Y}}^2 &= \mathbf{a}^\mathrm{T}\mathrm{W}^\mathrm{T}\Phi^\#\Phi\mathrm{W}\mathbf{a} - 2\mathbf{a}_0^\mathrm{T}\Phi^\#\Phi\mathrm{W}\mathbf{a} + \|\Phi\mathbf{a}_0\|_{\mathcal{Y}}^2 \\ &= \mathbf{a}^\mathrm{T}\Upsilon\mathbf{a} - 2\mathbf{a}_0^T\mathrm{W}\Upsilon\mathbf{a} + \|\Phi\mathbf{a}_0\|_{\mathcal{Y}}^2.\end{aligned} \tag{8.10}$$

Let $\check{\Upsilon} \in \mathbb{R}^r$ be the diagonal matrix containing the nonzero eigenvalues of $\Phi^\#\Phi$. Then since $\Upsilon\mathbf{a} = (\upsilon_1 a_1, \cdots, \upsilon_r a_r, 0, ..., 0)^\mathrm{T}$, we have that

$$\mathbf{a}^\mathrm{T}\Upsilon\mathbf{a} - 2\mathbf{a}_0^\mathrm{T}\mathrm{W}\Upsilon\mathbf{a} = \check{\mathbf{a}}^\mathrm{T}\check{\Upsilon}\check{\mathbf{a}} - 2\mathbf{a}_0^\mathrm{T}\check{\mathrm{W}}\check{\Upsilon}\check{\mathbf{a}}.$$

Consequently, injecting this into Equation (8.10) yields

$$\|\Phi\mathrm{W}\mathbf{a} - \Phi\mathbf{a}_0\|_{\mathcal{Y}}^2 = \|\Phi\check{\mathrm{W}}\check{\mathbf{a}} - \Phi\mathbf{a}_0\|_{\mathcal{Y}}^2. \tag{8.11}$$

Then, combining Equation (8.11) with Equation (8.9) concludes the proof. ■

As a consequence of Lemma 8.3, if L is $\|\cdot\|_{\mathcal{Y}}$-composed (Definition 8.1), we can predict only r coordinates corresponding to the eigenvectors associated to strictly positive eigenvalues of $\Phi^\#\Phi$. This hints as well that in order to reduce the dimension of the problem, it can make sense to consider only eigenvectors associated to eigenvalues which are greater than a threshold $\tau > 0$.

Dimensionality reduction and improved conditioning for the square loss

Now let us focus on the square loss. The corresponding problem is the following

$$\min_{\boldsymbol{\alpha}\in\mathbb{R}^{r\times n}} \frac{1}{n}\|\mathbf{y}-(\Phi\check{\mathrm{W}})_{(n)}\check{\mathbf{K}}\boldsymbol{\alpha}\|^2_{\mathcal{Y}^n} + \lambda\langle\boldsymbol{\alpha},\check{\mathbf{K}}\boldsymbol{\alpha}\rangle_{\mathbb{R}^{r\times n}}, \tag{8.12}$$

where $\check{\mathbf{K}}\in\mathcal{L}(\mathbb{R}^{r\times n})$ corresponds to the the following operator associated to the OVK $\check{\mathrm{K}}$ and the input observations

$$\check{\mathbf{K}}\colon \begin{pmatrix} \mathbb{R}^{r\times n} & \to \mathbb{R}^{r\times n} \\ \boldsymbol{\alpha} & \mapsto \left[\sum_{j=1}^n \check{\mathrm{K}}(x_i,x_j)\alpha_j\right]_{i=1}^n \end{pmatrix}. \tag{8.13}$$

All the derivations for the KPL ridge estimator performed in Section 5.2.2 are still relevant if we replace Φ by the linear operator $\Phi\check{\mathrm{W}}$ resulting in the following effective-rank estimator for KPL.

Corollary 8.4 (Effective-rank ridge estimator)**.** *The minimum in Problem 8.12 is achieved by any $\hat{\boldsymbol{\alpha}}\in\mathbb{R}^{r\times n}$ verifying*

$$\left(\check{\mathrm{K}}\check{\Upsilon}_{(n)}\check{\mathrm{K}} + n\lambda\check{\mathbf{K}}\right)\hat{\boldsymbol{\alpha}} = \check{\mathbf{K}}(\check{\mathrm{W}}^{\mathrm{T}}\Phi^{\#})_{(n)}\mathbf{y}. \tag{8.14}$$

Moreover if $\check{\mathbf{K}}$ is full rank then $\left(\check{\Upsilon}_{(n)}\check{\mathbf{K}} + n\lambda\mathbf{I}\right)$ is invertible and $\hat{\boldsymbol{\alpha}}$ is unique and defined as

$$\hat{\boldsymbol{\alpha}} = \left(\check{\Upsilon}_{(n)}\check{\mathbf{K}} + n\lambda\mathbf{I}\right)^{-1}(\check{\mathrm{W}}^{\mathrm{T}}\Phi^{\#})_{(n)}\mathbf{y}. \tag{8.15}$$

We define the effective-rank ridge estimator as

$$\check{h}^{\lambda} := \sum_{j=1}^n \check{\mathrm{K}}_{x_j}\hat{\alpha}_j. \tag{8.16}$$

Proof This is derived exactly as Proposition 5.13, noticing that

$$\check{\mathrm{W}}^{\mathrm{T}}\Phi^{\#}\Phi\check{\mathrm{W}} = \check{\Upsilon}.$$

■

Provided the OVK is separable as $\check{\mathrm{K}} = k_{\mathcal{X}}\check{\mathrm{B}}$, we can rewrite the above system as a discrete time Sylvester equation in Equation (5.22):

$$\check{\Upsilon}\check{\mathrm{B}}\boldsymbol{\alpha}\mathrm{K}_{\mathcal{X}} + n\lambda\boldsymbol{\alpha} = (\check{\mathrm{W}}^{\mathrm{T}}\Phi^{\#})_{(n)}\mathbf{y}, \tag{8.17}$$

and solve it in the same fashion using either a Sylvester solver or an eigendecomposition exploiting the Kronecker structure in $\mathcal{O}(n^3+r^3)$ time. Even though the r largest eigenvalues and their associated eigenvectors must be computed beforehand, it can be done much faster than computing the whole eigendecomposition if r is significantly smaller than d (Golub and Van Loan, 2013).

Example 8.5 (Case $\check{\mathrm{B}} = \mathrm{I}$). *In practice, we will generally take* $\check{\mathrm{B}} = \mathrm{I}$, *in which case the rows* $(\hat{\alpha}_{l\cdot})_{l=1}^{r}$ *of* $\hat{\boldsymbol{\alpha}}$ *can be computed separately for* $l \in [\![r]\!]$ *as*

$$\hat{\alpha}_{l\cdot} = \frac{1}{v_l}\Big(\mathrm{K}_{\mathcal{X}} + n\frac{\lambda}{v_l}\mathrm{I}\Big)^{-1}\Big((\check{\mathrm{W}}^{\mathrm{T}}\Phi^{\#})_{(n)}\mathbf{y}\Big)_{l\cdot}.$$

Therefore, only computing an eigendecomposition of $\mathrm{K}_{\mathcal{X}}$ *is sufficient and then the inverses* $\left(\left(\mathrm{K}_{\mathcal{X}} + n\frac{\lambda}{v_l}\mathrm{I}\right)^{-1}\right)_{l=1}^{r}$ *can be computed easily without adding significant terms to the computational complexity. This option is especially attractive to perform cross-validation on the regularization parameter* λ.

Remark 8.6 (Further rank-reduction). *We considered the eigenvectors of* $\Phi^{\#}\Phi$ *associated to strictly positive eigenvalues. However, in practice we can filter out small eigenvalues as well, for instance by defining a threshold, so as to further smooth the problem.*

8.2 Feature projection learning (FPL)

The key motivation behind projection learning is to circumvent the numerical issues created by the infinite-dimensional outputs. By predicting coefficients in an adapted dictionary, the numerical complexity linked to the outputs is greatly diminished. We now focus on reducing the complexity with respect to the number of observations n. As we have seen in Section 2.1, kernel methods are well-known for their capacity to model complex nonlinear dynamics within a sound mathematical framework. Nevertheless they suffer from a bad dependency (essentially cubic) in n. Several possibilities exist to circumvent this. We presented the two main ones in Section 2.1.4, and both boil down to using appropriate features in a linear model, either random Fourier or Nyström ones—see respectively Equation (2.15) and Example 2.20. Consequently, we propose to develop a linear version of projection learning, and indeed other non-kernel related features can be used as well.

Let us consider that we are given a set of input features $(\mathbf{x}_i)_{i=1}^{n} \in (\mathbb{R}^q)^n$ of dimension $q \in \mathbb{N}^*$ stacked in a matrix $\mathrm{X} \in \mathbb{R}^{q\times n}$ which are computed from the input observations. We then wish to study the linear projection learning problem with coefficients $\mathrm{C} \in \mathbb{R}^{q\times d}$:

$$\min_{\mathrm{C}\in\mathbb{R}^{d\times q}} \frac{1}{n}\sum_{i=1}^{n} L(y_i - \Phi(\mathrm{C}^{\mathrm{T}}\mathbf{x}_i)) + \Omega(\mathrm{C}), \tag{8.18}$$

where $\Omega : \mathbb{R}^{q\times d} \to \mathbb{R}$ is a regularization function and $L : \mathcal{Y} \to \mathbb{R}$ is a loss on $\mathcal{Y}$.

We propose first to derive an estimator in closed-form for the square loss with square norm regularization, the corresponding optimization problem is

$$\min_{\mathrm{C}\in\mathbb{R}^{q\times d}} \frac{1}{n}\|\Phi_{(n)}(\mathrm{C}^{\mathrm{T}}\mathrm{X}) - \mathbf{y}\|_{\mathcal{Y}^n}^2 + \lambda\|\mathrm{C}\|_{\mathbb{R}^{q\times d}}^2, \tag{8.19}$$

with $\lambda > 0$ controlling the intensity of the regularization. This is a convex problem, therefore we cancel the gradient with respect to C which is given by

$$G = \left(\left(\Phi^{\#}_{(n)}\Phi_{(n)}\right)(C^{\mathrm{T}}X) - \Phi^{\#}_{(n)}\mathbf{y}\right)X^{\mathrm{T}} + \lambda C^{\mathrm{T}}.$$

As we have seen in the proof of Proposition 5.13, $\Phi^{\#}_{(n)}\Phi_{(n)} = (\Phi^{\#}\Phi)_{(n)}$ which action is the same as that of $\Phi^{\#}\Phi$. Therefore we obtain the following matrix equation

$$(\Phi^{\#}\Phi)C^{\mathrm{T}}XX^{\mathrm{T}} + \lambda n C^{\mathrm{T}} = RX^{\mathrm{T}}, \tag{8.20}$$

where $R \in \mathbb{R}^{d\times n}$ is defined as in Remark 5.15: it contains the pairwise scalar product between the output observations and the elements of the dictionary.

Remark 8.7. *Unsurprisingly Equation (8.20) is very similar to its kernelized counterpart Equation (5.21) with XX^{T} playing the role of the intput kernel matrix $K_{\mathcal{X}}$ and the right side of the equation being RX^{T} instead of R. Those changes are however key since $XX^{\mathrm{T}} \in \mathbb{R}^{q\times q}$ which enable us to avoid the $\mathcal{O}(n^3)$ time cost and pay a $\mathcal{O}(q^3)$ cost instead.*

Equation (8.20) is a discrete time Sylvester equation. We propose however a solution based on an eigendecomposition similar to what we used in Example 8.5. Let us recall that $(\mathbf{w}_l, \upsilon_l)_{l=1}^{d}$ is an eigendecomposition of the matrix $\Phi^{\#}\Phi$, with $W = [\mathbf{w}_l]_{l=1}^{d} \in \mathbb{R}^{d\times d}$ and Υ is the diagonal matrix containing the eigenvalues in decreasing order. We make the change of variable $C^{\mathrm{T}} = WA$ with $A \in \mathbb{R}^{d\times q}$, then multiplying Equation (8.20) left by W^{T}, we obtain

$$\Upsilon AXX^{\mathrm{T}} + \lambda n A = W^{\mathrm{T}}RX^{\mathrm{T}}.$$

8.2.1 Efficient resolution in closed-form

We then notice that the above system is separable and can be solved separately for the different rows of A. Denoting by $\mathbf{a}_{l.}$ its l-th row and setting $B = W^{\mathrm{T}}RX^{\mathrm{T}}$ with rows $(\mathbf{b}_{l.})_{l=1}^{d}$, the above system is equivalent to

$$\begin{pmatrix} \upsilon_1\mathbf{a}_{1.} \\ \vdots \\ \upsilon_d\mathbf{a}_{d.} \end{pmatrix} XX^{\mathrm{T}} + \begin{pmatrix} \lambda n\mathbf{a}_{1.} \\ \vdots \\ \lambda n\mathbf{a}_{d.} \end{pmatrix} = \begin{pmatrix} \mathbf{b}_{1.} \\ \vdots \\ \mathbf{b}_{d.} \end{pmatrix}.$$

We can then drop the rows associated to null eigenvalues of $\Phi^{\#}\Phi$ as they yield equations which are not informative; this is another way to motivate the effective-rank approach that we introduced in Section 8.1. Therefore keeping only the r strictly positive eigenvalues, we have the system

$$\begin{pmatrix} \mathbf{a}_{1.}(XX^{\mathrm{T}} + \frac{\lambda n}{\upsilon_1}\mathrm{I}) \\ \vdots \\ \mathbf{a}_{r.}(XX^{\mathrm{T}} + \frac{\lambda n}{\upsilon_r}\mathrm{I}) \end{pmatrix} = \begin{pmatrix} \frac{\mathbf{b}_{1.}}{\upsilon_1} \\ \vdots \\ \frac{\mathbf{b}_{r.}}{\upsilon_r} \end{pmatrix}.$$

To solve this system, it suffices to invert the matrices $\left\{XX^{\mathrm{T}} + \frac{\lambda n}{\upsilon_r}\mathrm{I}\right\}_{l=1}^{r}$. Since they vary only by a scalar factor on the added identity matrix, this can be done in $\mathcal{O}(q^3)$ time by computing an eigendecomposition of XX^{T}. Let $\check{A}$ be the matrix regrouping the

obtained rows corresponding to nonzero eigenvalues, we can then recover C^{T} as $\check{\mathrm{W}}\check{\mathrm{A}}$. We refer to this approach as **efficient-rank FPL**.

Example 8.8 (Regularization in $\mathrm{Trace}(\mathrm{C}^{\mathrm{T}}\Phi^{\#}\Phi\mathrm{C})$)**.** *Problem 8.19 can however be regularized taking into account the projection operator Φ using the Hilbert-Schmidt norm of the operator $\Phi\mathrm{C} : \mathbb{R}^q \mapsto \mathcal{Y}$ given by* $\mathrm{Trace}(\mathrm{C}^{\mathrm{T}}\Phi^{\#}\Phi\mathrm{C})$.

$$\min_{\mathrm{C}\in\mathbb{R}^{q\times d}} \frac{1}{n}\|\Phi_{(n)}(\mathrm{C}^{\mathrm{T}}\mathrm{X}) - \mathbf{y}\|^2_{\mathcal{Y}^n} + \lambda\,\mathrm{Trace}(\mathrm{C}^{\mathrm{T}}\Phi^{\#}\Phi\mathrm{C}), \tag{8.21}$$

In that case, canceling the gradient as in Equation (8.20) we obtain

$$(\Phi^{\#}\Phi)\mathrm{C}^{\mathrm{T}}\mathrm{X}\mathrm{X}^{\mathrm{T}} + \lambda n\Phi^{\#}\Phi\mathrm{C}^{\mathrm{T}} = \mathrm{R}\mathrm{X}^{\mathrm{T}}, \tag{8.22}$$

which using the same change of variable $\mathrm{C}^{\mathrm{T}} = \mathrm{W}\mathrm{A}$ yields the equivalent equation

$$\Upsilon\mathrm{A}\mathrm{X}\mathrm{X}^{\mathrm{T}} + \lambda n\Upsilon\mathrm{A} = \mathrm{W}^{\mathrm{T}}\mathrm{R}\mathrm{X}^{\mathrm{T}}.$$

Therefore, we can drop in A *the rows corresponding to zero eigenvalues and obtain the solution*

$$\check{\mathrm{A}} = \check{\Upsilon}^{-1}\check{\mathrm{W}}^{\mathrm{T}}\mathrm{R}\mathrm{X}^{\mathrm{T}}(\mathrm{X}\mathrm{X}^{\mathrm{T}} + \lambda n\mathrm{I})^{-1},$$

and recover C^{T} as $\check{\mathrm{W}}\check{\mathrm{A}}$.

Remark 8.9 (Other losses)**.** *Problem 8.23 can be solved using any convex optimization iterative algorithm. If the loss L is a $\|\cdot\|_{\mathcal{Y}}$-composed loss as defined in Definition 8.1 and the regularization function is $\lambda\|\cdot\|^2_{\mathbb{R}^{q\times q}}$, exploiting Lemma 8.3, the same effective-rank trick can be used, and therefore we can solve*

$$\min_{\check{\mathrm{C}}\in\mathbb{R}^{r\times q}} \frac{1}{n}\sum_{i=1}^{n} \ell(\|y_i - \Phi(\check{\mathrm{W}}\check{\mathrm{C}}^{\mathrm{T}}\mathbf{x}_i)\|_{\mathcal{Y}}) + \|\check{\mathrm{C}}\|^2_{\mathbb{R}^{r\times q}}. \tag{8.23}$$

Remark 8.10 (FPL and Huber losses from Chapter 7)**.** *Provided the dictionary is properly chosen, the effective-rank trick should not be too necessary. Then, FPL can be used also with losses which are not $\|\cdot\|_{\mathcal{Y}}$-composed. In comparison with the dual approaches proposed in Chapter 7, FPL can be very advantageous. For instance, for the H^1_κ Huber loss, in Section 7.2.1 we optimize over $\boldsymbol{\alpha} \in \mathbb{R}^{n\times m}$, whereas in the FPL problem, we optimize over $\mathrm{C} \in \mathbb{R}^{q\times d}$ which is much more manageable. All the more so since FPL displays exactly the same robustness properties empirically as shown in Section 8.4.4.*

8.3 Dictionary selection

The effective-rank strategy is one possible way to deal with a redundant dictionary in projection learning. Another natural related question is, from a dictionary, can we extract automatically the atoms which are most relevant for the problem ? The linear version Problem 8.23 gives us direct access to the coefficients associated with each atom. Indeed, if the whole l-th column of the matrix C is null, the corresponding l-th atom ϕ_l will not intervene in the prediction function $\mathbf{x} \in \mathbb{R}^q \mapsto \Phi(\mathrm{C}^{\mathrm{T}}\mathbf{x})$. This idea is

interesting in itself, but it also enables us to reduce the dimension while using losses which are not of the $\|\cdot\|_{\mathcal{Y}}$-composed form (Definition 8.1).

8.3.1 Accelerated proximal gradient

This suggests the use of a structuring penalty to encourage atom selection. The idea of grouped variable selection has been proposed for the LASSO giving rise to the group LASSO (Cotter et al., 2005; Yuan and Lin, 2006) for which efficient algorithms exists–see for instance Fornasier and Rauhut (2008); Roth and Fischer (2008) and the thorough review of Rakotomamonjy (2011). A natural choice to enforce column-wise sparsity is then to penalize our objective by the composite $(1,2)$-norm on the columns of the coefficients $\mathrm{C} \in \mathbb{R}^{q\times d}$ (see *e.g.* Bach et al. 2012)

$$\|\mathrm{C}\|_{1,2} = \sum_{l=1}^{d} \|\mathbf{c}_l\|_{\mathbb{R}^q}.$$

We then study the problem

$$\min_{\mathrm{C}\in\mathbb{R}^{d\times q}} \frac{1}{n}\sum_{i=1}^{n} L_{y_i}(\Phi(\mathrm{C}^{\mathrm{T}}\mathbf{x}_i)) + \lambda\|\mathrm{C}\|_{1,2}, \tag{8.24}$$

where for $y \in \mathcal{Y}$, $L_{y_i}(y) = L(y - y_i)$. The penalty is convex and continuous, yet it is not differentiable, so we must use specific optimization strategies. We turn to accelerated proximal gradient descent (APGD) algorithms as in Chapter 7. The $\|\cdot\|_{1,2}$ is separable in the columns of the matrix, therefore its proximal operator boils down to applying the proximal operator of the $\|\cdot\|_{\mathbb{R}^q}$ norm separately to the columns of the matrix. The resulting operator is the well known block soft thresholding operator:

$$\mathrm{prox}_{\gamma\|\cdot\|_{1,2}}(\mathrm{C}) = \left[\left|1 - \frac{\gamma}{\|\mathbf{c}_l\|_{\mathbb{R}^q}}\right|_+ \mathbf{c}_l\right]_{l=1}^{d}.$$

However, if the dictionary is very redundant, we expect many atoms to be useless. Consequently, using APGD on its own may not be efficient. To exploit the expected sparsity, we recourse to *working sets* methods (see *e.g.* Nocedal and Wright 2006). At each step, optimization is performed on a reduced set of variables (the working set) and then we check if the corresponding partial solution is globally optimal. If it is not, new variables are added to the working set according to an heuristic to be determined, and so on until a global optimum is reached. In doing so, we avoid optimizing over the full set of variables. Examples of application in machine learning include multiple kernel learning (Bach, 2008), structured variable selection (Obizinski et al., 2010; Jenatton et al., 2011), sparse coding (Lee et al., 2007) or more recently sparse coding with nonconvex regularizers (Rakotomamonjy et al., 2022).

We propose to tackle Problem 8.24 in this way, however, we still need an inner solver, and for that we do use an APGD algorithm. Suppose that the loss is differentiable with gradient $\nabla L_{y_i} : \mathcal{Y} \to \mathcal{Y}$, then the gradient of the differentiable term in Problem 8.24 is

$$\frac{1}{n}\mathrm{X}\mathrm{G}(\mathrm{C})^{\mathrm{T}} \quad \text{with} \quad \mathrm{G}(\mathrm{C}) := \left[\Phi^{\#}\nabla L_{y_i}(\Phi(\mathrm{C}^{\mathrm{T}}\mathbf{x}_i))\right]_{i=1}^{n}. \tag{8.25}$$

Algorithm 8.1 APGD for linear projection learning with dictionary selection

input : Features matrix X, Projection operator Φ, Initial coefficients $\mathrm{C}^{(0)}$, regularization parameter λ, differentiable loss L, gradient step γ

init : $\mathrm{C}^{(-1)} = \mathrm{C}^{(0)}$

for *epoch* t *from* 0 *to* $T-1$ **do**

// gradient step

$\mathrm{A} = \mathrm{C}^{(t)} + \frac{t-2}{t+1}\left(\mathrm{C}^{(t)} - \mathrm{C}^{(t-1)}\right)$

$\mathrm{G}(\mathrm{A}) = \left[\Phi^{\#}\nabla L_{y_i}(\Phi(\mathrm{A}^{\mathrm{T}}\mathbf{x}_i))\right]_{i=1}^{n}$

$\mathrm{B} = \mathrm{A} - \gamma\left(\frac{1}{n}\mathrm{X}\mathrm{G}(\mathrm{A})^{\mathrm{T}}\right)$

// proximal step

for *column* $l \in [\![d]\!]$ **do**

$\mathbf{c}_l^{(t+1)} = \left|1 - \frac{\gamma\lambda}{\|\mathbf{b}_l\|_{\mathbb{R}^q}}\right|_+ \mathbf{b}_l$

return $\mathrm{C}^{(T)}$

Remark 8.11 (Partial observations)**.** *We present the algorithms using directly the vectors in $\mathcal{Y}$. However, provided enough observations are available, the quantities of interest can be estimated.*

Remark 8.12 (Integral loss and gradient estimation)**.** *If L is an integral loss with ground loss ℓ as in Example 5.20, for partially observed functions $(\boldsymbol{\theta}_i, \tilde{\mathbf{y}}_i)_{i=1}^n$ $\mathrm{G}(\mathrm{C})$ can be estimated as*

$$\frac{1}{m_i}\sum_{s=1}^{m_i} \ell\Big(y_i(\theta_{is}) - \boldsymbol{\phi}(\theta_{is})^{\mathrm{T}}\mathrm{C}^{\mathrm{T}}\mathbf{x}_i\Big)\boldsymbol{\phi}(\theta_{is}). \tag{8.26}$$

8.3.2 Working set algorithm

The APGD algorithm to solve Problem 8.24 is given in Algorithm 8.1. To use it within a working set framework, it must be run only on the relevant indices from the working set J. In practice we retain only the columns of C in J and use the projection operator for the dictionary containing the atoms which index is in J.

Our problem has the general form

$$\min_{\mathrm{C}\in\mathbb{R}^{q\times d}} f(\mathrm{C}) + \lambda\|\mathrm{C}\|, \tag{8.27}$$

where f is a differentiable and convex function and $\|\cdot\|$ is a norm on $\mathbb{R}^{q\times d}$.

To set up a working set algorithm however, we need a rule to check the global optimality of a solution computed on the working set. If global optimality is not reached, we also need a rule to choose the variables to add to the working set. We follow Bach et al. (2012, Chapter 6) and approximately monitor the duality gap at the current iterate C by checking whether

$$\|\nabla f(\mathrm{C})\|^* \leq \lambda, \tag{8.28}$$

where $\|\cdot\|^*$ is the dual norm of $\|\cdot\|$ (see Definition 2.45). In our case, $\|\cdot\| = \|\cdot\|_{1,2}$ and its dual norm is $\|\cdot\|_{\infty,2}$, therefore Equation (8.28) boils down to

$$\max_{l\in[\![d]\!]} \|(\nabla f(\mathrm{C}))_l\|_{\mathbb{R}^q} \leq \lambda. \tag{8.29}$$

Algorithm 8.2 Working set algorithm to solve Problem 8.24

input : Features matrix X, regularization parameter λ, differentiable loss L, gradient step γ

init : Initial working set $J = \{l_0\}$, random or $l_0 = \arg\max_{l \in [\![d]\!]} \frac{1}{n} \sum_{i=1}^{n} |\langle y_i, \phi_l \rangle_{\mathcal{Y}}|$.

$J^{(-1)} = J$, $r = \emptyset$

Initialize $\mathrm{C}_{J^{(-1)}}$

stop = False

while ***not*** *stop* **do**

  $\mathrm{C}_J^{(0)} = \left[\mathrm{C}_{J^{(-1)}}, \mathbf{0}_{\{r\}}\right]$ **or** for square loss, closed-form solution to Equation (8.20) restricted to J

  $\mathrm{C}_J = \text{Algorithm 8.1}(\mathrm{X}, \Phi_J, \mathrm{C}_J^{(0)}, \lambda, L, \gamma)$

  $\mathrm{C} = \left[\mathrm{C}_J, \mathbf{0}_{J^c}\right]$

  if $\|\nabla f(\mathrm{C})\|_{\infty,2} > \lambda$ **then**

    $r = \arg\max_{l \in J^c} \|(\nabla f(\mathrm{C}))_l\|_{\mathbb{R}^q}$

    $J^{(-1)} = J$

    $J = J \cup \{r\}$

  else

    stop = True

return C

Consequently, to increase the working set, it makes sense to add the atom with index l_0 for which $\|(\nabla f(\mathrm{C}))_{l_0}\|_{\mathbb{R}^q}$ is maximal. We detail the whole procedure in Algorithm 8.2. To simplify the presentation we do the following slight abuses of notation. For $J \subset [\![d]\!]$ and $\mathrm{C} \in \mathbb{R}^{q \times d}$:

- $\mathrm{C}_J \in \mathbb{R}^{q \times |J|}$ denotes the matrix C restricted to its columns with index is J, and similarly C_{J^c} denotes the same but for indices in the set complementary to J in $[\![d]\!]$.
- $\left[\mathrm{C}_J, \mathrm{C}_{J^c}\right]$ denotes the matrix which for $j \in J$, has the corresponding column of C_J and for $j \in J^c$, the corresponding one in J^c.

Remark 8.13 (Practical notes for the square loss)**.** *In Algorithm 8.2, we found that when using the square loss, instead of using the previous iterate as a warm starting point for Algorithm 8.1, it was much faster to use the closed-form solution–Equation (8.20)–to the problem with square loss and $\lambda_2 \|\cdot\|^2_{\mathbb{R}^{q \times d}}$ regularization with small λ_2. In our experiments, this resulted in a much lower number of inner iterations per atom addition, for significant $\lambda \|\cdot\|_{1,2}$ regularization, less than ten iterations were sufficient.*

8.4 Numerical experiments

In this section, we study in practice some of the proposed improvements. In all experiments the feature-based approach (Problem 8.23) is used, the features being Nyström ones (see Example 2.20).

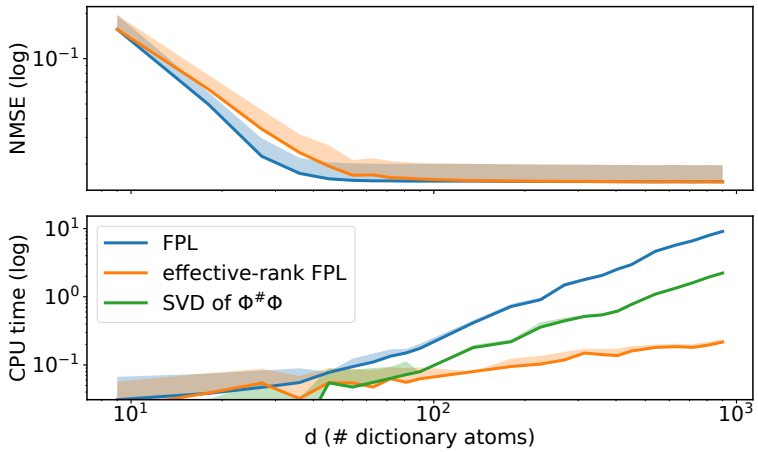


Figure 8.1: Effective rank FPL on synthetic data

8.4.1 Effective rank

In this section, we study in details the behavior of the effective-rank FPL estimator introduced in Section 8.2.1. We want to show it can be advantageous in term of compromise between accuracy and computational time.

Experimental setting. To that end we use the synthetic dataset based on Gaussian processes (GP) that we introduced in Section 7.4.1. As a brief reminder the output functions are random linear combinations of a set of four GPs drawn with the following standard deviations $\boldsymbol{\sigma}^{\text{out}} = (0.05, 0.1, 0.5, 0.7)$. To measure the efficiency of our estimators as the number of atoms in $\boldsymbol{\phi}$ (and its redundancy) increases, we use a dictionary of GP drawn with standard deviations $\boldsymbol{\sigma}^{\text{big}} = (0.001, 0.005, 0.01, 0.025, 0.05, 0.075, 0.1, 0.5, 0.7)$. These include the standard deviations from $\boldsymbol{\sigma}^{\text{out}}$, consequently as the number of drawn GPs increases, the resulting dictionary should work well for the problem. However, we also included some very low values. The corresponding realizations vary much too fast and therefore inflate the dictionary unnecessarily with noise atoms. Then, we solve the problem increasing the number of drawn processes: we draw l per standard deviation for l in $\{ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 15, 20, 25, 30, 35, 40, 45, 50, 60, 70, 80, 90, 100 \}$.

Optimization and tuning. On the one hand, for FPL we use the whole dictionary and solve Equation (8.20) with a Sylvester solver. On the other hand, for effective-rank FPL we employ the resolution strategy presented in Section 8.2.1 retaining only the eigenvalues/eigenvectors pairs of $\Phi^{\#}\Phi$ associated to eigenvalues above a given threshold. We set it in the experiments to $5 \times 10^{-5} \times d$ with d the number of atoms in the dictionary. We make it depend on d as the dimension structurally increases the magnitude of the largest eigenvalues. Finally, we select the regularization parameter λ through cross-validation for values in a geometric space with 50 values ranging from 10^{-8} to 10^{-2}. We repeat the whole experiment 10 times with different seeds for the different sources of randomness (draws of the GPs in the dictionary, generation of the dataset, draws of the Nyström features).

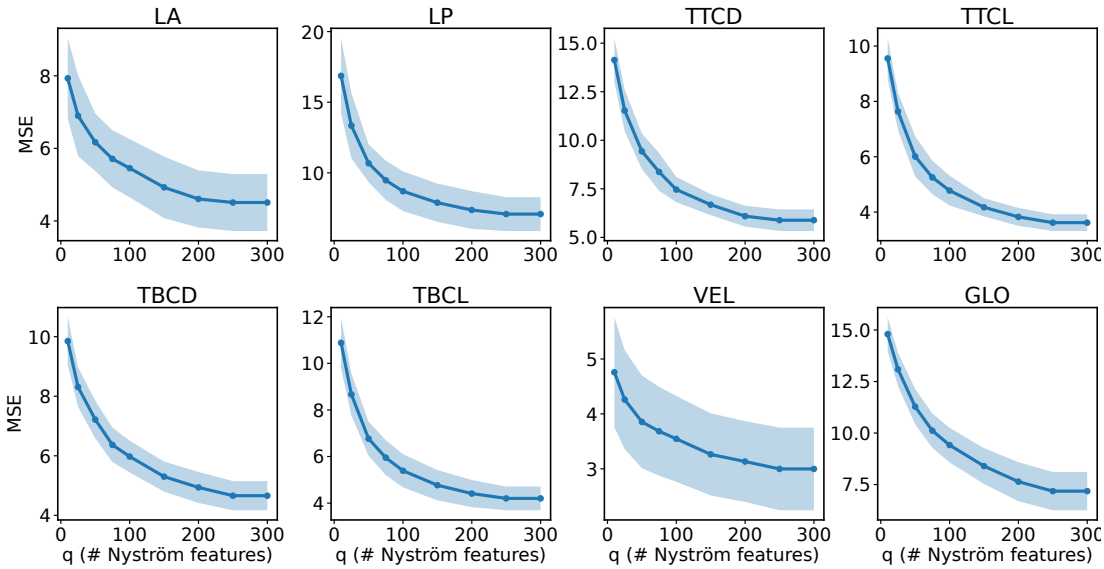


Figure 8.2: FPL with Nyström features on speech data

Results. We report in Figure 8.1 the average values as well as their standard deviation (light color bands above the curves). The performance of effective-rank FPL and FPL are very similar in terms of NMSE, and they could probably be the same if we worked a bit more on the selection of the threshold. As expected, when the number of atoms (and the redundancy of the dictionary) increases, the computational time for fitting FPL increases drastically faster than that for fitting effective-rank FPL (the CPU time is in log). Admittedly, to fit the latter we do need to compute eigenvalues/eigenvectors pairs of $\Phi^{\#}\Phi$, however the CPU time for doing so increases much slower than that to fit FPL. Moreover we only need to do this once per set of experiment, which is a great advantage for instance to tune other parameters.

8.4.2 Large scale

Experimental setting. We now check the efficiency of FPL using Nyström features on the speech dataset that we introduced in Section 5.3.4. We display the results for all eight vocal tracts increasing the number of Nyström features in Figure 8.2. We use the same learnt dictionary as in Section 5.3.4 fixing the number of atoms at 30. We select the regularization parameter through cross-validation considering λ in a geometric space of size 20 ranging from 10^{-10} to 10^{-5}. We display the mean values and standard deviation computed over 10 runs of the experiments with different seeds.

Results. As expected, using Nyström features we can reach a satisfying score not using all of the training observations (using about 2/3). We then get quite close to the score reached using the maximum number of Nyström features–equivalent to running KPL. We however note that for some vocal tracts (TTCL, TBCD and GLO), the convergence is a bit slower than what we are used to for a Nyström approximation. This can be explained by the nature of the data: observations corresponds to words and we do not have that many to start with. Therefore using only a reduced number of observations may make the problem challenging for certain vocal tracts.

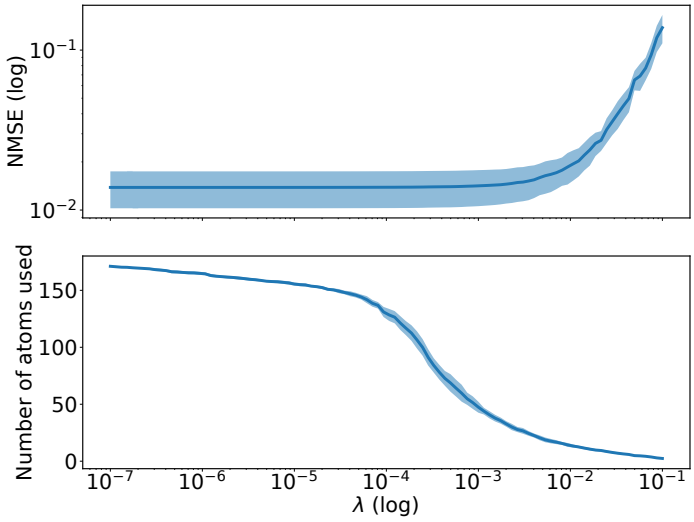


Figure 8.3: Dictionary selection using Algorithm 8.2 on synthetic data

8.4.3 Dictionary selection

We now wish to demonstrate how the $\|\cdot\|_{1,2}$ promotes selection of atoms from the dictionary. To that end we use the same setting as in Section 8.4.1: we work on the synthetic data and draw an highly redundant dictionary using 40 GPs per standard deviation in $\boldsymbol{\sigma}^{\text{big}}$. This results in a dictionary of size 360. We use the working set Algorithm 8.2 to solve Problem 8.24 with the square loss and we report the means and standard deviations over 10 runs of the experiments in Figure 8.3. It shows how the intensity of the mixed-norm regularization promotes the selection of the more relevant atoms for the problem. Indeed, we see that as λ increases, less and less atoms are used but the estimator remains accurate: the size of the dictionary can get down to 50 atoms without really degrading the NMSE.

8.4.4 FPL with robust losses

Experimental setting. As a complement we show how FPL can be combined with any differentiable loss. We consider the issue of robustness and use the Huber losses H^2_κ and H^1_κ introduced in Section 7.2. They are both differentiable once. We use the exact same experimental setting as in Section 7.4.3: on the synthetic data, we add global or local outliers and we make several of their parameters vary. These parameters are the proportion $\tau \in [0,1]$ of the training data being contaminated, the magnitude ζ of the global outliers and the proportion $\xi \in [0,1]$ of contaminated observations per function for the local outliers. For more details on the outliers, we refer to Section 7.4.1. We use a generic output dictionary: a Fourier basis including 40 frequencies. For optimization, we solve Problem 8.23 using an accelerated gradient descent combined with backtracking line search.

Tuning. We select through robust cross-validation the regularization parameter λ and the parameter κ of the losses. For the former, we consider values in a geometric grid of size 7 ranging from 10^{-7} to 10^{-5}. For the latter, we consider the same values as in Section 7.4.3 which are detailed in Appendix B.4.

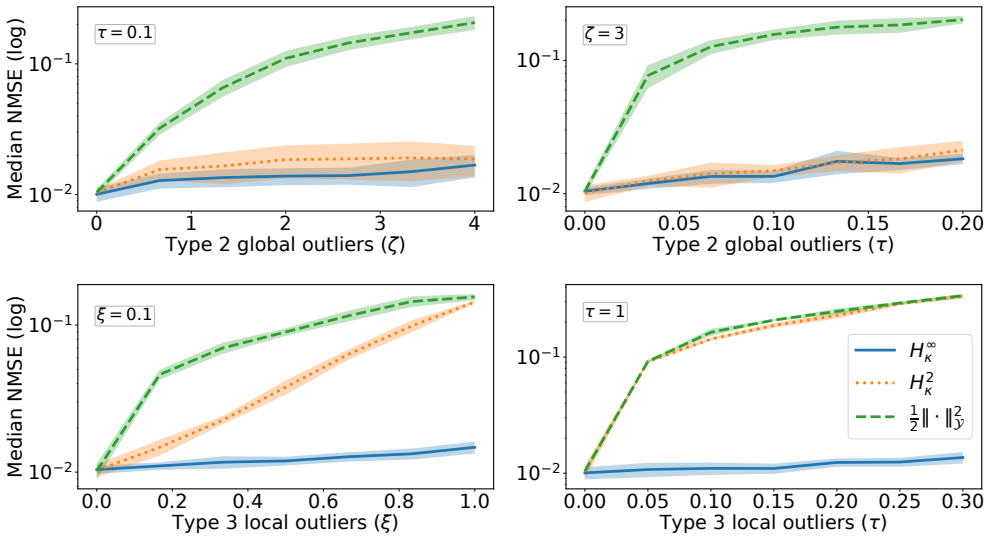


Figure 8.4: FPL with Huber losses and square loss

Results. We display in Figure 8.4 the median of the NMSEs obtained over 10 runs of the experiments as well as their median absolute deviation (MAD) which corresponds to the average absolute deviation to the median. We use these metrics because dealing with outliers, the results are naturally more scattered, and those quantities are clearer to read. As expected FPL can be made robust both to local and global outliers using those losses. The comments are the same as in Section 7.4.3: when the outliers are global, FPL with both Huber losses H_κ^2 and H_κ^1 is much more robust than FPL with the square loss. However, for local outliers, if all the observations are corrupted as in the bottom right panel ($\tau = 1$), the loss H_κ^2 is no more robust than the square one. It is because it filters the outliers in terms of their $\|\cdot\|_y$ norm. We also see this effect as τ increases with fixed ξ in the bottom left panel.

As highlighted in Remark 8.10, using the dual approach proposed in Section 7.2.1 we must optimize over coefficients $\boldsymbol{\alpha} \in \mathbb{R}^{n \times m}$ whereas here we optimize over $\mathrm{C} \in \mathbb{R}^{q \times d}$ which is much more manageable.

8.5 Conclusion

In this chapter we addressed several questions around the KPL framework. To deal with dictionaries which are too redundant, we proposed first to exploit an eigen-decomposition of the Gram matrix of the dictionary. We formulated a KPL problem in which we predict coordinates in the set of the most relevant eigenvectors. Doing so, we reduce both the dimension and the computational burden without making compromises on accuracy. Then, to improve the computational complexity further, we combined KPL with large scale kernel techniques. To that end, we formulated a linear projection learning problem and derived efficient ways of solving it, notably in closed-form when the square loss is used. We demonstrated empirically the efficiency of this estimator using Nyström features. For the same feature-based model, so as to achieve automatic selection of the relevant atoms, we regularized through the (1, 2)-mixed norm which encourages column-wise sparsity. In turn, this implies sparsity on the dictionary. We tackled this problem using an APGD algorithm and proposed to use

it within a working set framework to exploit the expected sparsity. To finish with, we highlight that the feature-based projection learning framework can be used with any differentiable loss, a possibility we demonstrate experimentally by combining it with functional Huber losses. This constitutes a very scalable functional output regression method: it uses features, so its complexity with respect to the number of inputs is limited while its complexity incurred by the functional outputs is limited through the use of a dictionary.

Conclusion and Perspectives

Method	Representation
FKRR-MC (Lian, 2007; Kadri et al., 2010)	Discrete
FKRR-EIG (Kadri et al., 2016)	T_{k_Θ} eigenfunctions
3BE (Oliva et al., 2015)	Orthogonal basis
KAM (Reimherr et al., 2018)	FPCA
FKR-HUB (Laforgue et al., 2020)	T_{k_Θ} eigenfunctions
KPL (Chapter 5)	Any dictionary
FKR-CONV-SP (Chapter 7)	Linear splines
FKR-CONV-EIG (Chapter 7)	T_{k_Θ} eigenfunctions
FPL (Chapter 8)	Any dictionary

Method	Inputs	loss	Fit complexity
FKRR-MC	Any	$1/2\|\cdot\|_{\mathcal{Y}}^2$	$\mathcal{O}(n^3 + m^3)$
FKRR-EIG	Any	$1/2\|\cdot\|_{\mathcal{Y}}^2$	$\mathcal{O}(n^3 + n^2md)$
3BE	Functions	$1/2\|\cdot\|_{\mathcal{Y}}^2$	$\mathcal{O}(q^3 + q^2d)$
KAM	Functions	$1/2\|\cdot\|_{\mathcal{Y}}^2$	$\mathcal{O}(n^2t^2 + d^2m^2 + n^3 + d^3)$
FKR-HUB	Any	H_κ^2	Strongly convex problem ($\mathbb{R}^{d\times n}$)
KPL	Any	$1/2\|\cdot\|_{\mathcal{Y}}^2$	$\mathcal{O}(d^3 + n^3)$
FKR-CONV-SP	Any	H_κ^1, H_κ^2, ℓ_ϵ^∞, ℓ_ϵ^2	Strongly convex problem ($\mathbb{R}^{m\times n}$)
FKR-CONV-EIG	Any	H_κ^2, ℓ_ϵ^2	Strongly convex problem ($\mathbb{R}^{d\times n}$)
FPL	Any	$1/2\|\cdot\|_{\mathcal{Y}}^2$	Truncated SVD of $\Phi^\#\Phi + \mathcal{O}(q^3)$
FPL	Any	Differentiable	Strongly convex problem ($\mathbb{R}^{d\times q}$)

Table: main characteristics of the FOR methods proposed in this thesis compared to these of existing ones.

In this thesis, we proposed new tools to solve functional output regression (FOR) problems nonlinearly. FOR is particularly challenging because the outputs lie in a functional Hilbert space which is generally infinite-dimensional. Therefore, a relevant approximation in finite dimension must be found. Usually, either the functional problem is solved in closed-form and then the solution is approximated, or the functions are smoothed upstream and then used to solve the problem. In terms of losses, the functional square loss is mainly used even though it can be sensitive to outliers.

Our main contribution is the framework of *projection learning*. It taps into the many possibilities offered to represent functions using dictionaries. However, it does so directly in the functional empirical risk minimization problem so that the representation is factored in producing relevant predictions. This technique can drastically reduce the computational complexity associated with the number of observations per output function.

We studied in detail *kernel projection learning (KPL)* where a vector-valued reproducing kernel Hilbert space (vv-RKHS) is taken as a hypothesis class for the representation coefficients. When the functional square loss is used, we derived two estimators in closed-form, one deals with fully-observed functions while the other is computed directly from the available discrete observations. We backed these estimators theoretically with excess risk bounds showing their consistency. On the practical side, we proposed an efficient procedure to compute these estimators for separable operator-valued kernels. To reduce the time complexity linked to the number of samples, we proposed *feature projection learning*, a linear version of projection learning that can be combined with large scale kernel features such as random Fourier features or Nyström features. The resulting further reduction in the time complexity makes it more manageable to use other functional losses. Finally, we introduced an efficient-rank estimator to deal with redundant dictionaries as well as an algorithm exploiting a structured-norm penalization to automatically select relevant atoms from the dictionary.

However, several possibilities around projection learning are still to be investigated. We have shown how to select a set of atoms from the dictionary common to all observations. Nevertheless, it would be interesting to select a set of atoms specific to each observation, or in other words to reach *input-dependent* sparsity. Theoretically, the excess risk bounds we derived could be improved to obtain better rates under more restrictive assumptions. We also analyzed the estimation procedure but not the approximation aspects linked to the use of a dictionary to represent the output functions. In terms of hypothesis classes, we studied in depth projection learning with vv-RKHSs. The idea could be extended to other classes of models. For instance, it would be simple to add a last dictionary layer to a neural network. Or projection learning could be combined with regression trees, since provided the square loss is used, the resulting splitting criterion can be computed in closed-form. From there, we could construct random-forests for Hilbert-valued regression. We also limited our scope to FOR problems, however projection learning can work for prediction in any separable Hilbert space. It could for instance be used to solve structured output prediction problems when the outputs are represented in a RKHS.

As a second contribution, we extended the possible losses for FOR focusing on regression in function-valued RKHSs. We introduced a family defined through infimal convolution to generalize the Huber and ϵ-insensitive losses for functions. These exploit the properties of functional p-norms to encourage robustness or sparsity with the parameter p determining the degree of locality of the property. The resulting empirical risk minimization problems are especially amenable to the use of Lagrange duality. We then proposed two possible finite-dimensional representations for the dual variables and solved the problem for certain values of the locality parameter p. We thus obtained estimators which are sparse or robust either in a local or global way.

Further investigations could focus on extending the possible values of p for which the problem is solvable in practice. We enforced sparsity or robustness both in a fully local and global way but did not find a reasonable solution to reach something in between. The bottleneck resides in the impossibility to project fast enough on the q-norm unit ball for $q \notin \{2, +\infty\}$. Therefore finding an efficient algorithm to perform such a projection could make this problem solvable. Lastly, it would be interesting to investigate other types of losses for functional outputs. For instance, an analogous to quantile regression for functions would be of special interest.

Part III

Applied contribution on wind energy

9

Prediction of wind power in the very short-term

Contents

The present chapter is dedicated to an applied contribution. More precisely, it studies several aspects of the prediction of wind speed and wind power in the very short-term using machine learning. By very short-term, we mean that the predictions range from nowcasts (almost immediate forecasts) to four hour forecasts. The initial motivation was to work on the theme of renewable energies and apply our methods for functional output regression. Consequently, we started working on a collaboration around a dataset regrouping several measurements from the sensors in the turbines of a company's (Zéphyr ENR) wind-farms at a high temporal resolution. The initial aim was to predict whole functions corresponding to predictions over a time interval. Nevertheless, for this particular applications, considering the problem as functional did not yield improvements over correctly tuned well-known machine learning methods. Consequently, we decided to leverage the work that was already done on using machine learning to predict wind speed and wind power in the short-term, and investigate several aspects that appeared of interest. We submitted this work to a specialized journal. Therefore, to maintain the coherence of the thesis, we add this contribution as an independent part, and we expose it as we submitted it and without making links to the rest of the thesis. It corresponds to the contributions of

- **D. Bouche**, R. Flamary, F. d'Alché-Buc, R. Plougonven, M. Clausel, J. Badosa and P. Drobinski. Wind power predictions from nowcasts to 4-hour forecasts: a learning approach with variable selection *Technical report*, 2022. (https://arxiv.org/abs/2204.09362). (submitted)

9.1 Introduction

The fast development of renewable energies is a necessity to mitigate climate changes (Masson-Delmotte et al., 2021). Wind energy has developed rapidly over the past three decades, with an average annual growth rate of 23.6% between 1990 and 2016 (IEA, 2018), and is now considered as a mature technology. The share of renewable energies in global electricity generation reached 29% in 2020, and is expected to keep growing fast in coming years (IEA, 2021) which raises a number of challenges, stemming from the variability and spatial distribution of the resource. Then, in order to facilitate the dynamic management of electricity networks, forecasts of wind energy require continual improvement. Short timescales, from a few minutes to a few hours, are of particular importance for operations.

To produce forecasts, one can rely on several distinct sources of information. On timescales of half a day to about a week, deterministic weather forecasts provide a representation on a grid of the atmospheric state, including wind speed near the surface. The skill of such numerical weather forecasts (NWP) models has continually increased over the past decades (Bauer et al., 2015), while their spatial resolution has also grown finer (down to few km). However, to predict at a given geographical location for time horizons shorter than a day, the use NWP models is impeded by two main difficulties being **(i)** the errors in the modeled wind and **(ii)** the relatively infrequent initiation of forecasts. The former result from both limited resolution and the impossibility to model local processes. For instance, for wind speed at an altitude of 100m is strongly affected by local small-scale features and turbulent motions, both of which remain beyond the spatial resolution that is achievable for NWP models. Regarding the second point, operational centers typically launch forecasts twice or four times per day, however the computation of the forecast itself as well as the preparation of its initial state require time and computational resources–see e.g. Kalnay (2003). As a result, many methodologies for forecasting short-term wind speed or wind power use only past local observations and focus on statistical methods–see e.g. the reviews from Tascikaraoglu and Uzunoglu (2014); Okumus and Dinler (2016); Liu et al. (2019) and references therein. Nevertheless, we know that NWP models can provide valuable information for the evolution of the atmosphere on larger scales–i.e. on the formation or passage of a low-pressure system and on the associated fronts.

It is therefore a natural idea to use both sources of information to train machine learning (ML) models: local deficiencies in NWP models can partly be overcome by *downscaling*; i.e. better estimating local variables from the knowledge from a model's outputs and past observations. Such efforts have been carried out for decades in meteorology and climatology, under different names. In a pioneering early study, Glahn and Lowry (1972) applied multilinear regressions trained on past observations to correct NWP errors. More recently, *model output statistics* has become common practice in operational weather forecast centers–see e.g. Wilson and Vallée (2002); Baars and Mass (2005). In recent years, ML methods have enhanced the performance of these post-processing steps–see e.g. Zamo et al. (2016); Goutham et al. (2021)).

Specifically for wind speed or wind power forecasting, several *hybrid models* combining successfully NWP outputs with local observations have been proposed. In terms of time horizons, the focus is mostly on forecasts beyond 1 hour with low resolution (generally one prediction per hour)–see e.g. Hoolohan et al. (2018, Table 1) and references therein. While for the shorter term, most existing hybrid methods rely on complex and deep architectures–see e.g. Han et al. (2022, Table 1) and references therein. Moreover, for all these methods only a very low number of local and NWP variables are used (most of the times, only the past observed wind speeds and the ones predicted by the NWP model).

In this paper, we study hybrid prediction of both wind speed and wind power. Our contributions are five-fold.

- We study the problem for time horizons ranging from 10 minutes to 4 hours at a high resolution (every 10 minutes). This allows us to study with precision when and how the transition from one source of information (past local observations) to the other (NWP forecasts) occurs. This setting has been introduced in Dupré et al. (2020) yet we extend it and use it to address the following new problems.

- We include many different outputs from a NWP model as they could provide broader information on the overall predicted dynamics to the ML model. We also include several local variables. We then focus on variable selection and study the evolution of the importance of the selected variables through time. This allows us to better understand the nature of the studied statistical relationship and to extract a usable set of relevant variables.

- We study five distinct wind farms which enables us to expose many similarities but also some site specificities and increases the statistical significance of our results.

- We investigate which type of ML methods are the most suited for hybrid prediction of wind speed and wind power.

- Many existing contribution focus either on wind power or wind speed prediction but not on the relation between them, whereas at all steps of the paper, we compare the direct (predict wind power) and indirect (wind speed predictions passed through a power curve) approaches.

In terms of methodology, we have two key focuses.

- We want this study to be usable by practitioners. To that end we concentrate on a reduced choice of well-known and efficient ML methods which scale well with the number of samples, and moreover provide all the needed elements for a straightforward implementation. We also put a particular emphasis on how we select our models.

- We want to ensure our results are statistically significant. To that end, we employ a thorough evaluation process. We study several sites over several periods of time (the number of samples is quite high per site) and for each location, we average the results over several data splits.

Figure 9.1: Cartography of the studied farms, BM (A), BO (B), MP (C), RE (D), VE (E)

In Section 9.2, we introduce the data set from Zéphyr ENR and detail our processing of the data. Section 9.3 is dedicated to the presentation of our methodology as well as to the introduction of the statistical learning tools. Then in Section 9.4 investigates variable selection. Finally in Section 9.5, exploiting all the previous results, we compare different well-known ML models as well as direct and indirect prediction for wind power.

Notation We introduce the following notation: for two integers $n_0, n_1 \in \mathbb{N}^*$, the set of strictly positive integers, we denote by $[\![n_0]\!]$ the set $\{1, ..., n_0\}$ and by $[\![n_0, n_1]\!]$ the set $\{n_0, ..., n_1\}$.

9.2 Data and context

In this Section, we introduce the dataset that we use (Section 9.2.1) as well as the pre-processing steps that we apply to it and the general evaluation methodology (Section 9.2.2).

9.2.1 Zéphyr ENR's dataset

Our first source of information consists of measurements made by sensors in the wind turbines (we call these in situ variables) whereas the second one consists of forecasts from the European Centre for Medium-Range Weather Forecasts (ECMWF). We study five wind farms in the northern half of France: Parc de Bonneval (BO), Moulin de Pierre (MP), Parc de Beaumont (BM), Parc de la Renardière (RE), and Parc de la Vènerie (VE). These wind farms are operated by the private company Zéphyr ENR and are described in details in (Dupré et al., 2020). We display their location on a map in Figure 9.1. Some are geographically close by–we can form the pairs (BO, MP) and (BM, RE)–while VE is isolated. Note that we left another available farm out of the study because it displayed signs of sensors deficiencies. On the one hand, the geographical topology of the surroundings for (BO, MP) and (BM, RE) are quite similar, they correspond to open fields with very few elevation variations. On the other hand, VE is surrounded by wooded countryside with slightly more elevation variations, which

Variable type	Altitude or pressure level	Variable	Unit
Surface	10m/100m	Zonal wind speed	ms^{-1}
		Meridional wind speed	ms^{-1}
	2m	Temperature	K
		Dew point temperature	K
	Surface	Skin temperature	K
		Mean sea level pressure	Pa
		Surface pressure	Pa
		Surface latent heat flux	Jm^{-2}
		Surface sensible heat flux	Jm^{-2}
		Boundary layer dissipation	Jm^{-2}
		Boundary layer height	m
Altitude	1000/925/850/700/500	Zonal wind speed	ms^{-1}
		Meridional wind speed	ms^{-1}
		Geopotential height	m^2s^{-2}
		Divergence	s^{-1}
		Vorticity	s^{-1}
		Temperature	K
Computed	10m/100m	Norm of wind speed	ms^{-1}
	10m to 925 hPa	Wind shear	ms^{-1}
		Temperature gradient	K

Table 9.1: ECMWF variables

Availability	Variable	Unit
All	Wind speed	ms^{-1}
All	Power output	kW
All	Wind direction	Degree
BO and BM	Temperature	Celsius degree

Table 9.2: In situ variables

Variable (source)	Abbreviation
Wind speed (in situ)	WS
Power output (in situ)	PW
Norm of wind speed at 100m (ECMWF)	F10
Norm of wind speed at 100m (ECMWF)	F100
Wind shear between 10m and 925 hPa (ECMWF)	DF
Boundary layer dissipation (ECMWF)	bld
Boundary layer height (ECMWF)	blh
Surface latent heat flux (ECMWF)	slhf

Table 9.3: Abbreviations for the variables used in the paper

may explain the differences that we observe between this farm and the others in Section 9.4 and Section 9.5.

For BO and VE we have three years of data (from 2015 to 2017) which amounts to a total of $n = 157680$ observations for BO. However, for VE we do not use the year 2016 because it encompasses sensor deficiencies, so we use $n = 105120$ observations. For BM and RE we have access to two years of data (from 2017 to 2018 for BM and from 2015 to 2016 for RE) which results in a total of $n = 105120$ observations, and finally for MP we have only one year (2017), which gives us of total of $n = 52560$ observations.

Several variables are available, we summarize the in situ ones in Table 9.2–note the temperature is available only for BO and BM. In order to encode the circular nature of the in situ wind direction we encode it using two trigonometric variables.

The ECMWF provides global forecasts issued by their NWP models. We followed Dupré et al. (2020): we extracted the day ahead forecast twice a day (at 0000UTC and 1200UTC) and included the same 47 variables as they do. These variables are either selected or computed so as to describe as well as possible the boundary layer, the wind parameters and the temperature in the lower troposphere. Table 9.1 presents the ECMWF variables we use. They can be either surface variables, altitude ones, or computed from other ECMWF variables. The spatial resolution of ECMWF forecasts is about 16 km (0.125 ° in latitude and longitude) and their temporal resolution is 1h, then to match that of the in situ variables (10 min), we linearly interpolate the ECMWF forecasts. To finish with we sum up the abbreviations for the variables mostly used in the paper in Table 9.3.

9.2.2 Preprocessing and evaluation methodology

In order to increase the statistical significance of our results, we average the outcomes of the experiments over different data splits. A split consists of 3 subsets from the dataset, a train subset (of size n_{train}), a validation one (size n_{val}) and a test one (of size n_{test}). In order to avoid overfitting, given a ML method and a set of possible parameter values, we first train the resulting models on the train set. Then we choose the model yielding the best score on the validation set. To finish with, we re-train this model on the concatenation of the train and validation set and report its score on the test set. So as to preserve time coherence, we build our splits in a rolling fashion. For instance for the first split we take the period $[\![n_{\text{train}}]\!]$ for training, the period $[\![n_{\text{train}}+1, n_{\text{train}}+n_{\text{val}}]\!]$ for validation and we test the models on the period $[\![n_{\text{train}}+n_{\text{val}}+1, n_{\text{train}}+n_{\text{val}}+n_{\text{test}}]\!]$. Then for the second split, the train period is $[\![n_{\text{train}}+n_{\text{val}}+n_{\text{test}}+1, 2n_{\text{train}}+n_{\text{val}}+n_{\text{test}}]\!]$, the validation one is $[\![2n_{\text{train}}+n_{\text{val}}+n_{\text{test}}+1, 2n_{\text{train}}+2n_{\text{val}}+n_{\text{test}}]\!]$ and so on. For the sizes of the windows, we set $n_{\text{train}} = 10000$, $n_{\text{val}} = 10000$ and $n_{\text{test}} = 10000$ (however, the last split generally contains around $5000 \leq n_{\text{test}} \leq 10000$ observations). Since the length of available data vary from farm to farm, we do not have the same number of splits for all the farms.

We pre-process the data in the following way. As the number of wind turbines per farm is quite low (6 for BM, 6 for BO, 3 for HC, 6 for MP, 6 for RE and 4 for VE) , we average the in situ data over the turbines for each farm. In all our experiments, we standardize both the input and the output variables (subtract the mean and divide by the standard deviation) using the training data. We do so both for in situ variables and ECMWF ones. Such operation is crucial for instance to avoid favoring some variables which are structurally bigger over others when using regularized ML models.

9.3 Methodology and machine learning tools

In this Section, we introduce our general methodology as well as the ML tools that we use for variable selection and forecasting.

9.3.1 Methodology

Dataset building. Let $m \in \mathbb{N}$ be the prediction length (the number of future wind speed or wind power values we want to forecast). For the ECMWF variables, we include the corresponding forecasts. However, in practice we found that including a bit more than that improved performances. To that end, we denote respectively by

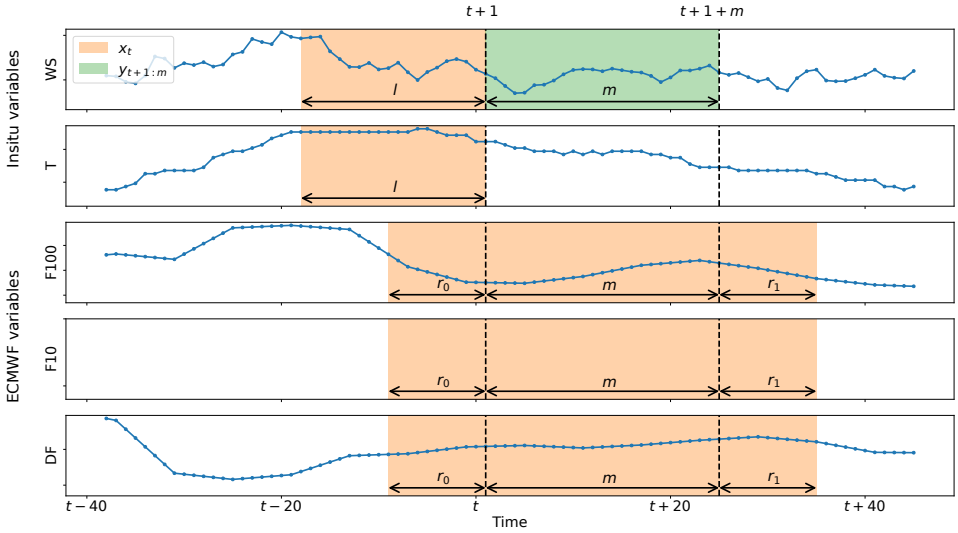


Figure 9.2: Summary of the time windows used for each source of data for wind speed prediction

$r_0 \in \mathbb{N}$ the number of past ECMWF predictions that we include, and by $r_1 \in \mathbb{N}$ the number of ECMWF predictions that we consider after m. For the in situ variables, we include a length $l \in \mathbb{N}$ of past observations. These different time windows are illustrated in Figure 9.2 for a reduced set of variables. For the all the variables and time windows, we concatenate the relevant observations $\mathbf{x}_t$ (these within the orange zones in Figure 9.2). From these, we want to produce a prediction for $\mathbf{y}_{t+1:m} \in \mathbb{R}^m$ (the m observations within the green zone in Figure 9.2). In practice, we use the following parameters which work well experimentally:

- we predict up to 4 hours, with a time sampling rate of 10 min, it means that $m = \frac{4\times60}{10} = 24$,
- for the in situ variables, we consider 3h of past observations, thus $l = \frac{3\times60}{10} = 18$,
- for the ECMWF variables we additionally use the predictions between 1.5h before and 1.5h after the time horizon of interest so $r_0 = r_1 = \frac{1.5\times60}{10}$.

ML models. We stick to ML models which are well-known and can scale well to a higher volume of data. Good results were obtained for one location (BO) from the studied dataset in Dupré et al. (2020) using linear regression with greedy forward stepwise variable selection (Hastie et al., 2001). Nevertheless, since we are interested in the importance of variables, we study also an alternative which select variables directly in the least square problem. The LASSO (Tibshirani, 1996) exploits the L1 penalty to induce sparsity in the regression coefficients, thus shrinking to zero the ones which are associated to the less relevant variables. Such methods can however be limited in that they can learn only linear dependencies between the input and output variables. Consequently, we study a nonlinear alternative: kernel ridge regression (KRR)–see for instance (Schölkopf and Smola, 2002; Shawe-Taylor and Cristianini, 2004). Finally, in order to include most families of ML models, we consider two other nonlinear methods: a tree-based boosting algorithm (Friedman, 2001)–we use XGBoost Chen and Guestrin (2016)–as well as a feed-forward neural network (NN). In

Section 9.3.2 we give more mathematical details on the general ML problem as well as on the methods that perform the best in the experimental section.

Variable selection. We have many in situ and ECMWF variables available (see Table 9.1 and Table 9.2). So as to improve the computational efficiency and and understand better what the models do, it is preferable to use only the most important variables. Ideally, we want to find a subset which is sufficient for a model to predict a statistically relevant target value from the input variables. In that sense a variable selection tool is necessarily model specific. Linear techniques will focus only on linear dependencies, whereas nonlinear ones will incorporate a much wider range of dependencies. Then we propose to use and interpret the results of one variable selection for each type. For the linear one we study the LASSO. For the nonlinear one, we use backward elimination using the Hilbert Schmidt Independence Criterion (Song et al., 2012). As opposed to the LASSO, it performs variable selection as an independent first step. The selected variables can then be used downstream to train any model. Then, for the nonlinear models (KRR, XG-Boost, feed-forward NN), we use the variables selected through this method. We give more detailed insights into the different techniques in Section 9.3.3.

9.3.2 Details on machine learning models

The input observations are the concatenation of the different variables on the time windows described in the previous section. We denote by $\mathcal{X} = \mathbb{R}^q$ the resulting input space, for some $q \in \mathbb{N}$. Our training data then consist of $((\mathbf{x}_t, \mathbf{y}_{t+1:m}))_{t=1}^n \in (\mathcal{X} \times \mathbb{R}^m)^n$ for some $n \in \mathbb{N}$, where we recall that $\mathbf{y}_{t+1:m} = (y_{t+1+m_0})_{m_0=1}^m$. Given a prediction function from a ML model class $f_{\mathbf{w}} : \mathcal{X} \to \mathcal{Y}$ parameterized by a vector $\mathbf{w} \in \mathcal{W}$, we want to minimize the average error on the training data:

$$\min_{\mathbf{w} \in \mathcal{W}} \frac{1}{n} \sum_{t=1}^{n} \|f_{\mathbf{w}}(\mathbf{x}_t) - \mathbf{y}_{t+1:m}\|_2^2 + \lambda \Omega(\mathbf{w}). \tag{9.1}$$

However, depending on the model, a penalty function $\Omega : \mathcal{W} \longrightarrow \mathbb{R}$ can be added in order to prevent overfitting or promote variable selection; its intensity is controlled by a parameter $\lambda > 0$.

In practice, instead of predicting all time horizons in one go as in Problem 9.1, we rather use separate models for each horizon in $[\![t+1, t+1+m]\!]$. That way we can tailor the different parameters for each horizon, which we found improved performances. Then in what follows, we consider a generic time horizon m and predict y_{t+1+m}.

Ordinary least squares (OLS). In forward stepwise variable selection (Hastie et al., 2001), at each step an OLS regression is solved for which the optimization problem reads:

$$\min_{\mathbf{w} \in \mathcal{W}, b \in \mathbb{R}} \frac{1}{n} \sum_{t=1}^{n} (\mathbf{w}^{\mathrm{T}} \mathbf{x}_t + b - y_{t+1+m})^2 \tag{9.2}$$

A well-known and simple closed form exist, which we use in practice.

LASSO. The optimization problem for the LASSO is the following:

$$\min_{\mathbf{w}\in\mathcal{W},b\in\mathbb{R}} \frac{1}{n}\sum_{t=1}^{n}(\mathbf{w}^{\mathrm{T}}\mathbf{x}_t + b - y_{t+1+m})^2 + \lambda\|\mathbf{w}\|_1,$$

where $\|\mathbf{w}\|_1$ is the sum of the absolute values of the coefficients $\mathbf{w}$. Many efficient algorithms exist to solve this convex yet non differentiable problem–see for instance Beck and Teboulle (2009). In practice we use the scikit-learn (Pedregosa et al., 2011) implementation with coordinate descent solver.

Kernel ridge regression (KRR). In KRR, we consider a class of models defined by a positive definite reproducing kernel $k : \mathcal{X}\times\mathcal{X} \longrightarrow \mathbb{R}$ which results in a unique associated reproducing kernel Hilbert space (RKHS). A most typical choice for k is the Gaussian kernel:

$$k_\gamma(\mathbf{x},\mathbf{x}') := \exp\Big(-\gamma(\|\mathbf{x}-\mathbf{x}'\|_2^2\Big).$$

We then seek our modeling function in this RKHS which we denote $\mathcal{H}_k$, each $h \in \mathcal{H}_k$ being a function from $\mathcal{X}$ to $\mathbb{R}$. For many kernels, this space constitutes a very rich class of modeling functions which can model nonlinear dependencies. The optimization problem reads:

$$\min_{h\in\mathcal{H}_k} \frac{1}{n}\sum_{t=1}^{n}(h(\mathbf{x}_t) - y_{t+1+m})^2 + \lambda\|h\|_{\mathcal{H}_k}^2 \tag{9.3}$$

where $\|\cdot\|_{\mathcal{H}_k}^2$ is the RKHS norm on $\mathcal{H}_k$, which measure in a sense the smoothness of functions in $\mathcal{H}_k$. Thanks to the Representer theorem, any solution to Problem 9.3 can be parameterized by a vector $\boldsymbol{\alpha} \in \mathbb{R}^n$:

$$h_{\boldsymbol{\alpha}} := \sum_{j=1}^{n}\alpha_j k(\mathbf{x}_j,\cdot),$$

which makes optimization in the RKHS amenable. For KRR the optimal coefficients $\widehat{\boldsymbol{\alpha}}$ can be found in close form:

$$\widehat{\boldsymbol{\alpha}} := (K + n\lambda I)^{-1}\mathbf{y}^{(m)},$$

with $\mathbf{y}^{(m)} := (y_{t+1+m})_{t=1}^n$, $I \in \mathbb{R}^{n\times n}$ the identity matrix and $K \in \mathbb{R}^{n\times n}$ with entries $K_{tj} = k(\mathbf{x}_t,\mathbf{x}_j)$.

In practice, to handle the large volume of training data, we use an approximated version of KRR. Nyström approximation (Williams and Seeger, 2001; Drineas and Mahoney W., 2005) exploits a random subset of points from the training data. Concretely, we sample randomly and uniformly without replacement $p \in \mathbb{N}$ indices $\{i_1,...,i_p\}$ among the integers in $[\![n]\!]$, and replace $h_{\widehat{\boldsymbol{\alpha}}}$–see e.g. (Rudi et al., 2015)–with:

$$\widetilde{h}_{\widetilde{\boldsymbol{\alpha}}} := \sum_{j=1}^{p}\widetilde{\alpha}_j k(\mathbf{x}_{i_j},\cdot),$$

where $\widetilde{\boldsymbol{\alpha}} \in \mathbb{R}^p$ is given by the following close form:

$$\widetilde{\boldsymbol{\alpha}} := (K_{np}^{\mathrm{T}} K_{np} + \lambda n K_{pp})^{\dagger} K_{np}^{\mathrm{T}} \mathbf{y}^{(m)}. \tag{9.4}$$

Where $A^{\dagger}$ denotes the Moore-Penrose pseudo-inverse of a matrix A, and $K_{np} \in \mathbb{R}^{n\times p}$ is defined by the entries $(K_{np})_{tj} := k(\mathbf{x}_t, \mathbf{x}_{i_j})$ and $K_{pp} \in \mathbb{R}^{p\times p}$ by the entries $(K_{pp})_{bj} = k(\mathbf{x}_{i_b}, \mathbf{x}_{i_j})$.

9.3.3 Details on variable selection

OLS with forward stepwise selection (OLS f-stepwise). When performing linear regression, variable selection can be performed in a greedy manner. First an intercept is fit to the data and then at each step we solve OLS problems–Problem 9.2–adding in turns each one of the remaining variables. We then keep the one which best improve the model according to some criterion. In Dupré et al. (2020), the Bayesian information criterion is used. However, in our experiments we rather used the improvement of the score on half of the validation set, as it led to better experimental performances.

LASSO. Provided the regularization intensity λ is well chosen, the L1 penalty of the LASSO shrinks the coefficients associated the variables which are less important towards zero. Then the model uses mostly the relevant variables and the magnitude of the coefficients can be looked at to deduce what these variables are. This is the type of analysis that we perform in Section 9.4.1.

Hilbert-Schmidt independence criterion (HSIC). The HSIC (Gretton et al., 2005) is an independence measure. Similarly to the KRR, it makes use of RKHSs to embed implicitly a set of observations into a high-dimensional space and consider a notion of independence in this space which allows for detection of nonlinear dependencies. More precisely, let us consider a positive-definite kernel $k : \mathcal{X}^2 \longrightarrow \mathbb{R}$ for the input observations and a one $g : (\mathbb{R}^m)^2 \longrightarrow \mathbb{R}$ for the output observations. For this variable selection technique, we consider all time horizons in $[\![t+1, t+1+m]\!]$ together as the kernelized framework allows for this. In practice, we estimate HSIC from the data as (Gretton et al., 2008):

$$\widehat{\mathrm{HSIC}} := \frac{1}{n^2}\mathrm{Trace}(HKHG),$$

where $H \in \mathbb{R}^{n\times n}$ is the centering matrix $H := \frac{1}{n}(I - \mathbf{1}\mathbf{1}^{\mathrm{T}})$ with $\mathbf{1} \in \mathbb{R}^n$ a vector full of ones and $I \in \mathbb{R}^{n\times n}$ the identity matrix. The matrices $K \in \mathbb{R}^{n\times n}$ and $G \in \mathbb{R}^{n\times n}$ are the kernel matrices:

$$\begin{aligned}(K)_{tj} &:= k(\mathbf{x}_t, \mathbf{x}_j),\\ (G)_{tj} &:= g(\mathbf{y}_{t+1:m}, \mathbf{y}_{j+1:m}).\end{aligned}$$

HSIC takes its values between 0 and 1, a value of 0 meaning independence and a value of 1 means full dependence.

However, to be able to compute the estimator for the large number of data points, we recourse to Nyström approximation as well (Zhang et al., 2018). We then sample randomly and uniformly without replacement $p \in \mathbb{N}$ indices $\{i_1, ..., i_p\}$ from the integers in $[\![n]\!]$ for the input observations and $p' \in \mathbb{N}$ ones $\{i'_1, ..., i'_{p'}\}$ for the output observations. We then define the Nyström features maps (Yang et al., 2012) (centered in the feature space using H):

$$\widehat{\Phi} := HK_{np}K_{pp}^{-\frac{1}{2}},$$
$$\widehat{\Psi} := HG_{np'}G_{p'p'}^{-\frac{1}{2}},$$

where the matrices K_{np} and K_{pp} are defined as for Equation (9.4) and the matrices $G_{np'}$ and $G_{p'p'}$ are defined similarly for the kernel g however based on the set of indices $\{i'_1, ..., i'_{p'}\}$. The Nyström HSIC estimator is then (Zhang et al., 2018):

$$\widetilde{\text{HSIC}} := \|\frac{1}{n}\widehat{\Phi}^{\mathrm{T}}\widehat{\Psi}\|_F^2,$$

where for a matrix A, the Frobenius norm is defined as $\|A\|_F^2 := \text{Trace}(A^{\mathrm{T}}A)$.

Backward selection with HSIC (BAHSIC). To perform variable selection, we start with all the available variables and then at each round, we compute the HSICs between the input variables and the target variable removing one input variable at a time. A given percentage of the input variables for which these HSICs are the highest are removed. We keep iterating in this way to rank the variables. Then, the ones removed the latest are the most important ones. The detailed algorithm corresponds to Algorithm 1 in Song et al. (2012). A forward version exists as well, however, the authors advocate the use of backward selection to avoid missing important variables. Finally as a side note, in practice we use as Gaussian kernels setting bandwidth following the recommendations from Song et al. (2012).

9.4 Importance of variables and their evolution through time

In this section, we study variable selection using the LASSO in Section 9.4.1 and BAHSIC in Section 9.4.2 to determine which variables are the most important and how their importance evolves through time.

9.4.1 Linear variable selection with LASSO

LASSO scores. We now describe the computations carried out to extract a subset of relevant variables suitable for interpretation from fitted LASSO models. In practice for each data split, we validated the regularization parameter λ on the validation set and obtained estimated LASSO coefficients. Now to reduce the number of variables we must rank them according to an importance metric based on these coefficients; we do so for each time horizon separately. So as to to avoid assigning more weight to models for which λ was selected small, for each farm and each data split we normalize the coefficients by the absolute value of the biggest one. As opposed to the grouped variable selection performed in Section 9.4.2, the observations through time for a given variable can be separated by the LASSO shrinking (a variable can be selected for instance at time t_0 but not at t_1). Consequently, we sum the normalized coefficients

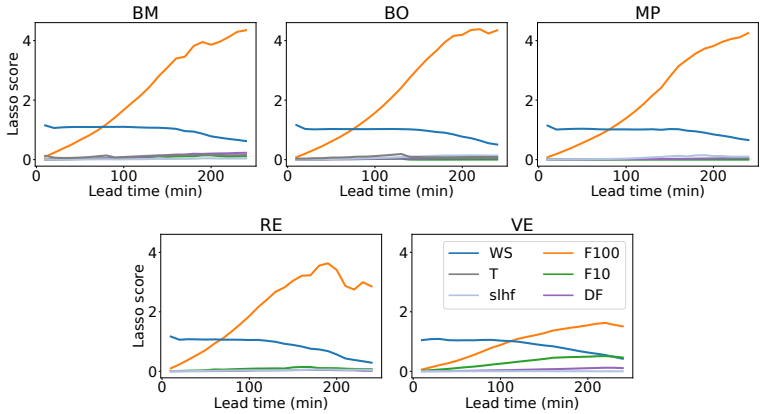


Figure 9.3: Linear variable selection with the LASSO (Wind speed as target)

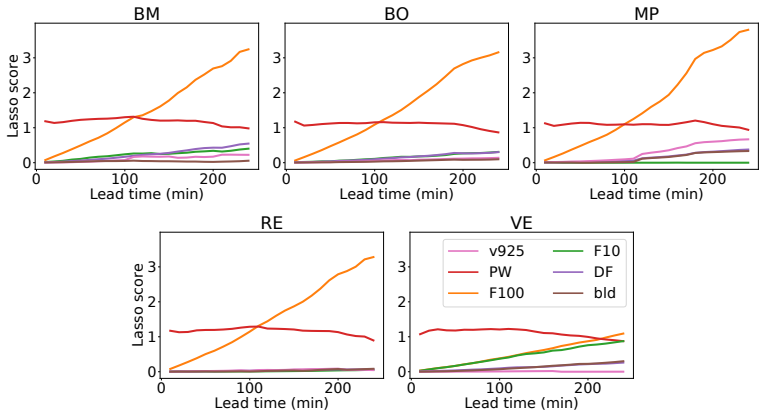


Figure 9.4: Linear variable selection with the LASSO (Wind power as target)

corresponding to different time instants for the same variable and in doing so, we obtain a single quantity per variable. Then we average these quantities over the data splits and call the resulting quantities LASSO scores. To sum up, at this point we have for each prediction horizon and each farm a set of such scores for each variable. Then, to select variables, we average these LASSO scores over farms. Finally, based on these average scores, for each prediction horizon, we keep the top 6 variables.

Interpretation. For these selected variables, we plot the evolution through time of the LASSO scores in Figure 9.3 for wind speed and in Figure 9.4 for wind power. We make the following key observations:

- At all locations, two variables are much more important than the rest. The in situ observed wind speeds (WS) and the ECMWF predicted wind speed at altitude 100m (F100) stand out for wind speed prediction. For wind power prediction, the in situ power production (PW) along with F100 are of particular importance. We can relate these results to the good performances of the LASSO for wind speed prediction in Section 9.5. Then if we look at the relative magnitudes of the coefficients, we can deduce that only using a linear combination of past local wind speeds (WS) and predicted wind speeds (F100), we can get an already good description of the future local wind speed.

- The location VE can be singled out from the others. Indeed the predicted wind speed at altitude 10m (F10) appears, and the forecasted wind speeds (F100 and F10) take longer to take over the in situ variables, especially when predicting wind power. This may be explained by a lesser representativity of the ECMWF forecasts for this location which may be linked to the elevation variations in the surroundings of the farm that we mentioned in Section 9.2.1.

As a concluding note, the dynamics of the local wind speed seem to be very well approximated by a simple linear model combining very few in situ variables and ECMWF ones. For predicting directly wind power however, we see in Section 9.5 the results are a bit less convincing, possibly due to the nonlinear aspect of the power curve.

9.4.2 Nonlinear variable selection with HSIC

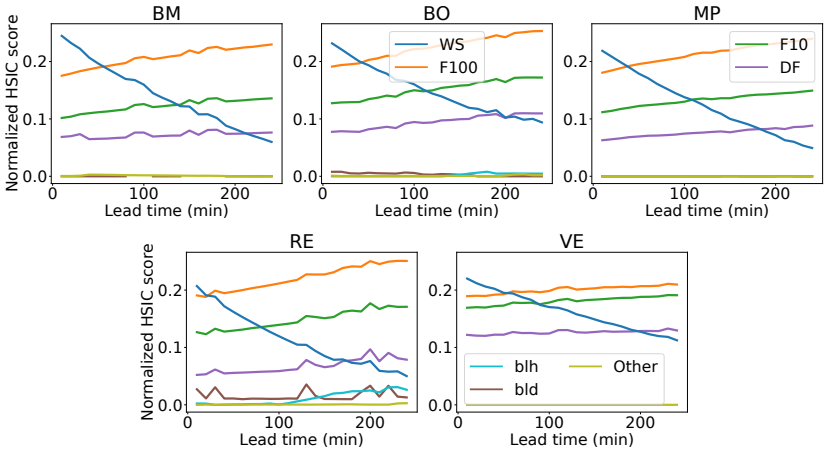


Figure 9.5: Nonlinear variable selection using HSIC (Wind speed as variable)

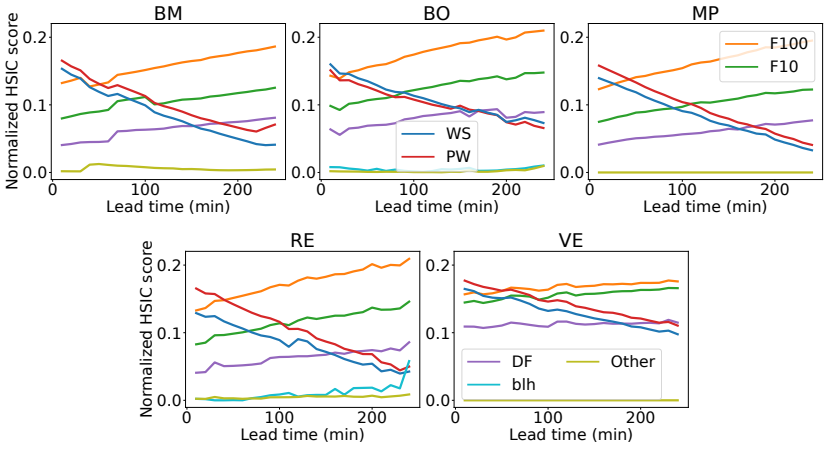


Figure 9.6: Nonlinear variable selection using HSIC (Wind power as target)

HSIC-score. For each farm and each split of the data we run BASHSIC on the training set until we have 5 variables left. Then for each variable we estimate the HSIC with that variable removed and normalize this value using the maximum HSIC value among these quantities. The normalized HSIC score appearing in the figures corresponds to 1 minus this score averaged over the training sets; the higher it is, the more

important the variable is. We display the results in Figure 9.5 for wind speed and in Figure 9.6 for wind power.

Global interpretation. We make the following key observations:

- As expected, the in situ variables are most relevant for the shortest time horizon and the ECMWF variables take progressively the lead for longer horizons. However, compared to linear feature selection, ECMWF variables take the lead faster here–between 10-50 minutes as opposed to 70-100 minutes for linear selection. The retained variables are mostly the same as the one selected by the LASSO (WS or WS and PW depending on the target) and F100. However, F10 and DF are now more systemically retained with a significant importance.

- As for the LASSO selection, probably due to the lesser accuracy of ECMWF forecasts for this location, we can single out VE where the importance of the in situ variable(s) decreases much less fast than at the other farms.

Presence of DF and F10. In comparison with linear selection, we have two more variables of interest (DF and F10). F10 describes the wind speed at lower levels and DF the wind shear near the surface. They thus bring useful information about the wind and its vertical shear, and likely help to correct deficiencies of the NWP model's description of wind at 100m. The fact that they appear here and not in the linear framework indicates a nonlinear relation, which is not surprising as the shear relates to the level of turbulence in the boundary layer. Additionally, the above results bring a fairly sharp answer to another question underlying this study. As the calculation of near-surface winds in NWP models involves parameterizations, they are not the most reliable output of NWP models. Consequently, one could expect that, informed about other aspects of the boundary layer and local wind realizations, a nonlinear method could capture better the relationship between the boundary layer and the near-surface winds. This is not the case: BAHSIC clearly select rather the wind variables as the best source of information. Over variable terrain (VE), wind speed at different heights (F10) are more used, suggesting that the NWP model indeed fails to accurately describe the wind shear. And yet variables describing the boundary layer (e.g. stratification) still remain unused or marginal. Over flat terrain, the wind speed at 100m height (F100) is the major source of information, which is positive and encouraging regarding the accuracy of NWP models.

9.5 Wind speed and wind power forecasting

In this Section we compare several ML models for predicting both wind speed and wind power, exploiting the variable selection techniques from the previous section. We include the two main baselines, namely persistence which predicts the last in situ observation and ECMWF which uses the F100 forecasts from the ECMWF. Note that the details of the parameters considered for tuning the models is available in Appendix C.

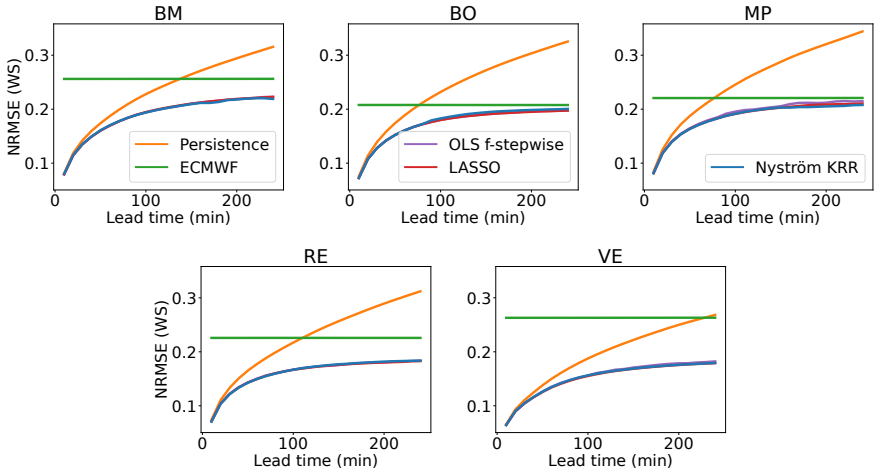


Figure 9.7: Average NRMSE at all time horizons for the three methods performing best overall according to Table 9.4 as well as for Persistence and ECMWF (wind speed as target)

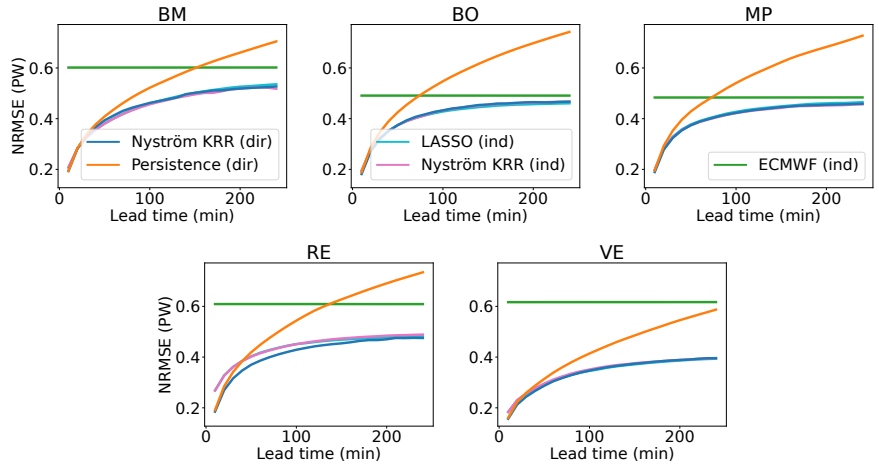


Figure 9.8: Average NRMSE at all time horizons for the three methods performing best overall according to Table 9.5 as well as for Persistence and ECMWF (wind power as target)

9.5.1 Experimental setup

Metrics. We evaluate our results using the normalized root mean squared error (NRMSE) as in Dupré et al. (2020). Let $(z_t)_{t=1}^n$ denote the realizations of a (scalar-valued) target variable. We define its global mean as $\bar{z} = \frac{1}{n}\sum_{t=1}^n z_t$. Given a set of predicted values $(\widehat{z_t})_{t=1}^n$, it is defined as:

$$\text{NRMSE} := \frac{\sqrt{\frac{1}{n}\sum_{t=1}^n (\widehat{z_t} - z_t)^2}}{\bar{z}}.$$

However, in order to compare the methods over the full time span, we need to introduce a new metric. If we were to simply average NRMSEs over time, then the resulting average would not make much sense because of the difference of magnitude between the errors at the different time horizons. Therefore, we propose to compare

Method (average rank)	BM	BO	MP	RE	VE
LASSO (1.8)	1.12	**0.13**	0.16	**0.14**	**0.16**
Nyström KRR (2.0)	**1.05**	0.36	**0.06**	0.19	0.21
OLS f-stepwise (2.8)	1.1	0.16	0.47	0.18	0.34
Feedforward NN (3.4)	1.08	0.46	0.37	0.45	0.66
XG Boost (5.0)	1.81	1.25	1.63	0.74	0.97
ECMWF (6.4)	7.81	3.63	3.75	6.66	11.37
Persistence (6.6)	5.59	6.71	7.02	6.7	4.5

Table 9.4: Average NRMSE degradation w.r.t. best predictor for wind speed prediction ($\times 10^{-2}$)

the NRMSE at each time horizon to a specific anchor reflecting what is achievable: the NRMSE of the best predictor for this time horizon. Let $\mathcal{F}$ be a given set of predictors– for instance when predicting wind speed we have $\mathcal{F} :=$ {Nyström KRR, LASSO, OLS f-stepwise, XG-Boost, Feedforward NN, Persistence, ECMWF}. Given a predictor $f \in \mathcal{F}$, a prediction horizon $m_0 \in [\![m]\!]$ and a data split s (among a total of $S \in \mathbb{N}$ data splits), let $\mathrm{NRMSE}_{s,m_0}^{(f)}$ denote the NRMSE of predictor f for the prediction horizon m_0 on the data split s. We then define the average NRMSE degradation of a predictor f_0 (with respect to the best predictor):

$$\frac{1}{mS}\sum_{s=1}^{S}\sum_{m_0=1}^{m}\left(\mathrm{NRMSE}_{s,m_0}^{(f_0)} - \min_{f\in\mathcal{F}} \mathrm{NRMSE}_{s,m_0}^{(f)}\right). \tag{9.5}$$

The best possible value is zero as it means that over all splits and over all horizons, the predictor was the best one.

Direct/indirect prediction. When we predict wind power, we consider two prediction techniques. Either we predict directly the wind power (direct approach) or we predict the wind speed which we transform using an estimated power curve in the same fashion as in (Dupré et al., 2020) (indirect approach). A theoretical power curve could be used as well, however, in this work we estimate it from the training WS and PW observations using median of nearest neighbors interpolation using 250 neighbors.

Model selection. We follow the methodology introduced in Section 9.3.1 and refer the reader to Section 9.3.2 for details on the ML methods. In practice, for each data split, the key parameters of the different methods are chosen using the validation set (the regularization parameters λ, the Gaussian kernel's γ for KRR, the number of variables for OLS f-stepwise, the architecture for the feedforward NN etc.). We provide the details of the considered parameters in the supplementary material.

9.5.2 Comparisons over the 10 minutes - 4 hours range

Overall efficiency of ML models. From a general perspective, our experiments show that combining a NWP model's outputs with local observations is very beneficial for predicting both wind speed and wind power at all the time horizons considered. To that end, Figure 9.7 displays the evolution of the NRSME for the two baselines (persistence and ECMWF) as well as for the three ML methods which performed best for wind speed prediction in Table 9.4 while Figure 9.8 displays the same for wind power prediction; the displayed ML methods being the top three ones from Table 9.5. For three farms (VE and to a lesser extend, BM and RE), even after four hours the improvement over ECMWF is still quite large. For BO and MP it becomes less important, yet

Type	Method (average rank)	BM	BO	MP	RE	VE
Direct	**Nyström KRR (2.4)**	4.01	1.2	0.63	**0.46**	**0.7**
Indirect	Nyström KRR (3.0)	**3.54**	1.19	**0.55**	2.97	1.2
Indirect	LASSO (3.4)	4.04	**0.8**	1.08	2.72	0.87
Direct	Feedforward NN (4.0)	3.56	1.71	2.29	1.08	1.67
Indirect	OLS f-stepwise (4.4)	3.7	0.87	1.53	3.42	1.61
Direct	XG Boost (direct) (5.2)	4.62	2.39	1.5	1.8	1.8
Direct	OLS f-stepwise (6.6)	4.15	3.01	3.56	2.42	3.0
Direct	LASSO (direct) (7.0)	4.91	3.15	3.46	2.54	2.46
Direct	Persistence (9.4)	12.08	15.48	14.91	14.25	9.88
Indirect	ECMWF (10.2)	18.93	8.66	8.02	19.74	28.53
Indirect	Persistence (10.4)	13.21	15.89	15.22	16.39	10.86

Table 9.5: Average NRMSE degradation w.r.t. best predictor for wind power prediction ($\times 10^{-2}$)

still present. The improvement can be quite dramatic for very short horizons (first 100 minutes or so), and a bit less important for longer time horizons. This is probably linked to the representativity of the NWP model's outputs which depends on the location.

Quantitative comparison. We now use the NRMSE degradation w.r.t. the best predictor–Equation (9.5)–to compare the methods. The results are displayed in Table 9.4 (WS as target) and in Table 9.5 (PW as target). On the one hand, for wind speed prediction, the LASSO is the best ranked method. Relating this to the results from Section 9.4.1, it shows that the dynamics of the wind speed can be very well described by a linear combination of few ECMWF and local variables (essentially past local wind speeds and forecasted wind speeds). It suggests that the important nonlinear dynamics are overall well captured in the ECMWF variables. On the other hand, it seems better to predict directly wind power and do so using the Nyström KRR. This is not surprising, as the power curve is a nonlinear function and so we expected the linear methods to struggle in direct prediction. Moreover, in direct prediction, we implicitly include the power curve into the learning problem. This is advantageous since for instance a model trained to predict wind speed first will be very eager to forecast well high values (failing to do so would incur a high error term). However to predict wind power, producing accurate forecasts for higher wind speeds is less critical, since in the power curve, the actual wind power as a function of the wind speed is thresholded.

We note that the feedforward NN does not beat indirect prediction with the LASSO. This suggests that the higher expressiveness of NNs beyond the ability to infer the nonlinearity of the power curve is not needed. The difference of performance with direct Nyström KRR is imputable to the optimization error and variability implied by non-convexity of NNs whereas for Nyström KRR the optimization error is close to non-existent thanks to the closed-form solution. XG-Boost also does not perform well, this can be explained by the use of time series as features: these are very correlated and high dimensional which can make tree-based models unstable (Gregorutti et al., 2017). Doing some more work on feature pre-processing should improve the results.

Horizon	Method (average rank)	BM	BO	MP	RE	VE
10 min	**Nyström KRR (1.8)**	8.0	**7.25**	8.15	**7.08**	6.44
	LASSO (2.0)	**7.91**	7.25	8.15	7.12	**6.43**
	OLS f-stepwise (2.2)	7.92	7.25	**8.14**	7.09	6.44
	Persistence (4.4)	8.11	7.55	8.43	7.39	6.56
	Feedforward NN (4.8)	8.26	7.55	8.33	7.37	6.8
	XG Boost (5.8)	8.72	8.26	9.57	7.61	6.77
	ECMWF (7.0)	25.61	20.78	22.06	22.57	26.3
1 hour	**Nyström KRR (1.6)**	17.01	**16.03**	**17.11**	**14.94**	13.48
	LASSO (1.8)	16.94	16.04	17.13	14.95	**13.4**
	OLS f-stepwise (2.6)	**16.92**	16.06	17.3	15.02	13.52
	Feedforward NN (4.0)	17.27	16.27	17.32	15.17	13.72
	XG Boost (5.0)	17.69	17.19	18.77	15.66	14.16
	Persistence (6.0)	18.68	18.81	20.09	17.73	15.0
	ECMWF (7.0)	25.61	20.78	22.06	22.58	26.3

Table 9.6: Average NRMSE for 10 minutes and 1 hour ahead wind speed prediction ($\times 10^{-2}$)

Horizon	Type	Method (average rank)	BM	BO	MP	RE	VE
10 min	**Direct**	**Nyström KRR(1.4)**	19.36	**18.16**	**18.94**	**18.49**	**15.7**
	Direct	OLS f-stepwise (2.0)	**19.04**	18.2	18.96	18.53	15.8
	Direct	LASSO (2.6)	19.06	18.22	19.01	18.56	15.78
	Direct	Persistence (4.4)	19.44	18.92	19.63	19.13	16.09
	Direct	Feedforward NN (4.6)	19.73	18.81	19.56	19.3	16.3
	Direct	XG Boost (6.0)	19.9	19.41	19.77	19.38	16.59
	Indirect	LASSO (7.6)	20.73	19.48	19.91	26.84	18.34
	Indirect	Nyström KRR (8.0)	20.86	19.51	19.87	26.8	18.36
	Indirect	OLS f-stepwise (8.4)	20.73	19.48	19.89	26.87	18.39
	Indirect	Persistence (10.0)	21.67	20.41	20.77	27.32	18.8
	Indirect	ECMWF (11.0)	60.15	49.09	48.34	60.91	61.66
1 hour	**Indirect**	**Nyström KRR (2.8)**	40.02	**39.04**	**38.74**	41.53	31.15
	Direct	Nyström KRR (3.0)	41.26	39.36	38.85	**38.59**	**30.46**
	Indirect	LASSO (3.4)	40.11	39.07	39.1	41.72	30.8
	Indirect	OLS f-stepwise (4.6)	**39.97**	39.16	39.37	42.2	31.36
	Direct	Feedforward NN (4.8)	40.51	39.64	39.58	39.44	31.16
	Direct	LASSO (5.2)	40.39	39.93	40.06	39.83	31.07
	Direct	OLS f-stepwise (5.6)	40.37	39.82	40.26	39.67	31.53
	Direct	XG Boost (6.6)	41.89	40.43	39.22	39.99	31.81
	Direct	Persistence (9.0)	43.58	45.52	45.44	44.84	33.88
	Indirect	Persistence (10.0)	44.68	46.01	45.99	47.36	35.08
	Indirect	ECMWF (11.0)	60.15	49.1	48.34	60.91	61.67

Table 9.7: Average NRMSE for 10 minutes and 1 hour ahead wind power prediction ($\times 10^{-2}$)

9.5.3 Zoom on 10 minutes and 1 hour ahead forecasting

We now propose to zoom in on on two time horizons of particular interest: 10 minutes and 1 hour ahead. We display the raw NRSMEs in Section 9.5.3 for WS and in Table 9.7 for PW.

For 10 minutes ahead prediction, persistence is unsurprisingly very efficient even though small yet significant improvements are already obtained by exploiting also ECMWF information with ML. For both the prediction of WS and PW, all three methods which beat persistence reach similar scores. For WS, these are the same as those

performing best overall in Table 9.4: Nyström KRR, LASSO and OLS f-stepwise. However, for PW, these are Nyström KRR (direct), OLS f-stepwise (direct), LASSO (direct). The first is the leading method in Table 9.4, but the other two are not. We explained their poor performance by the nonlinearity of the power curve which they cannot capture. Nevertheless for the very short-term, this is not an issue. This confirms our findings from Section 9.4: for 10 minutes ahead prediction, the last observed wind power is the most crucial information.

For 1 hour ahead prediction, the rankings of ML methods for both wind speed (Section 9.5.3) and wind power (Table 9.7) almost perfectly match the rankings of methods on the whole time span (Table 9.4 for WS and Table 9.5 for PW).

Overall, this analysis confirms that for both the very short term and the longer term Nyström KRR is a safe choice for wind speed and wind power prediction. For the latter the direct approach with this method should be preferred.

9.5.4 Computational times

Task	Method	Fit time (s)	Predict time (s)
Selection	Nyström BAHSIC	74.61 (73.38, 89.00)	-
Selection & regression	LASSO	1.39 (0.01, 5.03)	0.039 (0.036, 0.044)
Selection & regression	OLS f-stepwise	3.58 (2.92, 4.30)	0.037 (0.037, 0.043)
Regression	Nyström KRR	0.451 (0.414, 0.563)	0.332 (0.300, 0.433)
Regression	XgBoost	0.450 (0.405, 0.548)	0.058 (0.053, 0.070)
Regression	Feedforward NN	75.84 (72.70, 77.27)	0.029 (0.028, 0.031)

Table 9.8: Median (10 % quantile, 90 % quantile) of fit and predict computational times on laptop for direct wind power prediction on BO farm.

We now address the practical concern of computational times. To that end, we measure the time on a laptop to fit the different procedures and to produce the corresponding forecasts. We do so only for one wind farm (BO). We draw randomly 50 pairs containing a split from the dataset (see Section 9.2.2) and a parameter configuration among the ones we used. Then we time the procedures using these pairs. We display the median as well as the 10% and 90% quantiles of the obtained computational times in Table 9.8. To put these computational times into perspective, on the one hand the regression models have extra parameters to tune, and therefore many configurations must be tested. On the other hand Nyström BAHSIC seems expensive but no such tuning must be performed (it eliminates the variables gradually, therefore the ranking can be used to include more or less features afterwards).

9.6 Conclusion

We showed through experiments on several wind farms that we can improve very significantly short-term local forecasts of both wind speed and wind power by combining statistically a NWP model's outputs with local observations. To better understand how, we studied in details the evolution of the variables' importance using two metrics, a linear one based on LASSO coefficients and a nonlinear one using HSIC. Our global conclusion is that NWP wind variables are a very relevant source of information to complement local observations, even for the very short-term. To forecast wind speed, a parsimonious linear combination of NWP and local variables (with the

LASSO) yielded the best result. While to forecast wind power, direct prediction (no power curve involved) with a nonlinear method (Nyström KRR) using a few variables (selected with BAHSIC) is preferable. Beyond the ability to capture the nonlinearity of the power curve, it seems unnecessary to use more complex models which hints that NWP model's outputs describe sufficiently the other nonlinear dynamics involved. For future work, assessing the variability of the predictions, for instance by predicting conditional quantiles (Koenker and Hallock, 2001) which would inform us on the expected distribution of the predictions. This could help mitigate the intermittent effects of wind power production further.

Appendix

A Appendices for Chapter 5

Let us first explicitly give the product of Gaussian kernels that we used as kernel for KAM:

$$k^{(\text{add})} : ((\theta, \xi, u), (\theta', \xi', u')) \longmapsto \exp\left(\frac{-(\xi - \xi')^2}{\sigma_1^2}\right)\exp\left(\frac{-(\theta - \theta')^2}{\sigma_2^2}\right)\exp\left(\frac{-(u - u')^2}{\sigma_3^2}\right). \quad (6)$$

We recall also briefly the definition of a Laplace kernel that use

$$(\theta, \theta') \mapsto \exp\left(-\frac{\|\theta - \theta'\|_1}{\sigma}\right). \quad (7)$$

A.1 Additional experimental details: synthetic dataset

We compute the means over 10 runs with different train/test split for all experiments. For all the methods, λ is taken in a geometric grid of size 20 ranging from 10^{-9} to 10^{-4}. Moreover, we consider the following specific parameters.

- **KPL**. We take a truncated Fourier dictionary including 15 frequencies and use the separable kernel $\mathsf{K}(x, x') := k(x, x')\mathrm{I}$ with k a scalar-valued Gaussian kernel with standard deviation $\sigma_k = 20$ and $\mathrm{I} \in \mathbb{R}^{d \times d}$ the identity matrix. When using the logcosh loss, the parameter ν is set to $\nu = 25$ for the in two experiments related to outliers (so as to approach the absolute loss) and to $\nu = 10$ for the two other experiments.
- **3BE**. We use k a Gaussian kernel with standard deviation $\sigma_k = 3$. We use truncated Fourier bases as dictionaries, we include 10 and 15 frequencies respectively for the input dictionary and the output one.
- **KAM**. We use the kernel defined in Equation (6) taking $\sigma_1 = 0.2$, $\sigma_2 = 0.1$ and $\sigma_3 = 2.5$ and use $J = 20$ functional principal components.
- **FKRR**. We take a Gaussian kernel as input kernel with standard deviation parameter set as $\sigma_{k^{\text{in}}} = 20$. We use a Laplace kernel as output kernel—Equation (7)—, setting its parameter to $\sigma_{k^{\text{out}}} = 0.5$.

A.2 Additional experimental details: DTI dataset

The reported means and standard deviations are computer over 20 runs with different train/test split. For all methods (except KE) we center the output functions using the training examples and add back the corresponding mean to the predictions; and we consider values of λ in a geometric grid of size 25 ranging from 10^{-6} to 10^{-2}.

- **KE**. We use a Gaussian kernel with standard deviation in a regular grid ranging from 0.05 to 2 with 200 points.
- **KPL**. For the dictionary, we consider several families of Daubechies wavelets (Daubechies, 1996) with 2 or 3 vanishing moments and 4 or 5 dilatation levels. We use a separable kernel of the form $\mathsf{K}(x, x') = k(x, x')\mathrm{D}$ with k a Gaussian kernel with fixed standard deviation parameter $\sigma_k = 0.9$. The matrix D is a diagonal

matrix of weights decreasing geometrically with the scale of the wavelet at the rate $\frac{1}{b}$ (meaning for instance that at the j-th scale, the corresponding coefficients in the matrix are set to $\frac{1}{b^j}$). b is chosen in a grid ranging from 1 to 2 with granularity 0.1. When using the logcosh loss, we consider values of the parameter ν in $\{0.25, 0.5, 0.75, 1, 1.5, 2, 3, 4, 5, 10\}$.

- **3BE**. We test the same dictionaries of wavelets as for KPL for both the input and the output functions. We use 200 RFFs for the approximated KRRs; and consider standard deviation for the corresponding approximated Gaussian kernel in the grid $\{7.5, 10, 12.5, 15, 17.5, 20\}$.

- **KAM**. We use a product of Gaussian kernels defined in Equation (6) fixing $\sigma_1 = \sigma_2 = \sigma_3 = 0.1$. We consider including $d = 20$ and $d = 30$ principal components for the approximation.

- **FKRR**. We take a Gaussian kernel as input kernel with standard deviation parameter set as $\sigma_{k^{\text{in}}} = 0.9$. We use a Laplace kernel as output kernel—Equation (7)—choosing its parameter in $\sigma_{k^{\text{out}}} \in \{0.5, 0.75, 1, 1.25, 1.5, 1.75, 2, 3, 4, 5, 7.5, 10\}$.

A.3 Additional experimental details: speech data

MSE

The reported means and standard deviations are computed over 10 runs with different train/test split. For all methods, we consider values of λ in a geometric grid ranging from 10^{-12} to 10^{-5} of size 30 and try both centering and not centering the output functions. For ridge-DL-KPL, 1BE/ridge-Four-KPL and FKRR, we use the kernel from Equation (5.39) as input kernel taking $\sigma \in \{3, 4, 5, 7.5, 10\}$.

- **ridge-DL-KPL**. The dictionary ϕ is learnt by solving Problem 3.29 using Pedregosa et al. (2011)'s implementation and using a number of atoms fixed at 30.

- **1BE/ridge-Four-KPL**. We use a truncated Fourier basis as dictionary with included number of frequencies in the grid $\{20, 30, 40, 50\}$.

- **FKRR**. We use a Laplace kernel as output kernel—Equation (7). We consider the following values for its parameter: $\sigma_{k^{\text{out}}} \in \{0.005, 0.01, 0.05, 0.1, 0.125, 0.15\}$.

- **KAM**. We use the kernel defined above in Equation (5.40) for which we consider the following parameters values $\sigma_1 \in \{0.01, 0.05, 0.1, 0.5\}$, $\sigma_2 \in \{$ 0.0005, 0.001, 0.005, 0.01 $\}$ and $\sigma_3 \in \{$ 0.05, 0.1, 0.5, 1, 5$\}$. We consider also $J \in \{30, 40, 50\}$ functional PCAs.

Fitting times

Infrastructure and measurements details. So as to get better control over execution, we perform those experiments on a laptop rather than on the computing cluster used for the other experiments. This laptop is equipped with a 8th Generation Intel Core i7-8665U processor and 16 Gb of RAM. In Python, using the *multiprocessing* package, we execute the tasks in parallel, each on exactly one core of the CPU. We measure the corresponding CPU time using the *process_time()* function from the *time* package.

Parameters. Computation times necessarily depend on the choice of parameters. This dependence can be explicit for parameters determining the complexity of the problems (for instance the size of a dictionary or the size of an approximation grid). For such parameters, we use fixed values for each method which correspond either to the fixed values used or to those elected by cross-validation in the MSEs experiments; we detail those values below. Other parameters can influence the computational times through the conditioning of the problem. To account for this, we consider several values which we give below as well and report the corresponding means and standard deviations in the figure.

The computation times are averaged over 10 runs of the experiments with different shuffling of the dataset and over the VTs. For all methods, we consider values of λ in a geometric grid ranging from 10^{-12} to 10^{-5} of size 30 and center the output functions. For ridge-DL-KPL, 1BE/ridge-Four-KPL and FKRR, we use the kernel from Equation (5.39) as input kernel taking $\sigma = 3$.

- **ridge-DL-KPL**. The dictionary ϕ is learnt by solving Problem 3.29 using the **scikit-learn** implementation (Pedregosa et al., 2011)'s implementation and using a number of atoms fixed at 30.
- **1BE/ridge-Four-KPL**. We use a truncated Fourier basis as dictionary with 50 included frequencies, thus the size of the dictionary is $d = 99$ (cosinuses and sinuses are included plus a constant function).
- **FKRR**. We use a Laplace kernel as output kernel—Equation (7). We consider the following values for its parameter: $\sigma_{k^{\text{out}}} \in \{0.05, 0.1\}$.
- **KAM**. We use the kernel defined in Equation (5.40) for which we use the following parameters values: $\sigma_1 = 0.1$, $\sigma_2 = 0.05$ and $\sigma_3 = 1$. We take $J = 40$ functional PCAs.

B Appendices for Chapter 7

B.1 Proof of Proposition 7.10

Before going through the proof, let us recall Hölder's inequality.

Lemma .1 (Hölder's inequality)**.** *Let $p, q \in [1, +\infty]$ be conjugate exponents, in other words $\frac{1}{p} + \frac{1}{q} = 1$. Let Θ be a measurable space enriched with probability measure μ. Then for any $y, w : \Theta \to \mathbb{R}$ measurable functions one has*

$$\int_\Theta \left| y(\theta) w(\theta) \right| \mathrm{d}\mu(\theta) \leq \|y\|_p \|w\|_q.$$

Moreover, if $p \in]1, +\infty[$, $y \in \mathsf{L}^p(\Theta, \mu)$ and $w \in L^q(\Theta, \mu)$, then equality is attained if and only if $\left|f\right|^p$ and $\left|g\right|^p$ are linearly dependent in $\mathsf{L}^1(\Theta, \mu)$.

We now introduce a lemma useful to the proof of Proposition 7.10.

Lemma .2. *Let $p, q \in]1, +\infty[$ be conjugate exponents and $f \in \mathcal{Y}$ such that $1 < \|y\|_q < +\infty$. Then there exist $v \in \mathcal{Y}$ and $C > 0$ such that*

$$\langle y, v \rangle_{\mathcal{Y}} - \|v\|_p \geq C.$$

Moreover, one can choose h such that whenever $y(\theta) = 0$, $v(\theta) = 0$ also holds.

Proof Let $p, q \in]1, +\infty[$ be conjugate exponents and $y \in \mathcal{Y}$ such that $1 < \|y\|_q < +\infty$. We know that Hölder's inequality becomes an equality if and only if $\left|y\right|^q$ and $|w|^p$ are linearly dependent in $\mathsf{L}^1(\Theta, \mu)$. To that end, let $w\colon \Theta \to \mathbb{R}$ be defined as

$$w(\theta) = \operatorname{sign}(y(\theta))\left|y(\theta)\right|^{\frac{q}{p}} \quad \text{where} \quad \theta \in \Theta. \tag{8}$$

It is to be noted that w does not necessarily belong to $\mathcal{Y}$, yet it belongs to $\mathsf{L}^p(\Theta, \mu)$. By construction, we have

$$\int_\Theta y(\theta) w(\theta) \mathrm{d}\mu(\theta) = \|y\|_q \|w\|_p. \tag{9}$$

We consider a sequence $(w_n)_{n\in\mathbb{N}} \in \mathcal{Y}^{\mathbb{N}}$ such that $w_n(\theta) = \operatorname{sign}(w(\theta)) \min\left(\left|w(\theta)\right|, n\right)$ with $(n, \theta) \in \mathbb{N} \times \Theta$. As $\left|w_n(\theta)\right| \leq n$ for all $(n, \theta) \in \mathbb{N} \times \Theta$ and μ is a probability measure, the functions w_n belong to $\mathcal{Y}$. Since (i) $w_n(\theta) \xrightarrow{n\to\infty} w(\theta)$ for all $\theta \in \Theta$ and (ii) $\left|w_n(\theta)\right| \leq \left|w(\theta)\right|$ for any $n \in \mathbb{N}$ holds μ-almost everywhere, the dominated convergence theorem in $\mathsf{L}^p(\Theta, \mu)$ ensures that $\|w - w_n\|_p \xrightarrow{n\to\infty} 0$. Consequently, it holds that for all $n \in \mathbb{N}$,

$$\begin{aligned}
\left|\int_\Theta y(\theta) w(\theta) \mathrm{d}\mu(\theta) - \int_\Theta y(\theta) w_n(\theta) \mathrm{d}\mu(\theta)\right| &= \left|\int_\Theta y(\theta)[w(\theta) - w_n(\theta)] \mathrm{d}\mu(\theta)\right| \\
&\overset{(a)}{\leq} \int_\Theta \left|y(\theta)\right| \left|w(\theta) - w_n(\theta)\right| \mathrm{d}\mu(\theta) \\
&\overset{(b)}{\leq} \|y\|_q \|w - w_n\|_p.
\end{aligned}$$

In (a) we used that the absolute value of the integral can be upper bounded by the integral of the absolute value, in (b) the Hölder's inequality was invoked. Thus by $\|g - w_n\|_p \xrightarrow{n\to\infty} 0$ and $\|y\|_q < +\infty$, this means that $\langle y, w_n\rangle_{\mathcal{Y}} \xrightarrow{n\to\infty} \int_\Theta y(\theta) w(\theta) \mathrm{d}\mu(\theta) \overset{(9)}{=} \|y\|_q \|w\|_p$, and for all $\epsilon > 0$, there exist $N \in \mathbb{N}$ such that for all $n \geq N$, $\langle y, w_n\rangle_{\mathcal{Y}} > (\|y\|_q - \epsilon)\|w\|_p$. In particular for $\epsilon = \frac{\|y\|_q - 1}{2} > 0$, we have $\langle y, w_N\rangle_{\mathcal{Y}} > \frac{1 + \|y\|_q}{2}\|w\|_p$. Then,

$$\langle y, w_N\rangle_{\mathcal{Y}} - \|w_N\|_p \overset{(c)}{\geq} \langle y, w_N\rangle_{\mathcal{Y}} - \|w\|_p \overset{(d)}{\geq} \frac{1 + \|y\|_q}{2}\|w\|_p - \|w\|_p \geq \underbrace{\frac{\|y\|_q - 1}{2}\|w\|_p}_{>0}.$$

In (c) we used that $\|w_N\|_p \leq \|w\|_p$, (d) is implied by $\langle y, w_N\rangle_{\mathcal{Y}} > \frac{1+\|y\|_q}{2}\|w\|_p$. Taking $h = w_N$ and $C = \frac{\|y\|_q - 1}{2}\|w\|_p$ yields the announced result, by noticing that (8) shows that $y(\theta) = 0$ also implies $h(\theta) = w_N(\theta) = w(\theta) = 0$. ∎

We are now ready to prove Proposition 7.10, which we recall for completeness:

Proposition (Proposition 7.10)**.** *Let $\kappa > 0$, $p \in [1, +\infty]$, and q the conjugate exponent of p. Then for all $y \in \mathcal{Y}$,*

$$H_\kappa^p(y) = \frac{1}{2}\|\mathrm{Proj}_{\mathcal{B}_\kappa^q}(y)\|_{\mathcal{Y}}^2 + \kappa\|y - \mathrm{Proj}_{\mathcal{B}_\kappa^q}(y)\|_p.$$

Proof The proof is structured as follows. We first consider the case of $p = 1$, followed by $p \in]1, +\infty]$, and $p = +\infty$. The reasoning in all cases rely heavily on Hölder's inequality. Throughout the proof it is assumed that $y \in \mathcal{Y}$.

Case $p = 1$: The reasoning goes as follows: we show that $\|y\|_\infty \leq 1$ implies $\|\cdot\|_1^\star(y) = 0$, and $\|y\|_\infty > 1$ gives $\|\cdot\|_1^\star(y) = +\infty$, which allows one to conclude that $\|\cdot\|_1^\star = \chi_{\{\mathcal{B}_1^\infty\}}$.

- **When** $\|y\|_\infty \leq 1$: Exploiting Hölder's inequality, it holds that

$$\langle y, w\rangle_{\mathcal{Y}} \leq \|y\|_\infty \|w\|_1 \quad \text{for all } w \in \mathcal{Y}.$$

  Since $\|y\|_\infty \leq 1$, this implies that

$$\langle y, w\rangle_{\mathcal{Y}} - \|w\|_1 \leq 0 \quad \text{for all } w \in \mathcal{Y}.$$

  The supremum being attained for $w = 0$, we conclude that $\|\cdot\|_1^\star(y) = 0$.

- **When** $\|y\|_\infty > 1$: Let $A = \left\{\theta \in \Theta : \left|y(\theta)\right| > \frac{1+\|y\|_\infty}{2}\right\}$. By the definition of the essential supremum, $\mu(A) > 0$. We define $w\colon \Theta \to \mathbb{R}$ to be the function: $w(\theta) = \operatorname{sign}(y(\theta))$ if $\theta \in A$ and 0 otherwise. Since w is bounded, $w \in \mathcal{Y}$. Denoting by $t > 0$ a running parameter, it holds that

$$\begin{aligned}
\langle y, tw\rangle_{\mathcal{Y}} - \|tw\|_1 &\overset{(a)}{=} \langle y, tw\rangle_{\mathcal{Y}} - t\mu(A) = t\int_\Theta y(\theta)w(\theta)\mathrm{d}\mu(\theta) - t\mu(A) \\
&\overset{(a)}{=} t\int_A \left|y(\theta)\right|\mathrm{d}\mu(\theta) - t\mu(A) \\
&\overset{(b)}{\geq} t\mu(A)\frac{1+\|y\|_\infty}{2} - t\mu(A) = \underbrace{t\,\mu(A)\frac{\|y\|_\infty - 1}{2}}_{>0} \xrightarrow{t\to\infty} +\infty.
\end{aligned}$$

  In (a) we used the definition of g, (b) is implied by the fact that $\left|y(\theta)\right| > \frac{1+\|y\|_\infty}{2}$ for all $\theta \in A$. Thus $\|\cdot\|_1^\star(f) = +\infty$, which concludes the proof.

Case $p \in]1, +\infty[$: The reasoning proceeds as follows: we show that (i) $\|y\|_q \leq 1$ implies $\|\cdot\|_p^\star(y) = 0$, (ii) $1 < \|y\|_q < +\infty$ gives $\|\cdot\|_p^\star(y) = +\infty$, and (iii) $\|y\|_q = +\infty$ results in $\|\cdot\|_p^\star(y) = +\infty$. This allows us to conclude that $\|\cdot\|_p^\star = \chi_{\{\mathcal{B}_1^q\}}$.

- **When** $\|y\|_q \leq 1$: By Hölder's inequality, it holds that

$$\langle y, w\rangle_{\mathcal{Y}} \leq \|y\|_q \|w\|_p \text{ for all } w \in \mathcal{Y}.$$

  Exploiting $\|y\|_q \leq 1$, we get that

$$\langle y, w\rangle_{\mathcal{Y}} - \|w\|_p \leq 0 \text{ for all } w \in \mathcal{Y}.$$

  The supremum being reached for $w = 0$; we conclude that $\|\cdot\|_p^\star(y) = 0$.

- **When** $1 < \|y\|_q < +\infty$: According to Lemma .2, there exist $w \in \mathcal{Y}$ and $C > 0$ such that

$$\langle y, w\rangle_{\mathcal{Y}} - \|w\|_p \geq C.$$

  Denoting by $t > 0$ a running parameter, one arrives at

$$\langle y, tw\rangle_{\mathcal{Y}} - \|tw\|_p \geq tC \xrightarrow{t\to\infty} +\infty.$$

  This shows that $\|\cdot\|_p^\star(y) = +\infty$.

- **When** $\|y\|_q = +\infty$: We consider the sequence of functions $(y_n)_{n\in\mathbb{N}}$ defined as $y_n(\theta) = y(\theta)$ if $\left|y(\theta)\right| \leq n$ and $f_n(\theta) = 0$ otherwise, where $(n,\theta) \in \mathbb{N}\times\Theta$. Each y_n is bounded, thus belongs to $\mathsf{L}^q(\Theta,\mu)$, and the monotone convergence theorem applied to the functions $\left|y_n\right|^q$ states that $\|y_n\|_q \xrightarrow{n\to\infty} \|y\|_q = +\infty$. Thus, there exists $N \in \mathbb{N}$ such that $\|y_n\|_q > 1$. We can then apply Lemma .2 to get $w \in \mathcal{Y}$ and $C > 0$ such that

$$\langle y_N, w\rangle_{\mathcal{Y}} - \|w\|_p \geq C.$$

According to Lemma .2, $w(\theta) = 0$ whenever $y_N(\theta) = 0$, which ensures that

$$\langle y, w\rangle_{\mathcal{Y}} = \langle y_N, w\rangle_{\mathcal{Y}}.$$

Taking a running parameter $t > 0$, this means that

$$\langle y_N, tw\rangle_{\mathcal{Y}} - \|tw\|_p = \langle y, tw\rangle_{\mathcal{Y}} - \|tw\|_p \geq tC \xrightarrow{t\to\infty} +\infty,$$

which shows that $\|\cdot\|_q^\star(y) = +\infty$.

Case $p = +\infty$**:** The reasoning goes as follows: we show that $\|y\|_1 \leq 1$ implies $\|\cdot\|_\infty^\star(y) = 0$, and that $\|y\|_1 > 1$ gives $\|\cdot\|_\infty^\star(y) = +\infty$, which allows one to conclude that $\|\cdot\|_\infty^\star = \chi_{\{\mathcal{B}_1^1\}}$.

- **When** $\|y\|_1 \leq 1$: By applying Hölder's inequality we get that $\langle y, w\rangle_{\mathcal{Y}} \leq \|y\|_1\|w\|_\infty$ for all $w \in \mathcal{Y}$. Using the condition that $\|y\|_1 \leq 1$, this means that $\langle y, w\rangle_{\mathcal{Y}} - \|w\|_\infty \leq 0$ for all $w \in \mathcal{Y}$. Since the supremum is reached for $w = 0$, we get that $\|\cdot\|_\infty^\star(y) = 0$.

- **When** $\|y\|_1 > 1$: Let $w\colon \theta \mapsto \operatorname{sign}(y(\theta))$. Since w is bounded by 1, it belongs to $\mathcal{Y}$, and $\langle y, w\rangle_{\mathcal{Y}} = \|y\|_1$. Running a free parameter $t > 0$, this means that $\langle y, tw\rangle_{\mathcal{Y}} - t\|g\|_\infty = \underbrace{t\,(\|y\|_1 - 1)}_{>0} \xrightarrow{t\to\infty} +\infty$ which implies that $\|\cdot\|_\infty^\star(y) = +\infty$.

■

B.2 Proof of Proposition 7.8

Proposition (Proposition 7.8)**.** *Let* $\kappa > 0$, $p \in [1, +\infty]$, *and* q *the conjugate exponent of* p *(i.e.,* $\frac{1}{p} + \frac{1}{q} = 1$*). Then for all* $y \in \mathcal{Y}$,

$$H_\kappa^p(f) = \begin{cases} \frac{1}{2}\|y\|_{\mathcal{Y}}^2 & \text{if } \|y\|_q \leq \kappa \\ \frac{1}{2}\|\operatorname{Proj}_{\mathcal{B}_\kappa^q}(y)\|_{\mathcal{Y}}^2 + \kappa\|y - \operatorname{Proj}_{\mathcal{B}_\kappa^q}(y)\|_p & \text{otherwise.} \end{cases}.$$

Proof Let us introduce the notation $R(w) = \frac{1}{2}\|y - w\|_{\mathcal{Y}}^2 + \kappa\|w\|_p$ where $y \in \mathcal{Y}$, $w \in \mathcal{Y}$. Then

$$H_\kappa^p(y) \overset{(a)}{=} \inf_{w\in\mathcal{Y}} R(w) \overset{(b)}{=} R(\operatorname{prox}_{\kappa\|\cdot\|_p}(y)) \overset{(c)}{=} \frac{1}{2}\|\operatorname{Proj}_{\mathcal{B}_\kappa^q}(f)\|_{\mathcal{Y}}^2 + \kappa\|y - \operatorname{Proj}_{\mathcal{B}_\kappa^q}(y)\|_p, \tag{10}$$

where (a) follows from the definition of the infimal convolution, (b) is implied by that of the proximal operator using that $\kappa\|\cdot\|_p \in \Gamma_0(Y)$. (c) is a consequence of the Moreau decomposition (Lemma 2.53) as

$$\operatorname{prox}_{\kappa\|\cdot\|_p}(y) = y - \operatorname{prox}_{\left(\kappa\|\cdot\|_p\right)^\star}(y) \overset{(d)}{=} y - \operatorname{prox}_{\chi_{\{\mathcal{B}_\kappa^q\}}}(y) \overset{(e)}{=} y - \operatorname{Proj}_{\mathcal{B}_\kappa^q}(y), \tag{11}$$

where in (d) and (e) we used that

$$\left(\kappa\|\cdot\|_p\right)^\star \overset{(f)}{=} \chi_{\{\mathcal{B}_\kappa^q\}} \text{ with } \frac{1}{p}+\frac{1}{q}=1, \tag{12}$$

$$\operatorname{prox}_{\chi_{\{\mathcal{B}_\kappa^q\}}} \overset{(g)}{=} \operatorname{Proj}_{\mathcal{B}_\kappa^q}. \tag{13}$$

(f) follows from the facts listed in the 3rd and the 2nd line of Table 7.1:

$$\left(\kappa\|\cdot\|_p\right)^\star = \kappa\left(\|\cdot\|_p\right)^\star(\cdot/\kappa) = \kappa\chi_{\{\mathcal{B}_1^q\}}(\cdot/\kappa) = \chi_{\{\mathcal{B}_\kappa^q\}}.$$

(g) is implied by $\chi_{\{\mathcal{B}_\kappa^q\}} = \chi_{\{\mathcal{B}_1^q\}}(\cdot/\kappa)$, the precomposition rule of proximal operators ($\operatorname{prox}_{y(\alpha\cdot)} = \frac{1}{\alpha}\operatorname{prox}_{\alpha^2 f}(\alpha\cdot)$ holding for any $\alpha > 0$—see (2.2) in Parikh and Boyd (2014))—, and $\operatorname{prox}_{\chi_{\{\mathcal{B}_1^q\}}} = \operatorname{Proj}_{\mathcal{B}_1^q}$:

$$\begin{aligned}\operatorname{prox}_{\chi_{\{\mathcal{B}_\kappa^q\}}} &= \operatorname{prox}_{\chi_{\{\mathcal{B}_1^q\}}(\cdot/\kappa)} = \kappa\operatorname{prox}_{\frac{1}{\kappa^2}\chi_{\{\mathcal{B}_1^q\}}}(\cdot/\kappa)\\ &= \kappa\operatorname{prox}_{\chi_{\{\mathcal{B}_1^q\}}}(\cdot/\kappa) = \kappa\operatorname{Proj}_{\mathcal{B}_1^q}(\cdot/\kappa) = \operatorname{Proj}_{\mathcal{B}_\kappa^q}.\end{aligned}$$

Finally we note that $y = \operatorname{Proj}_{\mathcal{B}_\kappa^q}(y)$ is equivalent to $y \in \mathcal{B}_\kappa^q$ which by definition means that $\|y\|_q \le \kappa$. Therefore in that case, Equation (10) indeed simplifies to $\frac{1}{2}\|f\|_{\mathcal{Y}}^2$. ■

B.3 Proof of Proposition 7.18

Proposition (Proposition 7.18). *Let $\epsilon > 0$ and $p \in [1,+\infty]$. Then for all $y \in \mathcal{Y}$,*

$$\ell_\epsilon^p(y) = \frac{1}{2}\|y - \operatorname{Proj}_{\mathcal{B}_\epsilon^p}(y)\|_{\mathcal{Y}}^2.$$

Proof Let $R(g) = \frac{1}{2}\|y-w\|_{\mathcal{Y}}^2 + \chi_{\{\mathcal{B}_\epsilon^p\}}(w)$ where $y \in \mathcal{Y}$ and $w \in \mathcal{Y}$. Then

$$\ell_\epsilon^p(f) \overset{(a)}{=} \inf_{w\in\mathcal{Y}} R(w) \overset{(b)}{=} R\left(\operatorname{prox}_{\chi_{\{\mathcal{B}_\epsilon^p\}}}(y)\right) \overset{(c)}{=} R\left(\operatorname{Proj}_{\mathcal{B}_\epsilon^p}(y)\right) \overset{(d)}{=} \frac{1}{2}\|y - \operatorname{Proj}_{\mathcal{B}_\epsilon^p}(y)\|_{\mathcal{Y}}^2,$$

where (a) follows from the definition of the infimal convolution, (b) is implied by that of the proximal operator and by $\chi_{\{\mathcal{B}_\epsilon^p\}} \in \Gamma_0(\mathcal{Y})$, (c) is the consequence of $\operatorname{prox}_{\chi_{\{\mathcal{B}_\epsilon^p\}}}(y) = \operatorname{Proj}_{\mathcal{B}_\epsilon^p}(y)$ implied by Equation (7.13), in (d) the definition of R was applied. ■

B.4 Additional experimental details: synthetic data

We provide here the full details of the parameters used for the experiments on the toy dataset. For all experiments, we fix the parameter ρ^{in} of the input Gaussian kernel $k_{\mathcal{X}} : (x_1, x_2) \longmapsto \exp\left(-\rho \|x_0 - x_1\|_{\mathcal{X}}^2\right)$ to $\rho^{\text{in}} = 0.01$ and that of the output Gaussian kernel to $\rho^{\text{out}} = 100$. Indeed, since we are only given discrete observations for the input functions as well, we use the available observations to approximate the norms in the above kernels. For the experiments on robustness which results are displayed in Fig. **??** of the main paper, we select via cross-validation the regularization parameter λ and the κ parameters of the Huber loss, considering values in a geometric grid of size 10 ranging from 10^{-6} to 10^{-3} for λ and values in a geometric grid of size 25 ranging from 10^{-3} to 10^{-1} for κ.

B.5 Additional experimental details: DTI data

In this section we provide details regarding the experiments on the DTI dataset. For this dataset, we use a Gaussian kernel as input kernel and a Laplace kernel as output kernel, for the first we fix its parameter to $\rho^{\text{in}} = 1.25$, and for the second, defined as $k_\Theta : (\theta_1, \theta_2) \longmapsto \exp(-\rho^{\text{out}} \|x_0 - x_1\|_{\mathcal{X}})$, we fix its parameter to $\rho^{\text{out}} = 10$. We consider two values of λ, the first one ($\lambda = 10^{-5}$) is chosen too small for the square loss to highlight the additional sparsity-inducing regularization possibilities offered by the ϵ-insensitive loss through the parameter ϵ, while the second one ($\lambda = 10^{-3}$) corresponds to a near-optimal value for the square loss. We do cross-validate the parameters of the losses. For the loss ℓ_ϵ^2 we consider values of ϵ in a geometric grid of size 50 ranging from 10^{-3} to 10^{-1}, while for the loss ℓ_ϵ^∞, we search in a geometric grid of the same size, however ranging from 10^{-3} to $10^{-0.5}$. For the Huber losses H_κ^1 and H_κ^2, we search for κ using a geometric grid of size 50 ranging this time from 10^{-4} to 10^{-1}.

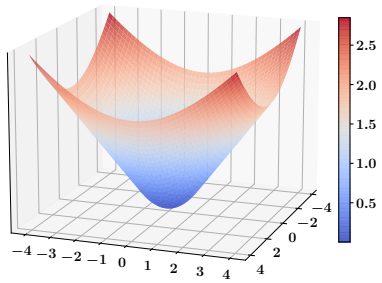
B.6 Additional experimental details: speech data

For all the experiments (with or without corruption), we select the parameter of the input kernel ρ^{in}, the regularization parameter and the parameters of the losses using cross-validation. We fix the parameter of the Laplace output kernel to $\rho^{\text{out}} = 10$. However, to reduce the computational burden, we perform the selection of the parameter ρ^{in} only for the square loss, and then take the corresponding values for the other losses. For this parameter values in a geometric grid of size 15 ranging from 10^{-2} to $10^{-0.5}$ are considered. For λ, the search space is a geometric grid of size 10 ranging from 10^{-10} to 10^{-6}. Finally, for the ϵ-insensitive loss, values of ϵ in a geometric grid of size 80 ranging from 10^{-5} to 10^{-1} are considered, while for the Huber losses we search for κ in a geometric grid of size 100 ranging from 10^{-7} to 1.

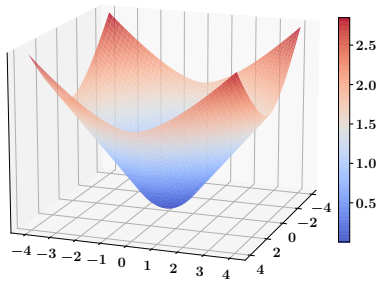
B.7 Additional losses illustrations

In this section, we plot several of the proposed losses when they are defined on $\mathbb{R}^2$ for several values of p.

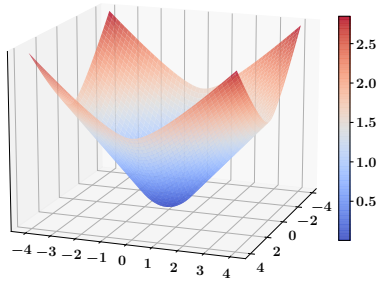
In Figure B.10 we highlight the influence of p on the shape of the ϵ-insensitive loss ℓ_ϵ^p defined on $\mathbb{R}^2$. We set $\epsilon = 1$ and consider values of $p \in \{1.01, 1.5, 2, 3, 5, +\infty\}$. We display $\ell_\epsilon^{1.01}$ in Figure B.10a, $\ell_\epsilon^{1.5}$ in Figure B.10b, ℓ_ϵ^2 in Figure B.10c, ℓ_ϵ^3 in Figure B.10d, ℓ_ϵ^5 in Figure B.10e and ℓ_ϵ^∞ in Figure B.10f.



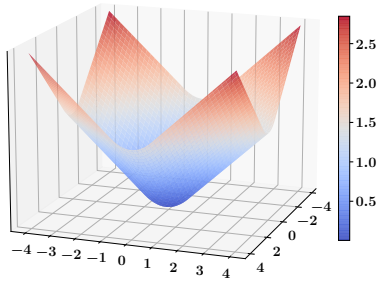
(a) H_κ^2 ($\kappa = 0.8$)



(b) $H_\kappa^{1.5}$ ($\kappa = 0.8$)



(c) $H_\kappa^{1.25}$ ($\kappa = 0.8$)



(d) H_κ^1 ($\kappa = 0.8$)

Figure B.9: Examples of the proposed Huber losses defined on $\mathbb{R}^2$ for different values of p.

Finally, in Figure B.9 we underline the influence that the parameter p has on our proposed Huber losses when it is defined on $\mathbb{R}^2$; we take $\kappa = 0.8$ and we display H_κ^2 in Figure B.9a, $H_\kappa^{1.5}$ in Figure B.9b, $H_\kappa^{1.25}$ in Figure B.9c and H_κ^1 in Figure B.9d.

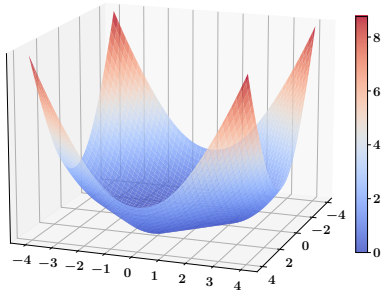
C Appendices for Chapter 9

This short appendix is dedicated to the full description of the parameters that we use in the experiments.

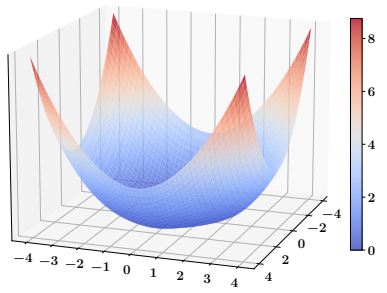
C.1 Importance of variables and their evolution through time

LASSO

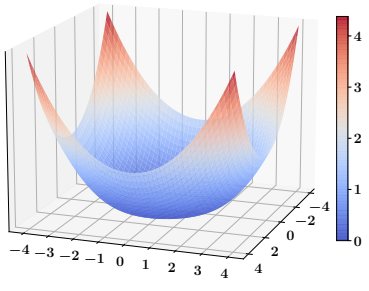
The main parameter of the LASSO is the regularization intensity λ. For each data split, we select it based on the NRMSE achieved on the validation set. We consider values in a geometric grid of size 30 ranging from 10^{-5} to 1.



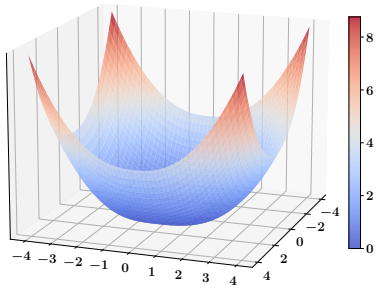
(a) $\ell_\epsilon^{1.01}$ $(\epsilon = 1)$



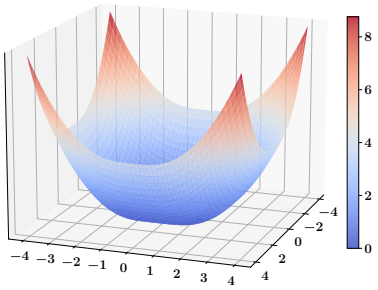
(b) $\ell_\epsilon^{1.5}$ $(\epsilon = 1)$



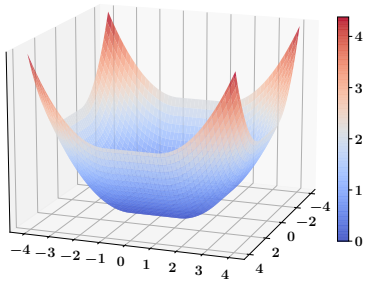
(c) ℓ_ϵ^{2} $(\epsilon = 1)$



(d) ℓ_ϵ^{3} $(\epsilon = 1)$



(e) ℓ_ϵ^{5} $(\epsilon = 1)$



(f) ℓ_ϵ^{∞} $(\epsilon = 1)$

Figure B.10: Examples of the proposed ϵ-insensitive losses defined on $\mathbb{R}^2$ for different values of p.

BAHSIC

We use a Gaussian kernel for both for the input kernel and the output one:

$$k_\gamma(\mathbf{z}, \mathbf{z}') := \exp\left(-\gamma(\|\mathbf{z} - \mathbf{z}'\|_2^2\right).$$

We follow Song et al. (2012) in the choice of the parameter γ. We standardized both our input and output data so we can apply their heuristic: set this parameter to $\frac{1}{2d}$ where d is the dimension of the inputs of the kernel. Then for the input kernel we have $d = q$ and for the output one $d = m$.

For the Nyström approximation, we use fewer points than for the KRR since as highlighted in Zhang et al. (2018), for detection of dependency, a fewer number of anchor points are generally sufficient. We then use 100 points for both the input and output approximation.

C.2 Wind speed and wind power forecasting

As a first general note, since we standardized all the variables, we consider the same parameter ranges for prediction of wind speed and wind power.

LASSO The fitted models used for interpretation in the variable selection section are the same that we use here (so the considered parameters are the same).

OLS f-stagewise We selected on the validation set the number of included variables. We consider the following number of variables: $\{5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 20\}$.

Nyström KRR We select both the input Gaussian kernel's γ parameter and the regularization parameter λ. We consider the following values:

- γ in a geometric space of length 30 ranging from 10^{-6} to 10^{-3}.
- λ in a geometric space of length 30 ranging from 10^{-4} to 5.

For the Nyström approximation, we use 300 sampled points.

Xg-Boost For Xg-Boost, we validate the trees' maximum depth considering values in $\{3, 4, 5, 6\}$ as well as the minimum loss reduction parameter for values in a geometric space of size 50 ranging from 10^{-7} to 50.

Feedforward NN We consider a NN with 3 hidden layers and validate the number of neurons per layer choosing among the possible values $\{(35, 20, 5), (50, 25, 10), (50, 35, 20), (75, 50, 25)\}$.

Bibliography

M. Abramowitz and I. A. Stegun. *Handbook of Mathematical Functions*. Dover, 1965. page 60

M. Aharon, M. Elad, and A. Bruckstein. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54(11):4311–4322, 2006. pages 66, 67

M. Álvarez, L. Rosasco, and N. Lawrence. Kernels for vector-valued functions: a review. *Foundations and Trends in Machine Learning*, 4(3):195–266, 2012. page 36

N. Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, pages 337–404, 1950. pages 25, 26

H. Avron, V. Sindhwani, J. Yang, and M. W. Mahoney. Quasi-monte carlo feature maps for shift-invariant kernels. *Journal of Machine Learning Research*, 17(120):1–38, 2016. page 33

J. Baars and C. Mass. Performance of National Weather Service forecasts compared to operational, consensus and weighted model output statistics. *Wea. Forecast.*, 20: 1034–1047, 2005. page 175

F. Bach. Exploring large feature spaces with hierarchical multiple kernel learning. In *Proceedings of the 21st International Conference on Neural Information Processing Systems (NIPS)*, pages 105–112, 2008. page 162

F. Bach. On the equivalence between kernel quadrature rules and random feature expansions. *Journal of Machine Learning Research*, 18(21):1–38, 2017. page 63

F. Bach, R. Jenatton, J. Mairal, and G. Obozinski. Optimization with sparsity-inducing penalties. *Found. Trends Mach. Learn.*, 4(1):1–106, 2012. pages 144, 162, 163

L. Baldassarre, L. Rosasco, and A. Barla. Multi-output learning via spectral filtering. *Machine Learning*, 87:259–301, 2012. pages 40, 109, 117

P. L. Bartlett, O. Bousquet, and S. Mendelson. Local rademacher complexities. *The Annals of Statistics*, 33(4):1497–1537, 2005. page 38

P. Bauer, A. Thorpe, and G. Brunet. The quiet revolution of numerical weather prediction. *Nature*, 525:47–55, 2015. page 175

H. H. Bauschke and P. L. Combettes. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Springer, 2 edition, 2017. pages 41, 42, 44, 88, 134, 135

A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009. pages 67, 139, 148, 182

A. Berlinet and C. Thomas-Agnan. *Reproducing Kernel Hilbert Spaces in Probability and Statistics*. Kluwer, 2004. page 24

R. Bhatia. *Matrix analysis*. Springer, 1997. page 115

T. Blumensath and M. E. Davies. Normalized iterative hard thresholding: Guaranteed stability and performance. *IEEE Journal of Selected Topics in Signal Processing*, 4(2): 298–309, 2010. page 68

G. Boente, M. Salibian-Barrera, and P. Vena. Robust estimation for semi-functional linear regression models. *Computational Statistics & Data Analysis*, 152:107041, 2020. page 130

O. Bousquet and A. Elisseeff. Stability and generalization. *Journal of Machine Learning Research*, 2:499–526, 2002. page 37

S. Boyd, S. P. Boyd, and L. Vandenberghe. *Convex optimization*. Cambridge University Press, 2004. pages 41, 42

L. Brogat-Motte, A. Rudi, C. Brouard, J. Rousu, and F. d'Alché Buc. Vector-valued least-squares regression under output regularity assumptions, 2023. (https://arxiv.org/abs/2211.08958). page 88

C. Brouard, F. d'Alché Buc, and M. Szafranski. Semi-supervised penalized output kernel regression for link prediction. In *International Conference on Machine Learning (ICML)*, pages 593–600, 2011. page 36

C. Brouard, M. Szafranski, and F. D'Alché-Buc. Input output kernel regression: supervised and semi-supervised structured output prediction with operator-valued kernels. *Journal of Machine Learning Research*, 17(1):6105–6152, 2016. pages 36, 37, 132

B. Cadre. Convergent estimators for the l1-median of Banach valued random variable. *Statistics: A Journal of Theoretical and Applied Statistics*, 35(4):509–521, 2001. page 130

T. T. Cai and M. Yuan. Optimal estimation of the mean function based on discretely sampled functional data: Phase transition. *The Annals of Statistics*, 39:2330–2355, 2011. page 49

A. Caponnetto and E. De Vito. Risk bounds for regularized least-squares algorithm with operator-valued kernels. Technical report, MIT, Computer Science and Artificial Intelligence Laboratory, 2005. page 112

A. Caponnetto and E. De Vito. Optimal rates for the regularized least-squares algorithm. *Foundations of Computational Mathematics*, pages 331–368, 2007. pages 38, 39, 40, 109, 115, 126

C. Carmeli, E. De Vito, and A. Toigo. Vector valued reproducing kernel Hilbert spaces of integrable functions and Mercer theorem. *Analysis and Applications*, 4(04):377–408, 2006. page 34

C. Carmeli, E. De Vito, A. Toigo, and V. Umanità. Vector valued reproducing kernel hilbert spaces and universality. *Analysis and Applications*, 8(1):19–61, 2010. pages 35, 88

P. G. Casazza. The art of frame theory. *Taiwanese journal of mathematics*, 4:129–201, 2000. page 108

T. Chen and C. Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794, 2016. page 180

A. Cohen, I. Daubechies, and J.-C. Feauveau. Biorthogonal bases of compactly supported wavelets. *Communications on Pure and Applied Mathematics*, 45(5), 1992. page 56

A. Cohen, I. Daubechies, and P. Vial. Wavelets on the interval and fast wavelet transforms. *Applied and Computational Harmonic Analysis*, 1(1):54–81, 1993. page 57

C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995. page 30

S. Cotter, B. Rao, K. Engan, and K. Kreutz-Delgado. Sparse solutions to linear inverse problems with multiple measurement vectors. *IEEE Transactions on Signal Processing*, 53:2477–2488, 2005. page 162

M. G. Cox. The numerical evaluation of b-splines. *IMA Journal of Applied Mathematics*, 10(2):134–149, 10 1972. pages 58, 59

C. Crambes, L. Delsol, and A. Laksaci. Robust nonparametric estimation for functional data. *Journal of Nonparametric Statistics*, 20(7):573–598, 2008. page 130

F. Cucker and S. Smale. On the mathematical foundations of learning. *Bulletin of the American Mathematical Society*, 39:1–49, 2001. pages 27, 28, 37, 62

M. Cuturi, K. Fukumizu, and J.-P. Vert. Semigroup kernels on measures. *Journal of Machine Learning Research*, 6:1169–1198, 2005. page 25

M. Cuturi, J.-P. Vert, O. Birkenes, and T. Matsui. A kernel for time series based on global alignments. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 413–416, 2007. page 25

I. Daubechies. *Ten Lectures on Wavelets*. SIAM, 1996. pages 55, 101, 196

C. de Boor. On calculating with b-splines. *Journal of Approximation Theory*, 6(1):50–62, 1972. pages 58, 59

C. de Boor. *A practical guide to Splines - Revised Edition*. Springer, 2001. pages 52, 58, 98

F. Demengel and G. Demengel. *Functional Spaces for the Theory of Elliptic Partial Differential Equations*. Springer, 2012. page 60

F. Dinuzzo, C. S. Ong, P. Gehler, and G. Pillonetto. Learning output kernels with block coordinate descent. In *Proceedings of the 28th International Conference on Machine Learning (ICML)*, pages 49–56, 2011. pages 74, 92

P. Drineas and M. Mahoney W. On the nystrom method for approximating a gram matrix for improved kernel-based learning. *Journal of Machine Learning Research*, 6: 2153–2175, 2005. pages 31, 182

H. Drucker, C. J. C. Burges, L. Kaufman, A. Smola, and V. Vapnik. Support vector regression machines. In *Advances in Neural Information Processing Systems*, volume 9, 1996. pages 30, 128

B. Dumitrescu and P. Irofti. *Dictionary Learning, Algorithms and Applications*. Springer, 2018. page 67

A. Dupré, P. Drobinski, B. Alonzo, J. Badosa, C. Briard, and R. Plougonven. Sub-hourly forecasting of wind speed and wind energy. *Renewable Energy*, 145:2373–2379, 2020. pages 176, 179, 180, 183, 188, 189

A. Dupré, P. Drobinski, J. Badosa, C. Briard, and P. Tankov. The economic value of wind energy nowcasting. *Energies*, 13(20):5266, 2020. page 177

M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image Processing*, 15(12):3736–3745, 2006. page 66

F. Ferraty and P. Vieu. The functional nonparametric model and application to spectrometric data. *Computational Statistics*, 17:545–564, 2002. page 80

F. Ferraty and P. Vieu. *Nonparametric Functional Data Analysis*. Springer, 2006. page 13

F. Ferraty, A. Laksaci, A. Tadj, and P. Vieu. Kernel regression with functional response. *Electronic Journal of Statistics*, 5:159–171, 2011. pages 71, 80, 97

M. Fornasier and H. Rauhut. Recovery algorithms for vector-valued data with joint sparsity constraints. *SIAM Journal on Numerical Analysis*, 46(2):577–613, 2008. page 162

J. H. Friedman. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5):1189–1232, 2001. page 180

H. Glahn and D. Lowry. The use of model output statistics (MOS) in objective weather forecasting. *J. App. Meteor.*, 11:1203–1211, 1972. page 175

G. Golub and C. Van Loan. *Matrix Computations*. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, 2013. page 158

N. Goutham, B. Alonzo, A. Dupré, R. Plougonven, R. Doctors, L. Liao, M. Mougeot, A. Fischer, and P. Drobinski. Using machine-learning methods to improve surface wind speed from the outputs of a numerical weather prediction model. *Boundary Layer Meteorology*, 179:133–161, 2021. page 175

M. S. Gowda and M. Teboulle. A comparison of constraint qualifications in infinite-dimensional convex programming. *SIAM Journal on Control and Optimization*, 28 (4):925–935, 1990. page 45

I. Gradshteyn and I. Ryzhik. In A. Jeffrey, editor, *Table of Integrals, Series, and Products*. Academic Press, 1980. page 28

P. J. Green and B. W. Silverman. *Nonparametric Regression and Generalized Linear Models: A roughness penalty approach*. Chapman and Hall/CRC, 1993. page 52

B. Gregorutti, B. Michel, and P. Saint-Pierre. Correlation and variable importance in random forests. *Statistics and Computing*, 27:659–678, 2017. page 190

A. Gretton, O. Bousquet, A. Smola, and B. Schölkopf. Measuring statistical dependence with hilbert-schmidt norms. In *Algorithmic Learning Theory*, pages 63–77. Springer Berlin Heidelberg, 2005. page 183

A. Gretton, K. Fukumizu, C. Teo, L. Song, B. Schölkopf, and A. Smola. A kernel statistical test of independence. In *Advances in Neural Information Processing Systems*, volume 20, 2008. page 183

Y. Han, L. Mi, L. Shen, C. Cai, Y. Liu, K. Li, and G. Xu. A short-term wind speed prediction method utilizing novel hybrid deep learning algorithms to correct numerical weather forecasting. *Applied Energy*, 312, 2022. page 176

T. Hastie, J. H. Friedman, and R. Tibshirani. *The Elements of Statistical Learning*. Springer, 2001. pages 180, 181

D. L. Hawkins. Some practical problems in implementing a certain sieve estimator of the gaussian mean function. *Communications in Statistics- Simulationas and Computations*, 18, 1989. page 28

L. Hoegaerts, J. A. Suykens, J. Vandewalle, and B. De Moor. Subset based least squares subspace regression in RKHS. *Neurocomputing*, 63:293–323, 2005. page 29

V. Hoolohan, A. S. Tomlin, and T. Cockerill. Improved near surface wind speed predictions using gaussian process regression combined with numerical weather predictions and observed meteorological data. *Renewable Energy*, 126:1043–1054, 2018. page 176

R. A. Horn and C. R. Johnson. *Topics in Matrix Analysis*. Cambridge University Press, 1991. page 92

P. J. Huber. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, pages 73–101, 1964. pages 17, 96, 128

M. Hubert, P. J. Rousseeuw, and P. Segaert. Multivariate functional outlier detection. *Statistical Methods and Applications*, 24:177–202, 2015. page 131

IEA. Renewables information. *International Energy Agency*, page 12, 2018. page 175

IEA. Renewables: Analysis and forecast to 2026. *International Energy Agency*, page 175, 2021. page 175

A. J. Izenman. Reduced-rank regression for the multivariate linear model. *Journal of Multivariate Analysis*, 5(2):248–264, 1975. page 156

R. Jenatton, J.-Y. Audibert, and F. Bach. Structured variable selection with sparsity-inducing norms. *Journal of Machine Learning Research*, 12(84):2777–2824, 2011. page 162

Y. Jiao and J.-P. Vert. The Kendall and Mallows kernels for permutations. In *International Conference on Machine Learning (ICML)*, pages 2982–2990, 2016. page 25

I. T. Jolliffe. *Principal Component Analysis*. Springer, 2002. page 64

H. Kadri, E. Duflos, P. Preux, S. Canu, and M. Davy. Nonlinear functional regression: a functional RKHS approach. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 9, pages 374–380, 2010. pages 13, 16, 18, 35, 36, 71, 72, 73, 97, 129, 130, 170

H. Kadri, M. Ghavamzadeh, and P. Preux. A generalized kernel approach to structured output learning. In *International Conference on Machine Learning (ICML)*, pages 471–479, 2013. page 36

H. Kadri, E. Duflos, P. Preux, S. Canu, A. Rakotomamonjy, and J. Audiffren. Operator-valued kernels for learning from functional response data. *Journal of Machine Learning Research*, 17:1–54, 2016. pages 13, 17, 18, 28, 35, 36, 71, 72, 73, 75, 97, 129, 130, 170

E. Kalnay. *Atmospheric modeling, data assimilation and predictability*. Cambridge University Press, 2003. page 175

I. Kalogridis and S. Van Aelst. Robust functional regression based on principal components. *Journal of Multivariate Analysis*, 173:393–415, 2019. page 130

K. Karhunen. *Über lineare Methoden in der Wahrscheinlichkeitsrechnung*. Annales Academiae Scientiarum Fennicae: Ser. A 1. Kirjapaino oy. sana, 1947. page 64

R. Koenker and K. F. Hallock. Quantile regression. *Journal of Economic Perspectives*, 15 (4):143–156, 2001. page 193

P. Kokoszka and M. Reimherr. *Introduction to Functional Data Analysis*. Chapman and Hall/CRC, 2017. page 49

N. M. Kriege, F. D. Johansson, and C. Morris. A survey on graph kernels. *Applied Network Science*, 5, 2020. page 25

S. Kumar, M. Mohri, and A. Talwalkar. Sampling methods for the nyström method. *Journal of Machine Learning Research*, 13(34):981–1006, 2012. page 31

P. Laforgue, A. Lambert, L. Brogat-Motte, and F. d'Alché Buc. Duality in RKHSs with infinite dimensional outputs: Application to robust losses. In *International Conference on Machine Learning (ICML)*, volume 9, pages 5598–5607, 2020. pages 13, 17, 18, 35, 36, 37, 78, 131, 134, 170

Q. Le, T. Sarlós, and A. Smola. Fastfood - computing Hilbert space expansions in loglinear time. In *International Conference on Machine Learning (ICML)*, volume 28, pages 244–252, 2013. page 33

H. Lee, A. Battle, R. Raina, and A. Y. Ng. Efficient sparse coding algorithms. In *Advances in Neural Information Processing Systems (NIPS) 19*, pages 801–808, 2007. pages 66, 162

Y.-J. Lee, W.-F. Hsieh, and C.-M. Huang. epsilon-SSVR: A smooth support vector machine for epsilon-insensitive regression. *IEEE Transactions on Knowledge & Data Engineering*, (5):678–685, 2005. page 131

Y. Li and T. Hsing. Uniform convergence rates for nonparametric regression and principal component analysis in functional/longitudinal data. *The Annals of Statistics*, 38(6):3321–3351, 2010. page 49

Z. Li, J.-F. Ton, D. Oglic, and D. Sejdinovic. Towards a unified analysis of random fourier features. *Journal of Machine Learning Research*, 22(108):1–51, 2021. pages 33, 109

H. Lian. Nonlinear functional models for functional responses in reproducing kernel Hilbert spaces. *Canadian Journal of Statistics*, pages 597–606, 2007. pages 13, 18, 36, 71, 72, 73, 97, 129, 130, 170

F. T. Liu, K. M. Ting, and Z.-H. Zhou. Isolation forest. In *2008 Eighth IEEE International Conference on Data Mining*, pages 413–422, 2008. page 130

H. Liu, C. Chen, X. Lv, X. Wu, and M. Liu. Deterministic wind energy forecasting: A review of intelligent predictors and auxiliary methods. *Energy Conversion and Management*, 195:328–345, September 2019. doi: 10.1016/j.enconman.2019.05.020. page 175

M. Loève. Fonctions aléatoires du second ordre. In *Processus Stochastiques et Mouvement Brownien*. 1st edition, 1948. page 64

J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online dictionary learning for sparse coding. In *Proceedings of the 26th Annual International Conference on Machine Learning (ICML)*, pages 689–696, 2009. page 66

S. Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, 2008. pages 54, 55, 56, 57

S. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41(12):3397–3415, 1993. page 67

R. A. Maronna and V. J. Yohai. Robust functional linear regression based on splines. *Computational Statistics & Data Analysis*, 65:46–55, 2013. page 130

V. Masson-Delmotte, P. Zhai, A. Pirani, S. Connors, C. Péan, S. Berger, N. Caud, Y. Chen, L. Goldfarb, M. I. Gomis, M. Huang, K. Leitzell, E. Lonnoy, J. Matthews, T. K. Maycock, T. Waterfield, O. Yelekçi, R. Yu, and B. Z. (eds.). IPCC, 2021: Summary for Policymakers. In *Climate Change 2021: The Physical Science Basis. Contribution of Working Group I to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change*, pages 1–42, 2021. page 175

W. F. Massy. Principal components regression in exploratory statistical research. *Journal of the American Statistical Association*, 60(309):234–256, 1965. page 156

G. Meanti, L. Carratino, L. Rosasco, and A. Rudi. Kernel methods through the roof: handling billions of points efficiently. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 14410–14422, 2020. page 31

M. Mehra. *Wavelets Theory and Its Applications*. Springer, 2018. page 55

Y. Meyer. Principe d'incertitude, bases hilbertiennes et algèbres d'opérateurs. In *Séminaire Bourbaki : volume 1985/86, exposés 651-668*, number 145-146 in Astérisque. Société mathématique de France, 1985. URL http://www.numdam.org/item/SB_1985-1986__28__209_0/. page 55

Y. Meyer. Ondelettes sur l'intervalle. *Revista Matemática Iberoamericana*, 7(2):115–133, 1991. page 57

Y. Meyer. *Wavelets and Operators*, volume 1 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, 1993. page 55

C. Micchelli and M. Pontil. On learning vector-valued functions. *Neural Computation*, 17:177–204, 2005. pages 36, 72, 107, 130

C. Micchelli, Y. Xu, and H. Zhang. Universal kernels. *Journal of Machine Learning Research*, 7:2651–2667, 2006. page 27

V. Mitra, Y. Ozbek, H. Nam, X. Zhou, and C. Y. Espy-Wilson. From acoustics to vocal tract time functions. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4497–4500, 2009. pages 101, 152

J. J. Moreau. Proximité et dualité dans un espace hilbertien. Technical report, 1965. (https://hal.archives-ouvertes.fr/hal-01740635). pages 42, 43

J. S. Morris. Functional regression. *The Annual Review of Statistics and Its Application*, 2:321–359, 2015. pages 13, 71, 77

E. A. Nadaraya. On estimating regression. *Theory of Probability & Its Applications*, 9 (1):141–142, 1964. page 80

S. Nagy, I. Gijbels, and D. Hlubinka. Depth-based recognition of shape outlying functions. *Journal of Computational and Graphical Statistics*, 26(4):883–893, 2017. page 130

T. V. Nguyen, R. K. W. Wong, and C. Hegde. Provably accurate double-sparse coding. *Journal of Machine Learning Research*, 20(141):1–43, 2019. page 68

J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, 2006. page 162

E. Novak, M. Ullrich, H. Woźniakowski, and S. Zhang. Reproducing kernels of sobolev spaces on $\setminus^d$ and applications to embedding constants and tractability. *Analysis and Applications*, 16(5):693–715, 2018. page 61

G. Obizinski, T. Ben, and M. I. Jordan. Joint covariate selection and joint subspace selection for multiple classification problems. *Statistics and Computing*, pages 231–252, 2010. page 162

I. Okumus and A. Dinler. Current status of wind energy forecasting and a hybrid method for hourly predictions. *Energy Conversion and Management*, 123:362–371, September 2016. page 175

J. Oliva, W. Neiswanger, B. Poczos, E. Xing, H. Trac, S. Ho, and J. Schneider. Fast function to function regression. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 38, pages 717–725, 2015. pages 13, 71, 72, 76, 97, 170

B. O'Donoghue and E. Candès. Adaptive restart for accelerated gradient schemes. *Foundations of Computational Mathematics*, 15:715–732, 2015. page 148

N. Parikh and S. Boyd. Proximal algorithms. *Found. Trends Optim.*, 1(3):127–239, 2014. pages 43, 132, 202

Y. Pati, R. Rezaiifar, and P. Krishnaprasad. Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition. In *Proceedings of 27th Asilomar Conference on Signals, Systems and Computers*, volume 1, pages 40–44, 1993. page 67

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. pages 182, 197, 198

G. Pedrick. Theory of reproducing kernels for Hilbert spaces of vector-valued functions. Technical report, University of Kansas, Department of Mathematics, 1957. pages 17, 33

J. Petrovich, R. M. L., and C. Daymont. Highly irregular functional generalized linear regression with electronic healthrecords. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 2018. page 51

S. Pezzulli and B. W. Silverman. Some properties of smoothed principal components analysis for functional data. *Computational Statistics*, 8:1–16, 1993. page 66

I. Pinelis and A. I. Sakhanenko. Remarks on inequalities for large deviation probabilities. *Theory of Probability and Its Applications*, 30:143–148, 1986. pages 40, 115

T. Qingguo. M-estimation for functional linear regression. *Communications in Statistics-Theory and Methods*, 46(8):3782–3800, 2017. page 130

A. Rahimi and B. Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1177–1184, 2007. pages 31, 32, 62, 76

A. Rakotomamonjy. Surveying and comparing simultaneous sparse approximation (or group-lasso) algorithms. *Signal Processing*, 91(7):1505–1526, 2011. page 162

A. Rakotomamonjy, R. Flamary, J. Salmon, and G. Gasso. Convergent working set algorithm for lasso with non-convex sparse regularizers. In *International Conference on Artificial Intelligence and Statistics*, volume 151, pages 5196–5211, 03 2022. page 162

J. O. Ramsay and C. J. Dalzell. Some tools for functional data analysis. *Journal of the Royal Statistical Society: Series B (Methodological)*, 53(3):539–561, 1991. pages 13, 58, 65

J. O. Ramsay and B. W. Silverman. *Functional Data Analysis*. Springer, first edition, 1997. page 13

J. O. Ramsay and B. W. Silverman. *Functional Data Analysis*. Springer, second edition, 2005. pages 17, 50, 52, 65, 70, 71, 77

J. O. Ramsay and B. W. Silverman. *Applied functional data analysis: methods and case studies*. Springer, 2007. page 16

C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. Adaptive Computation and Machine Learning, 2006. pages 28, 60

M. Reimherr, B. Sriperumbudur, and B. Taoufik. Optimal prediction for additive function on function regression. *Electronic Journal of Statistics*, 12:4571–4601, 2018. pages 13, 71, 72, 77, 79, 97, 170

G. C. Reinsel and R. P. V. Velu. *Multivariate Reduced-Rank Regression*. Springer, 1998. page 156

K. Richmond. *Estimating Articulatory Parameters from the Acoustic Speech Signal*. PhD thesis, The Center for Speech Technology Research, Edinburgh University, 2002. pages 101, 152

S. M. Robinson. Regularity and stability for convex multivalued functions. *Mathematics of Operations Research*, 1(2):130–143, 1976. page 45

R. T. Rockafellar. *Convex analysis*, volume 36. Princeton university press, 1970. page 41

R. T. Rockafellar. *Conjugate Duality and Optimization*. SIAM, 1974. page 45

V. Roth and B. Fischer. The group-lasso for generalized linear models: Uniqueness of solutions and efficient algorithms. In *Proceedings of the 25th International Conference on Machine Learning (ICML)*, pages 848–855, 2008. page 162

R. Rubinstein, M. Zibulevsky, and M. Elad. Efficient implementation of the k-svd algorithm using batch orthogonal matching pursuit. Technical report, Technion, Computer Science Department, 2008. (https://www.cs.technion.ac.il/~ronrubin/Publications/KSVD-OMP-v2.pdf). page 67

R. Rubinstein, M. Zibulevsky, and M. Elad. Double sparsity: Learning sparse dictionaries for sparse signal approximation. *IEEE Transactions on Signal Processing*, 58(3): 1553–1564, 2010. pages 67, 68

A. Rudi and L. Rosasco. Generalization properties of learning with random features. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 3218–3228, 2017. page 33

A. Rudi, R. Camoriano, and L. Rosasco. Less is more: Nyström computational regularization. In *International Conference on Neural Information Processing Systems (NIPS)*, pages 1657–1665, 2015. pages 31, 182

A. Rudi, L. Carratino, and L. Rosasco. Falkon: An optimal large scale kernel method. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 3891–3901, 2017. page 31

M. Sangnier, O. Fercoq, and F. d'Alché-Buc. Data sparse nonparametric regression with ϵ-insensitive losses. In *Asian Conference on Machine Learning (ACML)*, pages 192–207, 2017. page 131

I. J. Schoenberg. Contribution to the problem of approximation of equidistant data by analytic functions. *Quarterly of Applied Mathematics*, 4:45–99; 112–141, 1946. page 58

B. Schölkopf and A. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, 2002. pages 24, 63, 180

B. Schölkopf, K. Tsuda, and J.-P. Vert. *Kernel Methods in Computational Biology*. The MIT Press, 2004. page 25

H. L. Shang. A survey of functional principal component analysis. *AStA Advances in Statistical Analysis*, 98:121–142, 2014. page 65

J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004. pages 24, 180

H. Shin and S. Lee. An RKHS approach to robust functional linear regression. *Statistica Sinica*, pages 255–272, 2016. page 130

B. W. Silverman. Spline smoothing: The equivalent variable kernel method. *The Annals of Statistics*, 12(3):898–916, 1984. page 58

B. W. Silverman. Some aspects of the spline smoothing approach to non-parametric regression curve fitting. *Journal of the Royal Statistical Society. Series B (Methodological)*, 47(1):1–52, 1985. page 58

B. W. Silverman. Smoothed functional principal components analysis by choice of norm. *The Annals of Statistics*, 24(1):1–24, 1996. page 66

V. Sima. *Algorithms for Linear-Quadratic Optimization*. Chapman and Hall/CRC, 1996. pages 74, 92

S. Smale and D.-X. Zhou. Learning theory estimates via integral operators and their approximations. *Constructive Approximation*, pages 153–172, 2007. page 38

L. Song, A. Smola, A. Gretton, J. Bedo, and K. Borgwardt. Feature selection via dependence maximization. *Journal of Machine Learning Research*, 13:1393–1434, 2012. pages 181, 184, 206

B. Sriperumbudur and Z. Szabó. Optimal rates for random Fourier features. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1144–1152, 2015. page 32

B. K. Sriperumbudur, K. Fukumizu, and G. R. Lanckriet. Universality, characteristic kernels and rkhs embedding of measures. *Journal of Machine Learning Research*, 12 (70):2389–2410, 2011. page 27

G. Staerman, P. Mozharovskyi, C. on Stephan, and F. d'Alché Buc. Functional isolation forest. In *Asian Conference on Machine Learning (ACML)*, volume 101, pages 332–347, 2019. page 130

M. L. Stein. *Interpolation of Spatial Data*. Springer, 1999. page 61

I. Steinwart. *Journal of Machine Learning Research*, 2:67–93, 2001. pages 26, 27

I. Steinwart. Consistency of support vector machines and other regularized kernel classifiers. *IEEE Transactions on Information Theory*, 51(1):128–142, 2005. page 38

I. Steinwart and A. Christmann. *Support vector machines*. Springer Science & Business Media, 2008. page 24

B. L. Sturm and M. G. Christensen. Comparison of orthogonal matching pursuit implementations. In *Proceedings of the 20th European Signal Processing Conference (EUSIPCO)*, pages 220–224, 2012. page 67

J. Sulam, B. Ophir, M. Zibulevsky, and M. Elad. Trainlets: Dictionary learning in high dimensions. *IEEE Transactions on Signal Processing*, 64(12):3180–3193, 2016. page 68

A. Tascikaraoglu and M. Uzunoglu. A review of combined approaches for prediction of short-term wind speed and power. *Renewable and Sustainable Energy Reviews*, 34: 243–254, 2014. page 175

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996. pages 67, 180

A. N. Tikhonov and V. Y. Arsenin. *Solution of Ill-posed Problems*. Winston & Sons, 1977. page 15

R. Tuo and C. F. Jeff Wu. A theoretical framework for calibration in computer models: Parametrization, estimation and convergence properties. *SIAM/ASA Journal on Uncertainty Quantification*, 4(1):767–795, 2016. page 60

S. Ullah and C. F. Finch. Applications of functional data analysis: A systematic review. *BMC medical research methodology*, 13(1):1–12, 2013. pages 13, 16

G. Wahba. *Spline Models for Observational Data*. SIAM, 1990. page 61

J.-L. Wang, J.-M. Chiou, and H.-G. Müller. Functional data analysis. *Annual Review of Statistics and Its Application*, 3:257–295, 2016. pages 13, 70

R.-H. Wang. *Multivariate Spline Functions and Their Applications*. Springer Netherlands, Dordrecht, 2001. page 58

G. S. Watson. Smooth regression analysis. *Sankhyā: The Indian Journal of Statistics, Series A (1961-2002)*, 26(4):359–372, 1964. page 80

H. Wendland. *Scattered Data Approximation*. Cambridge University Press, 2004. pages 32, 59

C. K. I. Williams and M. Seeger. Using the Nyström method to speed up kernel machines. In *Advances in Neural Information Processing Systems (NIPS)*, pages 682–688, 2001. pages 31, 182

R. Williamson, A. Smola, and B. Scholkopf. Generalization performance of regularization networks and support vector machines via entropy numbers of compact operators. *IEEE Transactions on Information Theory*, 47(6):2516–2532, 2001. page 63

L. Wilson and M. Vallée. The Canadian Updateable Model Output Statistics (UMOS) systemm: Design and Development tests. *Wea. Forecast.*, 17:206–222, 2002. page 175

L. Xiao, L. Cai, W. Checkley, and C. Crainiceanu. Fast covariance estimation for sparse functional data. *Statistics and Computing*, 28:511–522, 2018. page 51

T. Yang, Y.-f. Li, M. Mahdavi, R. Jin, and Z.-H. Zhou. Nyström method vs random fourier features: A theoretical and empirical comparison. In F. Pereira, C. Burges, L. Bottou, and K. Weinberger, editors, *Advances in Neural Information Processing Systems (NIPS)*, volume 25, 2012. pages 31, 184

F. Yao, H.-G. Müller, and J.-L. Wang. Functional data analysis for sparse longitudinal data. *Journal of the American Statistical Association*, 100(470):577–590, 2005. page 51

M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1): 49–67, 2006. page 162

V. Yurinsky. *Sums and Gaussian Vectors*. Springer, 1995. pages 40, 115

M. Zamo, L. Bel, O. Mestre, and J. Stein. Improved gridded wind speed forecasts by statistical postprocessing of numerical models with block regression. *Weather and Forecasting*, 31:1929–1945, 2016. page 175

J. Zhang, A. May, T. Dao, and C. Ré. Low-precision random Fourier features for memory-constrained kernel approximation. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 1264–1274, 2019. page 33

Q. Zhang, S. Filippi, A. Gretton, and D. Sejdinovic. Large-scale kernel methods for independence testing. *Statistics and Computing*, 28, 1 2018. pages 184, 206

X. Zhang and J.-L. Wang. From sparse to dense functional data and beyond. *The Annals of Statistics*, 44(5):2281–2321, 2016. page 49

D.-X. Zhou. *Journal of Computational and Applied Mathematics*, 220:456–463, 2008. page 59

C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal. Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization. *ACM Transactions on Mathematical Software*, 1997a. page 97

H. Zhu, C. K. I. Williams, R. Rohwer, and M. Morciniec. Gaussian regression and optimal finite dimensional linear models. In *Neural Networks and Machine Learning*, pages 167–184, 1997b. page 28

H. Zhu, P. J. Brown, and J. S. Morris. Robust, adaptive functional regression in functional mixed model framework. *Journal of the American Statistical Association*, 106 (495):1167–1179, 2011. page 130

Titre : Améliorer la régression à valeurs fonctionnelles avec des noyaux reproduisant: rapidité, flexibilité et robustesse

Mots clés : régression fonctionnelle, noyaux à valeurs opérateurs, dictionnaires, pertes robustes

Résumé : L'augmentation du nombre et de la sophistication des appareils collectant des données permet de suivre l'évolution d'une multitude de phénomènes à des résolutions très fines. Cela étend le champ des applications possibles de l'apprentissage statistique. Un tel volume peut néanmoins devenir difficile à exploiter. Cependant quand leur nombre augmente, les données peuvent devenir redondantes. On peut alors chercher une représentation exploitant des propriétés du processus génératif.
Dans cette thèse, nous nous concentrons sur la représentation fonctionnelle. Bien sûr, les données sont toujours des observations discrètes. Néanmoins, si nous pensons que ces suites doivent être par exemple lisses ou de variations bornées, une telle représentation peut être à la fois plus fidèle et de dimension plus faible. Nous nous concentrons sur les modèles non-linéaires de régression à valeurs fonctionnelles (FOR) en utilisant une extension des espaces de Hilbert à noyau reproduisant pour les fonctions à valeurs vectorielles (vv-RKHS) qui constitue la clef de voûte de plusieurs méthodes existantes. Notre objectif est d'en proposer de nouvelles plus performantes sur le plan de la complexité calculatoire liée au caractère fonctionnel et/ou celui du choix de la fonction de perte.
Nous introduisons l'apprentissage de projection kernelisé (KPL) qui combine les vv-RKHSs et la représentation de signaux sur des dictionnaires. La perte demeure fonctionnelle, néanmoins le modèle prédit seulement un nombre fini de coordonnées. Nous bénéficions alors de la flexibilité de l'espace d'hypothèse tout en réduisant nettement la complexité liée aux sorties fonctionnelles. Pour la perte quadratique, nous introduisons deux estimateurs en forme close, l'un est adapté lorsque les fonctions de sortie sont observées totalement, et l'autre l'est lorsqu'elles ne le sont que partiellement. Nous montrons que chacun est consistant en termes d'excès de risque. Nous proposons aussi d'utiliser d'autres fonctions de perte différentiables, de combiner KPL avec les techniques de passage à l'échelle ou encore de sélectionner le dictionnaire via une pénalité structurée.
Une autre partie est dédiée au problème de FOR dans des vv-RKHS de fonctions à valeurs fonctionnelles en utilisant une famille de fonctions de pertes que nous introduisons comme définies à partir d'une convolution infimale. Celles-ci peuvent encourager soit la parcimonie soit la robustesse, le degré de localité de ces propriétés étant contrôlé via un paramètre dédié. Grâce à leur structure, ces pertes se prêtent particulièrement bien à la résolution par dualité lagrangienne. Nous surmontons alors les différents défis que pose la dimension infinie des variables duales en proposant deux représentations pour résoudre chaque problème dual numériquement.

Title : Function-valued regression with kernels: Improving speed, flexibility and robustness

Keywords : functional regression, operator-valued kernels, dictionaries, robust losses

Abstract : With the increasing ubiquity of data-collecting devices, a great variety of phenomena is monitored with finer and finer accuracy, which constantly expands the scope of Machine Learning applications. Dealing with such volume of data efficiently is however challenging. Fortunately, as measurements get denser, they may become gradually redundant. We can then greatly reduce the burden by finding a representation which exploits properties of the generating process and/or is tailored for the application at hand.
This thesis revolves around an aspect of this idea: functional data. Data indeed consist of discrete measurements, but sometimes thinking of these as functional, we can exploit prior knowledge on smoothness to obtain a better yet lower dimensional representation. The focus is on nonlinear models for functional output regression (FOR), relying on an extension of reproducing kernel Hilbert spaces for vector-valued functions (vv-RKHS), which is the cornerstone of many nonlinear existing FOR methods. We propose to challenge those in two aspects: their computational complexity with respect to the number of measurements per function and their focusing solely on the square loss.
To that end, we introduce the new framework of kernel projection learning (KPL) combining vv-RKHSs and representation of signals in dictionaries. The loss remains functional, however the model predicts only a finite number of representation coefficients. This approach retains the many advantages of vv-RKHSs yet greatly alleviates the computational burden incurred by the functional outputs. We derive two estimators in closed-form using the square loss, one for fully observed output functions and one for discretized ones. We show that both are consistent in terms of excess risk. We demonstrate as well the possibility to use other differentiable and convex losses, to combine this framework with large scale kernel methods and to automatically select the dictionary using a structured penalty.
In another contribution, we propose to solve the regression problem in vv-RKHSs of function-valued functions for the family of convoluted losses which we introduce. These losses can either promote sparsity or robustness with a parameter controlling the degree of locality of these properties. Thanks to their structure, they are particularly amenable to dual approaches which we investigate. We then introduce two representations to overcome the challenges posed by the functional nature of the dual variables and we propose corresponding algorithms to solve each dual problem.

Institut Polytechnique de Paris
91120 Palaiseau, France