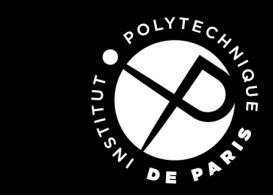
Learning function-valued functions in reproducible kernel Hilbert spaces with integral losses : Application to infinite task learning

Alex Lambert



INSTITUT POLYTECHNIQUE DE PARIS

NNT : 2021IPPAT016

Thèse de doctorat

Learning Function-Valued Functions in Reproducible Kernel Hilbert Spaces with Integral Losses : Application to Infinite Task Learning

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 07/07/2021, par

Alex Lambert

Composition du Jury :

Stephan Clémençon Professeur, Télécom Paris (LTCI)	Président
Hachem Kadri Associate Professor, Université Aix-Marseille (LIS)	Rapporteur
Dino Sejdinovic Associate Professor, University of Oxford (OxCSML)	Rapporteur
Marianne Clausel Professeure, Université de Lorraine (IECL)	Examinateur
Johan Suykens Professor, Katholieke Universiteit Leuven (ESAT-STADIUS)	Examinateur
Florence d'Alché-Buc Professeure, Télécom Paris (LTCI)	Directrice de thèse
Zoltàn Szabò Senior Researcher, École Polytechnique (CMAP)	Co-directeur de thèse
Stephane Canu Professeur, INSA de Rouen (LITIS)	Invité
Alessandro Rudi Chercheur, INRIA Paris (SIERRA)	Invité

Contents

Remerciements

À la mémoire de mon grand-père Jeannot.
Résistant, Pompier, Pêcheur, Cycliste.

Je souhaite débuter cette thèse en remerciant, de manière non exhaustive, les personnes qui m'ont accompagné et soutenu tout au long de ces années de doctorat.

Merci tout d'abord à mes deux directeurs de thèse, Florence et Zoltan, pour m'avoir offert la possibilité de faire de la recherche et m'en avoir transmis la passion. Merci pour votre implication, votre gentillesse, vous formez un duo remarquable et je m'estime chanceux d'avoir pu me former sous votre supervision. To Zoltan, who will leave France before having the time to learn the language, farewell. It's been a real pleasure working with you, and I wish you all the best in your journey.

I would like to thank Hachem Kadri and Dino Sejdinovic for having reviewed this manuscript, and Marianne Clausel, Stephan Clémençon, and Johan Suykens for being part of my jury. Many thanks also go to Alessandro Rudi and Stéphane Canu for accepting the invitation to my defense.

Merci à Mathurin et Pierre, deux très belles rencontres, que les aventures soient parisiennes, saclaysiennes, ou nippones vous avez toujours répondu à l'appel. Je n'oublie pas l'ami Kévin, et j'espère vous voir bientôt afin de compléter le quatuor de docteurs. Merci à Jache, fidèle acolyte, jamais le dernier pour explorer les ruelles de la butte aux Cailles ou rosser des petits jeunes (et moins jeunes, Charles et Quentin en ont fait les frais) au baby-foot. Merci aux collègues de Télécom, Alex Garcia, Mastane, Eugène, Anna, Pierre Ablin, Hamid et aux plus jeunes Guillaume, Anas, Sholom, Dimitri, Luc, Jayneel, Tamim: courage, la dernière ligne droite arrive vite.

Merci à mes co-auteurs, l'inénarrable Raymond B. , Lucho, Sanjeel, et Pierre encore: ce fut pour moi un plaisir de travailler avec vous, et quitte à cravacher la nuit entière, autant le faire en bonne compagnie.

Merci à mes amis télécommiens, Bryan, Prush, Chabert, Giano, toujours là pour m'épauler. Merci à Lanas et Solene pour ces trois années de colocation sous le signe de la bonne humeur (bonne humeur !), cela n'aurait pas eu la même saveur sans vous.

Merci à ma famille qui m'a toujours soutenu et m'a permis d'en arriver où j'en suis aujourd'hui. J'ai une pensée pour ma grand-mère Yvette et pour mes cousins, trop nombreux pour les citer tous ici mais avec qui j'ai du temps à rattraper (Léonie je te regarde). Merci à ma petite soeur Margot qui me permet de rester à jour sur le programme de mathématiques de licence, je suis très fier de toi. Merci à mes parents, soutiens sans faille tout au long de ma vie, dire que je vous dois beaucoup serait un euphémisme.

Merci à Estelle de partager ma vie et d'avoir supporté avec patience les aléas de cette thèse - les jours radieux arrivent.

Abstract

Kernel methods are regarded as a cornerstone of machine learning. They allow to model real-valued functions in expressive functional spaces, over which regularized empirical risk minimization problems are amenable to optimization and yield estimators whose statistical behavior is well studied. When the outputs are not reals but higher dimensional, vector-valued reproducible kernel Hilbert spaces (vv-RKHSs) based on operator-valued kernels (OVKs) provide similarly powerful spaces of functions, and have proven useful to tackle problems such as multi-task learning, structured prediction, or function-valued regression.

In this thesis, we introduce an original functional extension of multi-output learning called *infinite task learning* (ITL), that allows to jointly solve an infinite number of parameterized tasks, including for instance quantile regression, cost-sensitive classification and density level set estimation.

We propose a learning framework based on convex integral losses that encompasses the ITL problem and function-valued regression. Optimization schemes dedicated to solving the associated regularized empirical risk minimization problems are designed. By sampling the integral losses, we derive finite-dimensional representation of the solution under several choices of regularizers or shape constraints penalties, while keeping theoretical guarantees over their generalization capabilities. We also employ dualization techniques with the benefit of bringing desirable properties such as robustness or sparsity to the estimators thanks to the use of convoluted losses. Scalability issues are addressed by deriving optimization algorithms in the the context of approximated OVKs whose corresponding vv-RKHSs are of finite dimension. The use of trainable deep architectures composed by a neural network followed by a shallow kernel layer is also investigated as a way to learn the kernel used in practice on complex data such as images.

We apply these techniques to various ITL problems and to robust function-to-function regression, that are tackled in the presence of outliers. We also cast style transfer problems as a vectorial output ITL problem and demonstrate its efficiency in emotion transfer.

Résumé

Les méthodes à noyaux sont au coeur de l'apprentissage statistique. Elles permettent de modéliser des fonctions à valeurs réelles dans des espaces de fonctions à fort potentiel représentatif, sur lesquels la minimisation de risques empiriques régularisés est possible et produit des estimateurs dont le comportement statistique est largement étudié. Lorsque les sorties ne sont plus réelles mais de plus grande dimension, les Espaces de Hilbert à Noyaux Reproduisants à valeurs vectorielles (vv-RKHSs) basés sur des Noyaux à Valeurs Opérateurs (OVKs) fournissent des espaces de fonctions similaires et permettent de traiter des problèmes tels que l'apprentissage multi-tâche, la prédiction structurée ou la regression à valeurs vectorielles.

Dans cette thèse, nous étudions différents problèmes liés à l'apprentissage de modèles à valeurs fonctionnelles, un cas que nous traitons en modélisant les sorties comme vivant dans un espace de Hilbert de dimension infinie. Ceci nous permet notamment d'exploiter la théorie associée aux vv-RKHSs précédemment évoquée, avec l'aide de noyaux à valeurs opérateurs agissant sur ces mêmes espaces. Ces problèmes requièrent des fonctions de perte dédiées, définies sur des espaces fonctionnels qui diffèrent des fonctions de perte usuelles utilisées en dimension finie. En particulier, nous nous intéressons aux fonctions de perte convexes pouvant s'exprimer sous la forme d'une intégrale sur l'espace de définition des fonctions de sortie. Ces *pertes intégrales* constituent une extension naturelle des fonctions de perte utilisées en dimension finie, et permettent d'aborder la recherche de fonctions cibles comme équivalent à la minimisation d'un risque dans les vv-RKHSs.

Le cadre d'apprentissage que nous proposons repose sur la minimisation de risques empiriques régularisés en présence de telles fonctions de perte. De par la nature fonctionnelle des sorties, les problèmes d'optimisation en résultant souffrent d'écueuils supplémentaires à ceux déjà présents en dimension finie. En premier lieu, et contrairement au cadre fini-dimensionnel, il n'est pas garanti que les estimateurs bénéficient d'une représentation de taille finie sur une base naturelle de l'espace de fonction considéré. De plus, même si une telle décomposition était trouvée, les pertes intégrales sont accompagnées de problèmes de calculabilité: en effet les intégrales (et leurs gradients) ne peuvent le plus souvent pas être exprimées sous forme analytique et doivent être estimées.

Une première contribution de cette thèse est de fournir un ensemble de techniques d'optimisation à même de répondre à ces deux problématiques, tout en s'assurant que les algorithmes proposés ne soient pas trop gourmands en ressources de calcul. En reposant sur un échantillonage des pertes intégrales, nous obtenons une représentation de dimension finie des estimateurs pour différents choix de régularisation dans les vv-RKHSs, rendant possible l'utilisation d'algorithmes de descente de gradient dont la

convergence est bien étudiée. L'usage de la dualité lagrangienne vient compléter cette technique, en offrant un problème nouveau basé sur la transformée de Fenchel-Legendre des pertes intégrales, qui est explicitée pour quelques fonctions de perte utiles. Les difficultés liées à la représentation des variables duales sont traitées par l'usage de splines linéaires, qui conjuguées à l'approximation de l'opérateur intégral lié au noyaux rendent les problèmes d'optimisation solubles. Les problèmes de passages à l'échelle sont aussi traités par l'utilisation de noyaux approchés, dont les vv-RKHSs associés sont de dimension finie. Ceci permet de plus l'utilisation d'algorithmes de descente de gradient stochastique, les pertes intégrales étant interprétées comme des espérances. Enfin, nous proposons une architecture de modèle composée d'un réseau de neurone et d'une dernière couche à noyaux, qui rend possible l'apprentissage de représentations appropriées aux noyaux utiles dans les applications avec des données complexes comme les images.

Une seconde contribution de cette thèse est l'introduction d'une extension fonctionnelle originale du cadre multi-tâche appelée *Apprentissage d'un Continuum de Tâches* (ITL), qui permet de résoudre conjointement un continuum de tâches paramétrées, parmi lesquelles la régression quantile, la classification à coût assymétrique, ou l'estimation de niveaux de densité. Le cadre ITL est traité grâce aux techniques d'optimisation précédemment développées pour les pertes intégrales, avec quelques ajustements ayant pour but de gérer des contraintes souples sur la monotonicité des estimateurs dans le cas de la régression quantile notamment. Nous nous intéressons aussi aux capacités en généralisation des estimateurs, que nous contrôlons théoriquement en les étudiant sous l'angle de la stabilité uniforme. Enfin, cette approche fonctionnelle au cadre multi-tâche nous permet aussi de revisiter les problemes de transfert de style sous l'angle ITL, avec une application au transfert d'émotion.

Le dernier problème abordé dans cette thèse concerne la regression fonction-à-fonction robuste en présence de valeurs aberrantes. Nous substituons à la norme carrée des *pertes convoluées*, comme la perte ϵ-insensible ou la perte Huber, plus à même d'imposer des estimateurs parcimonieux ou robustes. Les problèmes d'optimisation sont traités dans le dual, où les synergies entre convolution infinitésimale et transformée de Fenchel-Legendre rendent possible le développement d'algorithmes de descente de gradient proximal. En particulier, certaines libertés dans la définition des pertes convoluées permettent l'émergence de différents types de parcimonie, ainsi que la résistance à différents types de valeurs aberrantes.

Notation

$:=$	Equal by definition
$\mathbb{R}_+$	Non-negative reals
$\mathbb{N}^*$	Positive integers: $\{1, 2, \ldots\}$
$[n]$	Set of integers from 1 to n ($\{1, \ldots, n\}$)
$\mathcal{X}$	Input space
Θ	Hyperparameter space. Compact $\subset \mathbb{R}^p$
$\mathcal{U}$	Hilbert space
$\mathcal{F}(\Theta, \mathcal{U})$	Set of functions from Θ to $\mathcal{U}$
$\mathcal{C}(\Theta, \mathcal{U}),\ \mathcal{C}^1(\Theta, \mathcal{U})$	Continuous, continuously differentiable functions from Θ to $\mathcal{U}$
$L^2[\Theta, \mu; \mathcal{U}], L^2[\Theta, \mu]$	Hilbert space of μ-integrable functions $\subset \mathcal{F}(\Theta, \mathcal{U})$; in the latter case: $\mathcal{U} = \mathbb{R}$
$\mathcal{Y}$	Output space assumed to be a Hilbert space $\subseteq L^2[\Theta, \mu; \mathcal{U}]$
$\langle \cdot, \cdot \rangle_{\mathcal{Y}},\ \lVert \cdot \rVert_{\mathcal{Y}}$	Scalar product and norm in $\mathcal{Y}$
$\mathcal{L}(\mathcal{U}, \mathcal{V}), \mathcal{L}(\mathcal{Y})$	Bounded linear operators from $\mathcal{U}$ to $\mathcal{V}$; $\mathcal{Y} := \mathcal{U} = \mathcal{V}$
Id	Identity operator on the ambient space
$A^\sharp$	Adjoint of operator $A \in \mathcal{L}(\mathcal{Y})$
$\operatorname{Ker} A$	Nullspace of a bounded linear operator A
$\operatorname{Im} A$	Image of a bounded linear operator A
$k_{\mathcal{X}} \colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$	Input kernel, scalar-valued
$k_\Theta \colon \Theta \times \Theta \to \mathbb{R}$	Hyperparameter kernel, scalar-valued
$K \colon \mathcal{X} \times \mathcal{X} \to \mathcal{L}(\mathcal{Y})$	Operator-valued kernel
$\mathcal{H}_K$	Vector-valued RKHS associated to K
ev_x	Evaluation map at point x

$\mathcal{M}_{n,m}(\mathbb{R}), \mathcal{M}_n(\mathbb{R})$	Set of real matrices of size $n \times m$; $m := n$
Id_n	Identity matrix of size $n \times n$
$\mathbf{A}_{i:}$	i^{th} row of matrix $\mathbf{A}$
Tr	Trace of operator or matrix
$\mathbf{A}^\top$	Transpose of matrix $\mathbf{A}$
$\|\cdot\|_{\mathrm{op}}$	Operator norm of operator or matrix
$\|\cdot\|_p$	ℓ_p-norm on vectors or functions for $p \in [1, +\infty]$
$\|\cdot\|_{p,q}$	Mixed norm: ℓ_q norm of the ℓ_p norm of the rows of the argument: $\|\mathbf{A}\|_{p,q} = \left\| \left(\|\mathbf{A}_{i:}\|_p \right)_i \right\|_q$.
$\langle \mathbf{A}, \mathbf{B} \rangle_{\mathrm{F}}$	Frobenius inner product of matrix $\mathbf{A}$ and $\mathbf{B}$
$\mathcal{B}_\epsilon^p$	Ball of radius ϵ for the ℓ_p-norm
$\mathcal{B}_\epsilon$	Ball of radius ϵ for the ambient Hilbert norm
$\mathrm{dom}(f)$	Domain of function f
$\Gamma_0(\mathcal{H})$	Proper, convex, lower-semicontinuous functions
$f^\star$	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
∂f	Subdifferential of function f
prox_f	Proximal operator of function f
$\mathrm{Proj}_{\mathcal{C}}$	Orthogonal projection on a closed convex set $\mathcal{C}$
relint	Relative interior
$\mathcal{O}$	Ordo
$\otimes$	Kronecker product of matrices, tensor product of Hilbert spaces or their elements
$\otimes_{m \in [M]} \mathbb{P}_m$	Product measure. When $\mathbb{P} = \mathbb{P}_1 = \ldots = \mathbb{P}_M$, we write $\mathbb{P}^{\otimes M}$.
$\mathbb{1}_S$	Characteristic function of the set S: 1 on S and 0 otherwise
$\|\cdot\|_+$	Positive part: $\|x\|_+ = \max(x, 0)$
$\times_m S_m$	Descartes product of the sets S_m
$\mathrm{Unif}([n])$	Uniform distribution on $[n]$

1

Motivation and Contributions

Contents

Due to the increasing complexity of collected data, which can come from many sensors, embedded devices, IOT, or any other data acquisition pipeline, real world applications are in dire need of machine learning systems able to deal with sophisticated data. Especially, when the collected data correspond to the recording of the behaviour of a phenomenon through time or space for instance, there is an interest for considering the collection of datapoints as observations of a function. One can think for example to data describing the trajectory of a plane through time, the evolution of CO_2 levels in the atmosphere, or the transmission of biochemical signals inside the brain.

These real world scenarii have motivated the research in *functional data analysis* (FDA, Ramsay and Silverman 1997) whose goal is to provide a framework dedicated to data modeled as functions. The potential scope of FDA is enormous and diverse (Ullah and Finch, 2013), and has attracted a great deal of attention (Wang et al., 2016).

In particular, early works focused on *semi-functional regression* where the explanatory variable is a function and the target variable is a scalar. The relationship between these two can then be modeled as linear (Cardot et al., 2003), nonlinear (Cardot et al., 2003) or hybrid (Aneiros-Pérez and Vieu, 2006), and many learning setting were investigated such as quantile regression (Cardot et al., 2005), robust estimation (Crambes et al., 2008; Azzedine et al., 2008) or variable selection (Aneiros et al., 2011).

More recently, the case of *functional output regression* (FOR) where both explanatory and target variables are functions has been under study. The FOR setting is more demanding than the semi-functional one, as it requires to model function-valued functions. The target variable can then belong to an infinite dimensional Hilbert space, a problem that needs to be addressed from a modeling point of view. This fundamental issue has lead to consider linear models (Morris, 2015), non-parametric modeling (Ferraty and Vieu, 2006), or kernel methods (Lian, 2007; Kadri et al., 2010; Ferraty et al.,

2011; Oliva et al., 2015; Kadri et al., 2016; Reimherr et al., 2018). The latter is of particular interest to us, and one goal of this thesis is to propose and study a general framework for function-valued regression beyond the square loss, and in the context of vector-valued RKHSs (Pedrick, 1957).

Apart from functional data, we have noticed that it can be interesting to have predictive models with function-valued outputs even if the output observations do not directly correspond to a function. In particular, in a large number of ML problems, the task at hand depends on some hyperparameter varying continuously and solving the problem for a continuum of values presents many interests. A functional view on these problems can be seen as an extension of *multi-task learning* (MTL; Evgeniou and Pontil 2004), which is the second motivation for this thesis. MTL consists in learning jointly a finite number of machine learning tasks with a single model, the underlying assumption being that solving these tasks together will bring better results than solving them independently. One can think for example to tasks such as *quantile regression* (QR, Koenker and Bassett Jr 1978), anomaly detection using *one class support vector machines* (OCSVM, Schölkopf et al. 2001b) or *cost-sensitive classification* (CSC, Elkan 2001) where the task is characterized by some hyperparameter encoding information about the target function: the quantile level in QR, the fraction of outliers in the OCSVM, or the relative importance associated to false positive and false negatives in CSC. The links between MTL and vector-valued regression are fertile (Álvarez et al., 2012), and MTL can be tackled using kernels for vector-valued functions (Micchelli and Pontil, 2005).

In the context of learning function-valued functions, a natural problem to consider is the joint learning of infinitely many tasks. Such problem has been proposed in the seminal paper (Takeuchi et al., 2013) for tasks whose loss function depends piecewise linearly on the hyperparameter. Extending this framework to a larger class of tasks and models is one of the motivation of this thesis.

The organization of this chapter is as follows: in Section 1.1 we present the type of machine learning tasks tackled in this thesis, and raise related research questions in Section 1.2. Our contributions are summarized in Section 1.3, with associated publications and preprints in Section 1.4.

1.1 Machine Learning Tasks

We introduce below two examples of supervised learning tasks, namely the *binary classification* and *least squares regression*.

Binary classification: The simplest example of a machine learning task is given by binary classification. In this setting, we are given two random variables $(\mathsf{X}, \mathsf{Y}) \in \mathcal{X} \times \{-1, 1\}$. X encodes the representation of the data, or *features* in the space $\mathcal{X}$, and Y is the corresponding *class*. The goal of supervised classification is to build a classifier $\hat{h} \colon \mathcal{X} \to \{-1, 1\}$ which attributes the right class to a realization of the random variable X, without knowing the probability distribution of (X, Y) and being only given a finite sample $(x_i, y_i)_{i=1}^n$ independently drawn from the probability distribution of (X, Y), called the training sample (Vapnik, 1999).

Theoretically, the ideal classifier involves the optimization problem

$$h^\dagger \in \operatorname*{arg\,min}_{h \text{ measurable}} \mathbb{E}_{(\mathsf{X},\mathsf{Y})} \left[\mathbb{1}_{\{-\mathsf{Y}\}}(h(\mathsf{X})) \right]. \tag{1.1}$$

The loss is called the $0-1$ *loss* as it returns 0 whenever the prediction is correct and 1 when it fails. The best decision rule is given by the *Bayes classifier*

$$h^\dagger(x) := \begin{cases} 1 & \text{if } \mathbb{P}(\mathsf{Y}=1 \,|\, \mathsf{X}=x) \geq \frac{1}{2} \\ 0 & \text{otherwise.} \end{cases}$$

However, given that the underlying distribution of (X, Y) is unknown, this rule cannot be applied in practice and one must trade Equation (1.1) against an empirical version based on the available training set $(x_i, y_i)_{i=1}^n$. Moreover, minimizing the $0-1$-loss is notoriously hard (can be NP-hard for many choices of $\mathcal{H}$, Ben-David et al. 2003), so that a convex upper bound such as the hinge loss can be used instead. Combined with a *margin based* approach (Vapnik, 1998), this results in

$$\hat{h} \in \operatorname*{arg\,min}_{h \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} \max\left(0, 1 - y_i h(x_i)\right) + \Omega(h),$$

where $\mathcal{H}$ is a set of possible candidates denoted *hypothesis space*, and $\Omega(h)$ is a *regularizer* used to avoid overfitting. This leads to the celebrated *support vector machines* (SVM) problem formulation and its separating hyperplane interpretation when using kernel models (Cortes and Vapnik, 1995).

Least squares regression: When the random variable Y is not binary but real-valued, inferring Y from X is called a *regression task*, the most well-known loss function for regression being the square loss. The goal is then to build a prediction function $\hat{h}\colon \mathcal{X} \to \mathbb{R}$ such that the residual $\mathsf{Y} - h^\dagger(\mathsf{X})$ is the smallest possible in expected square norm, having only access to a training dataset $(x_i, y_i)_{i=1}^n$. Minimizing the square loss has the benefit of estimating the conditional expectation of Y given X, as:

$$\mathbb{E}[\mathsf{Y}|\mathsf{X}] = h^\dagger(\mathsf{X}), \qquad\qquad h^\dagger \in \operatorname*{arg\,min}_{h \text{ measurable}} \mathbb{E}_{(\mathsf{X},\mathsf{Y})}\left[(\mathsf{Y} - h(\mathsf{X}))^2\right]$$

The theoretical best prediction at point x is achieved by $h^\dagger(x) = \mathbb{E}_{\mathsf{Y}|\mathsf{X}=x}\left[\mathsf{Y}\right]$. Again, this predictor is of little convenience in practice as one does not have access to the underlying distribution of (X, Y) and empirical versions based on the training dataset $(x_i, y_i)_{i=1}^n$ must be considered, such as the regularized empirical risk minimization problem

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

where $\mathcal{H}$ is hypothesis space and $\Omega(h)$ is a regularizer.

These two simple examples fall under the *regularized empirical risk minimization* umbrella (Tikhonov and Arsenin, 1977; Girosi et al., 1995), which consists in solving

$$\hat{h} \in \operatorname*{arg\,min}_{h \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} \ell(h(x_i), y_i) + \Omega(h),$$

where ℓ is a loss function used to measure the discrepancy between a prediction $h(x_i)$ and the ground truth y_i, and $\Omega(h)$ is a regularizer. Ideally, an estimator $\hat{h}$ should perform similarly on the *training dataset* $(x_i, y_i)_{i=1}^n$ and on new data sampled from

$\mathbb{P}_{(\mathsf{X},\mathsf{Y})}$. The *generalization capabilities* of the estimator can be controlled by bounding

$$\left| \mathbb{E}_{(\mathsf{X},\mathsf{Y})} \left[\ell(\hat{h}(\mathsf{X}), \mathsf{Y}) \right] - \frac{1}{n} \sum_{i=1}^{n} \ell(\hat{h}(x_i), y_i) \right| \tag{1.2}$$

in high probability (Vapnik, 1998).

1.1.1 Parameterized Tasks

Parameterized tasks arise when the target function to estimate depends on a parameter that impacts the task. We give below three examples of parameterized tasks.

Cost-sensitive classification: In the binary classification scenario, it may happen that the cost associated to making a mistake on one or the other class differs in real applications. For example in epidemiology, when designing a test to assess whether a person is contaminated by a virus or not, a false positive is of less impact than a false negative as the latter has dramatic consequence on the spread of the virus. It can then be of advantage to introduce an imbalanced coefficient that penalizes differently the two kinds of mistakes (Bach et al., 2006). A parameter $\theta \in [-1, 1]$ is capable of encoding such asymmetry, and gives rise to the task

$$\hat{h} \in \arg\min_{h \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} \left| \frac{\theta + 1}{2} - \mathbb{1}_{\{-1\}}(y_i) \right| \max(0, 1 - y_i h(x_i)) + \Omega(h).$$

This way when θ is close to 1, mistakes made on ground truth samples with class -1 are almost not penalized, so that the classifier focuses on being right on the class 1, and vice versa.

Quantile regression: In the regression setting, the conditional expectation may not suffice to give sufficiently good prediction of the output variable. One could think for example to data distributions where for all $x \in \mathcal{X}$, $\mathbb{P}_{\mathsf{Y}|\mathsf{X}=x}$ is bimodal and symmetric around 0, so that the conditional expectation $\mathbb{E}_{\mathsf{Y}|\mathsf{X}=x}[\mathsf{Y}] = 0$ would be a poor choice of prediction as it would not be meaningful with respect to the underlying data distribution. In such cases, one can resort to estimating the conditional quantiles of Y given X. Given a quantile level $\theta \in (0, 1)$, it is defined as

$$q(x) := \inf\{u \in \mathbb{R} \,:\, \mathbb{P}(\mathsf{Y} \leq u \,|\, \mathsf{X} = x) \geq \theta\}.$$

Estimating the conditional quantile can be tackled through its variational formulation

$$q(x) = \arg\min_{u \in \mathbb{R}} \mathbb{E}_{\mathsf{Y}}[\ell(\theta, u, \mathsf{Y}) \,|\, \mathsf{X} = x],$$

where ℓ is the so-called pinball loss (Koenker and Bassett Jr, 1978)

$$\ell(\theta, u, y) = \max(\theta(y - u), (\theta - 1)(y - u)).$$

ERM based algorithms then allow to estimate q in various function spaces (Steinwart et al., 2011).

Density level set estimation: In anomaly detection, the goal is to separate *inliers*, which are coherent observations, from *outliers* which are seen as anomalies in the data. Given an $\mathcal{X}$-valued random variable X, one aims at building a prediction function $\hat{h}\colon \mathcal{X} \to \{-1, 1\}$ used for deciding whether a new point is an inlier or an outlier. Given a desired fraction of outliers $\theta \in (0, 1)$, a natural way to achieve this goal is to estimate the *minimum volume sets* (MV-sets) associated to X (Polonik, 1997):

$$\mathrm{MV}(\theta) = \underset{G \subseteq \mathcal{X},\, G \text{ Borel}}{\arg\min} \left\{ \Lambda(G) \,|\, \mathbb{P}_{\mathsf{X}}(G) \geq \theta \right\},$$

where $\Lambda(G)$ is the Lebesgue measure of a Borel set G. MV-sets describe regions where X is most concentrated, and relate closely to the level sets of the density, in the sense that they describe the same sets but with a different parameterization (the level of the density versus the mass of the set with respect to $\mathbb{P}_{\mathsf{X}}$). Outlierness of a point x can then be characterized by its belonging to $\mathrm{MV}(\theta)$. In particular, MV-sets can be estimated using *one-class SVMs* (Schölkopf et al., 2001b).

1.1.2 Learning Vector-Valued Functions

While aforementioned tasks focus on predicting a single value, complex machine learning systems often require to predict jointly several values. This scenario typically occurs in the regression setting when the output variable Y is $\mathbb{R}^p$-valued, or when when one's goal is to solve jointly multiple tasks (Micchelli and Pontil, 2005). The latter is referred to as *multi-task learning* (MTL) and has attracted a great deal of attention in machine learning (Evgeniou et al., 2005) with far-reaching applications such as autonomous driving, climate science, or functional brain imaging.

In its most generic form, MTL proposes to solve jointly p tasks, where each task is encoded by a particular objective function, and observed dataset. Having access to p datasets $(X_j)_{j=1}^p$ where each $X_j := (x_{i,j}, y_{i,j})_{i=1}^{n_j} \in (\mathcal{X}_j \times \mathbb{R})^{n_j}$, the objective is then to build p estimators $(\hat{h}_j)_{j=1}^p$ such that each $\hat{h}_j\colon \mathcal{X}_j \to \mathbb{R}$ performs well on task j.

When the dataset is shared across the tasks, this simplifies to finding a vector-valued model $\hat{h}\colon \mathcal{X} \to \mathbb{R}^p$ such that each coordinate of $\hat{h}$ performs well on the corresponding task (Micchelli and Pontil, 2005). This simpler setting is also referred to as *multi-output learning* (Álvarez et al., 2012) and is the setting considered in this thesis.

Compared to independent learning of each task, the advantage of MTL is to take advantage of the similarity between the tasks to ensure consistency between them. For example in quantile regression, the quantiles are by definition nondecreasing with respect to θ, and that should be reflected in the estimator that jointly estimates the conditional quantiles associated to some $(\theta_j)_{j=1}^p$.

A simple way to jointly solve p tasks parameterized by $(\theta_j)_{j=1}^p$ and loss functions $(\ell(\theta_j, \cdot, \cdot))_{j=1}^p$ is then to sum the corresponding loss functions:

$$\hat{h} \in \underset{h \in \mathcal{H}}{\arg\min} \; \frac{1}{np} \sum_{i=1}^{n} \sum_{j=1}^{p} \ell(\theta_j, h(x_i)_j, y_i) + \Omega(h),$$

where $\mathcal{H} \subset \mathcal{F}(\mathcal{X}, \mathbb{R}^p)$ is a hypothesis space and Ω is some regularization term encoding similarities between tasks (Argyriou et al., 2008b,a; Baldassarre et al., 2012).

Relevant application-specific references include (Sangnier et al., 2017) for quantile regression and (Glazer et al., 2013) for density level set estimation. We finally want to mention alternatives to summing the loss functions such as *Pareto multi-task learning* which make use of multiobjective optimization (Lin et al., 2019).

1.1.3 Learning Function-Valued Functions

Learning function-valued functions is the goal of this thesis. Such scenario can be considered as an extension of the prior vector-valued case beyond Euclidean space $\mathbb{R}^p$ to generalized Hilbert spaces.

Functional output regression: The problem of learning function-valued function can appear in supervised learning when the output variable Y is a function itself. This happens for example in biomedical signal processing, epidemiology monitoring or climate science where the phenomena under study exhibit a functional nature, and the understanding of these phenomena depends on machine learning algorithms being able to reliably predict functional data (Ramsay and Silverman, 2007).

In this problem family the task is to regress to a functional output from a vectorial or functional input, a setting referred to as *functional output regression* (FOR). Assuming that Y takes values in a functional space $L^2[\Theta, \mu]$, where Θ is the domain of the realizations of Y and μ a probability measure on it, a natural way to tackle this problem is to estimate $\mathbb{E}[\mathsf{Y}|\mathsf{X}] = h^\dagger(\mathsf{X})$ where it is well-known that

$$h^\dagger = \operatorname*{arg\,min}_{h \in \mathcal{H}} \mathbb{E}_{(\mathsf{X},\mathsf{Y})} \left[\left\| \mathsf{Y} - h(\mathsf{X}) \right\|^2_{L^2[\Theta,\mu]} \right],$$

with $\mathcal{H}$ being the set of all measurable functions from $\mathcal{X}$ to $L^2[\Theta, \mu]$. As the underlying distribution is unknown, one can use regularized empirical risk minimization in suitable functional spaces to solve this problem, as proposed in the seminal works of Kadri et al. (2010, 2016).

Jointly learning infinitely many tasks: The problem of learning function-valued functions also emerges when considering the joint learning of infinitely many parameterized tasks. The goal is then to learn a mapping

$$\hat{h} \colon x \mapsto (\theta \mapsto \hat{h}(x)(\theta) \in \mathbb{R})$$

such that for all $\theta \in \Theta$, $\hat{h}(\cdot)(\theta)$ performs well on the task θ. Here Θ is a space encoding the tasks, such as the quantile level in quantile regression ($\Theta = (0, 1)$), the mass of the MV-sets in anomaly detection ($\Theta = (0, 1)$) or the asymmetry coefficient in cost-sensitive classification ($\Theta = [-1, 1]$).

The resulting optimization problem is

$$\hat{h} \in \operatorname*{arg\,min}_{h \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} \int_\Theta \ell(\theta, h(x_i)(\theta), y_i) \mathrm{d}\mu(\theta) + \Omega(h),$$

where μ is a probability measure describing the relative importance of each task, $\mathcal{H} \subset \mathcal{F}(\mathcal{X}, \mathcal{F}(\Theta, \mathbb{R}))$ is a suitable hypothesis space and $\Omega(h)$ is a regularizer. In their seminal work, Takeuchi et al. (2013) introduce the *parametric task learning* framework able to handle a continuum of tasks when the loss function has a piecewise linear dependency to

the θ parameter. By leveraging *solution path* techniques, they show that the associated optimization problem is tractable and yields a model whose dependency to θ is also piecewise linear.

We can notice that both problem families involve a class of loss functions that we denote *integral losses*, as the discrepancy between the function-valued prediction $h(x)$ and the observed output y is computed by means of an integration of the local loss functions $\ell(\theta, \cdot, \cdot)$ over the Θ space.

1.2 Research Questions

The goal of this thesis is to propose a general framework to learn function-valued functions, both to offer novel functional output regression tools and to extend multi-task learning to a continuum of tasks. The outputs are supposed to be functions over a domain Θ, and we consider a family of losses that write as an integral over this domain. This leads to the formulation of the corresponding regularized empirical risk minimization problems, with the additional requirement of picking a suitable hypothesis space. This hypothesis space is chosen to be a *vector-valued reproducible kernel Hilbert space* (vv-RKHS), a generalization of RKHSs allowing function-valued models whose full introduction is postponed to Chapter 2. Solving these optimization problems is not straightforward, and raises various challenging questions:

- How can we solve problems involving integral losses while they are not computable analytically?
- How do we represent the models on a computer despite the vv-RKHS being an infinite-dimensional functional space?
- What compromises can be accepted on the hypothesis space to scale to larger datasets?

For the specific functional output regression problem, existing techniques in the literature mostly focus on the square loss as a measure of the discrepancy between predictions and ground truth, which is known to be sensitive to outliers in the data. Especially, in a learning scenario where the data is contaminated with erroneous values, square loss based estimators tend to perform poorly. We suggest the following lines of research:

- How can we go beyond the square loss in functional output regression? Can we take advantage of dual approaches to allow sparse or robust estimators?

In Chapter 5 we develop a learning framework able to jointly learn infinitely many parameterized tasks, referred to as *infinite task learning* (ITL). Interesting questions then include:

- Which type of regularization can be used? How can we induce consistency between the tasks?
- What kind of generalization guarantees can we get for the ITL estimator?
- Can we design kernels and appropriate optimization algorithms suited to handle complex structured data such as images?

- Can we extend the ITL framework to more involved learning tasks with higher dimensional Θ?

1.3 Contributions

We now list the contributions of this thesis to the aforementioned problems, followed by the description of the organization of the manuscript. In particular, Chapter 2 is not included in these contributions as it is a background chapter on convex optimization and (operator-valued) kernel methods used throughout the manuscript.

- Chapter 3 develops optimization algorithms to solve regularized empirical risk minimization problems in vector-valued RKHSs, in the presence of integral losses. Closed-form solutions are obtained for the square loss based estimators in the fully and partially observed regime, extending known results to vectorial outputs. Primal optimization algorithms are proposed for general loss functions, relying either on some sampling of the integral loss that pertains the solution to a finite-dimensional subspace of the vv-RKHS, or on stochastic optimization algorithms made possible by the use of *random Fourier features*. Finally, dual methods are explored, with *compatibility conditions* between the loss and the vv-RKHS allowing for different representation of the solution and dedicated (proximal) gradient descent algorithms.
- Chapter 4 investigates the functional output regression problem with a focus on robustness and sparsity. To ensure these properties, new loss functions are designed with building blocks being integral losses and infimal convolutions. In particular, these losses extend the classical Huber and ϵ-insensitive losses to the functional output case. The resulting optimization problems are solved using dual methods, shown to be well-suited to the nature of the new losses and enabling proximal gradient descent algorithms.
- Chapter 5 introduces the *infinite task learning* framework, able to jointly learn a continuum of parameterized tasks and exemplified in quantile regression, cost-sensitive classification and density level set estimation. Different choices of regularizers are proposed, corresponding to regularization in vv-RKHS norm or in mixed L^2-RKHS norm, for which dedicated *double representer theorems* are stated. We analyse the generalization capabilities of the ITL estimator for the supervised setting using *uniform stability*. Finally, we show that *hybrid models* involving kernels and dedicated neural networks allow one to obtain more accurate predictions in the context of image processing.
- Chapter 6 casts a style transfer problem as an ITL problem with vectorial outputs. We consider the problem of emotion transfer for facial landmarks, for which a resolution in vv-RKHS is derived. We show that encoding emotions continuously in some embedding space makes sense for this problem and achieves good performance on two facial datasets, with the additional benefit over existing method of enjoying predictions over unseen emotions.

The algorithms developed in this thesis are gathered in the open source Python library torch_itl. It is designed as a high level python library exploiting pytorch's autodifferentiation capabilities and providing a scikit-like api. For now it supports quantile regres-

sion and vectorial infinite task learning (applied in the context of emotion transfer), but future release will include the rest of the proposed methods to enhance reproducibility.

1.4 Publications

These contributions have resulted in the following peer-reviewed publications and preprints ($\star$ indicates equal contribution)

- R. Brault*, **A. Lambert***, Z. Szabó, M. Sangnier and F. d'Alché-Buc. Infinite Task Learning in RKHSs. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 1294–1302, 2019.
- **A. Lambert***, R. Brault*, Z. Szabó, M. Sangnier and F. d'Alché-Buc. A Functional Extension of Multi-Output Learning In *International Conference on Machine Learning (ICML): Adaptive & Multitask Learning workshop (AMTL)*, 2019.
- 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)*, pages 5598-5607, 2020.
- **A. Lambert***, S. Parekh*, Z. Szabó, and F. d'Alché-Buc. Emotion Transfer Using Vector-Valued Infinite Task Learning *Technical report*, 2021. (https://arxiv.org/abs/2102.05075). (under review)

2

Background on Learning with Operator-Valued Kernels

"Kernels are dead! Long live kernels!"

Contents

In this chapter, we present several mathematical tools that will be used throughout the manuscript. In Section 2.1 we start with a brief summary on optimization of convex functionals in general Hilbert spaces, including Fenchel-Legendre conjugates, parametric duality and proximal operators. In Section 2.2 we focus on operator-valued kernels (OVKs) for machine learning. These classes give rise to expressive spaces of functions called vector-valued reproducible kernel Hilbert spaces (vv-RKHSs), allowing to model function-valued functions and chosen as hypothesis spaces in all further applications.

2.1 Convex Optimization Tools

Ultimately, machine learning consists in the minimization of a function, in the sense that the output of a learning algorithm is the solution of a minimization problem. Among these problems, convex ones are notoriously easier to deal with than nonconvex ones. While nonconvex optimization has to deal with the existence of potentially different local minima, convex optimization problems enjoy the property that any local minimum is also a global minimum. In this thesis, we are concerned with convex optimization problems, over a potentially infinite-dimensional Hilbert space denoted by $\mathcal{H}$. We refer to the monographs by *e.g.* Rockafellar (1970); Boyd et al. (2004); Bauschke et al. (2011), and adopt the notations and terminology of the latter as the optimization framework they design is suited to minimization problems over general Hilbert spaces. We begin by defining a class of functions ubiquitous in convex optimization.

Definition 2.1 (Proper, convex, lower semi-continuous functions)**.** *We denote by* $\Gamma_0(\mathcal{H})$ *the set of functions* $f : \mathcal{H} \to]-\infty, +\infty]$ *that are*

Table 2.1 – Useful Fenchel-Legendre conjugates, for any $f, g\colon \mathcal{H} \to \left]-\infty, +\infty\right]$.

Function	Fenchel-Legendre conjugate
$\frac{1}{2}\left\|\cdot\right\|_{\mathcal{H}}^2$	$\frac{1}{2}\left\|\cdot\right\|_{\mathcal{H}}^2$
$\left\|\cdot\right\|_p$	$\chi_{\mathcal{B}_1^{p^\star}}$ where $\frac{1}{p} + \frac{1}{p^\star} = 1$
ϵf	$\epsilon f^\star(\frac{\cdot}{\epsilon})$ for all $\epsilon > 0$
$f(\cdot - y)$	$f^\star + \langle \cdot, y\rangle_{\mathcal{H}}$ for all $y \in \mathcal{H}$
$f \,\square\, g$	$f^\star + g^\star$

- proper*:* $\operatorname{dom}(f) := \left\{x \in \mathcal{H} \,:\, f(x) < +\infty\right\} \neq \varnothing$*,*
- convex*:* $\forall x, y \in \mathcal{H}, \forall t \in [0,1], f(tx + (1-t)y) \leq tf(x) + (1-t)f(y)$*, and*
- lower semicontinuous*:* $\forall x \in \mathcal{H}, \underline{\lim}_{y \to x} f(y) \geq f(x)$*, where* $\underline{\lim}$ *denotes limit inferior.*

The Fenchel-Legendre conjugate of a function is an important notion for exploiting duality principles. In particular it plays an important role in convex machine learning as it is a key tool to solve variational problems.

Definition 2.2. *The* Fenchel-Legendre conjugate *of a function* $f\colon \mathcal{H} \to \left]-\infty, +\infty\right]$ *is defined by*

$$\forall x \in \mathcal{H}, \quad f^\star(x) := \sup_{y \in \mathcal{H}} \langle x, y\rangle_{\mathcal{H}} - f(y). \tag{2.1}$$

The Fenchel-Legendre conjugate of a function f is always convex. It is also involutive on Γ_0, meaning that $(f^\star)^\star = f$ for any $f \in \Gamma_0(\mathcal{H})$. We gather in Table 2.1 examples and properties of Fenchel-Legendre conjugates.

We now introduce the infimal convolution operator following Bauschke et al. (2011).

Definition 2.3 (Infimal Convolution)**.** *The* infimal convolution *of two functions* $f, g\colon \mathcal{H} \to \left]-\infty, +\infty\right]$ *is*

$$f \,\square\, g\colon \begin{pmatrix} \mathcal{H} \to [-\infty, +\infty] \\ x \mapsto \inf\limits_{x' \in \mathcal{H}} f(x - x') + g(x') \end{pmatrix}. \tag{2.2}$$

The infimal convolution operator is commutative, which means that $f \,\square\, g = g \,\square\, f$ for all $f, g\colon \mathcal{H} \to]-\infty, +\infty]$. It allows to define a smooth approximation of a potentially non-smooth function f through the *Moreau enveloppe* defined as $f \,\square\, \frac{1}{2\gamma}\left\|\cdot\right\|_{\mathcal{H}}^2$ where $\gamma > 0$. One key property of the infimal convolution operator is that it behaves nicely under Fenchel-Legendre conjugation, as detailed in the following proposition.

Proposition 2.4 (Bauschke et al. (2011), proposition 13.24)**.** *Let* $f, g\colon \mathcal{H} \to \left]-\infty, +\infty\right]$*. Then*

$$(f \,\square\, g)^\star = f^\star + g^\star.$$

We now define the proximal operator, used as a replacement for the classical gradient step in the presence of non-differentiable objective functions.

Definition 2.5 (Proximal Operator, Moreau (1965))**.** *The* proximal operator *(or proximal map) is defined as*

$$\forall (f, x) \in \Gamma_0 \times \mathcal{H}, \quad \operatorname{prox}_f(x) := \operatorname*{arg\,min}_{y \in \mathcal{H}} \; f(y) + \frac{1}{2} \|x - y\|_{\mathcal{H}}^2 . \tag{2.3}$$

One advantage of working with functions in Γ_0 is that the proximal operator is always well-defined. Its computation is efficient for many practical losses thanks to the following proposition.

Proposition 2.6 (Moreau decomposition)**.** *Let $f \in \Gamma_0(\mathcal{H})$ and $\gamma > 0$. Then*

$$\mathrm{Id} = \operatorname{prox}_{\gamma f}(\cdot) + \gamma \operatorname{prox}_{f^\star/\gamma}(\cdot/\gamma). \tag{2.4}$$

Example 2.7. *A particular case of interest is the proximal operator associated to the indicator function of a closed convex set $\mathcal{C} \subset \mathcal{H}$. It holds that for all $\gamma > 0$,*

$$\forall h \in \mathcal{H}, \; \operatorname{prox}_{\gamma \chi_{\mathcal{C}}(\cdot)}(h) = \mathrm{Proj}_{\mathcal{C}}(h)$$

where $\mathrm{Proj}_{\mathcal{C}}$ is the orthogonal projection on $\mathcal{C}$ in $\mathcal{H}$.

The proximal operator allows to solve *composite problems* of the form

$$\inf_{x \in \mathcal{H}} \; f(x) + g(x), \tag{2.5}$$

where $f, g \in \Gamma_0(\mathcal{H})$ are such that f is *Gâteaux differentiable* with C-Lipschitz continuous gradient for some $C > 0$, while g is not.

Definition 2.8 (Gâteaux differentiability)**.** *Let $f \colon \mathcal{H} \to \left]-\infty, +\infty\right]$ be proper and $x \in \operatorname{dom}(f)$. Let $y \in \mathcal{H}$. The* directional derivative *of f in the direction y is*

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

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

$$\forall y \in \mathcal{H}, \; f'(x, y) = \langle \nabla f(x), y \rangle_{\mathcal{H}}.$$

While in the finite dimensional case, existence of $\mathcal{C}^1$ partial derivatives ensures differentiablity of the function and existence of a gradient, this is no longer true in infinite dimension. Gâteaux differentiablity then helps to fill this void, and paves the way to first order optimization methods such as the *proximal gradient descent* algorithm, presented in Algorithm 2.1.

The subdifferential is a fundamental tool in convex analysis that generalizes the gradient to nonsmooth functions.

Definition 2.9 (Subdifferential)**.** *Let $f : \mathcal{H} \to \left]-\infty, +\infty\right]$. The* subdifferential *of f at $x \in \operatorname{dom}(f)$ is:*

$$\partial f(x) := \{u \in \mathbb{R}^d \, : \, \forall y \in \mathcal{H}, f(y) \geq f(x) + \langle u, y - x \rangle_{\mathcal{H}}\} \;, \tag{2.6}$$

i.e. *the set of "slopes" of all affine minorants of f which are exact at x.*

Algorithm 2.1 PROXIMAL GRADIENT DESCENT FOR PROBLEM 2.5

input : Lipschitz constant C, iteration number T
init : $x^{(0)}$
for $t = 1, \dots, T$ **do**
  $x^{(t)} = \operatorname{prox}_{\frac{1}{C}g}\left(x^{(t-1)} - \frac{1}{C}\nabla f(x^{(t-1)})\right)$
return $x^{(T)}$

If f is a convex Gâteaux differentiable function, the subdifferential at $x \in \operatorname{dom} f$ has only one element: $\nabla f(x)$. Subdifferentiability allows to generalize first-order optimality conditions to non-differentiable convex functions.

Proposition 2.10 (Fermat's rule)**.** *Let $f \in \Gamma_0(\mathcal{H})$. Then, for all $\hat{x} \in \mathcal{H}$:*

$$\hat{x} \in \operatorname*{arg\,min}_{x \in \mathcal{H}} f(x) \Leftrightarrow 0 \in \partial f(\hat{x}). \tag{2.7}$$

We are now ready to state the strong duality theorem in the context of minimization under affine equality constraints, which will be used when exploiting duality in vv-RKHSs.

Definition 2.11 (Lagrangian)**.** *Let $f \in \Gamma_0(\mathcal{H})$, $\mathcal{K}$ be a Hilbert space, $L \in \mathcal{L}(\mathcal{H}, \mathcal{K})$ and $b \in \mathcal{K}$. We are interested in solving*

$$\inf_{x \in \mathcal{H}} \underbrace{f(x) + \chi_{\{0\}}(Lx - b)}_{:= \mathcal{P}(x)} \tag{2.8}$$

that we refer to as the primal problem*. The Lagrangian associated to Problem 2.8 is*

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

We then refer to

$$\sup_{\alpha \in \mathcal{K}} \underbrace{\inf_{x \in \mathcal{H}} \mathcal{L}(x, \alpha)}_{:= \mathcal{D}(\alpha)}$$

as the dual problem*. Moreover, (x, α) is a* saddle point *if*

$$\mathcal{P}(x) = \mathcal{L}(x, \alpha) = \mathcal{D}(\alpha).$$

We can notice that $\mathcal{P}(x) = \sup_{\alpha \in \mathcal{K}} \mathcal{L}(x, \alpha)$ for $\forall x \in \mathcal{H}$. Denoting by $\mathcal{P}^*$ and $\mathcal{D}^*$ the respective optimal values of the primal and dual problems, *weak duality* ensures that $\mathcal{P}^* \geq \mathcal{D}^*$. It turns out that for a large class of problems, both values are equal, a setting refered to as *strong duality*.

Proposition 2.12 (Strong duality)**.** *Assume that there exist $x \in \operatorname{relint} \operatorname{dom} f$ such that $Lx - b = 0$. Then* strong duality *holds* i.e. *$\mathcal{P}^* = \mathcal{D}^*$.*

2.2 Kernel Methods for Machine Learning

In this section, we successively present a construction of RKHSs (Section 2.2.1) and vv-RKHSs (Section 2.2.2) and highlight their usefulness in machine learning.

2.2.1 Scalar-valued Kernels and RKHSs

Kernel methods stand as a cornerstone of machine learning. Not only they have played a preponderant role in extending linear models to nonlinear models but they come with a rigorous mathematical framework that eases the analysis of the learning algorithm (see Schölkopf and Smola (2002); Berlinet and Thomas-Agnan (2004); Steinwart and Christmann (2008) for in depth presentation of kernel methods for machine learning). Their development is strongly linked with convex optimization as practical algorithms make an extensive use of duality principles. They offer powerful modeling spaces, called reproducing kernel Hilbert spaces (RKHSs), which are Hilbert spaces of functions that enjoy a regularity property: convergence in the space implies pointwise convergence. Equivalently, this can be summarized by the requirement of continuity for all functions evaluations.

Definition 2.13. *Let $\mathcal{X}$ be any set and $\mathcal{H} \subset \mathcal{F}(\mathcal{X}, \mathbb{R})$ be a Hilbert space. $\mathcal{H}$ is said to be a* reproducing kernel Hilbert space *(RKHS) if and only if for all $x \in \mathcal{X}$, the following evaluation mapping at x is continuous.*

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

Definition 2.13 is of limited practical interest. Indeed, we seldom have access first to a RKHS $\mathcal{H}$, but rather construct it through the notion of *reproducing kernel* as presented below.

Definition 2.14. *A (scalar-valued) kernel on $\mathcal{X}$ is a function $k \colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ such that*

1. $\forall (x, z) \in \mathcal{X}^2, \quad k(x, z) = k(z, x)$, *and*
2. $\forall n \in \mathbb{N}^*, \ (x_i)_{i=1}^n \in \mathcal{X}^n, \ (\alpha_i)_{i=1}^n \in \mathbb{R}^n, \quad \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j k(x_i, x_j) \geq 0.$

Remark 2.15. *Throughout this thesis we consider kernels meant in the sense of positive definite functions while noting that the construction can be extended to indefinite kernels leading to Krein spaces (Ong et al., 2004; Huang et al., 2017).*

In the rest of this manuscript, we may with a slight abuse simply refer as kernel any scalar-valued kernel. Kernels can be constructed on sets $\mathcal{X}$ with very different structures including for instance permutations (Jiao and Vert, 2016), measures (Cuturi et al., 2005), graphs (Mahé and Vert, 2009), or time series (Cuturi et al., 2007). In case of $\mathcal{X} = \mathbb{R}^d$, we can mention the celebrated Gaussian kernel which reads

$$\forall (x, z) \in \mathbb{R}^2, \quad k(x, z) = e^{-\gamma \|x - z\|_2^2}, \quad (\gamma > 0).$$

The following theorem originating from (Aronszajn, 1950) states an equivalence between kernels and the existence of a *feature map* representation.

Theorem 2.16. *A function $k \colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a kernel if and only if there exists a Hilbert space $\mathcal{V}$ and a mapping $\phi \colon \mathcal{V} \to \mathcal{H}$ such that*

$$\forall (x, z) \in \mathcal{X}^2, \quad k(x, z) = \langle \phi(x), \phi(z) \rangle_{\mathcal{V}}. \tag{2.9}$$

The space $\mathcal{V}$ is then called a feature space *and the mapping ϕ a* feature map.

The following theorem links kernels to RKHSs, saying that any kernel can be associated to a unique RKHS, for that reason kernels are sometimes called *reproducing* kernels.

Theorem 2.17. *Let k be a kernel on $\mathcal{X}$. Then there exists a unique Hilbert space of functions $\mathcal{H}_k \subset \mathcal{F}(\mathcal{X}, \mathbb{R})$ such that*

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

The mapping $x \mapsto k(\cdot, x)$ is called the canonical feature map, with $\mathcal{V}$ in Equation (2.9) being $\mathcal{H}_k$ itself. If one gets a RKHS $\mathcal{H}$ from Definition 2.13, then the existence for all $x \in \mathcal{X}$ of a function $k(\cdot, x) \in \mathcal{H}$ stems from Riesz representation theorem, and taking $x \mapsto k(\cdot, x)$ as a feature map in Equation (2.9) ensures that $\mathcal{H}$ is the RKHS associated to the kernel k.

The above properties allow to define RKHSs on spaces $\mathcal{X}$ without any structural assumption. It turns out that endowing the input space with a metric structure and a probability measure leads to fertile connections with functional analysis and L^2 spaces.

Integral operators and RKHSs We begin by introducing the notion of *integral operator* associated to a kernel on a compact space Θ.

Definition 2.18 (Integral Operator)**.** *Let Θ be a compact metric space endowed with a Borel probability measure μ, and k be a continuous kernel on Θ. The integral operator $T_{k,\mu}$ is defined as*

$$T_{k,\mu} \colon \begin{pmatrix} L^2[\Theta, \mu] & \to & L^2[\Theta, \mu] \\ f & \mapsto & \left(\theta \mapsto \int_\Theta f(\theta') k(\theta, \theta') \mathrm{d}\mu(\theta')\right) \end{pmatrix}.$$

Remark 2.19. *When there is no ambiguity on the considered measure, we omit the dependence in μ and abbreviate $T_{k,\mu}$ as T_k.*

The continuity of k, combined with the compactness of Θ ensure that k is bounded on Θ, and thus T_k is continuous and compact. Moreover, the symmetry of k implies that T_k is self-adjoint. Finally, it turns out that T_k is positive (*i.e.* $\langle T_k f, f \rangle_{L^2[\Theta,\mu]} \geq 0$ for $\forall f \in L^2[\Theta, \mu]$). The spectral theorem for self-adjoint compact operators guarantees the existence of an at most countable family of functions $(\psi_j)_{j \in J}$ forming an orthonormal system in $L^2[\Theta, \mu]$ such that

$$\forall f \in L^2[\Theta, \mu], \quad T_k f = \sum_{j \in J} \lambda_j \langle f, \psi_j \rangle \psi_j, \tag{2.10}$$

where $(\lambda_j)_{j \in J}$ are the family of positive eigenvalues ordered so that $\lambda_1 \geq \lambda_2 \geq \cdots \geq 0$. Additionally, the eigenvalues $(\psi_j)_{j \in J}$ can be chosen to be continuous since T_k maps $L^2[\Theta, \mu]$ into the space $\mathcal{C}(\Theta, \mathbb{R})$ of continuous functions. This provides a *Mercer representation* of the kernel k, as stated below.

Theorem 2.20 (Mercer)**.** *Let Θ be a compact metric space endowed with a Borel probability measure μ such that* $\mathrm{supp}(\mu) = \Theta$*, and k be a continuous kernel on Θ. Let $(\lambda_j, \psi_j)_{j \in J}$ be the eigendecomposition of T_k with continuous eigenvectors. Then*

$$\forall (\theta, \theta') \in \Theta^2, \quad k(\theta, \theta') = \sum_{j \in J} \lambda_j \psi_j(\theta) \psi_j(\theta') \tag{2.11}$$

where the convergence is uniform on Θ^2 *and absolute for all* $(\theta, \theta') \in \Theta^2$.

Remark 2.21. *The compact assumption on* Θ *can be weakened to "*$\mathcal{H}_k$ *is compactly embedded in* $L^2[\Theta, \mu]$*" while still retaining most of Mercer's theorem, see Steinwart and Scovel (2012) for more details.*

While the eigendecomposition of T_k depends on μ, Equation (2.11) holds nevertheless. When $\text{supp}(\mu) = \Theta$, a nice byproduct of Mercer's theorem is to provide a description of $\mathcal{H}_k$ in terms of the decay rate of the Fourier coefficients in the basis $(\psi_j)_{j\in J}$. Indeed, it holds that

$$\mathcal{H}_k = \left\{ f \in L^2[\Theta, \mu] \,:\, \sum_{j\in J} \frac{\left(\langle f, \psi_j \rangle_{L^2[\Theta,\mu]}\right)^2}{\lambda_j} < +\infty \right\} \tag{2.12}$$

so that $h \in \mathcal{H}_k$ if and only if there exist $f \in L^2[\Theta, \mu]$ such that $T_k f = h$.

Exhibiting the exact eigendecomposition of T_k is a notoriously hard century-old problem for general k and μ (Stone, 1932; Klus et al., 2020). In some cases it can be carried out by solving a differential equation associated to the eigenvector problem, but this method can only be applied on an *ad hoc* basis.

Example 2.22 (Laplacian kernel eigendecomposition, Kadri et al. (2016))**.** *Let* $\Theta = [0, 1]$ *and* μ *be the Lebesgue measure. Let* $k(\theta, \theta') = e^{-|\theta-\theta'|}$*. The eigendecomposition of* T_k *is given by*

$$\lambda_j = \frac{2}{1 + c_j^2}, \quad \psi_j : \theta \mapsto c_j \cos(c_j\theta) + \sin(c_j\theta)$$

where $(c_j)_{j=1}^{\infty}$ *are solutions to the equation* $\cot(c) = \frac{1}{2}\left(c - \frac{1}{c}\right)$ *where* cot *is the* cotangent *function.*

If the exact eigendecomposition is impossible to get, one can resort to using approximate decomposition based on a sampling of the integral operator.

Example 2.23 (Approximate eigendecomposition, Hoegaerts et al. 2005)**.** *Let* k *be a continuous kernel on a compact metric space* Θ *endowed with a measure* μ*. Let* $m > 0$ *and* $(\theta_j)_{j=1}^m \overset{i.i.d.}{\sim} \mu$*. We consider the problem of finding a continuous eigenvector* ψ *of a sampled version of the integral operator* T_k *with eigenvalue* $\lambda > 0$*:*

$$\forall \theta \in \Theta, \ \frac{1}{m}\sum_{j=1}^{m} k(\theta, \theta_j)\psi(\theta_j) = \lambda\psi(\theta). \tag{2.13}$$

In particular, by evalutating Equation (2.13) at points $(\theta_j)_{j=1}^m$*, we get that* $(\lambda m, \psi(\theta_j)_{j=1}^m)$ *is a pair of eigenvalue/eigenvector associated to the Gram matrix* $\mathbf{K} \in \mathcal{M}_m(\mathbb{R})$ *defined by* $\mathbf{K}_{ij} = k(\theta_i, \theta_j)$ *for all* $(i, j) \in [m]^2$*. These can be computed using* e.g. *Singular Value Decomposition, and by backsubstitution in Equation (2.13) one gets an approximated eigenbasis of dimension at most* m *that can be used as a proxy in applications where the true eigenvectors of* T_k *are required.*

The decay rate of the eigenvalues $(\lambda_j)_{j\in J}$ determines "how large" the RKHS is; the slower the decay the larger the space is. An approach to measuring the modeling capacity of the RKHS then consists in answering the question "Can functions in the RKHS approximate any continuous function for the uniform convergence ?".

Definition 2.24 (Universal kernels, Micchelli et al. (2006))**.** *Let k be a kernel on a compact metric space Θ. k is said to be* universal *if $\overline{\mathcal{H}_k} = \mathcal{C}(\Theta, \mathbb{R})$ where the closure is taken with respect to the uniform convergence norm.*

Universal kernels are key in obtaining consistency of machine learning estimators, and relate to the integral operator the following way.

Proposition 2.25 (Carmeli et al. (2010))**.** *Let k be a kernel on a compact metric space Θ. Then k is universal if and only if $T_{k,\mu}$ is injective for all probability measures μ.*

The integral operator allows to embed a probability measure in a Hilbert space an idea that has lead to a very active subfield of machine learning (Smola et al., 2007; Fukumizu et al., 2008; Sriperumbudur et al., 2010; Sejdinovic et al., 2013). Indeed, $T_{k,\mu}(1) \in \mathcal{H}_k$ characterizes the distribution μ provided that the kernel is *characteristic*, a class of kernel encompassing in particular the universal kernels (see Sriperumbudur et al. (2011); Szabó and Sriperumbudur (2018) for an exhaustive classification of kernels), in which case the RKHS norm in $\mathcal{H}_k$ allows to discriminate between measures and paves the way to many modern statistical tests based on maximum mean discrepancy (Gretton et al., 2008, 2012).

Kernel quadrature rules RKHSs can be used to approximate integrals, a topic of interest for this thesis. Given a measure μ on a compact space $\Theta \subset \mathbb{R}^d$, the goal of *quadrature rules* is to find points $(\theta_j)_{j=1}^m \in \Theta$ and coefficients $(\eta_j)_{j=1}^m \in \mathbb{R}^m$ such that for some class of functions $\mathcal{H}$ the approximation

$$\forall h \in \mathcal{H}, \int_\Theta h(\theta) \mathrm{d}\mu(\theta) \approx \sum_{j=1}^m \eta_j h(\theta_j)$$

is reasonable. In particular, we want the approximation to have better convergence rates than the asymptotic $\mathcal{O}(m^{1/2})$ obtained by Monte-Carlo methods where one uses $(\theta_j)_{j=1}^m \overset{i.i.d.}{\sim} \mu$ and $\eta_j = \frac{1}{m}$. Taking $\mathcal{H}$ as the unit ball of some RKHS, this problem can be cast as the approximation of the mean embedding in the RKHS with well chosen elements (Smola et al., 2007). The resulting integration scheme $(\eta_j, \theta_j)_{j=1}^m$ is called a *kernel quadrature rule* and can enjoy asymptotic convergence rates faster than Monte-Carlo, for example of order $\mathcal{O}(m^{-r/d})$ in a Sobolev space of order r (Novak, 2006). Kernel quadrature rules enjoy rich connections with random feature expansions, among them the *random Fourier features* introduced later, which allows to express the optimal sampling measure of the $(\theta_j)_{j=1}^m$ as a *leverage score* depending on the kernel and μ (Bach, 2017). Finally we want to mention that it is possible to get convergence rates of the estimate given by kernel quadrature rules even when applied to functions outside the RKHS (Kanagawa et al., 2020).

Empirical risk minimization in RKHSs In the supervised learning setting, given a loss function $\ell\colon \mathbb{R}^2 \to \mathbb{R}$ (taken proper, convex, lower semi-continuous with respect to its first argument) and a joint random variable $(\mathsf{X}, \mathsf{Y}) \in \mathcal{X} \times \mathbb{R}$, an ideal prediction rule $h \in \mathcal{F}(\mathcal{X}, \mathbb{R})$ minimizes $\mathcal{R}(h) := \mathbb{E}_{(\mathsf{X},\mathsf{Y})}\ell(h(\mathsf{X}), \mathsf{Y})$ over the set of admissible,

i.e. measurable, prediction rules. Because the law of (X, Y) is unknown, one relies on i.i.d. samples $(x_i, y_i)_{i=1}^n$ and trades the expectation for the empirical mean, to which a regularization term is added to prevent overfitting. The set of admissible prediction rule can be chosen to be a RKHS $\mathcal{H}_k$, resulting in the optimization problem

$$\inf_{h \in \mathcal{H}_k} \quad \frac{1}{n} \sum_{i=1}^{n} \ell\left(h(x_i), y_i\right) + \frac{\lambda}{2} \left\|h\right\|_{\mathcal{H}_k}^2 \quad (\lambda > 0). \tag{2.14}$$

Problem 2.14 has received a great deal of attention in several different settings such as the square loss (kernel ridge regression, (Vapnik, 1998)), the pinball loss (Steinwart et al., 2011), support vector machines (SVM, Cortes and Vapnik 1995), least squares SVM (Suykens and Vandewalle, 1999) and many others. One advantage of RKHSs over other spaces is to benefit from a *representer theorem* for the solution. First introduced in Kimeldorf and Wahba (1971) for specific losses such as the square loss, it has been extended to handle any loss functions.

Theorem 2.26 (Representer, Steinwart and Christmann (2008))**.** *There is a unique solution $\hat{h} \in \mathcal{H}_k$ to Problem 2.14 and there exist $(\hat{\alpha}_i)_{i=1}^n \in \mathbb{R}^n$ such that*

$$\hat{h} = \sum_{i=1}^{n} \hat{\alpha}_i k(\cdot, x_i). \tag{2.15}$$

Remark 2.27. *The representer theorem still holds under weaker assumption, as long as the functional to minimize is convex, only involves the evaluation of the model at points $(x_i)_{i=1}^n$ and the regularizer is a nondecreasing function of $\left\|h\right\|_{\mathcal{H}_k}$ (Schölkopf et al., 2001a).*

Example 2.28 (Kernel Ridge Regression)**.** *Using the square loss $\ell(h(x), y) = \frac{1}{2}(h(x) - y)^2$, the solution of Problem 2.14 can be computed in closed form. Denoting by $\mathbf{K} \in \mathcal{M}_n(\mathbb{R})$ the Gram matrix associated to $(x_i)_{i=1}^n$ and kernel k, and by $\mathbf{y} = [y_i]_{i=1}^n \in \mathbb{R}^n$ the observed outputs, it holds that*

$$\hat{\boldsymbol{\alpha}} = \left(\mathbf{K} + \lambda n \operatorname{Id}\right)^{-1} \mathbf{y}, \tag{2.16}$$

where $\hat{\boldsymbol{\alpha}} = [\hat{\alpha}_i]_{i=1}^n \in \mathbb{R}^n$ encodes the coefficients in Equation (2.15).

Large scale learning The flexiblity of kernel methods often comes with a computation price; large-scale learning scenarii require dedicated techniques ta make the computations tractable. In particular, we can cite the Nyström method (Williams and Seeger, 2001) which proposes to approximately solve Equation (2.16) using only a subset of the columns of the Gram matrices. This amounts to a subsampling in the representer expression in Equation (2.15) and has allowed to scale to billions of points (Rudi et al., 2017; Meanti et al., 2020). Another popular method is random Fourier features (RFF) introduced in (Rahimi and Recht, 2007, 2008) for *shift-invariant* kernels.

Definition 2.29. *Let k be a kernel on $\mathcal{X}$. It is said that k is* shift-invariant *if there exists $k^0 \colon \mathcal{X} \to \mathbb{R}$ (called its* signature*) such that for all $(x, z) \in \mathcal{X}^2$, $k(x, z) = k^0(x - z)$.*

The RFF model stems from Bochner's theorem (Rudin, 1990), stating that any continuous, bounded and shift-invariant kernel $k \colon \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ can be expressed as the Fourier transform of some finite Borel measure ρ_k on $\mathbb{R}^d$, and the correspondence is

one-to-one:

$$k(x,z) = \int_{\mathbb{R}^d} \cos\Big(\big\langle \omega, x-z\big\rangle\Big)\,\mathrm{d}\rho_k(\omega).$$

Given some integer m and $(\omega_j)_{j=1}^m$ i.i.d. sampled from ρ_k, the kernel $\tilde{k}\colon \mathcal{X}\times\mathcal{X}\to\mathbb{R}$ defined by

$$\forall (x,z)\in\mathcal{X}^2,\quad \tilde{k}(x,z) = \frac{1}{m}\sum_{j=1}^m \cos\left(\langle \omega_j, x-z\rangle\right) \tag{2.17}$$

is then a reasonable approximation for k (see the work of Sriperumbudur and Szabó (2015) for optimal rates), and the associated RKHS $\mathcal{H}_{\tilde{k}}$ can be used as hypothesis space in machine learning problems.

The feature map associated to the kernel is then $\tilde{\phi}\colon \mathbb{R}^d\to\mathbb{R}^{2m}$ defined for all $x\in\mathbb{R}^d$ by

$$\tilde{\phi}(x) = \frac{1}{\sqrt{m}}(\cos(\omega_1^\top x),\dots,\cos(\omega_m^\top x),\sin(\omega_1^\top x),\dots,\sin(\omega_m^\top x))^\top. \tag{2.18}$$

Example 2.30 (RFF integral operator eigendecomposition)**.** *Let $\tilde{k}$ be a RFF kernel on $\mathbb{R}^d$ of type Equation (2.17) for some integer m and $(\omega_j)_{j=1}^m$ i.i.d. sampled from ρ_k. Let Θ be a compact subspace of $\mathbb{R}^d$, endowed with a probability measure μ. The eigendecomposition of $T_{\tilde{k}}$ is computable, as for any $f\in L^2[\Theta,\mu]$, it holds that*

$$\begin{aligned}\forall\theta\in\Theta,\ T_{\tilde{k}}f(\theta) &= \int_\Theta \tilde{k}(\theta,\theta')f(\theta')\mathrm{d}\mu(\theta')\\ &= \frac{1}{m}\sum_{j=1}^m \cos(\omega_j^\top\theta)\int_\Theta \cos(\omega_j^\top\theta')f(\theta')\mathrm{d}\mu(\theta')\\ &\qquad + \sin(\omega_j^\top\theta)\int_\Theta \sin(\omega_j^\top\theta')f(\theta')\mathrm{d}\mu(\theta').\end{aligned}$$

Plugging in some function $h=\sum_{j=1}^n a_j\cos(\omega_j^\top\theta)+b_j\sin(\omega_j^\top\theta)$ and writing the eigenvector problem, we get that h is an eigenvector of $T_{\tilde{k}}$ if and only if $(a_1,\dots a_m,b_1,\dots b_m)^\top$ is an eigenvector of the matrix $\Psi\in\mathcal{M}_{2m}(\mathbb{R})$ encoding the scalar products in $L^2[\Theta,\mu]$ of the cosines and sines, defined $\forall i,j\in[m]^2$ by

$$\begin{aligned}\Psi_{i,j} &= \int_\Theta \cos(\omega_i^\top\theta)\cos(\omega_j^\top\theta)\mathrm{d}\mu(\theta) & \Psi_{i+m,j+m} &= \int_\Theta \sin(\omega_i^\top\theta)\sin(\omega_j^\top\theta)\mathrm{d}\mu(\theta)\\ \Psi_{i+m,j} &= \int_\Theta \sin(\omega_i^\top\theta)\cos(\omega_j^\top\theta)\mathrm{d}\mu(\theta) & \Psi_{i,j+m} &= \int_\Theta \cos(\omega_i^\top\theta)\sin(\omega_j^\top\theta)\mathrm{d}\mu(\theta).\end{aligned}$$

Moreover, they also share the same eigenvalues up to the $\frac{1}{m}$ factor, so that Singular Value Decomposition can be applied on Ψ to get the eigendecomposition of $T_{\tilde{k}}$. Finally, notice that when μ is the Lebesgue measure, the coefficients of Ψ are computable in closed form, whereas they must be approximated for general μ.

RFF benefit from a large body of work, from pure computational aspects (Yang et al., 2014; Le et al., 2013; Zhang et al., 2019) to quantifying their effect in learning algorithms (Rudi and Rosasco, 2017; Li et al., 2019b).

Kernel Learning Though giving rise to expressive functional spaces, kernels are hard to tune in practice. The idea of *kernel learning* proposes to learn the kernel adapted to the task at hand. This can take many forms, including multiple kernel learning (Sonnenburg et al., 2006) which proposes to learn a convex combination of kernels, kernel alignment (Kandola et al., 2002) which gives a measure of adequacy between the kernel and the data, spectral kernel learning where the goal is to learn a shift invariant kernel from its fourier spectral measure stemming from Bochner's theorem (Oliva et al., 2016; Li et al., 2019a), and many others. In particular, we want to mention *deep kernel learning*, which consists in parameterizing a kernel by a neural architecture, and provides a way to train jointly the parameters of the estimator and the neural architecture. This idea has been introduced by Salakhutdinov and Hinton (2007), where authors use deep beliefs nets to learn covariance kernels for Gaussian processes. The pretraining of the network is carried out on a large, unsupervised dataset to learn relevant features, and the weights are fine-tuned for specific applications involving a smaller amount of data. The resulting kernel family is referred to as *deep kernels* (DK). Since then deep kernels have been exploited in various kernel-related tasks, such as maximum mean discrepancy for GAN discriminators (Li et al., 2017), density estimation via score matching in exponential families (Wenliang et al., 2018), semi supervised learning (Jean et al., 2018), non-parametric two-sample tests (Liu et al., 2020), among others. The resulting optimization problems are costly, as learning deep kernels involves the joint minimization in the parameters of the neural architecture and in the coefficients of the representer theorem. Thus, most approaches focus on scalable approaches by choosing a kernel that brings computational advantages. We can mention Yang et al. (2015) for image classification with random Fourier features approximated using the FastFood method Le et al. (2013), Wilson et al. (2016) for scaling Gaussian processes up to millions of points, Mehrkanoon et al. (2017); Mehrkanoon and Suykens (2018) for general hybrid RFF approaches. Deep kernel learning approaches allow to apply kernel methods to complex data such as images, as emphasized later in Section 5.4.4.

We now present the vv-RKHS extension to kernel methods used for modeling outputs which are not real-valued.

2.2.2 Operator-valued Kernels and vv-RKHSs

Vector-valued RKHSs (vv-RKHSs) are an extension of RKHSs allowing to model functions with outputs in a Hilbert space first developed in Pedrick (1957). While the definition and properties of vv-RKHSs are similar to the real output case (see Table 2.2 for a comparison between real and vector-valued cases), the kernels involved are not real-valued anymore but operator-valued. We refer to Carmeli et al. (2006, 2010) for an overview on the topic. In what follows, $\mathcal{Y}$ a separable Hilbert space.

Definition 2.31. *Let $\mathcal{H} \subset \mathcal{F}(\mathcal{X}, \mathcal{Y})$ be a Hilbert space. $\mathcal{H}$ is said to be a* vector-valued reproducing kernel Hilbert space *if and only if for all $x \in \mathcal{X}$, the following evaluation mapping at x is continuous.*

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

Similarly to the scalar case, vv-RKHSs can be constructed from OVKs.

Definition 2.32. *An operator-valued kernel (OVK) on $\mathcal{X}$ is a function $K : \mathcal{X} \times \mathcal{X} \to \mathcal{L}(\mathcal{Y})$ such that*

Table 2.2 – Equivalent notions between scalar (k) and operator-valued kernels (K).

	scalar-valued kernel	operator-valued kernel
kernel	$k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$	$K : \mathcal{X} \times \mathcal{X} \to \mathcal{L}(\mathcal{Y})$
positivity	$\sum_{i,j\in[n]} \alpha_i \alpha_j k(x_i, x_j) \geq 0$	$\sum_{i,j\in[n]} \left\langle K(x_i, x_j) y_i, y_j \right\rangle_{\mathcal{Y}} \geq 0$
function space	$\mathcal{H}_k = \overline{\text{Span}}\{k(\cdot, x) \,:\, x \in \mathcal{X}\}$	$\mathcal{H}_K = \overline{\text{Span}}\{K(\cdot, x)y \,:\, (x, y) \in \mathcal{X} \times \mathcal{Y}\}$
reproducing property	$h(x) = \left\langle h, k(\cdot, x) \right\rangle_{\mathcal{H}_k}$	$h(x) = K(\cdot, x)^{\#} h$
feature map	$\phi : \mathcal{X} \to \mathcal{V}$	$\Phi : \mathcal{X} \to \mathcal{L}(\mathcal{Y}, \mathcal{V})$
parametrization	$h = \left\langle \phi(\cdot), v \right\rangle_{\mathcal{V}}$	$h = \Phi(\cdot)^{\#} v$

1. $\forall (x, z) \in \mathcal{X}^2, \quad K(x, z) = K(z, x)^{\sharp}$, *and*
2. $\forall n \in \mathbb{N}^*, \ (x_i)_{i=1}^n \in \mathcal{X}^n, \ (y_i)_{i=1}^n \in \mathcal{Y}^n, \quad \sum_{i=1}^n \sum_{j=1}^n \langle y_i, K(x_i, x_j) y_j \rangle \geq 0.$

OVKs can also be characterized by the existence of a feature map.

Theorem 2.33. *A function $K \colon \mathcal{X} \times \mathcal{X} \to \mathcal{L}(\mathcal{Y})$ is an OVK if and only if there exist a Hilbert space $\mathcal{V}$ and a mapping $\Phi \colon \mathcal{X} \to \mathcal{L}(\mathcal{V}, \mathcal{Y})$ such that*

$$\forall (x, z) \in \mathcal{X}^2, \quad K(x, z) = \Phi(x)\Phi(z)^{\sharp}.$$

We again call Φ a feature map and $\mathcal{V}$ a feature space. Given an OVK K and $x \in \mathcal{X}$, $K_x \colon \mathcal{Y} \to \mathcal{F}(\mathcal{X}, \mathcal{Y})$ denotes the linear operator such that

$$\forall y \in \mathcal{Y}, \ \forall z \in \mathcal{X}, \quad K_x y(z) = K(x, z) y.$$

The following theorem describes the vv-RKHS associated to an OVK.

Theorem 2.34. *Let K be an OVK on $\mathcal{X}$. Then there exists a unique Hilbert space of functions $\mathcal{H}_K \subset \mathcal{F}(\mathcal{X}, \mathcal{Y})$ such that*

1. $\forall (x, y) \in \mathcal{X} \times \mathcal{Y}, \quad K_x y \in \mathcal{H}_K$, *and*
2. $\forall (h, x) \in \mathcal{H}_k \times \mathcal{X}, \quad h(x) = K_x^{\sharp} h.$

Remark 2.35. *If $\mathcal{H}$ is a vv-RKHS in the sense of Definition 2.31 then its associated kernel is given by $K(x, z) = ev_x ev_z^{\sharp}$.*

We now define the family of *separable* kernels which are among the simplest OVKs to work with.

Definition 2.36. *An operator-valued kernel $K \colon \mathcal{X} \times \mathcal{X} \to \mathcal{L}(\mathcal{Y})$ is called* separable *if there exists a scalar kernel $k \colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ and an operator $A \in \mathcal{L}(\mathcal{Y})$ such that*

$$\forall (x, z) \in \mathcal{X}^2, \quad K(x, z) = k(x, z) A.$$

Separable kernels are undoubtedly the most studied OVKs, for their simplicity and computational efficiency. Indeed the scalar product in the associated RKHS $\mathcal{H}_K$ takes a straightforward form on elementary elements of the form $K_x y$:

$$\left\langle K_{x_1} y_1, K_{x_2} y_2 \right\rangle_{\mathcal{H}_K} = \left\langle K(x_1, x_2) y_1, y_2 \right\rangle_{\mathcal{Y}} = k(x_1, x_2) \left\langle A y_1, y_2 \right\rangle_{\mathcal{Y}}$$

for all $(x_1, x_2, y_1, y_2) \in \mathcal{X}^2 \times \mathcal{Y}^2$.

Other notables classes of OVKs include the curl-free and div-free matrix-valued kernels used for vector fields learning (Macedo and Castro, 2010), the transformable matrix-valued kernels for multi-view data (Huusari et al., 2018) and the recent entangled or partial trace OVKs deriving from quantum computing concepts (Huusari and Kadri, 2021). It is also worth mentioning that extensions of multiple kernel learning to OVKs have been investigated in Kadri et al. (2012).

Example 2.37. *Let $k_\mathcal{X} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a kernel, and $\mathbf{A} \in \mathcal{M}_s(\mathbb{R})$ for some integer s, such that $\mathbf{A}$ is a positive self-adjoint operator. Let*

$$K : \begin{pmatrix} \mathcal{X} \times \mathcal{X} & \to & \mathcal{M}_s(\mathbb{R}) \\ (x, z) & \mapsto & k_\mathcal{X}(x, z)\mathbf{A} \end{pmatrix}.$$

Then K is an OVK on $\mathcal{X}$ and $\mathcal{H}_K \subset \mathcal{F}(\mathcal{X}, \mathbb{R}^s)$.

While in the scalar case the kernel associated to a RKHS is unique, in the operator-valued case one may find OVKs acting on different output space that lead to the "same" space of function up to an isometry in the sense defined below.

Definition 2.38. *Two Hilbert spaces $\mathcal{H}_1$, $\mathcal{H}_2$ are said to be* unitarily equivalent *if there exist $W \in \mathcal{L}(\mathcal{H}_1, \mathcal{H}_2)$ such that $WW^\sharp = \mathrm{Id}_{\mathcal{H}_2}$ and $W^\sharp W = \mathrm{Id}_{\mathcal{H}_1}$.*

Example 2.39 (Carmeli et al. (2010), example 6)**.** *Let $k_\mathcal{X} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ and $k_\Theta : \Theta \times \Theta \to \mathbb{R}$ be kernels. Let*

$$K : \begin{pmatrix} \mathcal{X} \times \mathcal{X} & \to & \mathcal{L}(\mathcal{H}_{k_\Theta}) \\ (x, z) & \mapsto & k_\mathcal{X}(x, z)\,\mathrm{Id}_{\mathcal{H}_{k_\Theta}} \end{pmatrix}.$$

Then K is an OVK on $\mathcal{X}$ and the corresponding vv-RKHS $\mathcal{H}_K$ is unitarily equivalent to $\mathcal{H}_{k_\mathcal{X}} \otimes \mathcal{H}_{k_\Theta}$.

Example 2.40 (Carmeli et al. (2010), example 7)**.** *Let $k_\mathcal{X} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ and $k_\Theta : \Theta \times \Theta \to \mathbb{R}$ be kernels. Assume that Θ is a compact metric space endowed with a probability measure μ such that $\mathrm{supp}(\mu) = \Theta$, and that k_Θ is continuous. Let*

$$K : \begin{pmatrix} \mathcal{X} \times \mathcal{X} & \to & L^2[\Theta, \mu] \\ (x, z) & \mapsto & k_\mathcal{X}(x, z)T_{k,\mu} \end{pmatrix}.$$

Then K is an OVK on $\mathcal{X}$ and the corresponding vv-RKHS $\mathcal{H}_K$ is unitarily equivalent to $\mathcal{H}_{k_\mathcal{X}} \otimes \mathcal{H}_{k_\Theta}$.

Remark 2.41. *The modeling spaces from Example 2.39 and Example 2.40 are essentialy the same spaces of functions as the scalar RKHS associated to the product kernel $k_\mathcal{X} \otimes k_\Theta$. What differs among these choices is whether we look at the elements of these spaces as functions (i) from $\mathcal{X}$ to $\mathcal{H}_{k_\Theta}$, (ii) from $\mathcal{X}$ to $L^2[\Theta, \mu]$, or (iii) from $\mathcal{X} \times \Theta$ to $\mathbb{R}$. We leverage this flexibility in the design of optimization algorithms in Chapter 3.*

Similarly to the scalar case, one can define the integral operator associated to an OVK $K : \Theta \times \Theta \to \mathcal{L}(\mathcal{Y})$ when $\mathcal{H}_K \subset \mathcal{C}(\Theta, \mathcal{Y})$ and K is bounded in operator norm on $\mathrm{supp}(\mu)$:

$$T_{K,\mu} : \begin{pmatrix} L^2[\Theta, \mu; \mathcal{Y}] & \to & L^2[\Theta, \mu; \mathcal{Y}] \\ f & \mapsto & \left(\theta \mapsto \int_\Theta K(\theta, \theta')f(\theta')\mathrm{d}\mu(\theta')\right) \end{pmatrix}. \tag{2.19}$$

Such object can be defined for the larger class of *Mercer kernels*, which are OVKs such that $x \mapsto \left\|K(x,x)\right\|_{\mathrm{op}}$ is locally bounded, and $\mathcal{H}_K \subset \mathcal{C}(\mathcal{X}, \mathcal{Y})$ (see Carmeli et al. (2010) for more details about this).

Proposition 2.42 (Carmeli et al. (2010))**.** *Let Θ be a compact metric space endowed with a Borel probability measure μ, and $K\colon \Theta \times \Theta \to \mathcal{L}(\mathcal{Y})$ be an OVK. Assume that $\mathcal{H}_K \subset \mathcal{C}(\Theta, \mathcal{Y})$ and that K is bounded in operator norm on* $\mathrm{supp}(\mu)$*. If $K(\theta,\theta)$ is a compact operator for all $\theta \in \Theta$, then $T_{K,\mu}$ is a compact operator.*

Proposition 2.42 allows to perform the eigendecomposition of $T_{K,\mu}$, and we get a similar characterization of $\mathcal{H}_K$ than the one in Equation (2.12). We refer the interested reader to Carmeli et al. (2010).

Regularized Empirical risk minimization in vv-RKHSs Let $(\mathsf{X}, \mathsf{Y}) \in \mathcal{X} \times \mathcal{Y}$ be random variables, we assume access to i.i.d. samples $(x_i, y_i)_{i=1}^n$. Given a proper, convex lower-semicontinuous loss function $L\colon \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ (with respect to its first argument), we consider the learning problem

$$\inf_{h \in \mathcal{H}_K} \frac{1}{n} \sum_{i=1}^{n} L(h(x_i), y_i) + \frac{\lambda}{2} \left\|h\right\|_{\mathcal{H}_K}^2 \quad (\lambda > 0), \tag{2.20}$$

where $K\colon \mathcal{X} \times \mathcal{X} \to \mathcal{L}(\mathcal{Y})$ is an OVK. Problem 2.20 arises when performing empirical risk minimization in vv-RKHSs for outputs that lie in a Hilbert space. In particular, we can cite applications to multi-output learning (Micchelli and Pontil, 2005; Álvarez et al., 2012), function-to-function regression (Kadri et al., 2016) and structured prediction where outputs are embeddings of structured objects into scalar RKHSs (Brouard et al., 2011; Kadri et al., 2013; Brouard et al., 2016). Similarly to the scalar case, it turns out that the minimizer of Problem 2.20 can be expressed using a representer theorem, this time with coefficients in $\mathcal{Y}$ themselves.

Theorem 2.43 (Representer, Micchelli and Pontil (2005))**.** *There exists a unique solution $\hat{h} \in \mathcal{H}_K$ to Problem 2.20, that can be written as*

$$\hat{h} = \sum_{i=1}^{n} K(\cdot, x_i)\hat{\alpha}_i \tag{2.21}$$

for some $(\hat{\alpha}_i)_{i=1}^n \in \mathcal{Y}^n$.

The benefit of Theorem 2.43 is to pertain the solution to Problem 2.20 to a specific subset of $\mathcal{H}_K$. The coefficients $(\hat{\alpha}_i)_{i=1}^n$ now belong to the Hilbert space $\mathcal{Y}$, which can pose representation problems for infinite dimensional $\mathcal{Y}$. This is discussed at length in Chapter 3.

Kernel methods have a rich history of exploiting parametric duality techniques to solve empirical risk minimization problem. In particular, these techniques translate well to OVK-based learning problems, and can be exploited to gain additional information about the coefficients $(\hat{\alpha}_i)_{i=1}^n$. Below we present a result originating from (Brouard et al., 2016) about the dualization of such problems with output in a general Hilbert space $\mathcal{Y}$. We adopt the notation $I_{\ell_i} : y \in \mathcal{Y} \mapsto I_\ell(y, y_i)$ for any $i \in [n]$.

Theorem 2.44 (Dualization, Brouard et al. (2016))**.** *The solution of Problem 2.20 is given by*

$$\hat{h} = \frac{1}{\lambda n} \sum_{i=1}^{n} K(\cdot, x_i)\hat{\alpha}_i, \tag{2.22}$$

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

$$\inf_{(\alpha_i)_{i=1}^n \in \mathcal{Y}^n} \sum_{i=1}^{n} I_{\ell_i}^{\star}(-\alpha_i) + \frac{1}{2\lambda n} \sum_{i,j=1}^{n} \left\langle \alpha_i, K(x_i, x_j)\alpha_j \right\rangle_{\mathcal{Y}}. \tag{2.23}$$

Again, solving Problem 2.23 may not be straightforward and we refer to Chapter 3 for dedicated techniques in the context of integral losses.

Large scale learning The vv-RKHS framework is even more challenging than the real-valued one from computational perspective. Approximated kernel schemes can however be applied to lighten it, as proposed by the *operator random Fourier features* (ORFF) methodology (Brault et al., 2019). This approach holds for shift-invariant OVK, and relies on an operator-valued version of Bochner's theorem, stating that any shift-invariant Mercer OVK $K\colon \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ can be expressed as the Fourier transform of some finite operator-valued borelian measure Q_K on $\mathbb{R}^d$, and the correspondence is one-to-one:

$$k(x, z) = \int_{\mathbb{R}^d} \cos\left(\langle \omega, x - z \rangle\right) \mathrm{d}Q_K(\omega).$$

Under mild regularity assumptions verified for separable OVKs (see Brault et al. (2016) for a precise statement), the operator-valued measure can be written $\mathrm{d}Q_K(\omega) = Q_K(\omega)\mathrm{d}\rho_K(\omega)$ for an operator-valued function Q_K and bounded Borel measure ρ_k. Given some integer m and $(\omega_j)_{j=1}^m$ i.i.d. sampled from ρ_K, the kernel $\widetilde{K}\colon \mathcal{X} \times \mathcal{X} \to \mathcal{L}(\mathcal{Y})$ defined by

$$\forall (x, z) \in \mathcal{X}^2, \quad \widetilde{K}(x, z) = \frac{1}{m} \sum_{j=1}^{m} \cos\left(\langle \omega_j, x - z \rangle\right) Q_K(\omega_j) \tag{2.24}$$

can be used as hypothesis space in machine learning problems. The feature map associated to the kernel is then $\widetilde{\Phi}\colon \mathbb{R}^d \to \mathcal{L}\left(\mathcal{Y}, \mathcal{Y}^{2m}\right)$ defined for all $x \in \mathbb{R}^d$ by

$$\widetilde{\Phi}(x) = \frac{1}{\sqrt{m}} \bigoplus_{i=1}^{m} \left(\cos\left(\omega_i^\top x\right) B(\omega_j)^\sharp \oplus \sin\left(\omega_m^\top x\right) B(\omega_j)^\sharp\right) \tag{2.25}$$

where $\forall j \in [m]$, $B(\omega_j)B(\omega_j)^\sharp = Q(\omega_j)$.

In the case of a separable kernel $K = k_{\mathcal{X}} A$, $\mathrm{d}Q_K = A\mathrm{d}\rho_k$ so that it suffices to exhibit a decomposition $BB^\sharp = A$ for some $B \in \mathcal{L}(\mathcal{Y})$ and the feature map writes

$$\widetilde{\Phi}(x) = \frac{1}{\sqrt{m}} \tilde{\phi}_{\mathcal{X}}(x) \otimes B^\sharp. \tag{2.26}$$

In particular Singh et al. (2020) use the ORFF methodology to learn stabilizable nonlinear dynamics.

2.3 Conclusion

In this chapter we provided a background in both convex optimization and vector-valued RKHSs theory, with focus on machine learning applications. In short, vv-RKHSs are extensions of RKHSs allowing to model functions with outputs in any Hilbert space, enabling the model of function-valued functions in particular. These spaces of functions are chosen as hypothesis spaces for the regularized empirical risk minimization problems raised in the rest of the thesis.

3

Optimization Schemes for Learning with Integral Losses

Contents

In this chapter, we address the general question of solving regularized empirical risk minimization problems in vv-RKHSs with functional outputs, in the presence of integral losses. After introducing the family of considered problems in Section 3.1, we devote Section 3.2 to the study of the special case of the square loss function, for which closed-form solutions exist. In Section 3.3 we investigate ways to solve the problems based on the primal formulation, either by means of a *double representer theorem* or by the use of *random features*. We then delve into dual algorithms in Section 3.4, designing techniques involving adapted basis related to the operator-valued kernels used in practice, or exploiting linear splines bases to represent the dual variables, rephrasing the problem as the minimization of an approximated quadratic form under specific linear constraints. This chapter ends with a conclusion in Section 3.5.

3.1 Problem Formulation

Let (X, Y) be a pair of random variables taking values in a product space $\mathcal{X} \times L^2[\Theta, \mu; \mathcal{U}]$. Depending on the application considered, $\mathcal{X}$ will be $\mathbb{R}^d$ or a functional space. The space $\Theta \subset \mathbb{R}^p$ is a compact domain endowed with a Borel probability measure μ, and $\mathcal{U} = \mathbb{R}^s$ for some integer $s > 0$. We begin by proposing a family of loss functions referred to as *integral losses* that involves an integration over the space Θ.

Integral Losses Losses on $L^2[\Theta, \mu; \mathcal{U}]$ can be constructed from a family of ground losses on $\mathcal{U}$ parameterized by $\theta \in \Theta$. Let $\ell \colon \Theta \times \mathcal{U} \times \mathcal{U} \to \mathbb{R}$ be such a family, and define a loss function I_ℓ on $L^2[\Theta, \mu; \mathcal{U}]^2$ by integration with respect to μ:

$$I_\ell(f, g) = \int_\Theta \ell(\theta, f(\theta), g(\theta)) \mathrm{d}\mu(\theta), \text{ for } f, g \in L^2[\Theta, \mu; \mathcal{U}]. \tag{3.1}$$

The integrated loss I_ℓ encodes the cost associated to the prediction of f given the ground truth g. As f and g are functions with domain Θ, the prediction error is measured pointwise ($\forall \theta \in \Theta$) as $\ell(\theta, f(\theta), g(\theta))$; in I_ℓ this error is aggregated on Θ with weighting μ (see Table 3.1 for a summarized view of all these parameters in different scenarii). We require that ℓ is a convex lower semi-continuous integrand; that is for all $\theta \in \Theta, v \in \mathcal{U}$, $\ell(\theta, \cdot, v)$ is lower semi-continuous and convex. The integrand is also chosen to be proper: $\forall (\theta, u, v) \in \Theta \times \mathcal{U}^2, \ell(\theta, u, v) \neq -\infty$ and is not everywhere $+\infty$. We assume in the sequel and refer to these requirements jointly as *normal convex integrand*. To avoid indefinite integral issues, by convention $I_\ell(f, g) = +\infty$ if any of the positive or negative part of the integrand is infinite.

One of the challenges to tackle is that even computing $I_\ell(f, g)$ in Equation (3.1) is not straightforward. Our **goal** is to minimize the risk (or in practice its empirical version) over a hypothesis class $\mathcal{H}$ (detailed below)

$$\inf_{h \in \mathcal{H}} \mathcal{R}(h) := \mathbb{E}_{(\mathsf{X}, \mathsf{Y})}[I_\ell(h(\mathsf{X}), \mathsf{Y})]. \tag{3.2}$$

It is instructive to consider a few examples for Problem 3.2; the corresponding $(\mathcal{X}, \Theta, \mathcal{U}, \ell)$ choices are summarized in Table 3.1.

- **Functional output regression** (FOR, Chapter 4): In this problem family the task is to regress to a functional output ($\Theta = [0, 1]$ is endowed with a probability measure μ, $\mathcal{U} = \mathbb{R}$) from a vectorial ($\mathcal{X} = \mathbb{R}^d$) or functional input ($\mathcal{X} = L^2[\Theta_0, \mu_0]$). A natural way to tackle this problem is to estimate $\mathbb{E}[\mathsf{Y}|\mathsf{X}] = h^\dagger(\mathsf{X})$ where it is well-known that

$$h^\dagger = \arg\min_{h \in \mathcal{H}} \mathbb{E}_{(\mathsf{X}, \mathsf{Y})} \left[\left\| \mathsf{Y} - h(\mathsf{X}) \right\|^2_{L^2[\Theta, \mu]} \right],$$

  with $\mathcal{H}$ being the set of all measurable functions from $\mathcal{X}$ to $L^2[\Theta, \mu]$. As $\|f\|^2_{L^2[\Theta,\mu]} = \int_\Theta f^2(\theta) \, \mathrm{d}\mu(\theta)$ for any $f \in L^2[\Theta, \mu]$, finding $h^\dagger$ reduces to Problem 3.2 with $\ell(\theta, u, v) = \frac{1}{2}(v - u)^2$.
- **Joint quantile regression** (JQR, Section 5.4): While classically quantiles are regressed for a single level (Koenker and Bassett Jr, 1978), one can consider the learning problem of simultaneously learning multiple quantiles (Sangnier et al., 2016) or that of the whole quantile function (Brault et al., 2019). Particularly, let $\Theta = [0, 1]$, $\mathcal{U} = \mathbb{R}$ and consider a pair of random variables $(\mathsf{X}, \mathsf{Y}) \in \mathbb{R}^d \times \mathbb{R}$ or $(\mathsf{X}, \mathsf{Y}) \in \mathbb{R}^{d_1 \times d_2} \times \mathbb{R}$. Here the r.v. Y is real valued, but is identified with a random variable of constant functions in $L^2[\Theta, \mu]$. The θ-conditional quantile of $\mathsf{Y}|\mathsf{X}$ for $\theta \in (0, 1)$ is defined as

$$q(x)(\theta) = \inf\{u \in \mathbb{R} \,:\, \mathbb{P}(\mathsf{Y} \leq u \,|\, \mathsf{X} = x) \geq \theta\} = \arg\min_{u \in \mathbb{R}} \mathbb{E}_{\mathsf{Y}}[\ell(\theta, u, \mathsf{Y}) \,|\, \mathsf{X} = x], \tag{3.3}$$

  where in the variational description ℓ is the so-called pinball loss

$$\ell(\theta, u, v) = \max\left(\theta(v - u), (\theta - 1)(v - u)\right).$$

Table 3.1 – Considered problem family: examples. QR: quantile regression. CSC: cost-sensitive classification. FOR: functional output regression. ET: emotion transfer.

Task	$\mathcal{X}$	Θ	$\mathcal{U}$	$\ell(\theta, u, v)$
QR (Section 5.4)	$\mathbb{R}^d$	[0,1]	$\mathbb{R}$	$\max(\theta(v-u), (\theta-1)(v-u))$
CSC (Section 5.5)	$\mathbb{R}^d$	$[0,1]$	$\mathbb{R}$	$\left\lvert\theta - \mathbb{1}_{\{-1\}}(v)\right\rvert \max(0, 1-uv)$
FOR (Chapter 4)	$\mathbb{R}^d$ or $L^2[\Theta_0, \mu_0]$	$[0,1]$	$\mathbb{R}$	$\frac{1}{2}(u-v)^2$
ET (Chapter 6)	$\mathbb{R}^d$ or $\mathbb{R}^{d_1 \times d_2}$	$\subset \mathbb{R}^p$	$\mathcal{X}$	$\frac{1}{2}\|u-v\|_{\mathcal{U}}^2$

The variational form in Equation (3.3) implies that for any $h \in \mathcal{H}$

$$\int_\Theta \mathbb{E}_{\mathsf{X}}[\mathbb{E}_{\mathsf{Y}}[\ell(\theta, q(\mathsf{X})(\theta), \mathsf{Y}) \,|\, \mathsf{X}]] \mathrm{d}\mu(\theta) \leq \int_\Theta \mathbb{E}_{\mathsf{X}}[\mathbb{E}_{\mathsf{Y}}[\ell(\theta, h(\mathsf{X})(\theta), \mathsf{Y}) \,|\, \mathsf{X}]] \mathrm{d}\mu(\theta).$$

This means that $R(q) \leq R(h)$ for any $h \in \mathcal{H}$ in Problem 3.2.

- **Cost-sensitive classification** (CSC, Section 5.5): Let $\Theta = [0,1]$, $\mathcal{U} = \mathbb{R}$ and (X, Y) be random variables taking values in $\mathbb{R}^d \times \{-1, 1\}$. We denote c_+ (resp. c_-) the cost associated to a false positive (resp. negative). The cost-sensitive hinge loss is defined as by $\ell_{\text{hinge}}(c_+, c_-, y, y') = (c_+ \mathbb{1}_{\{1\}}(y') + c_- \mathbb{1}_{\{-1\}}(y')) \max(0, 1 - yy')$ where $(y, y') \in \mathbb{R} \times \{-1, 1\}$. Given $x \in \mathcal{X}$, a classification rule is obtained by minimizing $\mathbb{E}_{\mathsf{Y}}[\ell(c_+, c_-, s, Y) \,|\, \mathsf{X} = x]$ for $s \in \mathbb{R}$, and predicting $\operatorname{sign}(s)$ (Bach et al., 2006). Defining the asymmetry coefficient $\theta = \frac{c_+}{c_+ + c_-}$ and restraining the values of (c_+, c_-) to the line $c_+ + c_- = 1$ the training can be performed jointly for all values of $\theta \in [0,1]$ by considering (3.2) with $\ell(\theta, u, v) = \left\lvert\theta - \mathbb{1}_{\{-1\}}(v)\right\rvert \max(0, 1-uv)$, μ encoding the user importance to different values of θ, and identifying the random variable $\mathsf{Y} \in \{-1, 1\}$ with a random variable of constant functions in $L^2[\Theta, \mu]$ (Brault et al., 2019).
- **Emotion transfer** (ET, Chapter 6): Let $\Theta \subset \mathbb{R}^p$ be a compact set corresponding to an embedding space for emotions endowed with a probability μ measuring the importance of each emotion, and Y be a random variable taking its values in $\mathcal{C}(\Theta, \mathcal{U})$ where $\mathcal{U} = \mathbb{R}^d$ or $\mathcal{U} = \mathbb{R}^{d_1 \times d_2}$. The random variable Y encodes the trajectory of a phenomenon (facial landmarks, face picture) with respect to the emotions. Let ϑ be a random variable on Θ with probability ν, independent from Y, and consider $\mathsf{X} = \mathsf{Y}(\vartheta)$ the $\mathcal{X}$-valued random variable ($\mathcal{X} = \mathcal{U}$). Estimating Y from X can be tackled by finding $\mathbb{E}[\mathsf{Y}|\mathsf{X}] = h^\dagger(\mathsf{X})$ where

$$h^\dagger = \operatorname*{arg\,min}_{h \in \mathcal{H}} \mathbb{E}_{(\mathsf{X},\mathsf{Y})} \left[\left\| \mathsf{Y} - h(\mathsf{X}) \right\|^2_{L^2[\Theta, \mu; \mathcal{U}]} \right],$$

with $\mathcal{H}$ being the set of all measurable function from $\mathcal{X}$ to $L^2[\Theta, \mu; \mathcal{U}]$. Therefore, finding $h^\dagger$ reduces to Problem 3.2 with $\ell(\theta, u, v) = \frac{1}{2}\|v - u\|^2_{\mathcal{U}}$.

Hypothesis Space We make use of the modeling choices described in Example 2.37 and Example 2.39, given some kernels $k_{\mathcal{X}}$, k_Θ, and output similarity encoding matrix $\mathbf{A}$. Let $\mathcal{H}_G$ be the vv-RKHS associated with the decomposable OVK $G = k_\Theta \mathbf{A}$. Elements in $\mathcal{H}_G$ model the $\Theta \mapsto \mathcal{U}$ mapping. We then specify $\mathcal{H}_K$ to be the vv-RKHS associated

to kernel $K = k_{\mathcal{X}} \operatorname{Id}_{\mathcal{H}_G}$. Elements $h \in \mathcal{H}_K$ model the relation

$$h \colon \mathcal{X} \mapsto (\underbrace{\Theta \mapsto \mathcal{U}}_{\in \mathcal{H}_G}).$$

As seen in Remark 2.41, there is an equivalence between the views $K = k_{\mathcal{X}} \operatorname{Id}_{\mathcal{H}_G}$ and $K = k_{\mathcal{X}} T_G$ where T_G is the integral operator associated to G defined in Equation (2.19). Depending on the situation at hand, we may use one or the other representation, and denote by a unified notation $\mathcal{Y}$ the output space of functions in $\mathcal{H}_K$ (either $\mathcal{H}_G$ or $L^2[\Theta, \mu; \mathcal{U}]$).

Choosing $\mathcal{H} := \mathcal{H}_K$ in Problem 3.2 one arrives at

$$\inf_{h \in \mathcal{H}_K} \mathcal{R}(h) := \mathbb{E}_{(\mathsf{X},\mathsf{Y})} \left[I_\ell(h(\mathsf{X}), \mathsf{Y}) \right]. \tag{3.4}$$

In practice, we do not have access to the true distribution of (X, Y), but we are given i.i.d. samples gathered in a dataset $\mathcal{S} := (x_i, y_i)_{i=1}^n$. These samples define the empirical risk

$$\mathcal{R}_{\mathcal{S}}(h) := \frac{1}{n} \sum_{i=1}^n I_\ell(h(x_i), y_i),$$

and one can solve

$$\inf_{h \in \mathcal{H}_K} \mathcal{R}_{\mathcal{S}}(h) + \frac{\lambda}{2} \|h\|_{\mathcal{H}_K}^2 \quad (\lambda > 0). \tag{3.5}$$

The term $\frac{\lambda}{2} \|h\|_{\mathcal{H}_K}^2$ acts as a regularizer. It can be interpreted two ways: from a statistical learning point of view, it is used to prevent over-fitting whereas from an optimization point of view it brings strong convexity to the problem and ensures the coercivity of the functional to minimize.

One of the advantages of working with vv-RKHSs is to have a fairly general *representer theorem* which ensures that the solution belongs to a specific subset of $\mathcal{H}_K$ (see Theorem 2.43). Particularly, we know that Problem 3.5 admits a unique solution $\hat{h} \in \mathcal{H}_K$ which can be written as

$$\hat{h} = \sum_{i=1}^n K(\cdot, x_i) \hat{\alpha}_i \tag{3.6}$$

for some $(\hat{\alpha}_i)_{i=1}^n \in \mathcal{Y}^n$.

When $\mathcal{Y}$ is finite-dimensional, this is enough to parameterize the solution and represent it on a computer. In our case, $\mathcal{Y}$ is a potentially infinite-dimensional functional space raises an additional challenge on the parameterization of the $(\hat{\alpha}_i)_{i=1}^n$ challenging. Therefore plugging Equation (3.6) into Problem 3.5 is not a viable choice to directly solve it, as we would be left with an optimization problem over an infinite-dimensional space $\mathcal{Y}^n$.

Parametric duality can be exploited in the context of vv-RKHSs (see Theorem 2.44) and the dual problem takes the form

$$\inf_{(\alpha_i)_{i=1}^n \in \mathcal{Y}^n} \sum_{i=1}^n I_{\ell_i}^\star(-\alpha_i) + \frac{1}{2\lambda n} \sum_{i,j=1}^n \left\langle \alpha_i, K(x_i, x_j) \alpha_j \right\rangle_{\mathcal{Y}}, \tag{3.7}$$

Table 3.2 – Summary of proposed algorithms. GD: (sub) gradient descent. SGD: stochastic (sub) gradient descent. PGD: proximal gradient descent.

Section	type	$\mathcal{Y}$	parameterization	loss	algorithm
3.2.1	closed form	$L^2[\Theta, \mu; \mathcal{U}]$	eigenbasis of T_G	square loss	analytic
3.2.2	closed form	$\mathcal{H}_G$	double representer	square loss	analytic
3.3.1	primal	$\mathcal{H}_G$	double representer	sampled	GD
3.3.2	primal	$\mathcal{H}_G$	ORFF	any	SGD
3.4.2	dual	$L^2[\Theta, \mu; \mathcal{U}]$	eigenbasis of T_G	Assumption 3.16	GD
3.4.3	dual	$L^2[\Theta, \mu; \mathcal{U}]$	linear splines	Assumption 3.18	PGD

where $I_{\ell_i} : y \in \mathcal{Y} \mapsto I_\ell(y, y_i)$ for any $i \in [n]$. Given that $(\hat{\alpha}_i)_{i=1}^n$ are solution to Problem 3.7, the resulting estimator is

$$\hat{h} = \frac{1}{\lambda n} \sum_{i=1}^{n} K(\cdot, x_i)\hat{\alpha}_i, \tag{3.8}$$

One can notice that Theorem 2.44 provides a representer expression, with the additional benefit of having access to a *dual* problem, which can be easier to solve than the direct plug-in of Equation (3.6) into Problem 3.5. The search for the solution in $\mathcal{H}_K$ is again transferred to the search for optimal coefficients in $\mathcal{Y}^n$, which may be intractable in practice for infinite-dimensional $\mathcal{Y}$.

Solving Problem 3.5 is thus a challenging task, and requires dedicated techniques. Two questions are raised here:

- How can we represent the coefficients $(\alpha_i)_{i=1}^n$ so that the resulting estimator admits a finite-dimensional parametrization ?
- How can we learn the best estimator when we cannot even compute exactly each $I_\ell(h(x_i), y_i)$?

These two questions are interlinked, and we propose in this chapter several algorithms to solve Problem 3.5. In Section 3.2 we focus on the particular case of the square loss, for which estimators enjoy closed-form solution. This is an extension of existing results on kernel ridge regression for real-valued output functions. In Section 3.3, we devise general optimization algorithms for any integral loss, using the primal version of Problem 3.5. In particular, we show that sampling the integral loss guarantees a finite parametrization of the estimator by means of a *double representer theorem* and allows at the same time the approximate computation of the loss function, enabling gradient descent algorithms. We then consider operator random Fourier features (ORFF) models that naturally enjoy finite parameterization and propose a stochastic gradient descent algorithm bypassing the need to compute each $I_\ell(h(x_i), y_i)$. Finally, Section 3.4 is devoted to the exploration of dual algorithms. We investigate properties of the Fenchel-Legendre conjugate of integral losses on $\mathcal{Y}$, and show that the flexibility of the vv-RKHS modeling is well-suited to their computation. We then propose to solve Problem 3.7 by leveraging appropriate representation bases for the dual variables, either based on the eigendecomposition of T_G or using linear splines depending on the compatibility between the loss function and the output vv-RKHS $\mathcal{H}_G$.

3.2 The Special Square Loss Case

In the following, we derive closed-form expressions for the solution of the regularized empirical risk minimization with the square loss in two different settings: in case of modeling the observations $(y_i)_{i=1}^n$ as functions (Section 3.2.1) or to assume access to a sampled version of these (Section 3.2.2). Ultimately, even when modeling the outputs as functions they are represented as multi-dimensional vectors and relevant quantities have to be estimated from these vectors. This section corresponds to the extension of (Lian, 2007; Kadri et al., 2010, 2016) beyond real-valued output functions.

3.2.1 Functional Observation Case

In the case where $\ell(\theta, u, v) = \frac{1}{2}\|u - v\|_{\mathcal{U}}^2$, one can notice that

$$I_\ell(f, g) = \frac{1}{2}\left\|f - g\right\|_{L^2[\Theta,\mu;\mathcal{U}]}^2 \text{ for } \forall f, g \in L^2[\Theta, \mu; \mathcal{U}].$$

Taking $\mathcal{Y} = L^2[\Theta, \mu; \mathcal{U}]$ and leveraging the view $K = k_{\mathcal{X}} T_G$, leads to an easy-to-compute Fenchel-Legendre transform:

$$I_{\ell_i}^\star(\alpha) = \frac{1}{2}\|\alpha\|_{\mathcal{Y}}^2 + \langle \alpha, y_i \rangle_{\mathcal{Y}}$$

for all $\alpha \in \mathcal{Y}$. This is a direct consequence of the squared norm being the only fixed point of the Fenchel-Legendre transform (Bauschke et al., 2011), as well as the fact that for any $f \colon \mathcal{Y} \to \mathbb{R}$ and $y \in \mathcal{Y}$, $f(\cdot - y)^\star = f^\star + \langle \cdot, y \rangle_{\mathcal{Y}}$. Consequently, Problem 3.7 writes as

$$\inf_{(\alpha_i)_{i=1}^n \in \mathcal{Y}^n} \sum_{i=1}^n \frac{1}{2}\left\|\alpha_i\right\|_{\mathcal{Y}}^2 - \langle \alpha_i, y_i \rangle_{\mathcal{Y}} + \frac{1}{2\lambda n} \sum_{i,j=1}^n k_{\mathcal{X}}(x_i, x_j) \left\langle \alpha_i, T_G \alpha_j \right\rangle_{\mathcal{Y}}. \tag{3.9}$$

Before diving deeper into the solution, we introduce two useful notations: we denote by $\mathbf{K}_{\mathcal{X}} \in \mathcal{M}_n(\mathbb{R})$ the Gram matrix of points $(x_i)_{i=1}^n$ with kernel $k_{\mathcal{X}}$ and by $\mathbf{y} = \left[y_i\right]_{i=1}^n \in \mathcal{Y}^n$ the aggregation of the observed outputs $(y_i)_{i=1}^n$.

Proposition 3.1 (Closed-Form Functional Ridge)**.** *The solution $\hat{h}$ of Problem 3.5 with the square loss is given by*

$$\hat{h} = \frac{1}{\lambda n} \sum_{i=1}^n k_{\mathcal{X}}(\cdot, x_i) T_G \hat{\alpha}_i,$$

where $\hat{\boldsymbol{\alpha}} := \left[\hat{\alpha}_i\right]_{i=1}^n \in \mathcal{Y}^n$ is the solution of the linear system

$$\left(\mathbf{K}_{\mathcal{X}} \otimes T_G + \lambda n \operatorname{Id}_{\mathcal{Y}^n}\right) \hat{\boldsymbol{\alpha}} = n\lambda \mathbf{y}. \tag{3.10}$$

Proof Using the $\mathbf{K}_{\mathcal{X}}$ and $\mathbf{y}$ notations, define

$$\mathcal{J}(\boldsymbol{\alpha}) = \frac{1}{2}\|\boldsymbol{\alpha}\|_{\mathcal{Y}^n} - \langle \boldsymbol{\alpha}, \mathbf{y} \rangle_{\mathcal{Y}^n} + \frac{1}{2\lambda n}\left\langle \boldsymbol{\alpha}, \left(\mathbf{K}_{\mathcal{X}} \otimes T_G\right)\boldsymbol{\alpha} \right\rangle_{\mathcal{Y}^n}, \quad \boldsymbol{\alpha} \in \mathcal{Y}^n.$$

Using this notation Problem 3.9 corresponds to solving $\inf_{\boldsymbol{\alpha} \in \mathcal{Y}^n} \mathcal{J}(\boldsymbol{\alpha})$. $\mathcal{J}$ is a (strongly) convex Gâteaux differentiable function (see Definition 2.8), and

$$\nabla \mathcal{J}(\boldsymbol{\alpha}) = \boldsymbol{\alpha} - \mathbf{y} + \frac{1}{\lambda n}\left(\mathbf{K}_{\mathcal{X}} \otimes T_G\right)\boldsymbol{\alpha}.$$

Setting the gradient to zero at optimum $\hat{\boldsymbol{\alpha}} \in \mathcal{Y}^n$, one gets

$$\hat{\boldsymbol{\alpha}} - \mathbf{y} + \frac{1}{\lambda n}\left(\mathbf{K}_\mathcal{X} \otimes T_G\right)\hat{\boldsymbol{\alpha}} = \mathbf{0}$$

which translates to

$$\left(\mathbf{K}_\mathcal{X} \otimes T_G + \lambda n \operatorname{Id}_{\mathcal{Y}^n}\right)\hat{\boldsymbol{\alpha}} = \lambda n \mathbf{y}.$$

■

Remark 3.2 (Relationship to Kadri et al. (2016))**.** *The problem considered in Kadri et al. (2016) is a real-valued function-to-function regression problem using the squared norm in $L^2[0,1]$ ($\mathcal{U} = \mathbb{R}$, μ is the Lebesgue measure on $\Theta = [0,1]$). The authors leverage a representer theorem associated to a directional derivative argument to obtain a closed-form solution which corresponds to the solution presented here with $\mathbf{A} = 1 \in \mathbb{R}$.*

Solving Equation (3.10) can be carried out at the price of an operator inversion:

$$\hat{\boldsymbol{\alpha}} = \lambda n \left(\mathbf{K}_\mathcal{X} \otimes T_G + \lambda n \operatorname{Id}_{\mathcal{Y}^n}\right)^{-1} \mathbf{y}. \tag{3.11}$$

As a reminder, for the OVK G considered here we have $T_G = \mathbf{A} \otimes T_{k_\Theta}$. Computing the inverse operator in Equation (3.11) is intractable in practice for the large majority of output kernels k_Θ because of the complexity of T_{k_Θ}. However, one can turn to spectral methods to get an approximate solution. This idea was first exploited in Kadri et al. (2016) in the setting $\mathcal{U} = \mathbb{R}$, $\Theta = [0,1]$ and can be adapted to the more general case proposed here. It relies on the following observation: the eigen decomposition of $\mathbf{K}_\mathcal{X} \otimes T_G$ can be obtained from the eigen decompositions of $\mathbf{K}_\mathcal{X}$ and T_G. Indeed, if $(v, \psi) \in \mathbb{R}^n \times \mathcal{Y}$ are respectively eigenvectors of $\mathbf{K}_\mathcal{X}$ and T_G with associated eigenvalues (σ, λ) then $v \otimes \psi$ is an eigenvector of $\mathbf{K}_\mathcal{X} \otimes T_G$ with eigenvalue $\sigma\lambda$ and all eigenvectors of $\mathbf{K}_\mathcal{X} \otimes T_G$ can be obtained this way.

Let us assume access to the eigen decomposition $(\lambda_j, \psi_j)_{j=1}^m$ of rank m of the integral operator T_G, similarly obtained using the eigen decomposition of $\mathbf{A}$ and T_{k_Θ}. A discussion about the eigen decomposition of T_{k_Θ} is presented in Section 2.2.1. Let $(\sigma_i, v_i)_{i=1}^n \in (\mathbb{R} \times \mathbb{R}^n)^n$ be the eigen decomposition of the matrix $\mathbf{K}_\mathcal{X}$ obtained by for instance using SVD. We propose to use

$$\hat{\boldsymbol{\alpha}} = \sum_{i=1}^{n}\sum_{j=1}^{m} \frac{1}{\lambda + \sigma_i\lambda_j} \langle \mathbf{y}, v_i \otimes \psi_j \rangle_{\mathcal{Y}^n} v_i \otimes \psi_j. \tag{3.12}$$

The cost associated to the eigen decomposition of $\mathbf{K}_\mathcal{X}$ (respectively $\mathbf{A}$) scales as $\mathcal{O}(n^3)$ (respectively $\mathcal{O}(s^3)$). Finding the first m pairs of (eigenvalue, eigenvector) of T_G can be hard and costly for general output kernels k_Θ. One can then turn to using approximate eigen decomposition as presented in Example 2.23 or use the eigen decomposition of a Random Fourier Feature model from Example 2.30.

Remark 3.3. *Equation (3.12) involves the scalar products $\langle y_i, \psi_j \rangle_\mathcal{Y}$, which have to be estimated from the data.*

We now investigate ways to solve Problem 3.5 in the square loss setting when one does not assume full access to the observed output functions $(y_i)_{i=1}^n$ referred to as the *partial observation setting*.

3.2.2 Partially Observed Case

In various cases, due to the functional nature of the $(y_i)_{i=1}^n$, one cannot assume to have access to the whole functions (in the analogous signal sense, or being able to query at any $\theta \in \Theta$), and that makes the quantity $\mathcal{R}_{\mathcal{S}}$ impossible to compute. This will be the case in particular for the emotion transfer setting in Chapter 6. In this *partially observed setting*, we assume that each y_i is observed at certain number ($m \in \mathbb{N}^*$) of locations $(\theta_{ij})_{j=1}^m$.

Remark 3.4 (General Case)**.** *One could have considered that the number of observed locations varies with the numbering of the observation i, so that each y_i is observed at locations $(\theta_{ij})_{j=1}^{m_i}$. To keep the notations simple, we consider that the number of locations observed per sample is fixed ($m = m_i$ for all i).*

We moreover assume that for $\forall i \in [n]$, $(\theta_{ij})_{j=1}^m \overset{\text{i.i.d.}}{\sim} \mu$, and define the *sampled empirical risk* as

$$\widetilde{\mathcal{R}}_{\mathcal{S}}(h) := \frac{1}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} \frac{1}{2} \left\| y_i(\theta_{ij}) - h(x_i)(\theta_{ij}) \right\|_{\mathcal{U}}^2 . \tag{3.13}$$

. This choice ensures that $\widetilde{\mathcal{R}}_{\mathcal{S}}(h)$ converges towards $\mathcal{R}(h)$ when the number of samples and locations grows towards infinity.

Remark 3.5 (Re-weighting Scheme)**.** *In the case where the locations are not sampled with probability μ, then there is no convergence of the sampled empirical risk towards the true risk. To remedy this bottleneck, one can consider a re-weighted sampled empirical risk of the form*

$$\widetilde{\mathcal{R}}_{\mathcal{S}}(h) := \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{m} \frac{\eta_{ij}}{2} \left\| y_i(\theta_{ij}) - h(x_i)(\theta_{ij}) \right\|_{\mathcal{U}}^2 ,$$

where $(\eta_{ij})_{i,j=1}^{n,m}$ encodes a re-weighting scheme ensuring that the proposed sampled empirical risk converges towards the risk in Problem 3.4. The solution developed below can then be adapted to such settings.

We now introduce the problem based on the minimization of the sampled empirical risk:

$$\inf_{h \in \mathcal{H}_K} \widetilde{\mathcal{R}}_{\mathcal{S}}(h) + \frac{\lambda}{2} \|h\|_{\mathcal{H}_K}^2 . \tag{3.14}$$

It is advantageous in this case to adopt the view $K = k_{\mathcal{X}} \operatorname{Id}_{\mathcal{H}_G}$ as it helps in the derivation of the dual problem of Problem 3.14.

Proposition 3.6. *The solution of Problem 3.14 is given by*

$$\hat{h}(x)(\theta) = \frac{1}{\lambda nm} \sum_{i=1}^{n} \sum_{j=1}^{m} k_{\mathcal{X}}(x, x_i) k_{\Theta}(\theta, \theta_j) \mathbf{A} \hat{\alpha}_{ij}, \quad \forall (x, \theta) \in \mathcal{X} \times \Theta, \tag{3.15}$$

with $(\hat{\alpha}_{ij})_{i,j=1}^{n,m} \in \mathcal{U}^{nm}$ being the solution of the dual problem

$$\begin{aligned} \inf_{(\alpha_{ij})_{i,j=1}^{n,m} \in \mathcal{U}^{nm}} & \sum_{i=1}^{n} \sum_{j=1}^{m} \frac{1}{2} \left\| \alpha_{ij} \right\|_{\mathcal{U}}^2 - \left\langle y_i(\theta_{ij}), \alpha_{ij} \right\rangle_{\mathcal{U}} \\ & + \frac{1}{2\lambda nm} \sum_{i_1, i_2=1}^{n} \sum_{j_1, j_2=1}^{m} k_{\mathcal{X}}(x_{i_1}, x_{i_2}) k_{\Theta}(\theta_{i_1 j_1}, \theta_{i_2 j_2}) \left\langle \alpha_{i_1 j_1}, \mathbf{A} \alpha_{i_2 j_2} \right\rangle_{\mathcal{U}} . \end{aligned} \tag{3.16}$$

Proof We introduce slack variables $\boldsymbol{\xi} := (\xi_{ij})_{i,j=1}^{n,m} \in \mathcal{U}^{nm}$ so that Problem 3.14 can be rewritten as

$$\inf_{h \in \mathcal{H}_K, \boldsymbol{\xi} \in \mathcal{U}^{nm}} \sum_{i,j=1}^{n,m} \ell_{ij}(\xi_{ij}) + \frac{\lambda nm}{2} \left\|h\right\|_{\mathcal{H}_K}^2 \\ s.t.\ \xi_{ij} = h(x_i)(\theta_{ij}) \quad \forall (i,j) \in [n] \times [m]$$

where we adopt the notation $\ell_{ij} = \left\| \cdot - y_i(\theta_{ij}) \right\|_{\mathcal{U}}^2$. Denoting by $\boldsymbol{\alpha} := (\alpha_{ij})_{i,j=1}^{n}$ the dual variables, the Lagrangian associated to the problem is

$$\mathcal{L}\left(h, \boldsymbol{\xi}, \boldsymbol{\alpha}\right) = \sum_{i,j=1}^{n,m} \ell_{ij}(\xi_{ij}) + \frac{\lambda nm}{2} \left\|h\right\|_{\mathcal{H}_K}^2 + \left\langle \alpha_{ij}, \xi_{ij} - h(x_i)(\theta_{ij}) \right\rangle_{\mathcal{U}}. \tag{3.17}$$

The dual function then writes as

$$\begin{aligned} g(\boldsymbol{\alpha}) &= \inf_{h \in \mathcal{H}_K, \boldsymbol{\xi} \in \mathcal{U}^{nm}} \mathcal{L}\left(h, \boldsymbol{\xi}, \boldsymbol{\alpha}\right) \\ &= \inf_{\boldsymbol{\xi} \in \mathcal{U}^{nm}} \sum_{i,j=1}^{n,m} \ell_{ij}(\xi_{ij}) + \left\langle \alpha_{ij}, \xi_{ij} \right\rangle_{\mathcal{U}} + \inf_{h \in \mathcal{H}_K} \frac{\lambda nm}{2} \left\|h\right\|_{\mathcal{H}_K} - \sum_{i,j=1}^{n,m} \left\langle \alpha_{ij}, h(x_i)(\theta_{ij}) \right\rangle_{\mathcal{U}}. \end{aligned}$$

In the first term, we can recognize the Fenchel-Legendre conjugate of the squared norm:

$$\inf_{\boldsymbol{\xi} \in \mathcal{U}^{nm}} \sum_{i,j=1}^{n,m} \ell_{ij}(\xi_{ij}) + \left\langle \alpha_{ij}, \xi_{ij} \right\rangle_{\mathcal{U}} = - \sum_{i,j=1}^{n,m} \ell_{ij}^{\star}(-\alpha_{ij}) = - \left(\sum_{i,j=1}^{n,m} \frac{1}{2} \left\|\alpha_{ij}\right\|_{\mathcal{U}}^2 - \left\langle \alpha_{ij}, y_i(\theta_{ij}) \right\rangle_{\mathcal{U}} \right).$$

For the second term, differentiating with respect to h yields the optimum

$$\hat{h} = \frac{1}{\lambda nm} \sum_{i,j=1}^{n,m} K(\cdot, x_i) G(\cdot, \theta_{ij}) \alpha_{ij}$$

by using the reproducing property successively in $\mathcal{H}_K$ and $\mathcal{H}_G$. It then holds that

$$\begin{aligned} &\frac{\lambda nm}{2} \left\|\hat{h}\right\|_{\mathcal{H}_K} - \sum_{i,j=1}^{n,m} \left\langle \alpha_{ij}, \hat{h}(x_i)(\theta_{ij}) \right\rangle_{\mathcal{U}} \\ &\qquad = -\frac{1}{2\lambda nm} \sum_{i_1, i_2 = 1}^{n} \sum_{j_1, j_2 = 1}^{m} k_{\mathcal{X}}(x_{i_1}, x_{i_2}) k_{\Theta}(\theta_{i_1 j_1}, \theta_{i_2 j_2}) \left\langle \alpha_{i_1 j_1}, \mathbf{A} \alpha_{i_2 j_2} \right\rangle. \end{aligned}$$

Finally the dual problem writes as

$$\sup_{\boldsymbol{\alpha} \in \mathcal{U}^{nm}} g(\boldsymbol{\alpha}) = - \inf_{\boldsymbol{\alpha} \in \mathcal{U}^{nm}} -g(\boldsymbol{\alpha})$$

and by plugging in the expression of $g(\boldsymbol{\alpha})$ one arrives at Problem 3.16. ■

Proposition 3.6 allows to parameterize the solution to Problem 3.14 by a finite number of elements of the finite-dimensional space $\mathcal{U}$. Thus, the solution can be represented on a computer, and solving Problem 3.16 boils down to the minimization of a quadratic form, solvable in closed-form.

Particularly, let us introduce three matrices encoding information about the dataset and dual variables. Let $\mathbf{K} \in \mathcal{M}_{nm}(\mathbb{R})$ be the Gram matrix associated to the problem, whose entries are defined as

$$\mathbf{K}_{m(i_1-1)+j_1, m(i_2-1)+j_2} = k_{\mathcal{X}}(x_{i_1}, x_{i_2}) k_{\Theta}(\theta_{i_1 j_1}, \theta_{i_2 j_2}), \quad (i_1, i_2) \in [n]^2, (j_1, j_2) \in [m]^2. \tag{3.18}$$

Let $\mathbf{Y} \in \mathcal{M}_{nm,s}(\mathbb{R})$ be a matrix gathering observations

$$\forall (i,j) \in [n] \times [m], \quad \mathbf{Y}_{m(i-1)+j,:} = y_i(\theta_{ij})^\top. \tag{3.19}$$

The dual variables $(\alpha_{ij})_{i,j=1}^{n,m} \in \mathcal{U}^{nm}$ are collected in a matrix $\boldsymbol{\alpha} \in \mathcal{M}_{nm,s}(\mathbb{R})$ such that

$$\forall (i,j) \in [n] \times [m], \quad \boldsymbol{\alpha}_{m(i-1)+j,:} = \alpha_{ij}^\top.$$

Using these notations, Problem 3.16 writes as

$$\inf_{\boldsymbol{\alpha} \in \mathcal{M}_{nm,s}(\mathbb{R})} \operatorname{Tr}\left(\frac{1}{2} \boldsymbol{\alpha}\boldsymbol{\alpha}^\top - \boldsymbol{\alpha}\mathbf{Y}^\top + \frac{1}{2\lambda nm} \mathbf{K}\boldsymbol{\alpha}\mathbf{A}\boldsymbol{\alpha}^\top \right). \tag{3.20}$$

It turns out that Problem 3.20 is akin to a *Sylvester equation* for which solutions can be computed in closed-form, as it is summarized in the following lemma.

Lemma 3.7 (Optimization task for $\boldsymbol{\alpha}$)**.** *The solution $\hat{\boldsymbol{\alpha}}$ of Problem 3.20 is the solution of the following linear equation*

$$\mathbf{K}\hat{\boldsymbol{\alpha}}\mathbf{A} + \lambda nm \hat{\boldsymbol{\alpha}} = \lambda nm \mathbf{Y}. \tag{3.21}$$

When $\mathbf{A} = \mathrm{Id}_s$, the solution is analytic:

$$\hat{\boldsymbol{\alpha}} = \lambda nm \left(\mathbf{K} + \lambda nm \, \mathrm{Id}_{nm} \right)^{-1} \mathbf{Y}. \tag{3.22}$$

Proof The statement follows by setting the gradient of the objective function to zero. Indeed,

$$\nabla \left(\boldsymbol{\alpha} \mapsto \frac{1}{2} \operatorname{Tr}\left(\boldsymbol{\alpha}\boldsymbol{\alpha}^\top \right) \right) = \boldsymbol{\alpha}, \qquad \nabla \left(\boldsymbol{\alpha} \mapsto \operatorname{Tr}\left(-\boldsymbol{\alpha}\mathbf{Y}^\top \right) \right) = -\mathbf{Y},$$
$$\nabla \left(\boldsymbol{\alpha} \mapsto \frac{1}{2\lambda nm} \operatorname{Tr}\left(\mathbf{K}\boldsymbol{\alpha}\mathbf{A}\boldsymbol{\alpha}^\top \right) \right) = \frac{1}{\lambda nm} \mathbf{K}\boldsymbol{\alpha}\mathbf{A}.$$

■

Remark 3.8. *Lemma 3.7 writes as a generalization of the method developed in Lian (2007); Kadri et al. (2010) to the vector-valued outputs case and non-regular grid.*

If the matrix $\mathbf{A}$ is not the identity matrix, one can still solve Equation (3.21) by using dedicated solvers (El Guennouni et al., 2002). We now move on to the general case of Problem 3.5 when the loss function no longer permits explicit closed-form solution.

3.3 Solving in the Primal

In this section, we consider the general case of any proper, convex lower semicontinuous integral loss and propose optimization algorithms based on the primal formulation of Problem 3.5, that reads as

$$\inf_{h \in \mathcal{H}_K} \frac{1}{n} \sum_{i=1}^{n} I_\ell(h(x_i), y_i) + \frac{\lambda}{2} \|h\|^2_{\mathcal{H}_K}.$$

The classical representer theorem from Theorem 2.43 states that there exist $(\hat{\alpha}_i)_{i=1}^n \in \mathcal{Y}^n$ such that the solution $\hat{h}$ to Problem 3.5 takes the form

$$\hat{h} = \sum_{i=1}^{n} K(\cdot, x_i)\hat{\alpha}_i.$$

However in our case, $\mathcal{Y}$ is a (potentially) infinite-dimensional Hilbert space, which raises the problem of being able to provide an expression of $\hat{h}$ that can be explicitly calculated. Moreover, the computation of the loss function I_ℓ is not straightforward, as it involves an integral over the Θ domain. To circumvent these difficulties, we propose below various optimization algorithms dedicated to solving Problem 3.5. Section 3.3.1 is devoted to solving an approximated problem based on a sampling of the integral loss I_ℓ for which a finite-dimensional parametrization can be obtained by means of a *double representer theorem*, killing two birds with one stone. In Section 3.3.2, we explore the use of Random Fourier Features, with the double benefit of lowering the computational cost associated to the solution as well as benefiting by construction of a finite-dimensional representation of the solution.

3.3.1 Sampling Schemes and Representer Theorems

Sampling I_ℓ appears as a natural approach to solve Problem 3.5. It provides the immediate benefit of making the loss function computable; the resulting *sampled empirical risk* is

$$\widetilde{\mathcal{R}}_{\mathcal{S}}(h) := \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{m} \eta_{ij} \ell(\theta_{ij}, h(x_i)(\theta_{ij}), y_i(\theta_{ij})), \tag{3.23}$$

where $(\theta_{ij}, \eta_{ij})_{i,j=1}^{n,m}$ encodes information about a chosen sampling scheme. We are now interested in solving an approximated counterpart of Problem 3.5 based on using Equation (3.23) as a proxy for the empirical risk:

$$\inf_{h \in \mathcal{H}_K} \widetilde{\mathcal{R}}_{\mathcal{S}}(h) + \frac{\lambda}{2} \|h\|^2_{\mathcal{H}_K}. \tag{3.24}$$

Below we discuss various valid sampling schemes before stating the double representer theorem which allows to represent numerically the solution of Problem 3.24.

Monte-Carlo Sampling: The Monte-Carlo (MC) method simply consists in sampling i.i.d. random variables $(\theta_j)_{j=1}^m \overset{\text{i.i.d}}{\sim} \mu$ and estimating the integral I_ℓ by its empirical counterpart

$$\widetilde{I}_\ell(f, g) := \frac{1}{m} \sum_{j=1}^{m} \ell(\theta_j, f(\theta_j), g(\theta_j)).$$

The MC methods have a probabilistic error scaling as $\mathcal{O}\left(\frac{1}{\sqrt{m}}\right)$ (with constants depending on the variance of the integrand), which is independent from the dimensionality of Θ. As $\mathcal{R}_{\mathcal{S}}$ consists in the sum of n terms involving I_ℓ, a MC sampling scheme for $\mathcal{R}_{\mathcal{S}}$ is described by some $(\theta_{ij})_{i,j=1}^{n,m}$ where $\forall i \in [n]$, $(\theta_{ij})_{j=1}^{m} \overset{\text{i.i.d.}}{\sim} \mu$. By setting $\forall (i,j) \in [n] \times [m], \eta_{ij} = \frac{1}{m}$ we arrive at Equation (3.23). It is worth noting at this point that using $m = 1$ is an interesting choice: it amounts to a very loose approximation of each $I_\ell(h(x_i), y_i)$ but as the number of samples grows it still holds that $\lim_{n\to\infty} \mathcal{R}_{\mathcal{S}}(h) = \mathcal{R}(h)$.

Quasi Monte-Carlo Sampling: The idea behind quasi Monte-Carlo (QMC) sampling is to use a low discrepency sequence to estimate the integral. Such sequences provide successively finer uniform partitions of the compact space Θ and guarantee that the proportion of points falling in a subset $B \subseteq \Theta$ is close to $\mu(B)$. We give an example of such sequence called the *Sobol* sequence (Sobol, 1967), exemplified in case of $\Theta = [0,1]$. Start with $\theta_1 = \frac{1}{2}$. Then build $\theta_2 = \frac{1}{4}$, and $\theta_3 = \frac{3}{4}$. Then $\theta_4 = \frac{1}{8}$, $\theta_5 = \frac{3}{8}$, and so on as to obtain a regular covering of Θ. Such sequence can be adapted to higher dimensional Θ.

Let $\eta_j = \frac{1}{m} F^{-1}(\theta_j)$ where μ is assumed to be absolutely continuous *w.r.t.* the Lebesgue measure and F is its associated *c.d.f.* , and $(\theta_j)_{j=1}^m$ is a QMC sequence. We then define

$$\widetilde{I}_\ell(f,g) = \sum_{j=1}^{m} \eta_j \ell(\theta_j, f(\theta_j), g(\theta_j)).$$

The QMC methods has an approximation error scaling as $\mathcal{O}\left(\frac{(\log m)^p}{m}\right)$, which is faster that what is obtained using MC for many practical values of (m,p). However, the constants involve the *Hardy-Kraus* variation of the integrand, which is often hard to estimate. More details about this are given in Section 5.3. The QMC sampling scheme leads to Equation (3.23) with $(\eta_{ij}, \theta_{ij})_{i,j=1}^{n,m}$ that are independent of i and simply consists in stacking the same locations and coefficients defined by μ and the Sobol sequence.

Kernel Quadrature Rules: Quadrature rules are suited for low-dimensional Θ. They are described in Section 2.2.1 and lead to an approximation

$$\widetilde{I}_\ell(f,g) = \sum_{j=1}^{m} \eta_j \ell(\theta_j, f(\theta_j), g(\theta_j)),$$

where $(\eta_j, \theta_j)_{j=1}^m$ are the set of weights and locations produced by the quadrature rule. By stacking n of them we recover Equation (3.23).

To solve Problem 3.24, we adopt the view $K = k_{\mathcal{X}} \operatorname{Id}_{\mathcal{H}_G}$, as it allows to apply a representer theorem on the coefficients $(\alpha_i)_{i=1}^n \in \mathcal{Y}^n$ themselves, as expressed in the theorem below.

Theorem 3.9 (Double Representer)**.** *Problem 3.24 has a unique solution $\hat{h}$ and it takes the form*

$$\hat{h}(x)(\theta) = \sum_{i=1}^{n} \sum_{j=1}^{m} k_{\mathcal{X}}(x, x_i) k_\Theta(\theta, \theta_{ij}) \mathbf{A} \hat{\alpha}_{ij}, \quad \forall (x,\theta) \in \mathcal{X} \times \Theta \tag{3.25}$$

for some coefficients $\hat{\alpha}_{ij} \in \mathcal{U}$ with $i \in [n]$ and $j \in [m]$.

Proof Let $\mathcal{J}(h) = \widetilde{\mathcal{R}}_{\mathcal{S}}(h) + \frac{\lambda}{2}\left\|h\right\|^2_{\mathcal{H}_K}$ denote the objective function to minimize. For all $g \in \mathcal{H}_G$, let $K_x g$ denote the function defined by $(K_x g)(t) = K(t,x)g \; \forall t \in \mathcal{X}$. Similarly, for all $\alpha \in \mathcal{U}$, $G_\theta \alpha$ stands for the function $t \mapsto G(t,\theta)\alpha$ where $t \in \Theta$. Let us take the finite-dimensional subspace

$$E = \operatorname{span}\left(K_{x_i} G_{\theta_{ij}} \alpha \,:\, i \in [t], j \in [m], \alpha \in \mathcal{U}\right).$$

The space $\mathcal{H}_K$ can be decomposed as E and its orthogonal complement: $E \oplus E^\perp = \mathcal{H}_K$. The existence of $\hat{h}$ follows from the coercivity of $\mathcal{J}$ (i.e. $\mathcal{J}(h) \to +\infty$ as $\left\|h\right\|_{\mathcal{H}_K} \to +\infty$) which is the consequence of the quadratic regularizer and the lower boundedness of ℓ. Uniqueness comes from the strong convexity of the objective. Let us decompose $\hat{h} = \hat{h}_E + \hat{h}_{E^\perp}$, and take any $\alpha \in \mathcal{U}$. Then $\forall (i,j) \in [n] \times [m]$,

$$\left\langle \hat{h}_{E^\perp}(x_i)(\theta_{ij}), \alpha \right\rangle_{\mathcal{U}} \overset{(a)}{=} \left\langle \hat{h}_{E^\perp}(x_i), G_{\theta_{ij}}\alpha \right\rangle_{\mathcal{H}_G} \overset{(b)}{=} \left\langle \hat{h}_{E^\perp}, \underbrace{K_{x_i} G_{\theta_{ij}} \alpha}_{\in E} \right\rangle_{\mathcal{H}_K} \overset{(c)}{=} 0.$$

(a) follows from the reproducing property in $\mathcal{H}_G$, (b) is a consequence of the reproducing property in $\mathcal{H}_K$, and (c) comes from the decomposition $E \oplus E^\perp = \mathcal{H}_K$. This means that $\hat{h}_{E^\top}(x_i)(\theta_{ij}) = 0 \; \forall (i,j) \in [n] \times [m]$, and hence $\widetilde{\mathcal{R}}_{\mathcal{S}}(\hat{h}) = \widetilde{\mathcal{R}}_{\mathcal{S}}(\hat{h}_E)$. Since

$$\lambda \left\|\hat{h}\right\|^2_{\mathcal{H}_K} = \lambda \left(\left\|\hat{h}_E\right\|^2_{\mathcal{H}_K} + \left\|\hat{h}_{E^\perp}\right\|^2_{\mathcal{H}_K} \right) \geq \lambda \left\|\hat{h}_E\right\|^2_{\mathcal{H}_K}$$

we conclude that $\hat{h}_{E^\top} = 0$ and get that there exist coefficients $\hat{\alpha}_{ij} \in \mathcal{U}$ such that

$$\hat{h} = \sum_{i=1}^{n} \sum_{j=1}^{m} K_{x_i} G_{\theta_{ij}} \hat{\alpha}_{ij}.$$

This evaluates for all $(x, \theta) \in \mathcal{X} \times \Theta$ to

$$\hat{h}(x)(\theta) = \sum_{i=1}^{n} \sum_{j=1}^{m} k_{\mathcal{X}}(x, x_i) k_\Theta(\theta, \theta_{ij}) \mathbf{A} \hat{\alpha}_{ij}$$

as claimed in Equation (3.25). ■

The benefit of Theorem 3.9 is to provide a finite-dimensional parameterization of the solution $\hat{h}$. Equation (3.25) then allows to write the predictions of the model in a compact form involving the Gram matrix of the problem and the similarity matrix $\mathbf{A}$. To that end, define $\mathbf{K} \in \mathcal{M}_{nm}(\mathbb{R})$ to be the Gram matrix whose entries are defined as

$$\mathbf{K}_{m(i_1-1)+j_1, m(i_2-1)+j_2} = k_{\mathcal{X}}(x_{i_1}, x_{i_2}) k_\Theta(\theta_{i_1 j_1}, \theta_{i_2 j_2}), \; (i_1, i_2) \in [n]^2, (j_1, j_2) \in [m]^2. \tag{3.26}$$

Then, denoting by $\boldsymbol{\alpha} \in \mathcal{M}_{nm,s}$ the matrix encoding information about the coefficients such that for all $i, j \in [n] \times [m], \hat{\boldsymbol{\alpha}}_{i(m-1)+j,:} = \hat{\alpha}_{ij}^\top$, it holds that

$$\hat{h}(x_i)(\theta_{ij}) = (\mathbf{K}\hat{\boldsymbol{\alpha}}\mathbf{A})_{i(m-1)+j,:}, \quad (i,j) \in [n] \times [m]. \tag{3.27}$$

Remark 3.10. *Depending on the chosen sampling, the Gram matrices associated to the problem is structured differently. In particular, in the QMC and the quadrature rule schemes, the sampling locations are the same for all samples and the Gram matrix* $\mathbf{K}$ *has a tensorial structure* $\mathbf{K} = \mathbf{K}_{\mathcal{X}} \otimes \mathbf{K}_\Theta$ *which can be used to speed up computations, and reduce the memory footprint.*

Associated to the fact that $\widetilde{\mathcal{R}}_{\mathcal{S}}$ can be computed (the integration has been traded against a summation), we now have all ingredients needed to apply classical optimization tools to solve Problem 3.24. Indeed, defining

$$L(\mathbf{r}) := \frac{1}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m}\ell(\theta_{ij}, \mathbf{r}_{m(i-1)+j,:}, y_i(\theta_{ij})), \quad \mathbf{r} \in \mathcal{M}_{nm,s}$$

we can restate Problem 3.24 as

$$\inf_{\boldsymbol{\alpha}\in\mathcal{M}_{nm,s}} L(\mathbf{K}\boldsymbol{\alpha}\mathbf{A}) + \frac{\lambda}{2}\operatorname{Tr}\left(\mathbf{K}\boldsymbol{\alpha}\mathbf{A}\boldsymbol{\alpha}^\top\right). \tag{3.28}$$

If one assumes access to the subgradients of ℓ, then by the separability of L it holds that,

$$\partial L(\mathbf{r})_{m(i-1)+j,:} = \partial\ell(\theta_{ij}, \cdot, y_i(\theta_{ij}))(\mathbf{r}_{m(i-1)+j,:}), \quad \forall \mathbf{r} \in \mathcal{M}_{nm,s}, \forall (i,j) \in [n]\times[m].$$

Problem 3.28 can be then be tackled by (sub)gradient descent methods making use of

$$\partial\left(\boldsymbol{\alpha} \mapsto L(\mathbf{K}\boldsymbol{\alpha}\mathbf{A})\right) = \mathbf{K}\partial L(\mathbf{K}\boldsymbol{\alpha}\mathbf{A})\mathbf{A}, \quad \nabla\left(\boldsymbol{\alpha} \mapsto \frac{\lambda}{2}\operatorname{Tr}\left(\mathbf{K}\boldsymbol{\alpha}\mathbf{A}\boldsymbol{\alpha}^\top\right)\right) = \lambda\mathbf{K}\boldsymbol{\alpha}\mathbf{A}.$$

This allows to perform various gradient descent algorithms; the specific technique can be for instance vanilla (sub)gradient descent, accelerated gradient descent (Nesterov, 1983), or quasi-Newton methods (Zhu et al., 1997), depending on the regularity properties of ℓ.

3.3.2 Random Features Based Learning

Random Fourier features (RFF) were initially introduced for shift-invariant scalar kernels (Rahimi and Recht, 2007) before being extended to shift-invariant OVKs (Brault et al., 2016) under the *operator random Fourier features* (ORFF) framework. We refer to Section 2.2 for an overview on the topic. They allow to lower the computational burden associated to solving empirical risk minimization problems by providing finite-dimensional functional spaces whose functions approximate those in the original hypothesis space. In our case, applied to the OVK G, they also provide the additional benefit of bypassing the need to prove a representer theorem for the coefficients $(\alpha_i)_{i=1}^n$ in Equation (3.6), since by construction the RKHSs associated to random feature maps are finite-dimensional.

We choose to work in output with an approximated OVK $\widetilde{G}$ obtained by applying the ORFF methodology. We call $\widetilde{\Phi}\colon \Theta \to \mathcal{L}(\mathcal{U}, \mathcal{V})$ its feature map where $\mathcal{V}$ is the associated finite-dimensional feature space. As a reminder, $\mathcal{V}$ is of dimension $2ms$ where m is the chosen number of random features chosen to approximate the kernel k_Θ. We refer to Equation (2.25) for the precise construction of this feature map, the key property exploited is that each function $\alpha \in \mathcal{H}_{\widetilde{G}}$ can be written

$$\alpha = \widetilde{\Phi}(\cdot)^\sharp v \tag{3.29}$$

for some $v \in \mathcal{V}$. This provides a finite-dimensional parametrization of the space $\mathcal{H}_{\widetilde{G}}$, which we shall use in combination with Theorem 2.43. Indeed, we know that the solution of Problem 3.5 applied to the vv-RKHS associated to $\widetilde{K} = k_\chi \operatorname{Id}_{\mathcal{H}_{\widetilde{G}}}$ writes as

$$\hat{h} = \sum_{i=1}^{n} k_\chi(\cdot, x_i)\hat{\alpha}_i \tag{3.30}$$

for some $(\hat{\alpha}_i)_{i=1}^n \in \mathcal{H}_{\widetilde{G}}$. By exploiting Equation (3.29), Problem 3.5 becomes

$$\inf_{(v_i)_{i=1}^n \in \mathcal{V}^n} \sum_{i=1}^n I_\ell \left(\sum_{j=1}^n k_\mathcal{X}(x_i, x_j) \widetilde{\Phi}(\cdot)^\sharp v_j, y_i \right) + \frac{\lambda}{2} \sum_{i,j=1}^n k_\mathcal{X}(x_i, x_j) \left\langle v_i, v_j \right\rangle_\mathcal{V}. \tag{3.31}$$

This can be expressed in a more compact form by using tensor products. Indeed, by writing $\mathbf{v} := (v_i)_{i=1}^n \in \mathcal{V}^n$ one gets that

$$\sum_{j=1}^n k_\mathcal{X}(x_i, x_j) \widetilde{\Phi}(\cdot)^\sharp v_j = \left[\left(\mathbf{K}_\mathcal{X} \otimes \widetilde{\Phi}(\cdot)^\sharp \right) \mathbf{v} \right]_i, \ \forall i \in [n]. \tag{3.32}$$

We can then recognize in Problem 3.31 a general minimization problem of the form

$$\min_{\mathbf{v} \in \mathcal{V}^n} \ \mathbb{E}_{\theta \sim \mu} \underbrace{\left[\frac{1}{n} \sum_{i=1}^n \ell \left(\theta, \left[\left(\mathbf{K}_\mathcal{X} \otimes \widetilde{\Phi}(\theta)^\sharp \right) \mathbf{v} \right]_i, y_i(\theta) \right) + \frac{\lambda}{2} \operatorname{Tr} \left(\mathbf{K}_\mathcal{X} \mathbf{v} \mathbf{v}^\top \right) \right]}_{:= \mathcal{J}(\theta, \mathbf{v})}. \tag{3.33}$$

These so-called stochastic optimization tasks are often encountered in machine learning, and can be solved using gradient methods based on stochastic approximations (Bottou, 1991).

We propose in Algorithm 3.1 a randomized optimization algorithm based on stochastic gradient descent to obtain $\hat{\mathbf{v}}$.

Algorithm 3.1 Stochastic Gradient Descent with ORFFs

input : ORFF feature map $\widetilde{\Phi}$, input Gram matrix $\mathbf{K}_\mathcal{X}$, gradient steps $(\gamma_t)_{t=0}^{T-1}$
init : $\mathbf{v}^{(0)} = \mathbf{0} \in \mathcal{V}^n$
for *epoch* t *from* 0 *to* $T-1$ **do**
  `// sampling step`
  Sample $\theta \sim \mu$
  `// gradient step`
  $\mathbf{v}^{(t+1)} = \mathbf{v}^{(t)} - \gamma_t \partial_\mathbf{v} \mathcal{J}(\theta, \mathbf{v}^{(t)})$
return $\mathbf{v}^{(T)}$

Remark 3.11. *This algorithm describes the vanilla stochastic gradient descent. Depending on the regularities of $\mathcal{J}$, one can turn to more involved optimization algorithm employing* e.g. *Nesterov acceleration (Nesterov, 1983) or any improvement over the vanilla version. The stepsizes $(\gamma_t)_{t=0}^{T-1}$ are also to be chosen depending on the problem to ensure convergence of the algorithm, in our case the objective is strongly convex thanks to the regularizer so that a sufficiently small constant stepsize ensures convergence (Bottou et al., 2018).*

Double ORFF: If some application stems into large-scale learning, with n being so big that Algorithm 3.1 becomes impossible to employ, one can make use of the RFF trick on both kernels $k_\mathcal{X}$ and G. This is possible when the input space $\mathcal{X}$ is the Euclidean space $\mathbb{R}^d$, which we assume in what follows. This amounts to working in the vv-RKHS associated to the OVK $\widetilde{K} = \tilde{k}_\mathcal{X} \operatorname{Id}_{\mathcal{H}_{\widetilde{G}}}$ where $\widetilde{G}$ and $\tilde{k}_\mathcal{X}$ are respectively ORFF and RFF

approximations of kernels $k_{\mathcal{X}}$ and G. This vv-RKHS is of finite dimension $4m_{\mathcal{X}}m_{\Theta}s$ where $m_{\mathcal{X}}$ and m_{Θ} are the number of random features used for approximating $k_{\mathcal{X}}$ and k_{Θ}. We denote by $\tilde{\phi}_{\mathcal{X}}$ and $\widetilde{\Phi}_{\Theta}$ the corresponding finite-dimensional feature maps, the dependence to each input space being made explicit for clarity. There is then a one-to-one correspondence between function in $\mathcal{H}_{\widetilde{K}}$ and $\mathcal{M}_{2m_{\mathcal{X}},2m_{\Theta}s}(\mathbb{R})$: $\forall h \in \mathcal{H}_{\widetilde{K}}$, $\exists \boldsymbol{\beta} \in \mathcal{M}_{2m_{\mathcal{X}},2m_{\Theta}s}(\mathbb{R})$ such that

$$h(x)(\theta) = \tilde{\phi}_{\mathcal{X}}(x)^{\sharp}\boldsymbol{\beta}\widetilde{\Phi}_{\Theta}(\theta), \quad (x,\theta) \in \mathcal{X} \times \Theta. \tag{3.34}$$

With this full parametrization of the model, Problem 3.5 writes as

$$\min_{\boldsymbol{\beta} \in \mathcal{M}_{2m_{\mathcal{X}},2m_{\Theta}s}(\mathbb{R})} \mathbb{E}_{\theta\sim\mu}\mathbb{E}_{i\sim\mathrm{Unif}([n])} \underbrace{\left[\ell\left(\theta, \tilde{\phi}_{\mathcal{X}}(x_i)^{\sharp}\boldsymbol{\beta}\widetilde{\Phi}_{\Theta}(\theta), y_i(\theta)\right) + \frac{\lambda}{2}\,\mathrm{Tr}\left(\boldsymbol{\beta}\boldsymbol{\beta}^{\top}\right)\right]}_{:=\mathcal{J}_i(\theta,\boldsymbol{\beta})}. \tag{3.35}$$

One can employ a doubly stochastic gradient descent to solve Problem 3.5 as proposed in Algorithm 3.2. The algorithm is doubly stochastic as it proposes to first sample $i \in [n]$, and then perform a SGD step with loss function $\mathcal{J}_i(\theta, \boldsymbol{\beta})$. Previous attempts at doubly stochastic algorithms in kernel based learning include Dai et al. (2014), where the random features are drawn sequentially as the algorithm goes on - contrary to their setting, we work with a fixed number of these random features and the double stochasticity comes from recognizing the objective function as an expectation over $\theta \sim \mu$.

Algorithm 3.2 Doubly Stochastic Gradient Descent with ORFFs

input : ORFF feature map $\widetilde{\Phi}_{\Theta}$, RFF feature map $\tilde{\phi}_{\mathcal{X}}$ gradient steps $(\gamma_t)_{t=0}^{T-1}$
init : $\boldsymbol{\beta}^{(0)} = \mathbf{0} \in \mathbb{R}^{2m_{\mathcal{X}}\times 2m_{\Theta}s}$
for t *from* 0 *to* $T-1$ **do**
  // sampling step
  Sample $i \sim \mathrm{Unif}([n])$
  Sample $\theta \sim \mu$
  // gradient step
  $\boldsymbol{\beta}^{(t+1)} = \boldsymbol{\beta}^{(t)} - \gamma_t \partial_{\boldsymbol{\beta}} \mathcal{J}_i(\theta, \boldsymbol{\beta}^{(t)})$
return $\boldsymbol{\beta}^{(T)}$

3.4 Solving in the Dual

We now investigate dual algorithms for regularized empirical risk minimization in the presence of integral losses. As a reminder, Problem 3.7 writes as

$$\inf_{(\alpha_i)_{i=1}^n \in \mathcal{Y}^n} \sum_{i=1}^n I_{\ell_i}^{\star}(-\alpha_i) + \frac{1}{2\lambda n}\sum_{i,j=1}^n \left\langle \alpha_i, K(x_i,x_j)\alpha_j \right\rangle_{\mathcal{Y}},$$

We can see here that the objective function in Problem 3.7 is composed of two terms, the first term being related to the Fenchel-Legendre conjugate of the integral loss I_{ℓ} and to the data, and the second term being a quadratic form involving the dual variables $(\alpha_i)_{i=1}^n \in \mathcal{Y}^n$ and the OVK K.

The difficulties encountered while solving Problem 3.5 are still visible here: the search for the $(\alpha_i)_{i=1}^n \in \mathcal{Y}^n$ cannot *a priori* be reduced to a finite-dimensional optimization problem. Also, it is unclear if the first term can be computed numerically, a challenge similar to what was observed in Problem 3.5 with integral losses. Finally, evaluating the quadratic form in the right term involves knowing how to compute each $\left\langle \alpha_i, K(x_i, x_j)\alpha_j \right\rangle_{\mathcal{Y}}$ which may not be straightforward.

To tackle these challenges, we first begin in Section 3.4.1 by providing general results on the Fenchel-Legendre conjugate of integral losses on vv-RKHSs. Then in Section 3.4.2 we propose a solution to Problem 3.7 based on expressing dual variables in a basis suitable to both the OVK K and the loss function. This is feasible only when there is some sort of compatibility between the loss function and the output vv-RKHS $\mathcal{H}_G$ as detailed in Assumption 3.16. Finally in Section 3.4.3 we develop optimization algorithms to be used when there is no such compatibility. The basis in which to express the dual variables is to be picked so as to make possible the use of proximal algorithms that depend on the loss function. The quadratic term in Problem 3.7 is then approximated to make the computations possible.

3.4.1 Fenchel-Legendre Conjugate of Integral Losses

To design dual optimization algorithm, it is crucial that we know how to compute the Fenchel-Legendre conjugate of an integral loss. It turns out that under certain assumptions, the Fenchel-Legendre conjugate of an integral loss is the integral of the Fenchel-Legendre conjugate, as was emphasized in the seminal paper Rockafellar (1974). To that end, two key assumptions must be verified. The first one is that ℓ is a *normal convex integrand*, which we have assumed throughout this chapter, while the second one is the following:

Assumption 3.12. *The Hilbert space of functions $\mathcal{Y}$ is* decomposable*, which means that there exists an increasing sequence $(\Theta_k)_{k\in\mathbb{N}}$ of subsets of Θ with $\cup_{k=0}^{\infty}\Theta_k = \Theta$ such that for $\forall k \in \mathbb{N}$, for all bounded measurable functions $f\colon \Theta_k \to \mathbb{R}$, and for all $g \in \mathcal{Y}$, the measurable function that coincides with f on Θ_k and with g on $\Theta \setminus \Theta_k$ belongs to $\mathcal{Y}$.*

In a sense, Assumption 3.12 ensures that the space $\mathcal{Y}$ is big enough so that the Fenchel-Legendre conjugate of the integral loss is the integral of the Fenchel-Legendre conjugate, as detailed in the following proposition.

Proposition 3.13 (Fenchel-Legendre conjugate of Integral Losses, Rockafellar (1976))**.** *Let $\ell\colon \Theta \times \mathcal{U} \times \mathcal{U} \to \left]-\infty, +\infty\right]$ be a normal convex integrand. If $\mathcal{Y}$ satisfies Assumption 3.12 then for all $y \in \mathcal{Y}$,*

$$I_\ell(\cdot, y)^\star = I_{\ell^\star}(\cdot, y). \tag{3.36}$$

where $\ell^\star\colon \Theta \times \mathcal{U} \times \mathcal{U} \to]-\infty, +\infty]$ is the integrand defined for $\forall(\theta, u, v) \in \Theta \times \mathcal{U}^2$ as $\ell^\star(\theta, u, v) = \ell(\theta, \cdot, v)^\star(u)$.

It turns out that assuming Assumption 3.12 is strict in the sense that many commonly-used small spaces of functions do not satisfy it: linear models, polynomial functions, Gaussian RKHSs, and so on. However, $L^2[\Theta, \mu; \mathcal{U}]$ for any probability measure μ verifies Assumption 3.12, which suggests to adopt the view $\mathcal{Y} = L^2[\Theta, \mu; \mathcal{U}]$ and the associated OVK $K = k_\mathcal{X} T_G$.

We can however wonder what would happen if we tried to compute $I_\ell(\cdot, y)^\star$ with $\mathcal{Y} = \mathcal{H}_G$. The following lemma provides a response with additional assumptions of universality of kernel G and continuity of I_ℓ.

Proposition 3.14. *Let k_Θ be a universal kernel, and $\mathbf{A} \in \mathcal{M}_s(\mathbb{R})$ an invertible matrix. Let $\mathcal{H}_G$ be the vv-RKHS associated to the kernel $G = k_\Theta \mathbf{A}$. Assume that μ has full support, in other words* $\operatorname{supp} \mu = \Theta$. *Assume finally that I_ℓ is continuous. Then for all $y \in \mathcal{H}_G$*

$$\sup_{g \in \mathcal{H}_G} \langle g, \alpha \rangle_{\mathcal{H}_G} - I_\ell(g, y) = I_{\ell^\star}(\cdot, y) \circ T_G^{-1}(\alpha), \tag{3.37}$$

where $T_G^{-1}(\alpha)$ is the only function in $L^2[\Theta, \mu; \mathcal{U}]$ such that $T_G(T_G^{-1}(\alpha)) = \alpha$.

Proof We can remark that $\operatorname{Im}(T_G) = \mathcal{H}_G$ since $\operatorname{supp} \mu = \Theta$. Moreover, $\mathbf{A}$ is injective, and universality of k_Θ implies that T_{k_Θ} is injective (see Proposition 2.25), so that $T_G = \mathbf{A} \otimes T_{k_\Theta}$ is injective. Thus $T_G^{-1}(\alpha) \in L^2[\Theta, \mu; \mathcal{U}]$ is well-defined for all $\alpha \in \mathcal{H}_G$. Then for $\forall g \in \mathcal{H}_G$,

$$\begin{aligned}
\langle g, \alpha \rangle_{\mathcal{H}_G} - I_\ell(g, y) &= \left\langle g, T_G \circ T_G^{-1}(\alpha) \right\rangle_{\mathcal{H}_G} - I_\ell(g, y) \\
&= \left\langle g, T_G^{-1}(\alpha) \right\rangle_{L^2[\Theta, \mu; \mathcal{U}]} - I_\ell(g, y)
\end{aligned}$$

where we used the fact that $T_G^\sharp$ is the canonical inclusion from $\mathcal{H}_G$ to $L^2[\Theta, \mu; \mathcal{U}]$ (Carmeli et al. (2010), Proposition 3). Since $\mathcal{H}_G$ is dense in $L^2[\Theta, \mu; \mathcal{U}]$ and $g \mapsto \left\langle g, T_G^{-1}(\alpha) \right\rangle_{L^2[\Theta, \mu; \mathcal{U}]} - I_\ell(g, y)$ is continuous, this results in Equation (3.37). ■

The result in Proposition 3.14 endorses the modeling choice $\mathcal{Y} = L^2[\Theta, \mu]$, since it avoids the difficulty brought in by the term involving T_G^{-1} and additional assumptions. One can then directly work with dual variables $(\alpha_i)_{i=1}^n \in L^2[\Theta, \mu; \mathcal{U}]$ and solve Problem 3.7. Before delving into this solution, we list below the Fenchel-Legendre conjugates of a various integral losses used in practice.

Square Loss: When $\ell(\theta, u, v) = \frac{1}{2} \|u - v\|_{\mathcal{U}}^2$ then $\ell^\star(\theta, u, 0) = \ell(\theta, \cdot, 0)^\star(u) = \frac{1}{2} \|u\|_{\mathcal{U}}^2$ and by integrating it along Θ with probability μ we recover what was used in Section 3.2. The case with nonzero v can be deduced by translation properties of the Fenchel-Legendre conjugate (see Table 2.1).

Pinball loss: This corresponds to $\mathcal{U} = \mathbb{R}$, $\Theta = [0, 1]$ and

$$\ell(\theta, u, v) = \max\left(\theta(v - u), (\theta - 1)(v - u)\right).$$

In this case, for $\forall \theta \in \Theta$,

$$\ell^\star(\theta, u, 0) = \ell(\theta, \cdot, 0)^\star(u) = \chi_{\{\theta - 1 \leq \cdot \leq \theta\}}(-u).$$

So that for $\forall (\theta, v) \in \Theta \times \mathcal{U}$,

$$\ell^\star(\theta, u, v) = \ell(\theta, \cdot, 0)^\star(u) - uv = \chi_{\{\theta - 1 \leq \cdot \leq \theta\}}(-u) - uv$$

by using the translation properties of the Fenchel-Legendre conjugate. By integrating it, one gets for all $(\alpha, y) \in \mathcal{Y}^2$:

$$I_{\ell^\star}(\alpha, y) = \chi_{\mathcal{C}}(-\alpha) - \langle \alpha, y \rangle_{\mathcal{Y}},$$

where $\mathcal{C} = \Big\{ g \in \mathcal{Y} \,:\, \theta - 1 \leq g(\theta) \leq \theta \text{ holds } \mu \text{ a.e} \Big\}$.

Cost-sensitive hinge loss: In this case, $\mathcal{U} = \mathbb{R}$, $\Theta = [0,1]$ and

$$\ell(\theta, u, v) = \Big|\theta - \mathbb{1}_{\{-1\}}(v)\Big| \max(0, 1 - uv).$$

As we shall see in Chapter 5, in the application to cost-sensitive classification the data $(y_i)_{i=1}^n$ are constant functions being equal to ± 1. We therefore only propose to compute $I_\ell(\cdot, 1)^\star$ and $I_\ell(\cdot, -1)^\star$. For all $(\theta, u) \in \Theta \times \mathbb{R}$, it holds that

$$\ell^\star(\theta, u, 1) = \ell(\theta, \cdot, 1)^\star(u) = \chi_{[0,\theta]}(-u)$$

so by integration over Θ one arrives at

$$I_\ell^\star(\alpha, 1) = \chi_{\mathcal{C}^1}(-u),$$

where $\mathcal{C}^1 = \Big\{g \in \mathcal{Y} \,:\, 0 \leq g(\theta) \leq \theta \text{ holds } \mu \text{ a.e}\Big\}$. Similarly, we get

$$I_\ell^\star(\alpha, -1) = \chi_{\mathcal{C}^{-1}}(-u)$$

where $\mathcal{C}^{-1} = \Big\{g \in \mathcal{Y} \,:\, 0 \leq g(\theta) \leq 1 - \theta \text{ holds } \mu \text{ a.e}\Big\}$.

We now move on to the solution of Problem 3.7 when the problem benefits from *compatibility* between the integral operator T_G and the loss functions $I_{\ell_i^\star}$.

3.4.2 Integral Operator Eigenbasis Representation

As seen in Proposition 2.42, T_G is a compact operator that admits an eigendecomposition

$$T_G = \sum_{j \in J} \lambda_j \psi_j \psi_j^\sharp,$$

where $(\lambda_j)_{j \in J} \in \mathbb{R}_+^{|J|}$ is a non-increasing sequence of eigenvalues with limit 0 and $(\psi_j)_{j=1}^\infty$ is an orthonormal family of $L^2[\Theta, \mu; \mathcal{U}]$.

Remark 3.15. *The eigenvalues $(\lambda_j)_{j \in J}$ are not to be confounded with the regularization parameter λ despite the similar notation.*

While we have no guarantee that the sequence $(\hat{\alpha}_i)_{i=1}^n$ has a finite expansion on this basis, we propose to perform the search of the dual variables $(\alpha_i)_{i=1}^n$ in the subset of $L^2[\Theta, \mu; \mathcal{U}]$ spanned by the first m eigenvectors of this basis, transforming the Problem 3.7 into a finite-dimensional optimization problem. The rationale behind this approach is that as $j \to \infty$, the influence of a particular direction ψ_j in the model will vanish, because

$$\Big\|T_G \psi_j\Big\|_{\mathcal{Y}} = \lambda_j \Big\|\psi_j\Big\|_{\mathcal{Y}} = \lambda_j \underset{j \to \infty}{\to} 0.$$

This approach can only be applied when there is compatibility between the loss function and the output RKHS $\mathcal{H}_G$. Indeed, for Problem 3.7 to be tractable, one needs to be able to compute the terms $I_{\ell_i}(\alpha_i)$ based solely on the coefficients of the dual variables in the eigenbasis $(\psi_j)_{j=1}^m$. We propose below an assumption that quantifies such behavior.

Assumption 3.16 (Basis Compatibility)**.** *The eigenbasis* $(\psi_j)_{j=1}^m$ *is* compatible *with the integral loss* I_ℓ *in the sense that for all* $i \in [n]$*, there exists* $L_i \colon \mathbb{R}^m \to \mathbb{R}$ *such that*

$$\forall \alpha \in \mathbb{R}^m, I_{\ell_i^\star}\left(-\sum_{j=1}^m \alpha_j \psi_j\right) = L_i(\alpha, y_i). \tag{3.38}$$

Remark 3.17. *One major drawback of this approach is that the eigenbasis must be compatible with the computation of the Fenchel-Legendre conjugate, and Assumption 3.16 is seldom verified in practice. For example, it is verified in case of the squared loss, but not for the pinball or cost-sensitive hinge loss. However, the techniques developed here are useful beyond integral losses, in the context of* convoluted *losses. This will be exploited in Chapter 4 to gain sparsity using* ϵ*-insensitive losses or robustness through the Huber loss.*

We remark here that there are multiple choices for obtaining the system $(\psi_j)_{j=1}^m$, as presented in Section 2.2. In particular, the use of approximated eigenbasis Example 2.23 or Random Fourier Features Example 2.30 can be beneficial to the user. Working with a truncated basis is equivalent to solving Problem 3.7 in the vv-RKHS associated to kernel

$$\widetilde{K}(x,z) := k_\mathcal{X}(x,z) \sum_{j=1}^m \lambda_j \psi_j \psi_j^\sharp.$$

Problem 3.7 can then be rewritten as

$$\inf_{\boldsymbol{\alpha} \in \mathcal{M}_{n,m}(\mathbb{R})} \sum_{i=1}^n \underbrace{I_{\ell_i^\star}\left(-\sum_{j=1}^m \alpha_{ij}\psi_j\right)}_{L_i\left(\boldsymbol{\alpha}_{i:}, y_i\right) \text{ by (3.38)}} + \frac{1}{2\lambda n} \operatorname{Tr}\left(\mathbf{K}_\mathcal{X} \boldsymbol{\alpha} \boldsymbol{\Lambda} \boldsymbol{\alpha}^\top\right), \tag{3.39}$$

where $\boldsymbol{\Lambda} = \operatorname{diag}(\lambda_j)_{j=1}^m$ encodes information about the eigenvalues of T_G and $\mathbf{K}_\mathcal{X}$ is the input kernel Gram matrix over the data points $(x_i)_{i=1}^n$. This optimization problem can then be tackled by different means depending on the regularity of the problem. We now explore in Section 3.4.3 optimization algorithms dedicated to solving Problem 3.7 when we cannot make Assumption 3.16.

3.4.3 Proximal Algorithms and Approximated Quadratic Forms

The solution of Problem 3.7 proposed in Section 3.4.2 relies on the strong Assumption 3.16. In practice, this assumption is seldom verified and one cannot compute each $I_{\ell_i^\star}(-\alpha_i)$ based on the coefficients of α_i in an orthonormal basis. Thus, the approach consisting in choosing first a basis adapted to the quadratic form is flawed here, as the resulting problem is not amenable to optimization. We propose to work the other way around: choose a basis adapted to the $I_{\ell_i}^\star$, and then find a way to compute the quadratic form, at the price of an approximation.

Assumption 3.18. *Let* $\mathcal{S}$ *be a finite-dimensional subspace of* $L^2[\Theta, \mu; \mathcal{U}]$*, so that each function* $\alpha \in \mathcal{S}$ *can be parameterized by a vector* $v \in \mathbb{R}^m$*. We say that* $\mathcal{S}$ *is* compatible *with* I_ℓ *if for all* $i \in [n]$*, there exist* $R_i \colon \mathbb{R}^m \to \mathbb{R}$ *and* $Q_i \colon \mathbb{R}^m \to \left]-\infty, +\infty\right]$ *so that*

1. *for all* $\alpha \in \mathcal{S}$ *parameterized by* $v \in \mathbb{R}^m$*,* $I_{\ell_i}^\star(\alpha) = R_i(v) + Q_i(v)$*,*

2. *R_i is differentiable, and*

3. $\mathrm{prox}_{\gamma Q_i}$ *is computable in closed form for all* $\gamma > 0$.

Assumption 3.18 is designed so that Problem 3.7 writes as a *composite problem* whose solution can be performed using proximal gradient descent algorithm (see Section 2.1 for reminders on these concepts). As an example, we consider the specific quantile regression scheme associated to the integral pinball loss. In this setting, $\Theta = [0,1]$ and $\mathcal{U} = \mathbb{R}$ are one-dimensional. The Fenchel-Legendre conjugate of the integral pinball loss computed in Section 3.4.1 reads

$$I_{\ell_i^\star}(\alpha) = -\langle \alpha, y_i \rangle_{\mathcal{Y}} + \chi_{\mathcal{C}}(-\alpha), \quad (\alpha, i) \in \mathcal{Y} \times [n]$$

with $\mathcal{C} = \Big\{ g \in \mathcal{Y} \,:\, \theta - 1 \leq g(\theta) \leq \theta \text{ holds } \mu \text{ a.e} \Big\}$. Moreover,

$$\mathrm{prox}_{\gamma \chi_{\mathcal{C}}(\cdot)}(\alpha) = \mathrm{Proj}_{\mathcal{C}}(\alpha)$$

for all $\gamma > 0$ and $\alpha \in \mathcal{Y}$. Thus, to satisfy Assumption 3.18 a suitable finite-dimensional representation space $\mathcal{S}$ must satisfy the two properties

1. for all $(\alpha, i) \in \mathcal{S} \times [n]$, if α is encoded by a vector $v \in \mathbb{R}^m$ then $R_i(v) = -\langle \alpha, y_i \rangle_{\mathcal{Y}}$ is computable (or at least can be approximated)

2. for all $\alpha \in \mathcal{S}$, $\mathrm{Proj}_{\mathcal{C}}(\alpha)$ can be computed in closed form.

Taking a closer look at this projection operator, we notice that

$$\forall \alpha \in \mathcal{Y},\ \mathrm{Proj}_{\mathcal{C}}(\alpha) \colon \begin{pmatrix} \Theta & \to & \mathbb{R} \\ \theta & \mapsto & \max(\theta - 1, \min(\alpha(\theta), \theta)). \end{pmatrix} \tag{3.40}$$

We see here why the eigendecomposition of T_G is not a suitable basis: $\mathrm{Proj}_{\mathcal{C}}(\alpha)$ involve a *pointwise* projection of each $\alpha(\theta)$ on the corresponding interval $[\theta - 1, \theta]$ so that $\mathrm{Span}\{(\psi_j)_{j=1}^m\}$ may not stable with respect to this projection. This suggests to choose a representation for the dual variables that allows a *pointwise* control on the functions. We propose to use linear splines in this case, as their shape matches that of the boundaries of $\mathcal{C}$.

Linear Splines A linear spline $\alpha \in \mathcal{Y}$ is encoded by a set of ordered locations $(\theta_j)_{j=1}^m \in \Theta^m$ and by a vector $\boldsymbol{\alpha} := (\alpha(\theta_j))_{j=1}^m \in \mathbb{R}^m$. We denote by $\mathcal{S}_m$ the set of linear splines obtained with those fixed locations. In between locations can be computed using linear interpolation:

$$\forall \theta \in [\theta_j, \theta_{j+1}], \quad \alpha(\theta) = \alpha(\theta_j) + \frac{\alpha(\theta_{j+1}) - \alpha(\theta_j)}{\theta_{j+1} - \theta_j}(\theta - \theta_j).$$

The first remark to be made about restricting dual variables to be linear splines is that linear splines can approximate any function in $\mathcal{Y}$, so that it is a rich enough representation to work with. Moreover, although restricting the dual variables to be linear splines may appear a limitation, it turns out that in the estimator

$$\hat{h} = \frac{1}{\lambda n} \sum_{i=1}^n k_{\mathcal{X}}(\cdot, x_i) T_G \hat{\alpha}_i$$

the splines are smoothed by means of the integral operator T_G, so that the model itself is not piecewise linear in θ, only the dual variables are. The drawback of this method is that one cannot compute exactly the quadratic terms $\langle \alpha_i, T_G \alpha_j \rangle_{\mathcal{Y}}$. We propose to approximate them as presented below.

Approximated Quadratic Forms: Let α, β be linear splines encoded by locations $(\theta_j)_{j=1}^m \in \Theta^m$ distributed i.i.d. according to μ and the vectors of function evaluations $\boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathbb{R}^m$. Let $\mathbf{K}_\Theta$ be the Gram matrix associated to locations $(\theta_j)_{j=1}^m$ and kernel k_Θ. Then

$$\left(T_{k_\Theta} \beta\right)(\theta) = \int_\Theta k_\Theta(\theta, \theta') \beta(\theta) \mathrm{d}\mu(\theta') \approx \frac{1}{m} \sum_{j=1}^m k_\Theta(\theta, \theta_j) \beta(\theta_j), \quad \theta \in \Theta,$$

and

$$\left\langle \alpha, T_{k_\Theta} \beta \right\rangle_{\mathcal{Y}} = \left\langle \alpha, \int_\Theta k_\Theta(\cdot, \theta') \beta(\theta) \mathrm{d}\mu(\theta') \right\rangle_{\mathcal{Y}} \approx \frac{1}{m^2} \underbrace{\sum_{i,j=1}^m k_\Theta(\theta_i, \theta_j) \alpha(\theta_i) \beta(\theta_j)}_{\boldsymbol{\alpha}^\top \mathbf{K}_\Theta \boldsymbol{\beta}}$$

We thus propose to solve the following problem

$$\inf_{(\alpha_i)_{i=1}^n \in \mathcal{S}_m} \sum_{i=1}^n \chi \mathcal{C}(\alpha_i) + \underbrace{\sum_{i=1}^n \langle \alpha_i, y_i \rangle_{\mathcal{Y}} + \frac{1}{2\lambda n m^2} \operatorname{Tr}\left(\mathbf{K}_\chi \boldsymbol{\alpha} \mathbf{K}_\Theta \boldsymbol{\alpha}^\top\right)}_{:= \mathcal{J}(\boldsymbol{\alpha})}, \tag{3.41}$$

where given some splines $(\alpha_i)_{i=1}^n \in \mathcal{S}_m$, we denote by $\boldsymbol{\alpha} \in \mathcal{M}_{n,m}(\mathbb{R})$ the matrix encoding row-wise the values associated to each spline. We present in Algorithm 3.3 a proximal gradient algorithm for solving Problem 3.41, that reads as a projected gradient descent: for each epoch we perform one step of gradient descent on $\mathcal{J}$, followed by the proximal step that consists in the projection on the feasible set defined by $\mathcal{C}$.

Algorithm 3.3 Proximal Gradient Descent with Linear Splines for Quantile Regression

input : Splines locations $(\theta_j)_{j=1}^m \overset{\text{i.i.d.}}{\sim} \mu$, Gram matrices $\mathbf{K}_\chi$, $\mathbf{K}_\Theta$
init : $\boldsymbol{\alpha} = \mathbf{0} \in \mathcal{M}_{n,m}(\mathbb{R})$
for *epoch from* 1 *to* T **do**
  // gradient step
  $\boldsymbol{\alpha} = \boldsymbol{\alpha} - \nabla \mathcal{J}(\boldsymbol{\alpha})$
  // proximal step
  for *row* i *from* 1 *to* n **do**
    for *column* j *from* 1 *to* m **do**
      $\boldsymbol{\alpha}_{ij} = \mathrm{Proj}_{[\theta_j - 1, \theta_j]}(\boldsymbol{\alpha}_{ij})$
return $\boldsymbol{\alpha}$

Remark 3.19. *While in the general case the* $\langle \alpha_i, y_i \rangle_{\mathcal{Y}}$ *have to be estimated, as we will see in Chapter 5 the* Infinite Task Learning *framework uses constant functions as observed outputs* $(y_i)_{i=1}^n$*. This makes the exact computation of these scalar products*

possible when μ is the Lebesgue measure, as for a spline $\alpha \in \mathcal{S}_m$

$$\begin{aligned}\langle \alpha, y_i \rangle_{\mathcal{Y}} &= y_i(0) \int_\Theta \alpha(\theta) \mathrm{d}\theta = y_i(0) \sum_{j=1}^{m-1} \int_{\theta_j}^{\theta_{j+1}} \alpha(\theta) \mathrm{d}\theta \\ &= y_i(0) \sum_{j=1}^{m-1} \frac{1}{2}(\alpha(\theta_{j+1}) + \alpha(\theta_j))(\theta_{j+1} - \theta_j).\end{aligned}$$

This algorithm can be adapted to the integral cost-sensitive hinge loss with a slightly different projector suited to Section 3.4.1 and Section 3.4.1. The general idea of the algorithm can work in higher dimensions: provided that some set $\mathcal{S}$ allows a pointwise control on the dual variables and Assumption 3.18 is verified, the quadratic term can be approximated and a proximal gradient algorithm is possible.

3.5 Conclusion

In this chapter, we provide optimization algorithms to solve regularized empirical risk minimization problems in vv-RKHSs in the presence of integral losses. Beyond closed-form solutions for the square loss, we address the question of representing the functional coefficients by several means. Primal approaches include the *double representer theorem*, which provides a workable parametrization of the solution to the price of an approximation on the integral loss, and *random Fourier features* amenable to optimization via stochastic algorithms. Dual techniques are also developed, and involve some compatibility conditions between the vv-RKHS and the loss for the representation of the dual variables to be viable. In particular, dual approaches pave the way to extend this framework beyond integral losses, by using the infimal convolution operator to design new loss functions. Finally, we want to emphasize the compatiblity of the developed algorithms with hybrid models involving the composition of a neural architecture and a kernel, allowing to tackle tasks with complex input data. This is exemplified in Section 5.4.4 for quantile regression with input being images.

4

Robust Functional Output Regression

Contents

In this chapter, we propose a framework dedicated to regressing functional outputs with a focus on robustness. To do so, we extend the integral loss framework introduced in Chapter 3 to handle *convoluted losses*. This goal is achieved by exploiting duality principles which are well-suited to the nature of these loss functions. After a brief introduction in Section 4.1, we state the problem to solve in Section 4.2. In Section 4.3 we propose a methodology that handles different variants of the Huber loss, known to induce robustness to outliers. In Section 4.4 we study optimization problems based on ϵ-insensitive losses which are akin to bring sparsity to the estimator. Numerical experiments on real and synthetic datasets are gathered in Section 4.5, and conclusions are drawn in Section 4.6.

4.1 Introduction

Due to the increased availability of streaming data, learning to predict complex objects has attracted a great deal of attention in machine learning. Biomedical Signal Processing, Epidemiology Monitoring or Climate Science are examples of interdisciplinary research fields where the phenomena under study exhibit a functional nature, and the understanding of these phenomena depends on machine learning algorithms being able to reliably handle functional data. Functional data analysis (FDA, Ramsay and Silverman 1997; Wang et al. 2016) has been devoted to such setting, where one assumes that the measurements of the underlying phenomena come numerous enough so that

modeling it with functions makes sense. In particular, Functional Output Regression (FOR) specializes to regression problems where the output variable is a function, the input variable being either finite-dimensional or of a functional nature themselves.

The simplest way to design an algorithm is then to model a linear dependency between the inputs and the outputs (Morris, 2015), at the price of being unable to cope with complex dependency within the data. To remedy this bottleneck, nonlinear approaches have been developed in recent years. In nonparametric statistics, Ferraty et al. (2011) propose a Banach-valued version of the Nadaraya-Watson estimator. Kernel methods have proven useful to tackle this problem as well, with works involving tri-variate regression problem in RKHSs (Reimherr et al., 2018), approximated kernel ridge regression (KRR) using orthonormal bases (Oliva et al., 2015), and in the operator-valued kernel literature a function-valued KRR with double representer theorem (Lian, 2007), solvers based on a sampling of the functional norm (Kadri et al., 2010) or purely functional methods relying on approximate inversion of integral operators (Kadri et al., 2016). We can also mention recent techniques using vv-RKHSs to learn finite-dimensional coefficients expressing the functional outputs with the help of a dictionary basis (Bouche et al., 2021).

Most of these applications focus on the square loss, which is known to induce estimates of the conditional expectation of the functional outputs given the input data. However, defective sensors or malicious attacks can lead to erroneous or contaminated measurements of the phenomena that result in functional outliers, and using the square loss has the major drawback of providing estimators which are sensitive to these outliers, producing unreliable prediction systems. In the scalar-valued case, variations of the square loss such as the Huber loss (Huber, 1964) or ϵ-insensitive losses (Lee et al., 2005) have been introduced (among other techniques) as a way to mitigate such sensitivity. In the FDA setting, robustness aspects have been investigated using Bayesian methods (Zhu et al., 2011), trading the mean for the Banach-valued median (Cadre, 2001), using bounded loss functions (Maronna and Yohai, 2013), or leveraging principal component analysis (Kalogridis and Van Aelst, 2019). We can also mention works involving the Huber loss function for the semi-functional setting (functional input variable, scalar output variable; Crambes et al. 2008; Shin and Lee 2016; Qingguo 2017; Boente et al. 2020).

In the operator-valued kernel literature, extension of ϵ-insensitive losses to the (finite) vector-valued regression setting has been proposed by Sangnier et al. (2017). Using convex optimization tools such as the infimal convolution operator and parametric duality leads to efficient solvers and provide sparse estimators, an idea later exploited in (Laforgue et al., 2020) where a generalization of this approach to infinite-dimensional outputs encompassing both the Huber and ϵ-insensitive losses is proposed. In this chapter, we extend the results from Laforgue et al. (2020) in the FOR setting and build a robust framework based on parametric duality in vv-RKHSs with functional outputs, adapting the dual optimization algorithms from Chapter 3 to the setting of convoluted losses.

4.2 Problem Setting

In the functional output regression (FOR) setting, the goal is to regress to a functional output Y taking its values in $\mathcal{Y} := L^2[\Theta, \mu]$ where $\Theta := [0, 1]$ is endowed with a probability measure μ, from an input variable X that either takes its values in the Euclidean

space $\mathcal{X} = \mathbb{R}^d$ or in the Hilbert space $\mathcal{X} = L^2[\Theta, \mu]$ itself. A natural way to tackle this problem is to estimate $\mathbb{E}[\mathsf{Y}|\mathsf{X}] = h^\dagger(\mathsf{X})$ where it is well-known that

$$h^\dagger = \operatorname*{arg\,min}_{h \in \mathcal{H}} \ \mathbb{E}_{(\mathsf{X},\mathsf{Y})} \left[\left\| \mathsf{Y} - h(\mathsf{X}) \right\|_{\mathcal{Y}}^2 \right] \tag{4.1}$$

with $\mathcal{H}$ being the set of all measurable functions from $\mathcal{X}$ to $\mathcal{Y}$. The existence of $h^\dagger$ is guaranteed by Theorem 10.2.2 from Dudley (2002) as $\mathcal{Y}$ is a Polish space (*i.e.* a separable completely metrizable topological space).

In practice, the law of (X, Y) is unknown, and one has only information about it via i.i.d. samples $(x_i, y_i)_{i=1}^n$. Since working with the set of all measurable functions from $\mathcal{X}$ to $\mathcal{Y}$ is notoriously hard, we restrict the hypothesis space to be a *vector-valued reproducing kernel Hilbert space* (vv-RKHS) $\mathcal{H}_K \subset \mathcal{F}(\mathcal{X}, \mathcal{Y})$ associated to some *operator-valued kernel* (OVK) $K \colon \mathcal{X} \times \mathcal{X} \to \mathcal{L}(\mathcal{Y})$. We assume that K is chosen to be a decomposable kernel of the form $K = k_{\mathcal{X}} T_{k_\Theta}$ with scalar-valued kernels $k_{\mathcal{X}} \colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ and $k_\Theta \colon \Theta \times \Theta \to \mathbb{R}$. We refer to Section 2.2.2 for more details about such functional spaces. We then arrive to the following regularized empirical risk minimization problem

$$\inf_{h \in \mathcal{H}_K} \frac{1}{n} \sum_{i=1}^n \left\| y_i - h(x_i) \right\|_{\mathcal{Y}}^2 + \frac{\lambda}{2} \left\| h \right\|_{\mathcal{H}_K}^2, \tag{4.2}$$

where $\lambda > 0$ is a hyperparameter controling the strength of the regularization in the vv-RKHS. Problem 4.2 has been introduced and solved in Kadri et al. (2016) with a closed-form estimator

$$\hat{h} := \sum_{i=1}^n K(\cdot, x_i) \hat{\alpha}_i, \tag{4.3}$$

with optimal coefficients $\hat{\boldsymbol{\alpha}} := [\hat{\alpha}_i]_{i=1}^n \in \mathcal{Y}^n$ given by

$$\hat{\boldsymbol{\alpha}} = \left(\mathbf{K}_{\mathcal{X}} \otimes T_{k_\Theta} + \lambda n \operatorname{Id} \right)^{-1} \mathbf{y}, \tag{4.4}$$

where $\mathbf{K}_{\mathcal{X}}$ is the Gram matrix associated to the input data $(x_i)_{i=1}^n$ and kernel $k_{\mathcal{X}}$, and $\mathbf{y} = [y_i]_{i=1}^n$ is the vector aggregating the functional responses $(y_i)_{i=1}^n$.

One main advantage of working with the square loss is to benefit from a closed-form solution. This is due to the fact that the gradient of the objective function linearly depends on the residuals $y_i - h(x_i)$, reducing Problem 4.2 to the solution of a linear system. This somehow innocent and positive property has dramatic consequences in the presence of contaminated data, as the estimator will be strongly influenced by a large shift in the data caused by some measurement error or malicious actor. One can then resort to using losses different than the square loss, that provide more robust estimators.

Remark 4.1. *The notation here slightly differs from Chapter 3 where the loss functions were taking two arguments. The reason behind this was to encompass two kinds of residuals under the same umbrella: the regression case $y - h(x)$ and the classification case $yh(x)$. Since this chapter is devoted to regression, the loss function L in Problem 4.5 takes only one argument which is $y - h(x)$.*

Given a general loss function $L \colon \mathcal{Y} \to \mathbb{R}$, we consider the problem

$$\inf_{h \in \mathcal{H}_K} \frac{1}{n} \sum_{i=1}^n L(y_i - h(x_i)) + \frac{\lambda}{2} \left\| h \right\|_{\mathcal{H}_K}^2 \tag{4.5}$$

Table 4.1 – Summary of proposed algorithms.

Section	L	$L^\star$	parameterization	proximal operator
4.3.1	vectorial Huber 4.9	$\frac{1}{2}\left\|\cdot\right\|_{\mathcal{Y}}^2 + \chi_{\mathcal{B}_\kappa}$	eigenbasis of T_{k_Θ}	$\mathrm{Proj}_{\mathcal{B}_\kappa}$
4.3.2	integral Huber 4.10	$\frac{1}{2}\left\|\cdot\right\|_{\mathcal{Y}}^2 + \chi_{\mathcal{B}_\kappa^\infty}$	linear splines	$\mathrm{Proj}_{\mathcal{B}_\kappa^\infty}$
4.4.1	vectorial ϵ-ridge 4.18	$\frac{1}{2}\left\|\cdot\right\|_{\mathcal{Y}}^2 + \epsilon\left\|\cdot\right\|_{\mathcal{Y}}$	eigenbasis of T_{k_Θ}	BST
4.4.2	integral ϵ-ridge 4.19	$\frac{1}{2}\left\|\cdot\right\|_{\mathcal{Y}}^2 + \epsilon\left\|\cdot\right\|_1$	linear splines	ST

and recall the formulation of the associated dual problem (see Theorem 2.44)

$$\inf_{(\alpha_i)_{i=1}^n \in \mathcal{Y}^n} \sum_{i=1}^n L_i^\star(-\alpha_i) + \frac{1}{2\lambda n}\sum_{i=1}^n\sum_{j=1}^n k_{\mathcal{X}}(x_i, x_j)\left\langle \alpha_i, T_{k_\Theta}\alpha_j\right\rangle_{\mathcal{Y}}, \tag{4.6}$$

where $f^\star$ is the Fenchel-Legendre conjugate of any function $f\colon \mathcal{Y} \to \left]-\infty, +\infty\right]$ (see Definition 2.2) and by convention $L_i = L(y_i - \cdot)\ \forall i \in [n]$. The solution of Problem 4.5 then takes the form

$$\hat{h} = \frac{1}{\lambda n}\sum_{i=1}^n k_{\mathcal{X}}(\cdot, x_i) T_{k_\Theta}\hat{\alpha}_i$$

with $(\hat{\alpha}_i)_{i=1}^n$ being the solution of Problem 4.6.

Among interesting losses, we can mention the Huber loss (Huber, 1964) which is known to induce robust estimators, or ϵ-insensitive losses (Lee et al., 2005) leading to sparse solutions. A particularity of these losses is that they write as the infimal convolution of two loss functions, which can be used to our advantage in Problem 4.6. Indeed, constructing suitable loss functions L can be carried out using the infimal convolution operator (see Definition 2.3): given two loss functions f and g, taking $L = f \,\square\, g$ yields a new loss function that can be used in Problem 4.5. In the duality framework, this type of loss functions is particularly appealing as the Fenchel-Legendre conjugate of L can be expressed as a simple addition $L^\star = f^\star + g^\star$ in Problem 4.6. This way, to design a loss function of interest, we can for example take $f = \frac{1}{2}||\cdot||_{\mathcal{Y}}^2$ and convolve it with a function g that brings an interesting property (sparsity, robustness) to the estimator resulting from solving Problem 4.6. This results in dual problems with objective function being the a smooth quadratic part, to which is added a non smooth term corresponding to $g^\star$. This idea was first developed in the operator-valued kernel literature in Sangnier et al. (2017), where it was applied to ϵ-insensitive losses in vv-RKHSs with finite-dimensional outputs and then extended to infinite-dimensional outputs in Laforgue et al. (2020).

We propose to deepen this approach in the FOR case, and introduce two families of problems that each bring special properties to the estimator. In Section 4.3, we develop optimization algorithms to enforce robust estimators by using different versions of the *Huber loss*. Then in Section 4.4, we apply this methodology to *ϵ-insensitive loss* functions, for which estimators benefit from sparsity properties. A summary of all these settings is given in Table 4.1. Finally, numerical experiments are presented in Section 4.5.2 to illustrate the effectiveness of the approach in different outlier scenarii.

4.3 Robust Estimators with Huber Loss

The Huber loss has been introduced by Huber (1964) to provide a robust estimate of the mean of a random variable. In the real output case, it is defined as follows.

Definition 4.2. *The real Huber loss with parameter $\kappa > 0$ is defined as*

$$\forall s \in \mathbb{R}, \quad \ell_{H,\kappa}(s) := \begin{cases} \frac{1}{2}s^2 & \text{if } |s| \leq \kappa \\ \kappa\left(|s| - \frac{\kappa}{2}\right) & \text{otherwise.} \end{cases}$$

Due to its asymptotic behavior as $\kappa|\cdot|$, the Huber loss is useful when the training data is heavy-tailed or contains outliers. It turns out that $\ell_{H,\kappa}$ can be written in the more compact form

$$\ell_{H,\kappa} = \frac{1}{2}(\cdot)^2 \,\square\, \kappa|\cdot|. \tag{4.7}$$

This allows to easily compute its Fenchel-Legendre transform as detailed in the following proposition.

Proposition 4.3. *The Fenchel-Legendre transform of the real Huber loss $\ell_{H,\kappa}$ is given by*

$$\forall s \in \mathbb{R}, \quad \ell^\star_{H,\kappa}(s) = \frac{1}{2}s^2 + \chi_{[-\kappa,\kappa]}(s). \tag{4.8}$$

Proof This is a simple application of Proposition 2.4 to $f = \frac{1}{2}(\cdot)^2$ and $g = \kappa|\cdot|$, whose Fenchel-Legendre transforms can be found in Table 2.1. ■

Extending the Huber loss to higher dimension can be done multiple ways. A first one would consist in replacing $\frac{1}{2}(\cdot)$ by $\frac{1}{2}\|\cdot\|^2_{\mathcal{Y}}$ and $\kappa|\cdot|$ by $\kappa\|\cdot\|_{\mathcal{Y}}$ in Equation (4.7), which we refer to as the vectorial Huber loss.

Definition 4.4. *The* vectorial Huber loss *of parameter $\kappa > 0$ is given by*

$$L_{H,\kappa} := \kappa\|\cdot\|_{\mathcal{Y}} \,\square\, \frac{1}{2}\|\cdot\|^2_{\mathcal{Y}}, \tag{4.9}$$

or again:

$$\forall y \in \mathcal{Y}, \quad L_{H,\kappa}(y) = \begin{cases} \frac{1}{2}\|y\|^2_{\mathcal{Y}} & \text{if } \|y\|_{\mathcal{Y}} \leq \kappa \\ \kappa\left(\|y\|_{\mathcal{Y}} - \frac{\kappa}{2}\right) & \text{otherwise.} \end{cases}$$

An illustration of the vectorial Huber loss is provided in Figure 4.1 for dimensions 1 and 2, in dimension 1 it coincides with the real Huber loss. Another way to extend the Huber loss to $\mathcal{Y}$ is to consider the integral loss associated to the real Huber loss (see Section 3.1 for more details about integral losses), as defined below.

Definition 4.5. *The* integral Huber loss *of parameter $\kappa > 0$ is given by*

$$\forall y \in \mathcal{Y}, \quad I_{\ell_{H,\kappa}}(y) := \int_\Theta \ell_{H,\kappa}(y(\theta))\mathrm{d}\mu(\theta). \tag{4.10}$$

In Section 4.3.1, we propose a solution of Problem 4.6 with the vectorial Huber loss, before considering the use of the integral Huber loss in Section 4.3.2. Both losses add a different term in Problem 4.6 that constrains the dual variables, thus resulting in robustness of the estimator to outliers.

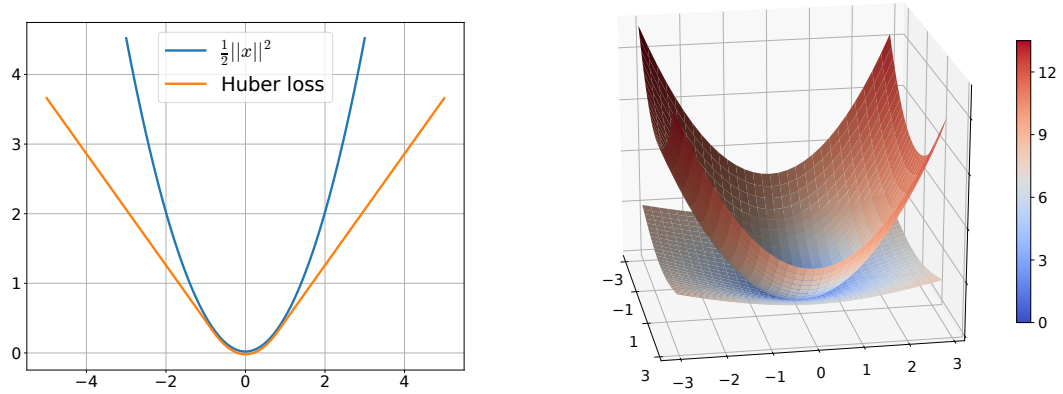


Figure 4.1 – Standard square loss and Huber loss in 1 and 2 dimensions ($\kappa = 0.8$).

4.3.1 Vectorial Huber Loss

Before proposing a solution to Problem 4.5 with the vectorial Huber loss, we quickly provide the expression of its Fenchel-Legendre transform.

Proposition 4.6. *The Fenchel-Legendre transform of the vectorial Huber loss $L_{H,\kappa}$ is given by*

$$\forall y \in \mathcal{Y}, \quad L^\star_{H,\kappa}(y) = \frac{1}{2}\|y\|^2_{\mathcal{Y}} + \chi_{\mathcal{B}_\kappa}(y) \tag{4.11}$$

where $\mathcal{B}_\kappa$ is the ball with radius κ according to norm $\|\cdot\|_{\mathcal{Y}}$.

Proof This is again an application of Proposition 2.4 to $f = \frac{1}{2}\|\cdot\|^2_{\mathcal{Y}}$ and $g = \kappa\|\cdot\|_{\mathcal{Y}}$, whose Fenchel-Legendre transforms can be found in Table 2.1. ■

We are now ready to exemplify Problem 4.6 for the vectorial Huber loss scenario.

Proposition 4.7. *The dual of Problem 4.5 in the vectorial Huber loss case writes as*

$$\inf_{(\alpha_i)_{i=1}^n \in \mathcal{Y}^n} \sum_{i=1}^n \frac{1}{2}\|\alpha_i\|^2_{\mathcal{Y}} - \langle \alpha_i, y_i\rangle_{\mathcal{Y}} + \chi_{\mathcal{B}_\kappa}(\alpha_i) + \frac{1}{2\lambda n}\sum_{i=1}^n\sum_{j=1}^n k_{\mathcal{X}}(x_i,x_j)\left\langle \alpha_i, T_{k_\Theta}\alpha_j\right\rangle_{\mathcal{Y}}. \tag{4.12}$$

Proof This is the direct instantiation of Problem 4.6 with Fenchel-Legendre transform given by Proposition 4.6, combined with the properties of the Fenchel-Legendre transform with respect to the translation operator (see Table 2.1). ■

We first notice that Problem 4.12 is similar to the dual problems tackled using integral losses in Section 3.4. Its solution can be performed using the techniques developed in Section 3.4.2. Indeed, Assumption 3.16 which assess of the compatibility between the loss function and the integral operator T_{k_Θ} is satisfied here: all the terms involved in Problem 4.12 can be computed provided that we possess a representation of the $(\alpha_i)_{i=1}^n$ in an orthonormal basis. We thus represent the dual variables in the truncated eigenbasis $(\lambda_j, \psi_j)_{j=1}^m$ associated to T_{k_Θ}. We introduce the notation $\mathbf{R} \in \mathcal{M}_{n,m}(\mathbb{R})$ such that $\mathbf{R}_{ij} = \langle y_i, \psi_j\rangle_{\mathcal{Y}}$ and $\mathbf{\Lambda} = \operatorname{diag}(\lambda_j)_{j=1}^m \in \mathcal{M}_m(\mathbb{R})$. Denoting by $\boldsymbol{\alpha} \in \mathcal{M}_{n,m}(\mathbb{R})$ the matrix encoding the coefficients of the dual variables, Problem 4.12 can be rephrased as

$$\inf_{\boldsymbol{\alpha}\in\mathcal{M}_{n,m}(\mathbb{R})} \underbrace{\operatorname{Tr}\left(\frac{1}{2}\boldsymbol{\alpha}\boldsymbol{\alpha}^\top - \boldsymbol{\alpha}\mathbf{R}^\top + \frac{1}{2\lambda n}\mathbf{K}_{\mathcal{X}}\boldsymbol{\alpha}\mathbf{\Lambda}\boldsymbol{\alpha}^\top\right)}_{:=\mathcal{J}(\boldsymbol{\alpha})} \quad s.t. \ \|\boldsymbol{\alpha}\|_{2,\infty} \leq \kappa. \tag{4.13}$$

Algorithm 4.1 Projected Gradient Descent for Vectorial Huber

input : Gram matrix $\mathbf{K}_\mathcal{X}$, matrix of eigenvalues $\mathbf{\Lambda}$, data scalar product matrix $\mathbf{R}$, regularization parameter λ, Huber parameter κ, gradient step γ

init : $\boldsymbol{\alpha}^{(0)} = \mathbf{0} \in \mathbb{R}^{n \times m}$ or $\boldsymbol{\alpha}^{(0)} = \lambda n \left(\mathbf{K}_\mathcal{X} \otimes \mathbf{\Lambda} + \lambda n \, \mathrm{Id}_{nm}\right)^{-1} \mathbf{R}$

for *epoch* t *from* 0 *to* $T-1$ **do**

  // gradient step

  $\boldsymbol{\alpha}^{(t+1)} = \boldsymbol{\alpha}^{(t)} - \gamma \left(\boldsymbol{\alpha}^{(t)} + \frac{1}{\lambda n} \mathbf{K}_\mathcal{X} \boldsymbol{\alpha}^{(t)} \mathbf{\Lambda} - \mathbf{R}\right)$

  // projection step

  for *row* i *from* 1 *to* n **do**

    $\boldsymbol{\alpha}_{i:}^{(t+1)} = \min\left(\frac{\kappa}{\left\|\boldsymbol{\alpha}_{i:}^{(t+1)}\right\|_2}, 1\right) \boldsymbol{\alpha}_{i:}^{(t+1)}$

return $\boldsymbol{\alpha}^{(T)}$

We recognize in Problem 4.13 a composite optimization problem (see Problem 2.5) that can be tackled using proximal gradient descent. Let $\gamma > 0$ and $\boldsymbol{\beta} \in \mathbb{R}^m$, it holds that

$$\mathrm{prox}_{\gamma \chi_{\mathcal{B}_\kappa}(\cdot)}(\boldsymbol{\beta}) = \mathrm{Proj}_{\mathcal{B}_\kappa}(\boldsymbol{\beta}) = \min\left(\kappa / \left\|\boldsymbol{\beta}\right\|_2, 1\right) \boldsymbol{\beta},$$

hence by the separability of $\boldsymbol{\alpha} \mapsto \sum_{i=1}^n \chi_{\mathcal{B}_\kappa}(\boldsymbol{\alpha}_{i:})$ the proximal step is obtained by projecting each row of $\boldsymbol{\alpha}$ onto $\mathcal{B}_\kappa$. The proximal gradient descent algorithm is then akin to a projected gradient descent because of the particular form of the proximal operator. The algorithm is summarized in Algorithm 4.1. The resulting estimator is then given by

$$\hat{h} = \frac{1}{\lambda n} \sum_{i=1}^n \sum_{j=1}^m \lambda_j \hat{\alpha}_{ij} k_\mathcal{X}(\cdot, x_i) \psi_j.$$

Remark 4.8. *When κ is large, one recovers the unconstrained ridge regression problem from Kadri et al. (2016), whose practical solution is performed in the truncated eigenbasis $(\psi_j)_{j=1}^m$ (see Section 3.2.1 for details). The solution is described by the closed-form expression*

$$\hat{\boldsymbol{\alpha}} = \lambda n \left(\mathbf{K}_\mathcal{X} \otimes \mathbf{\Lambda} + \lambda n \, \mathrm{Id}_{nm}\right)^{-1} \mathbf{R}. \tag{4.14}$$

This can be used as an initialization point in Algorithm 4.1 if one has enough computational power. It does not correspond to a feasible dual point (except for large enough κ) but after one epoch the projection step will ensure that the iterates satisfies $\|\boldsymbol{\alpha}\|_{2,\infty} \leq \kappa$.

Remark 4.9. *The gradient step γ to choose can be estimated from the parameters of the problem. Indeed, for guaranteed convergence, one must set $\gamma < \frac{2}{C}$ where C is the Lipschitz constant associated to the gradient of the objective function $\mathcal{J}$. Here,*

$$\nabla \mathcal{J}(\boldsymbol{\alpha}) = \boldsymbol{\alpha} + \frac{1}{\lambda n} \mathbf{K}_\mathcal{X} \boldsymbol{\alpha} \mathbf{\Lambda} - \mathbf{R}$$

which is C-Lipschitz with

$$C = 1 + \frac{1}{\lambda n} \left\|\mathbf{K}_\mathcal{X}\right\|_{\mathrm{op}} \lambda_1.$$

4.3.2 Integral Huber Loss

The solution of Problem 4.5 with the integral Huber loss follows similar steps, exploiting the corresponding Fenchel-Legendre transform whose expression is given below.

Proposition 4.10. *The Fenchel-Legendre transform of the integral Huber loss $I_{\ell_{H,\kappa}}$ is given by*

$$\forall y \in \mathcal{Y}, \quad I^\star_{\ell_{H,\kappa}}(y) = \frac{1}{2}\|y\|^2_{\mathcal{Y}} + \chi_{\mathcal{B}^\infty_\kappa}(y), \tag{4.15}$$

where $\mathcal{B}^\infty_\kappa = \left\{ y \in \mathcal{Y} \,:\, |y(\theta)| \leq \kappa \ , \ \mu \text{ a.e.} \right\}$ is the ball of radius κ for the ∞-norm.

Proof This is an application of Proposition 3.13 which states that $I^\star_{\ell_{H,\kappa}} = I_{\ell^\star_{H,\kappa}}$ combined with Proposition 4.3 that gives an expression for $\ell^\star_{H,\kappa}$. ∎

We can now present Problem 4.6 for the integral Huber loss scenario.

Proposition 4.11. *The dual to Problem 4.5 in the integral Huber loss case writes*

$$\inf_{(\alpha_i)_{i=1}^n \in \mathcal{Y}^n} \sum_{i=1}^n \frac{1}{2}\|\alpha_i\|^2_{\mathcal{Y}} - \langle \alpha_i, y_i \rangle_{\mathcal{Y}} + \chi_{\mathcal{B}^\infty_\kappa}(\alpha_i) + \frac{1}{2\lambda n}\sum_{i=1}^n\sum_{j=1}^n k_{\mathcal{X}}(x_i, x_j)\left\langle \alpha_i, T_{k_\Theta}\alpha_j \right\rangle_{\mathcal{Y}}. \tag{4.16}$$

Proof This is the direct instantiation of Problem 4.6 with Fenchel-Legendre transform given by Proposition 4.10, combined with the properties of the Fenchel-Legendre transform with respect to the translation operator (see Table 2.1). ∎

In contrast to what was obtained in Section 4.3.1, Assumption 3.16 is no longer satisfied, and we cannot perform the solution of Problem 4.16 in the eigenbasis associated to T_{k_Θ}. Indeed, given $\alpha_i = \sum_{j=1}^m \alpha_{ij}\psi_j$ it is not possible to evaluate $\chi_{\mathcal{B}^\infty_\kappa}(\alpha_i)$ based on the sole coefficients $(\alpha_{ij})_{j=1}^m$, nor is it possible to easily project α_i onto $\mathcal{B}^\infty_\kappa$. We can then turn to the method developed in Section 3.4.3 and represent the dual variables as linear splines, at the cost of approximating the quadratic term involving T_{k_Θ}. Indeed, a proximal gradient algorithm involves a projection on the feasible set $\mathcal{B}^\infty_\kappa$ which has linear borders: it thus seems appropriate to use a linear spline basis for the representation of the dual variables $(\alpha_i)_{i=1}^n$. Given a set of locations $(\theta_j)_{j=1}^m \overset{\text{i.i.d.}}{\sim} \mu$, we denote by $\mathcal{S}_m$ the set of linear splines with locations $(\theta_j)_{j=1}^m$, and encode the dual variables $(\alpha_i)_{i=1}^n$ by a matrix $\boldsymbol{\alpha} \in \mathcal{M}_{n,m}(\mathbb{R})$ such that the rows of $\boldsymbol{\alpha}$ correspond to the values of the dual variables at locations $(\theta_j)_{j=1}^n$. Finally, we approximate $\langle \alpha_i, y_i \rangle_{\mathcal{Y}} \approx \frac{1}{m}\sum_{j=1}^m \alpha_i(\theta_j) y_i(\theta_j)$ and store the observed outputs $y_i(\theta_j)$ in a matrix $\mathbf{Y} \in \mathcal{M}_{n,m}(\mathbb{R})$. Problem 4.16 then writes as

$$\inf_{\boldsymbol{\alpha} \in \mathcal{M}_{n,m}(\mathbb{R})} \operatorname{Tr}\left(\frac{1}{2}\boldsymbol{\alpha}\boldsymbol{\alpha}^\top - \boldsymbol{\alpha}\mathbf{Y}^\top + \frac{1}{2\lambda nm}\mathbf{K}_{\mathcal{X}}\boldsymbol{\alpha}\mathbf{K}_\Theta\boldsymbol{\alpha}^\top \right) \quad s.t. \ \|\boldsymbol{\alpha}\|_\infty \leq \kappa. \tag{4.17}$$

A proximal gradient descent algorithm to solve Problem 4.17 is presented in Algorithm 4.2, where projecting on $\mathcal{B}^\infty_\kappa$ is equivalent to projecting each value $\boldsymbol{\alpha}_{ij}$ onto the interval $[-\kappa, \kappa]$. The resulting estimator is given by

$$\forall x \in \mathcal{X}, \ \hat{h}(x) = \frac{1}{\lambda nm}\sum_{i=1}^n\sum_{j=1}^m \hat{\alpha}_{ij} k_{\mathcal{X}}(x, x_i) k_\Theta(\cdot, \theta_j).$$

Algorithm 4.2 Projected Gradient Descent for Integral Huber

input : Gram matrices $\mathbf{K}_\mathcal{X}, \mathbf{K}_\Theta$, data matrix $\mathbf{Y}$, regularization parameter λ, Huber parameter κ, gradient step γ

init : $\boldsymbol{\alpha}^{(0)} = \mathbf{0} \in \mathbb{R}^{n \times m}$ or $\boldsymbol{\alpha}^{(0)} = \lambda nm \left(\mathbf{K}_\mathcal{X} \otimes \mathbf{K}_\Theta + \lambda nm \, \mathrm{Id}_{nm}\right)^{-1} \mathbf{Y}$

for *epoch* t *from* 0 *to* $T-1$ **do**

  // gradient step

  $\boldsymbol{\alpha}^{(t+1)} = \boldsymbol{\alpha}^{(t)} - \gamma \left(\boldsymbol{\alpha}^{(t)} + \frac{1}{\lambda nm} \mathbf{K}_\mathcal{X} \boldsymbol{\alpha}^{(t)} \mathbf{K}_\Theta - \mathbf{Y}\right)$

  // projection step

  for *row* i *from* 1 *to* n **do**

    for *column* j *from* 1 *to* m **do**

      $\alpha_{ij}^{(t+1)} = \operatorname{sign}\left(\alpha_{ij}^{(t+1)}\right) \min\left(\kappa, \left|\alpha_{ij}^{(t+1)}\right|\right)$

return $\boldsymbol{\alpha}^{(T)}$

4.4 Sparse Estimators with ϵ-Insensitive Losses

Let us recall the important notion of ϵ-insensitive losses. Learning with ϵ-insensitive losses forces the estimator to neglect small errors, preventing from overfitting and inducing some form of regularization, which will be highlighted later in the associated dual problems. As proposed in Sangnier et al. (2017); Laforgue et al. (2020) we extend them from $\mathbb{R}^p$ to any Hilbert space $\mathcal{Y}$.

Definition 4.12. *Let $L : \mathcal{Y} \to \mathbb{R}_+$ be a convex loss such that $L(0) = 0$, and $\epsilon > 0$. The ϵ-insensitive version of L, denoted L_ϵ, is defined by $L_\epsilon(y) = (L \,\square\, \chi_{\mathcal{B}_\epsilon})(y)$, or again:*

$$\forall y \in \mathcal{Y}, \quad L_\epsilon(y) = \begin{cases} 0 & \text{if } \|y\|_\mathcal{Y} \leq \epsilon \\ \inf\limits_{\|d\|_\mathcal{Y} \leq 1} L(y - \epsilon d) & \text{otherwise.} \end{cases}$$

In other terms, $L_\epsilon(y)$ is the smallest value of L within the ball of radius ϵ centered at y.

Remark 4.13. *The definition of L_ϵ depends on the metric used to characterize $\mathcal{B}_\epsilon$. We use the $\|\cdot\|_\mathcal{Y}$ norm as a natural choice, but one could envision variations of the ϵ-insensitive losses based on different metrics, leading to an alternative diverse family of dual problems.*

In general, it is not possible to find an analytic expression for L_ϵ. However, the ϵ-insensitive version of the square loss enjoys a closed-form representation as detailed below.

Definition 4.14. *We define the* vectorial ϵ-ridge loss *to be the ϵ-insensitive version of the square loss:*

$$\|\cdot\|^2_{\mathcal{Y},\epsilon} := \|\cdot\|^2_\mathcal{Y} \,\square\, \chi_{\mathcal{B}_\epsilon}(\cdot) = \max(\|\cdot\|_\mathcal{Y} - \epsilon, 0)^2. \tag{4.18}$$

The vectorial ϵ-ridge loss is illustrated in Figure 4.2 for dimension 1 and 2. Similarly to what was done with the Huber loss in Section 4.3, we can define an interesting loss on $\mathcal{Y}$ by integrating the local real ϵ-ridge losses.

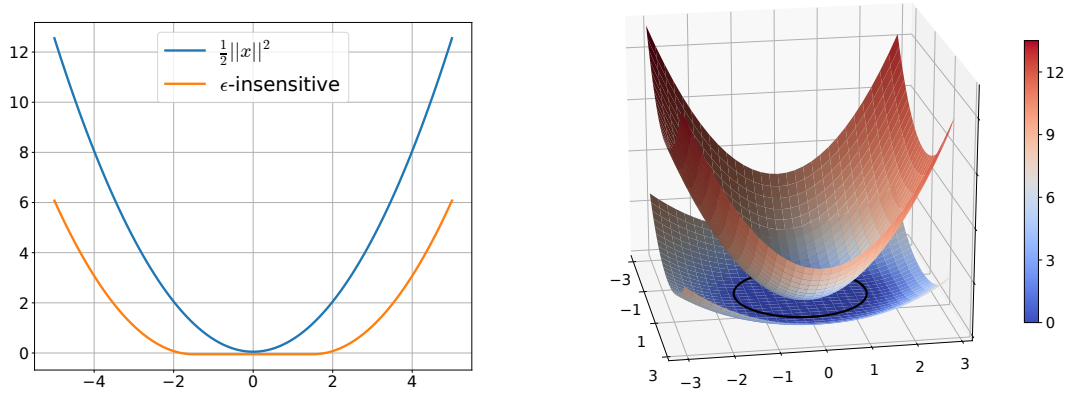


Figure 4.2 – Standard and ϵ-insensitive versions of the square loss in 1 and 2 dimensions ($\epsilon = 1.5$).

Definition 4.15. *We define the* integral ϵ-ridge loss *to be the integral loss associated to the ϵ-insensitive version of the real square loss:*

$$\forall y \in \mathcal{Y}, \quad I_{\frac{1}{2}(\cdot)_\epsilon^2}(y) := \int_\Theta \max(|y(\theta)| - \epsilon, 0)^2 \mathrm{d}\mu(\theta). \tag{4.19}$$

Having defined these two losses, we are ready to dive into the associated Problem 4.5. We start in Section 4.4.1 by proposing a solution method for the vectorial ϵ-insensitive ridge loss, before considering in Section 4.4.2 the integral ϵ-insensitive ridge loss function.

4.4.1 Vectorial ϵ-Ridge

The solution of Problem 4.6 with the vectorial ϵ-ridge is similar to its Huber counterpart presented in Section 4.3.1. We begin by giving the Fenchel-Legendre conjugate of $\|\cdot\|_{\mathcal{Y},\epsilon}^2$.

Proposition 4.16. *The Fenchel-Legendre conjugate of the vectorial ϵ-insensitive ridge loss is given by*

$$\forall y \in \mathcal{Y}, \quad \left(\|\cdot\|_{\mathcal{Y},\epsilon}^2\right)^\star (y) = \frac{1}{2}\|y\|_{\mathcal{Y}}^2 + \epsilon\|y\|_{\mathcal{Y}}. \tag{4.20}$$

Proof This is again an application of Proposition 2.4 applied to $f = \frac{1}{2}\|\cdot\|_{\mathcal{Y}}^2$ and $g = \chi_{\mathcal{B}_\epsilon}(\cdot)$, whose Fenchel-Legendre conjugates can be found in Table 2.1. ■

We can now apply this to Problem 4.6 in the vectorial ϵ-insensitive ridge loss scenario as stated below.

Proposition 4.17. *The dual to Problem 4.5 in the vectorial ϵ-insensitive ridge loss case writes as*

$$\inf_{(\alpha_i)_{i=1}^n \in \mathcal{Y}^n} \sum_{i=1}^n \frac{1}{2}\|\alpha_i\|_{\mathcal{Y}}^2 - \langle \alpha_i, y_i \rangle_{\mathcal{Y}} + \epsilon\|\alpha_i\|_{\mathcal{Y}} + \frac{1}{2\lambda n}\sum_{i=1}^n\sum_{j=1}^n k_{\mathcal{X}}(x_i, x_j)\left\langle \alpha_i, T_{k_\Theta}\alpha_j \right\rangle_{\mathcal{Y}}. \tag{4.21}$$

Proof This is the direct instantiation of Problem 4.6 with Fenchel-Legendre conjugate given by Proposition 4.16, combined with the properties of the Fenchel-Legendre conjugate with respect to the translation operator (see Table 2.1). ■

Algorithm 4.3 Proximal Gradient Descent for Vectorial ϵ-Ridge

input : Gram matrix $\mathbf{K}_{\mathcal{X}}$, matrix of eigenvalues $\mathbf{\Lambda}$, data scalar product matrix $\mathbf{R}$, regularization parameter λ, ridge parameter ϵ, gradient step γ

init : $\boldsymbol{\alpha}^{(0)} = \mathbf{0} \in \mathbb{R}^{n \times m}$ or $\boldsymbol{\alpha}^{(0)} = \lambda n \left(\mathbf{K}_{\mathcal{X}} \otimes \mathbf{\Lambda} + \lambda n \, \mathrm{Id}_{nm}\right)^{-1} \mathbf{R}$

for *epoch* t *from* 0 *to* $T-1$ **do**

  `// gradient step`

  $\boldsymbol{\alpha}^{(t+1)} = \boldsymbol{\alpha}^{(t)} - \gamma \left(\boldsymbol{\alpha}^{(t)} + \frac{1}{\lambda n} \mathbf{K}_{\mathcal{X}} \boldsymbol{\alpha}^{(t)} \mathbf{\Lambda} - \mathbf{R}\right)$

  `// proximal step`

  for *row* i *from* 1 *to* n **do**

    $\boldsymbol{\alpha}_{i:}^{(t+1)} = \mathrm{BST}\left(\boldsymbol{\alpha}_{i:}^{(t+1)}, \gamma\epsilon\right)$

return $\boldsymbol{\alpha}^{(T)}$

We can notice that Assumption 3.16 is again satisfied here, and exploit the same methodology from Section 3.4.2 as in Section 4.3.1. Keeping the same notations, one can rephrase Problem 4.21 as

$$\inf_{\boldsymbol{\alpha} \in \mathcal{M}_{n,m}(\mathbb{R})} \underbrace{\mathrm{Tr}\left(\frac{1}{2}\boldsymbol{\alpha}\boldsymbol{\alpha}^\top - \boldsymbol{\alpha}\mathbf{R}^\top + \frac{1}{2\lambda n}\mathbf{K}_{\mathcal{X}}\boldsymbol{\alpha}\mathbf{\Lambda}\boldsymbol{\alpha}^\top\right)}_{:= \mathcal{J}(\boldsymbol{\alpha})} + \epsilon \left\|\boldsymbol{\alpha}\right\|_{2,1}. \tag{4.22}$$

Problem 4.22 consists in the minimization of the finite-dimensional quadratic form $\mathcal{J}$ with multi-task Lasso regularization $\epsilon \left\|\cdot\right\|_{2,1}$. We can adapt the proximal gradient descent introduced in Algorithm 4.1 by choosing the suitable proximal operator which is the block soft thresholding (BST) operator, defined for all $x \in \mathbb{R}^m$ by

$$\mathrm{BST}(x, \epsilon) := \left|1 - \frac{\epsilon}{\left\|x\right\|_2}\right|_+ x.$$

The algorithm is summarized in Algorithm 4.3. Remark 4.9 about the calibration of the gradient step γ is still valid here.

4.4.2 Integral ϵ-Ridge

The solution of Problem 4.5 with the integral ϵ-ridge loss follows closely the steps in Section 4.3.2. We begin by computing the associated Fenchel-Legendre conjugate.

Proposition 4.18. *The Fenchel-Legendre conjugate of the integral ϵ-ridge loss is given by*

$$\forall y \in \mathcal{Y}, \quad \left(I_{\frac{1}{2}(\cdot)_\epsilon^2}\right)^\star (y) = \frac{1}{2}\left\|y\right\|_{\mathcal{Y}}^2 + \epsilon \int_\Theta |y(\theta)| \mathrm{d}\mu(\theta). \tag{4.23}$$

Proof This is an application of Proposition 3.13 which states that $I^\star_{\frac{1}{2}(\cdot)_\epsilon^2} = I_{\left(\frac{1}{2}(\cdot)_\epsilon^2\right)^\star}$. ■

We can now state Problem 4.6 for the integral ϵ-ridge loss scenario.

Proposition 4.19. *The dual to Problem 4.5 in the integral ϵ-ridge loss case can be written as*

$$\inf_{(\alpha_i)_{i=1}^n \in \mathcal{Y}^n} \sum_{i=1}^n \frac{1}{2} \left\|\alpha_i\right\|_{\mathcal{Y}}^2 - \langle \alpha_i, y_i \rangle_{\mathcal{Y}} + \epsilon \int_\Theta |\alpha_i(\theta)| \mathrm{d}\mu(\theta) + \frac{1}{2\lambda n} \sum_{i=1}^n \sum_{j=1}^n k_{\mathcal{X}}(x_i, x_j) \left\langle \alpha_i, T_{k_\Theta} \alpha_j \right\rangle_{\mathcal{Y}}. \tag{4.24}$$

Proof This is the direct instantiation of Problem 4.6 with Fenchel-Legendre conjugate given by Proposition 4.18, combined with the properties of the Fenchel-Legendre conjugate with respect to the translation operator (see Table 2.1). ■

In contrast to what was obtained in Section 4.4.1, Assumption 3.16 is no longer satisfied. This is due to the term $\epsilon \int_\Theta |\alpha_i(\theta)| \mathrm{d}\mu(\theta)$ in Problem 4.24 which cannot be computed from a representation of the dual variables in an orthonormal basis. We thus propose to rely on the splines methodology developed in Section 3.4.3, and begin by expressing the proximal operator further used in the proximal descent algorithm: as we recognize that

$$\epsilon \int_\Theta |\alpha_i(\theta)| \mathrm{d}\mu(\theta) = \epsilon \left\|\alpha_i\right\|_1$$

the proximal operator of the non-differentiable part can easily be computed using Moreau decomposition Proposition 2.6

$$\forall \gamma > 0, \quad \mathrm{prox}_{\gamma\epsilon\|\cdot\|_1} = \mathrm{Id} - \gamma \, \mathrm{prox}_{1/\gamma \underbrace{\left(\epsilon \|\cdot\|_1\right)^\star}_{\chi_{\mathcal{B}_\epsilon^\infty}}} \left(\frac{\cdot}{\gamma}\right) = \mathrm{Id} - \gamma \, \mathrm{Proj}_{\mathcal{B}_{\gamma\epsilon}^\infty}(\cdot).$$

Thus linear splines are adapted to the problem in the sense of Assumption 3.18, as they provide *pointwise control* of the dual variables, suited to the projection on $\mathcal{B}_{\gamma\epsilon}^\infty$. Using the same notations as in Section 4.3.2, an approximated Problem 4.24 writes as

$$\inf_{\boldsymbol{\alpha} \in \mathcal{M}_{n,m}(\mathbb{R})} \mathrm{Tr}\left(\frac{1}{2}\boldsymbol{\alpha}\boldsymbol{\alpha}^\top - \boldsymbol{\alpha}\mathbf{Y}^\top + \frac{1}{2\lambda nm}\mathbf{K}_{\mathcal{X}}\boldsymbol{\alpha}\mathbf{K}_\Theta\boldsymbol{\alpha}^\top\right) + \epsilon \left\|\boldsymbol{\alpha}\right\|_1. \tag{4.25}$$

We recognize in Problem 4.25 the minimization of a quadratic form with Lasso penalty $\epsilon \left\|\boldsymbol{\alpha}\right\|_1$. We propose to solve it using a proximal gradient algorithm presented in Algorithm 4.4. The proximal step involves the soft-thresholding operator

$$\mathrm{ST}(x, \epsilon) := \mathrm{sign}(x) \left||x| - \epsilon\right|_+,$$

which is known to induce sparsity in the iterates. The resulting estimator is given by

$$\forall x \in \mathcal{X}, \ \hat{h}(x) = \frac{1}{\lambda nm} \sum_{i=1}^n \sum_{j=1}^m \hat{\alpha}_{ij} k_{\mathcal{X}}(x, x_i) k_\Theta(\cdot, \theta_j).$$

4.5 Numerical Experiments

In this section, we present numerical experiments to illustrate the benefits of learning with convoluted losses. We consider the setting of *function-to-function regression*, where both the input variable X and the output variable Y are functions. The experiments relied on two datasets, one real and one synthetic.

Algorithm 4.4 Proximal Gradient Descent for Integral ϵ-ridge

input : Gram matrices $\mathbf{K}_\mathcal{X}, \mathbf{K}_\Theta$, data matrix $\mathbf{Y}$, regularization parameter λ, Huber parameter κ, gradient step γ

init : $\boldsymbol{\alpha}^{(0)} = \mathbf{0} \in \mathbb{R}^{n \times m}$ or $\boldsymbol{\alpha}^{(0)} = \lambda nm \left(\mathbf{K}_\mathcal{X} \otimes \mathbf{K}_\Theta + \lambda nm \, \mathrm{Id}_{nm}\right)^{-1} \mathbf{Y}$

for *epoch* t *from* 0 *to* $T-1$ **do**

  // gradient step

  $\boldsymbol{\alpha}^{(t+1)} = \boldsymbol{\alpha}^{(t)} - \gamma \left(\boldsymbol{\alpha}^{(t)} + \frac{1}{\lambda nm} \mathbf{K}_\mathcal{X} \boldsymbol{\alpha}^{(t)} \mathbf{K}_\Theta - \mathbf{Y}\right)$

  // proximal step

  for *row* i *from* 1 *to* n **do**

    for *column* j *from* 1 *to* m **do**

      $\alpha_{ij}^{(t+1)} = \mathrm{ST}\left(\alpha_{ij}^{(t+1)}, \gamma\epsilon\right)$

return $\boldsymbol{\alpha}^{(T)}$

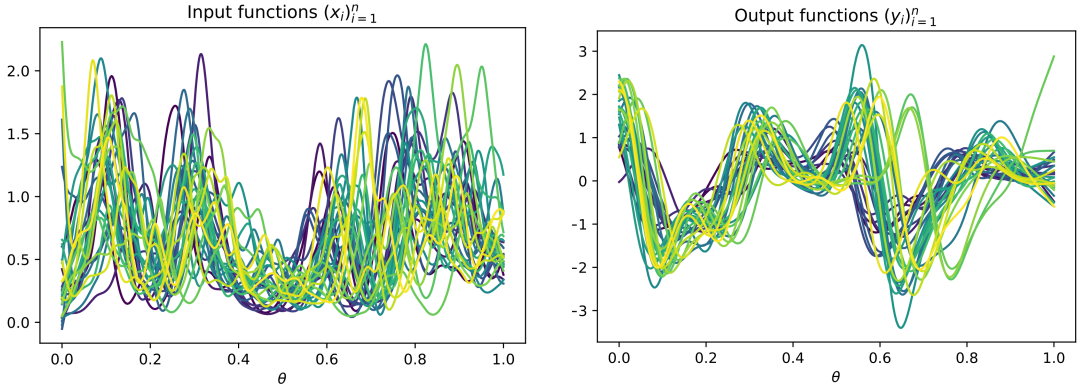


Figure 4.3 – Lip dataset.

Real dataset We choose to work with the *Lip* dataset from Ramsay (2004), which depicts the movement of the lip when a subject pronounces the word "bob" inside the phrase "say bob again". More precisely, the input observations $(x_i)_{i=1}^n$ are Electromyogram (EMG) recordings of a muscle that depresses the lower lip while the output observations $(y_i)_{i=1}^n$ describe the acceleration of the lower lip of the speaker, both seen as functions of the time. These functions are observed each millisecond, which forms a regular grid of size 641 considering the duration of the experiments; we rescaled this grid so that the domain of these functions is $\Theta := [0, 1]$. The underlying goal is to map the EMG signals onto the movements of the lip, allowing for a better understanding of how the brain controls the diction. We have access to $n = 32$ such recordings, whose illustration can be found in Figure 4.3.

Synthetic dataset To design a challenging task for our functional output regression framework, we adapted a synthetic dataset from Bouche et al. (2021). A pair of (input, output) functions is built as follows. We first draw $r = 4$ Gaussian processes (GPs) denoted by $(\mathrm{GP}_j^{\mathrm{in}})_{j=1}^r$ with mean 0 and Gaussian covariances of various bandwidths which will be used to model the input functions, and r other GPs to model the output functions, this time denoted by $(\mathrm{GP}_j^{\mathrm{out}})_{j=1}^r$. Both sets of GPs were kept fixed, and to create each pair of samples (x_i, y_i), we draw $(a_{ij})_{j \in [r]} \in \mathbb{R}^r$ uniformly in $[-2, 2]^r$, and

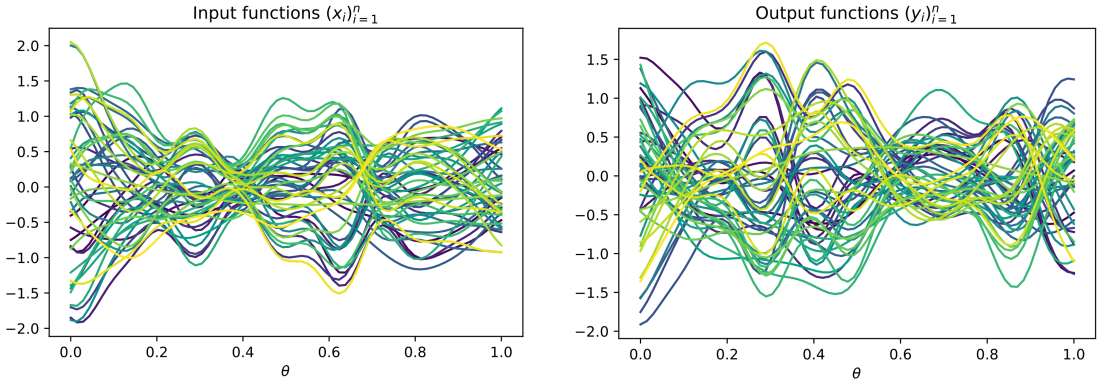


Figure 4.4 – Synthetic dataset. 50 samples are shown.

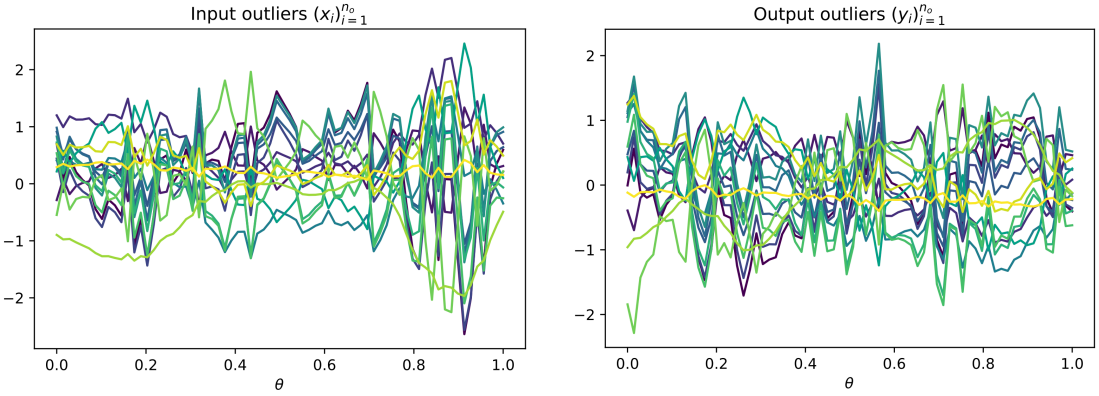


Figure 4.5 – Synthetic dataset global outliers. 15 samples are shown.

define

$$\forall \theta \in \Theta,\ x_i(\theta) = \sum_{j=1}^{r} a_{ij} \mathrm{GP}_j^{\mathrm{in}}(\theta), \qquad y_i(\theta) = \sum_{j=1}^{r} a_{ij} \mathrm{GP}_j^{\mathrm{out}}(\theta).$$

Examples of such functions can be found in Figure 4.4.

Outliers generation To illustrate the robustness properties brought by the use of the Huber and ϵ-insensitive losses, we propose to add two kinds of outliers to the aforementioned datasets. The first type of outliers, that we refer to as *local outliers*, model inconsistency in the data acquisition pipeline on a local level: we take a sample from the training set and add some noise to the measurements of the functions at m_o number of locations $(\tilde{\theta}_j)_{j=1}^{m_o}$ chosen randomly. The second type of outliers consists in *global outliers*, that is functions whose shape do not match the ones of the training samples. In particular, we create them the following way: for the lip dataset, we consider the case of a data acquisition error where an input function x_i is observed correctly, but associated to an output function $\tilde{y}_i = -s_o y_i$ where s_o is the scale of the outlier, fixed during the experiment and y_i is the normal observation. This is chosen to maximize the disalignment between inliers and outliers. For the synthetic dataset, we create new samples with different Gaussian processes in input and output, see Figure 4.5 for an illustration. We also use s_o to refer to the scale of the outliers, in the sense that the coefficients $(a_{ij})_{j\in[r]} \in \mathbb{R}^r$ are drawn uniformly in $[-s_o, s_o]^r$.

Model Specification We use a decomposable kernel with respective input and output kernels being $k_{\mathcal{X}}$ and k_Θ. The former is chosen to be an integrated Gaussian kernel, *i.e.* $k_{\mathcal{X}}(x,z) = \int_\Theta e^{-\gamma_{\mathcal{X}}[x(\theta)-z(\theta)]^2}$. The bandwidth $\gamma_{\mathcal{X}}$ is chosen via cross validation. For k_Θ, we chose a Gaussian kernel with bandwidth γ_Θ also chosen by cross validation. Since algorithms dealing with the vectorial Huber and ϵ-insensitive losses involve the eigendecomposition of T_{k_Θ}, in these cases we use a random Fourier features (RFF) approximation of the kernel for which the eigendecomposition of the associated integral operator is easy to get (see Example 2.30). The number of RFFs is set to 25 which provides functional spaces of dimension 50.

4.5.1 Huber Losses

In this section, we demonstrate the benefits of the two approaches developed in Section 4.3.1 and Section 4.3.2 in the global outlier scenario detailed above for both the real and synthetic dataset. We want to answer the following question:

- "Does the use of a Huber loss while training improves the mean square error at the testing phase when the training data is contaminated with global outliers?"

Real dataset Because of the limited amount of data provided for the *Lip* experiments, we are able to compute the leave-one-out (LOO) mean square error associated to the vectorial and integral Huber loss regressors. The setting is as follows: we augment the dataset with 4 global outliers (12.5%) generated by the procedure described above with various scale levels $s_o \in \{0.5, 1, 2, 5\}$. With a slight abuse, the hyperparameters $\lambda, \gamma_{\mathcal{X}}, \gamma_\Theta$ were computed once for each scale level using LOO cross validation for the ridge regressor, and kept fixed for the various settings obtained by trading the square loss against a Huber loss, thus results presented here are pessimistic *w.r.t.* the performances of the Huber loss estimators. For a given scale level, as κ grows, the constraint in the dual brought by the Huber loss becomes void, and we recover the ridge regression solution from (Kadri et al., 2016) in the vectorial Huber case, and from (Lian, 2007) in the integral Huber case. This phenomenon is illustrated in Figure 4.6, we observe that there exist a range of κ for which the LOO error (computed with the square loss) is better than the one of a model trained with a square loss itself, which suggests that in presence of outliers the dual constraint on the norm of the coefficients of the model helps to achieve robustness.

Synthetic dataset For the synthetic dataset, the procedure is roughly the same, except that it is now cheap to have access to new test samples, so that we do not use LOO error but rather measure the efficiency of the estimator on a fresh test set drawn from the same distribution as the inliers. We consider 250 normal observations from the dataset illustrated in Figure 4.4, and augment the training dataset with 25 outliers generated as in Figure 4.5 with various scale levels $s_o \in \{2, 4, 6, 10\}$. We then tune the hyperparameters by cross-validation on the gathered dataset, and report in Figure 4.7 mean square error obtained on a fresh test set generated from the inliers distribution. Numerically, there still exists a range of κ for which the Huber estimator performs better, but it is harder to see it with eye. This is certainly due to the fact that in this case it is harder to differentiate between the inliers and the outliers, whereas for the real dataset usecase the outliers were designed to maximize the confusion brought in to the estimator.

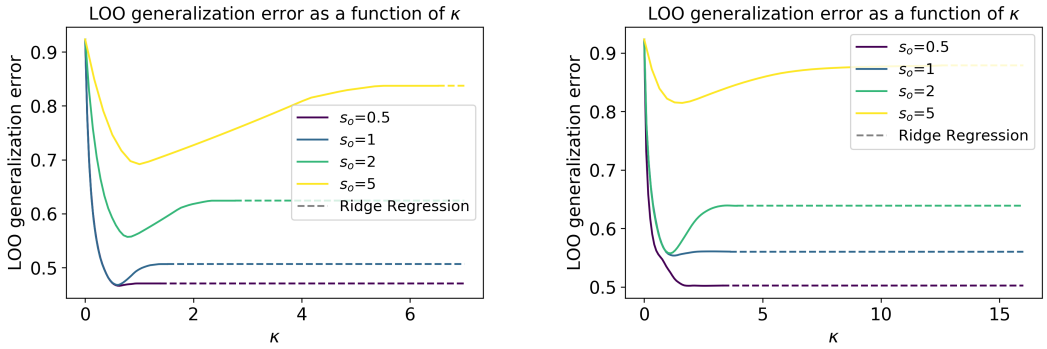


Figure 4.6 – Leave-one-out (LOO) generalization error when learning with Huber loss and contaminated real data for several scales of outliers $s_o \in \{0.5, 1, 2, 5\}$. Left: vectorial Huber loss. Right: integral Huber loss. For large enough Huber loss parameter κ the estimates (solid) coincide with that of the ridge solution (dashed).

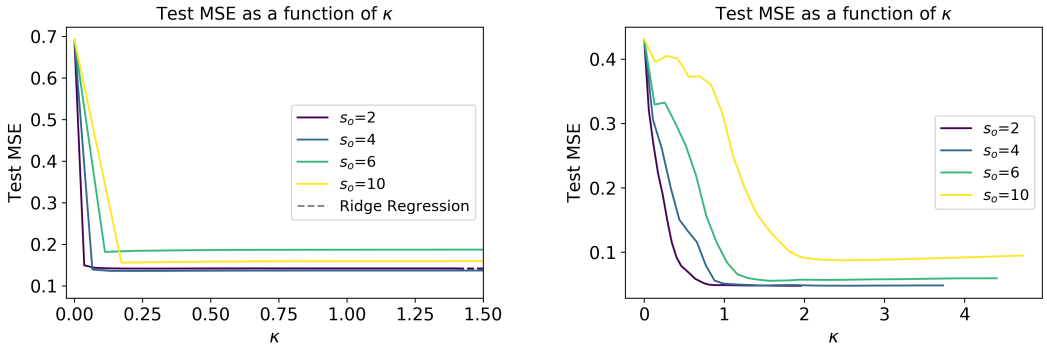


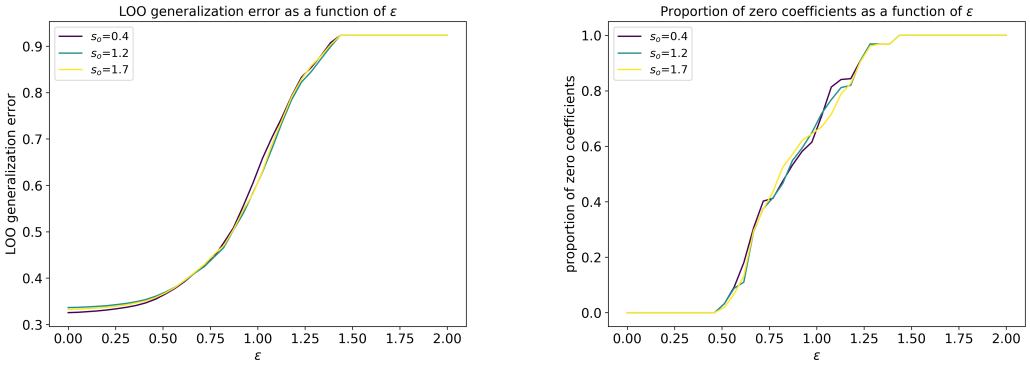
Figure 4.7 – Test MSE when learning with Huber loss and contaminated synthetic data for several scales of outliers $s_o \in \{2, 4, 6, 10\}$. Left: vectorial Huber loss. Right: integral Huber loss.

4.5.2 ϵ-insensitive Losses

In this section, we present numerical experiments to illustrate the effectiveness of the approaches developed in Section 4.4. In particular, we want to answer the following question:

- "At what price (in terms of test error) can we get sparse estimators?"

Here, the sparsity is to be understood in terms of the percentage of zero coefficients. As seen in the corresponding optimization problems, the vectorial and integral ϵ-insensitive losses induce sparsity as they respectively bring in Group-LASSO and LASSO penalization terms to the dual problem. When $\epsilon = 0$, we recover the unconstrained ridge regression solution, and as ϵ grows, the optimal coefficients become sparser. We expect that as ϵ grows, there is a degradation of the performances of the model in terms of test mean square error. Intuitively, the ϵ-insensitive ridge loss does not penalize the residuals whose norm is smaller than ϵ, and we propose to experiment with these losses in the context of local outliers, when some amount of noise is artificially added to the data measurement points. We present an illustration of this behavior for the vectorial ϵ-ridge, the two estimators behaving similarly.



(a) LOO error.

(b) Sparsity of the coefficients.

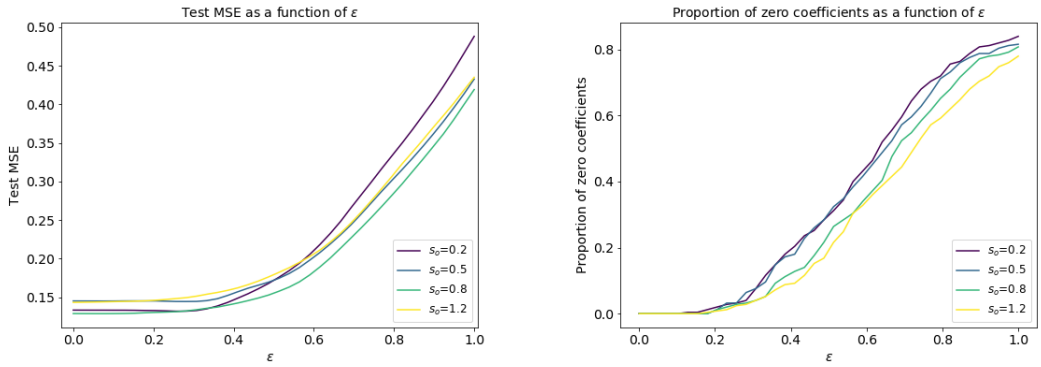
Figure 4.8 – LOO error and sparsity when learning with vectorial ϵ-ridge loss and contaminated real data for several scales of local outliers $s_o \in \{0.2, 0.5, 0.8, 1.2\}$. Left: Test MSE. Right: Proportion of zero coefficients.

Real Dataset Similarly to the Huber case, we used LOO error as a measure of performance and the same hyperparameters $(\lambda, \gamma_{\mathcal{X}}, \gamma_{\Theta})$. For the vectorial ϵ-ridge case, we plot in Figure 4.8 the LOO error and the proportion of zero coefficients. Since the problem is akin to a group lasso, this proportion corresponds to the number of training samples used in the estimator, as the $\|\cdot\|_{2,1}$ penalty forces some lines of the coefficient matrix to be zero. The local outliers were obtained by added randomly a Gaussian noise with standard deviation $s_o \in \{0.2, 0.5, 0.8, 1.2\}$ at 20% of the locations associated to 50% of the training data. As expected, the LOO error degrades when ϵ grows. We observe little influence of the scale of the outliers on this experiment.

Synthetic Dataset We used test MSE on fresh test samples to assess the performance of the estimator. The local outliers were obtained by adding Gaussian noise with standard deviation $s_o \in \{0.4, 1.2, 1.7\}$ at 20% of the locations associated to 50% of the training data. The results are illustrated in Figure 4.9, we observe similarly to the real dataset case that as ϵ grows, the coefficients of the estimators grow sparser and the test MSE degrades.

4.6 Conclusion

In this chapter, we proposed a functional output regression framework based on duality in vv-RKHS that allows to go beyond the square loss case. The use of convoluted losses, particularly suited to duality thanks to their compatibility with the Fenchel-Legendre transform, produces a family of problems that read in the dual as the original regularized empirical risk minimization problem plus a term akin to a penalty, depending on the chosen loss. This term can enforce robustness, as seen with the family of the Huber losses, or sparsity when choosing the family of ϵ-insensitive losses. To obtain them, we convolve the square loss with a loss that is norm dependent, inviting to investigate in future work the choice of the norm used in the right term of the convolution. This would likely lead to different ways to enforce sparsity or robustness.



(a) Test MSE.

(b) Sparsity of the coefficients.

Figure 4.9 – Test MSE and sparsity when learning with vectorial ϵ-ridge loss and contaminated synthetic data for several scales of local outliers $s_o \in \{0.4, 1.2, 1.7\}$. Left: Test MSE. Right: Proportion of zero coefficients.

5

Infinite Task Learning

Contents

In this chapter, we propose to exploit the general framework of learning with integral losses from Chapter 3 to adopt a new angle to multi-task learning. Multi-task learning consists in leveraging dependency across tasks to jointly solve multiple task with a single vector-valued model (Micchelli and Pontil, 2005). We extend the idea of multiple tasks to a continuum of *parameterized tasks*, and present a principled way to jointly learn these tasks referred to as *infinite task learning* (ITL).

After a brief introduction in Section 5.1, we devote Section 5.2 to the definition of the ITL framework, before studying its generalization capabilities in Section 5.3. In Section 5.4 we apply the ITL methodology to quantile regression and demonstrate its efficiency on both synthetic and real-world datasets. Section 5.5 is devoted to the cost-sensitive classification problem, and finally Section 5.6 explores the unsupervised density level set estimation problem under the ITL angle. Conclusions are drawn in Section 5.7.

5.1 Introduction

Several fundamental problems in machine learning and statistics can be phrased as the minimization of a loss function described by a hyperparameter. The hyperparameter might capture numerous aspects of the problem: (i) the tolerance *w.r.t.* outliers as the ϵ-insensitivity in support vector regression (SVR; Drucker et al. 1997), (ii) importance of smoothness or sparsity such as the weight of the l_2-norm in Tikhonov regularization (Tikhonov and Arsenin, 1977), l_1-norm in LASSO (Tibshirani, 1996), or more general structured-sparsity inducing norms (Bach et al., 2012), (iii) density level set estimation (DLSE), see for example one-class support vector machines (OCSVM), (iv) confidence as exemplified by quantile regression (QR), or (v) importance of different decisions as implemented by cost-sensitive classification (CSC). In various cases including QR, CSC or DLSE, one is interested in solving the parameterized task for several hyperparameter values. Multi-task learning (MTL; Evgeniou and Pontil 2004) provides a principled way of benefiting from the relationship between similar tasks while preserving local properties of the algorithms: ν-property in DLSE (Glazer et al., 2013) or quantile property in QR (Takeuchi et al., 2006).

A natural extension from the traditional multi-task setting is to provide a prediction tool being able to deal with *any* value of the hyperparameter. In their seminal work, (Takeuchi et al., 2013) extended multi-task learning by considering an infinite number of parameterized tasks in a framework called parametric task learning. Assuming that the loss is piecewise affine in the hyperparameter, the authors were able to get the whole solution path through parametric programming, relying on techniques developed by Hastie et al. (2004).[1] In this chapter, we relax the affine model assumption on the tasks as well as the piece-wise linear assumption on the loss, and take a different angle. We propose infinite task learning (ITL) within the framework of function-valued function learning to handle a continuum number of parameterized tasks. For that purpose we leverage tools from operator-valued kernels and the associated vv-RKHS. The idea is that the output is a function over the hyperparameter space, modelled as an element of a scalar-valued RKHS. Properties of this kernel, *e.g.* continuity, give an explicit control on the relationship between tasks, and manipulating output functions in a RKHS also enables us to consider incorporate specific constraints on their nature. In the studied framework each task is described by a (scalar-valued) RKHS over the input space which is capable of dealing with nonlinearities. The resulting ITL formulation relying on vv-RKHS specifically encompasses existing multi-task approaches including joint quantile regression (Sangnier et al., 2016) or multi-task variants of density level set estimation (Glazer et al., 2013) by encoding a continuum of tasks.

5.2 From Parameterized to Infinite Task Learning

In this section, we gradually define our goal by moving from single parameterized tasks (Section 5.2.1) to infinite task learning (ITL; Section 5.2.3) through multi-task learning (MTL; Section 5.2.2).

[1] Alternative optimization techniques to deal with countable or continuous hyperparameter spaces could include semi-infinite (Stein, 2012) or bi-level programming (Wen and Hsu, 1991).

5.2.1 Learning Parameterized Tasks

A *supervised parametrized task* is defined as follows. Let $(\mathsf{X}, \mathsf{Y}) \in \mathcal{X} \times \mathbb{R}$ be a random variable with joint distribution $\mathbb{P}_{\mathsf{X},\mathsf{Y}}$ which is assumed to be fixed but unknown. We have access to n i.i.d. observations called training samples: $\mathcal{S} := (x_i, y_i)_{i=1}^n \sim \mathbb{P}_{\mathsf{X},\mathsf{Y}}^{\otimes n}$. Let Θ be the domain of hyperparameters, and $\ell \colon \Theta \times \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ be a loss function. Let $\mathcal{H}_{k_\mathcal{X}} \subset \mathcal{F}\left(\mathcal{X}; \mathbb{R}\right)$ denote our hypothesis class; it is assumed to be a RKHS with kernel $k_\mathcal{X} \colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. For a given θ, the goal is to estimate the minimizer of the expected risk

$$\mathcal{R}(\theta, h) := \mathbb{E}_{\mathsf{X},\mathsf{Y}}[\ell(\theta, h(\mathsf{X}), \mathsf{Y})] \tag{5.1}$$

over $\mathcal{H}_{k_\mathcal{X}}$, using the training sample $\mathcal{S}$. This task can be addressed by solving the regularized empirical risk minimization problem

$$\min_{h \in \mathcal{H}_{k_\mathcal{X}}} \mathcal{R}_\mathcal{S}(\theta, h) + \Omega(h), \tag{5.2}$$

where $\mathcal{R}_\mathcal{S}(\theta, h) := \frac{1}{n}\sum_{i=1}^n \ell(\theta, h(x_i), y_i)$ is the empirical risk and $\Omega : \mathcal{H}_{k_\mathcal{X}} \to \mathbb{R}$ is a regularizer. Below we give three examples.

Quantile regression (QR): In this setting $\theta \in \left(0, 1\right)$. For a given hyperparameter level θ, in QR the goal is to predict the θ-quantile of the real-valued output conditional distribution $\mathbb{P}_{\mathsf{Y}|\mathsf{X}}$. The task can be tackled using the pinball loss (Koenker and Bassett Jr, 1978) defined in Equation (5.3) and illustrated in Figure 5.1.

$$\begin{aligned} \ell(\theta, h(x), y) &= \max(\theta(y - h(x)), (\theta - 1)(y - h(x))), \\ \Omega(h) &= \frac{\lambda}{2}\left\|h\right\|_{\mathcal{H}_{k_\mathcal{X}}}^2 \quad (\lambda > 0). \end{aligned} \tag{5.3}$$

Cost-sensitive classification (CSC): Our next example considers binary classification ($\mathsf{Y} \in \{-1, 1\}$) where a (possibly) different cost is associated with each class, as it is often the case in medical diagnosis or default detection (Japkowicz and Stephen, 2002; Elkan, 2001). The sign of $h \in \mathcal{H}_{k_\mathcal{X}}$ yields the estimated class and in cost-sensitive classification one takes

$$\begin{aligned} \ell(\theta, h(x), y) &= \left|\frac{\theta + 1}{2} - \mathbb{1}_{\{-1\}}(y)\right| \max(0, 1 - yh(x)), \\ \Omega(h) &= \frac{\lambda}{2}\left\|h\right\|_{\mathcal{H}_{k_\mathcal{X}}}^2 \quad \lambda > 0. \end{aligned} \tag{5.4}$$

The $\theta \in [-1, 1]$ hyperparameter captures the trade-off between the importance of correctly classifying the samples having -1 and $+1$ labels. When θ is close to -1, the obtained h focuses on classifying well class -1, and vice-versa. Typically, it is desirable for a physician to choose *a posteriori* the value of the hyperparameter at which she wants to predict. Since this cost can rarely be considered to be fixed, this motivates to learn one model giving access to all hyperparameter values.

Density level sets estimation (DLSE): Examples of parameterized tasks can also be found in the unsupervised setting. For instance in outlier detection, the goal is to separate outliers from inliers. A classical technique to tackle this task is the celebrated *one-class support vector machines* (OCSVM; Schölkopf et al. 2000). OCSVM has a

free parameter $\theta \in (0, 1]$, which can be proven to be an upper bound on the fraction of outliers. When using a Gaussian kernel with a bandwidth converging to zero, OCSVM consistently estimates density level sets (Vert and Vert, 2006). This unsupervised learning problem can be empirically described by the minimization of a regularized empirical risk $\mathcal{R}_{\mathcal{S}}(\theta, t, h) + \Omega(h)$, solved *jointly* over $h \in \mathcal{H}_{k_{\mathcal{X}}}$ *and* $t \in \mathbb{R}$ with

$$\ell(\theta, h(x), t) = -t + \frac{1}{\theta} \left| t - h(x) \right|_+ , \qquad \Omega(h) = \frac{1}{2} \left\| h \right\|^2_{\mathcal{H}_{k_{\mathcal{X}}}} .$$

5.2.2 Solving a Finite Number of Tasks as Multi-Task Learning

In all the aforementioned problems, one is rarely interested in the choice of a single hyperparameter value (θ) and associated risk $\Big(\mathcal{R}_{\mathcal{S}}(\theta, \cdot)\Big)$, but rather in the joint solution of multiple tasks. The naive approach of solving the different tasks independently can easily lead to inconsistencies. A principled way of solving many parameterized tasks has been cast as a MTL problem (Evgeniou et al., 2005) which takes into account the similarities between tasks and helps providing consistent solutions. For example it is possible to encode the similarities of the different tasks in MTL through an explicit constraint function (Ciliberto et al., 2017). In the ITL approach, the similarity between tasks is designed in an implicit way through the use of a kernel on the hyperparameters. Moreover, in contrast to MTL, in our case the input space and the training samples are the same for each task; a task is specified by a value of the hyperparameter. This setting is sometimes referred to as multi-output learning (Álvarez et al., 2012).

Formally, assume that we have m tasks described by parameters $(\theta_j)_{j=1}^m$. The idea of multi-task learning is to minimize the sum of the local loss functions $\mathcal{R}_{\mathcal{S}}(\theta_j, \cdot)$, *i.e.*

$$\operatorname*{arg\,min}_{h \in \mathcal{H}} \sum_{j=1}^{m} \mathcal{R}_{\mathcal{S}}(\theta_j, h_j) + \Omega(h),$$

where the individual tasks are modelled by the real-valued h_j functions, the overall $\mathbb{R}^m$-valued model is the vector-valued function $h \colon x \mapsto (h_1(x), \ldots, h_m(x))$ belonging to some hypothesis space $\mathcal{H}$, and Ω is a regularization term encoding similarities between tasks.

It is instructive to consider two concrete examples:

- In joint quantile regression one can use the regularizer to encourage that the predicted conditional quantile estimates for two similar quantile values are similar. This idea forms the basis of the approach proposed by Sangnier et al. (2016) who formulates the joint quantile regression problem in a vector-valued reproducing kernel Hilbert space with an appropriate decomposable kernel that encodes the links between the tasks. The obtained solution shows less quantile curve crossings compared to estimators not exploiting the dependencies of the tasks as well as an improved accuracy.
- A multi-task version of DLSE has been presented by Glazer et al. (2013) with the goal of obtaining nested density level sets as θ grows. Similarly to joint quantile regression, it is crucial to take into account the similarities of the tasks in the joint model to efficiently solve this problem.

5.2.3 Towards Infinite Task Learning

In the following, we propose a novel framework called infinite task learning (ITL) in which we learn a function-valued function $h \in \mathcal{H}_K$ where $\mathcal{H}_K \subset \mathcal{F}(\mathcal{X}, \mathcal{F}(\Theta, \mathbb{R}))$ is a vv-RKHS. Our goal is to be able to handle new tasks after the learning phase and thus, not to be limited to given predefined values of the hyperparameter. Regarding this goal, our framework generalizes the *parametric task learning* (PTL) approach introduced by Takeuchi et al. (2013), by allowing a wider class of models and relaxing the hypothesis of piece-wise linearity of the loss function. Moreover a nice byproduct of this vv-RKHS based approach is that one can benefit from the functional point of view, design new regularizers and impose various constraints on the whole continuum of tasks, *e.g.* ,

- The continuity of the $\theta \mapsto h(x)(\theta)$ function is a natural desirable property: for a given input x, the predictions on similar tasks should also be similar.
- Another example is to impose a shape constraint in QR: the conditional quantile should be increasing *w.r.t.* the hyperparameter θ. This requirement can be imposed through the functional view of the problem.
- In DLSE, to get nested level sets, one would want that for all $x \in \mathcal{X}$, the decision function $\theta \mapsto \mathbb{1}_{\mathbb{R}_+}(h(x)(\theta) - t(\theta))$ changes its sign only once.

To keep the presentation simple, in the sequel we are going to focus on ITL in the supervised setting; we dedicate Section 5.6 to the unsupervised task of DSLE.

We introduce the integral loss function

$$I_\ell(h(x), y) := \int_\Theta \ell(\theta, h(x)(\theta), y) \mathrm{d}\mu(\theta), \tag{5.5}$$

where $\ell \colon \Theta \times \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is a loss function and μ is a probability measure on Θ which encodes the importance of the prediction at different hyperparameter values. Without prior information and for compact Θ, one may consider μ to be uniform. The true and empirical risks read then as

$$\mathcal{R}(h) := \mathbb{E}_{\mathsf{X},\mathsf{Y}}\left[I_\ell(h(\mathsf{X}), \mathsf{Y})\right], \qquad \mathcal{R}_{\mathcal{S}}(h) := \frac{1}{n}\sum_{i=1}^{n} I_\ell(h(x_i), y_i). \tag{5.6}$$

Intuitively, minimizing the expectation of the integral over θ in a rich enough space corresponds to searching for a pointwise minimizer $x \mapsto \hat{h}(x)(\theta)$ of the parametrized tasks introduced in Equation (5.1) with, for instance, the implicit space constraint that $\theta \mapsto \hat{h}(x)(\theta)$ is a continuous function for each input x. We show in Proposition 5.25 that this is precisely the case in QR.

Interestingly, the empirical counterpart of the true risk minimization can now be considered with a much richer family of penalty terms:

$$\min_{h \in \mathcal{H}_K} \mathcal{R}_{\mathcal{S}}(h) + \Omega(h). \tag{5.7}$$

Here, $\Omega(h)$ can be a weighted sum of various penalties

- imposed directly on h such as $\frac{\lambda}{2}\left\|h\right\|^2_{\mathcal{H}_K}$ for $\lambda > 0$, in which case we recognize in Problem 5.7 the type of problems whose resolution has been discussed at length in Chapter 3 or

- integrated constraints on either $\theta \mapsto h(x)(\theta)$ or $x \mapsto h(x)(\theta)$ such as

$$\int_{\mathcal{X}} \Omega_1(h(x)(\cdot))\mathrm{d}\mathbb{P}(x) \text{ or } \int_{\Theta} \Omega_2(h(\cdot)(\theta))\mathrm{d}\mu(\theta)$$

  which allow the property enforced by Ω_1 or Ω_2 to hold pointwise on $\mathcal{X}$ or Θ respectively.

It is worthwhile to see a concrete example before turning to the statistical analysis in Section 5.3: in quantile regression, the monotonicity assumption of the $\theta \mapsto h(x)(\theta)$ function can be encoded by choosing Ω_1 as

$$\Omega_1(f) = \lambda_{nc} \int_{\Theta} \left| -f'(\theta) \right|_+ \mathrm{d}\mu(\theta)$$

when the output RKHS $\mathcal{H}_{k_\Theta}$ is populated with differentiable functions and $\lambda_{nc} > 0$ controls the strength of the soft constraint.

5.3 Generalization Analysis through Uniform Stability

In this section, we study the *generalization gap* (or *excess risk*) associated to the ITL method. The goal here is to control with high probability the quantity

$$\mathcal{E}(\hat{h}) := \mathcal{R}(\hat{h}) - \mathcal{R}_{\mathcal{S}}(\hat{h}), \tag{5.8}$$

where $\hat{h}$ is the estimator resulting from the ITL method in the supervised setting:

$$\hat{h} = \underset{h \in \mathcal{H}_K}{\arg\min} \ \frac{1}{n} \sum_{i=1}^{n} I_\ell(h(x_i), y_i) + \lambda \left\| h \right\|^2_{\mathcal{H}_K}. \tag{5.9}$$

Remark 5.1. *The reader can notice that the regularization parameter here is λ and not $\frac{\lambda}{2}$ as it helps readability of the generalization bound.*

When $\mathcal{E}(\hat{h})$ is small, the learned estimator performs comparably on unseen data drawn with probability $\mathbb{P}_{(\mathsf{X},\mathsf{Y})}$, whereas when it is large, the estimator *overfits*, meaning it performs poorly on new data. A myriad of techniques have been developed to study the generalization error (Steinwart and Christmann, 2008; Zhou, 2002; Rudi and Rosasco, 2017), extending the seminal work of (Vapnik, 1992). In the particular kernel based learning, we can invoke the *Rademacher complexity* tool (Bartlett and Mendelson, 2002) which allows to bound $\sup_{h \in \mathcal{H}_K} |\mathcal{E}(h)|$ and benefits from a large body of work, with extensions to vector-valued learning (Maurer and Pontil, 2016). However, these extensions rely on the assumption of trace class OVKs which is not verified for the kernel $K = k_{\mathcal{X}} \operatorname{Id}_{\mathcal{H}_{k_\Theta}}$. Moreover, bounding $\sup_{h \in \mathcal{H}} |\mathcal{E}(h)|$ appears unnecessarily hard, since we are only interested in bounding $\mathcal{E}(\hat{h})$.

These considerations lead us to work in the framework of *uniform stability* introduced in (Bousquet and Elisseeff, 2002). Uniform stability asks the question: "what happens when we learn an estimator with a slightly modified dataset ?" and answers by saying that when the two resulting estimators are close, then it is possible to bound the excess risk of the output of the algorithm.

Definition 5.2. *Let $\mathcal{S} = (x_i, y_i)_{i=1}^n$ be the training data. We call $\mathcal{S}^i$ the training data $\mathcal{S}^i = ((x_1, y_1), \dots, (x_{i-1}, y_{i-1}), (x_{i+1}, y_{i+1}), \dots, (x_n, y_n))$, $i \in [n]$.*

In the following, $L \colon \mathcal{H}_K \times \mathcal{X} \times \mathbb{R}$ is a loss function whose dependence *w.r.t.* to the model h is explicit, and the risks are

$$\mathcal{R}(h) = \mathbb{E}_{(\mathsf{X},\mathsf{Y})}\left[L(h, \mathsf{X}, \mathsf{Y})\right], \qquad \mathcal{R}_{\mathcal{S}}(h) = \frac{1}{n}\sum_{i=1}^{n} L(h, x_i, y_i).$$

Remark 5.3. *To fit in this framework, we will choose either*

$$L(h, x, y) = I_\ell(h(x), y) \quad \text{or} \quad L(h, x, y) = \widetilde{I}_\ell(h(x), y) \tag{5.10}$$

depending on whether we analyse the theoretical ITL estimator (see Section 5.3.1) or the one obtained from the practical solution with a sampled empirical risk and double representer theorem (see Section 5.3.2).

Definition 5.4. *A learning algorithm mapping a dataset $\mathcal{S}$ to an estimator $\hat{h}_{\mathcal{S}}$ is said to be β-uniformly stable with respect to the loss function L if for all $n \in \mathbb{N}^*$, $i \in [n]$, and $\mathcal{S}$ training set,*

$$\left\| L(\hat{h}_{\mathcal{S}}, \cdot, \cdot) - L(\hat{h}_{\mathcal{S}^i}, \cdot, \cdot) \right\|_\infty \leq \beta. \tag{5.11}$$

Assumption 5.5. *There exists $\xi \geq 0$ such that for all $(x, y) \in \mathcal{X} \times \mathcal{Y}$ and training set $\mathcal{S}$, $L(\hat{h}_{\mathcal{S}}, x, y) \leq \xi$.*

Having defined these concepts, we are now ready to state the generalization bound from (Bousquet and Elisseeff, 2002) that allows to quantify the excess risk of an estimator provided by a learning algorithm.

Proposition 5.6. *(Bousquet and Elisseeff, 2002) Let $\mathcal{S} \mapsto \hat{h}_{\mathcal{S}}$ be a learning algorithm with uniform stability β with respect to a loss L satisfying Assumption 5.5. Then for all $n \geq 1$ and $\delta \in (0, 1)$, with probability at least $1 - \delta$ on the drawing of the samples, it holds that*

$$\mathcal{E}\left(\hat{h}_{\mathcal{S}}\right) \leq 2\beta + (4n\beta + \xi)\sqrt{\frac{\log(1/\delta)}{n}}.$$

Proposition 5.6 ensures that whenever $\sqrt{n\beta} \underset{n\to\infty}{\to} 0$, the true risk $\mathcal{R}(\hat{h}_{\mathcal{S}})$ is controlled by its empirical version $\mathcal{R}_{\mathcal{S}}(\hat{h}_{\mathcal{S}})$, and the bounds are tight when β scales as $\frac{1}{n}$. The bounds associated to uniform stability have recently benefited from sharper results proposed by Feldman and Vondrak (2018) and later Bousquet et al. (2020). We present it below for completeness.

Proposition 5.7. *(Bousquet et al., 2020) Let $\mathcal{S} \mapsto \hat{h}_{\mathcal{S}}$ be a learning algorithm with uniform stability β with respect to a loss L satisfying Assumption 5.5. Then for all $n \geq 1$ and $\delta \in (0, 1)$, with probability at least $1 - \delta$ on the drawing of the samples, it holds that*

$$\left|\mathcal{E}\left(\hat{h}_{\mathcal{S}}\right)\right| \leq c_1 \beta \log_2(n) \log(e/\delta) + c_2 \xi \sqrt{\frac{\log(e/\delta)}{n}}$$

where $c_1 = 12\sqrt{2}e$ and $c_2 = 4e$.

This bound shows both *sub-Gaussian* ($\sqrt{\log e/\delta}$ term) and *sub-exponential* ($\log e/\delta$ term) regimes and can be of great interest for β that decrease slower than $\frac{1}{n}$ (see Bousquet et al. (2020) for a discussion about this). In our case as we will see β scales as $\frac{1}{n}$; we present the results implied by Proposition 5.6 and note the analysis can be adapted to Proposition 5.7. This is the strength of the uniform stability framework, all we have to do is prove that the algorithms are β-stable, and we can profit from both bounds.

Uniform stability and vv-RKHS learning We now exploit results from Kadri et al. (2016) (we can also mention the initial work of Audiffren and Kadri (2013)), who studied the uniform stability of learning algorithms based on empirical risk minimization with vv-RKHS norm regularization. There is a slight difference between their setting and ours, since they use losses defined for some y in the output space of the vv-RKHS, but this difference has no impact on the validity of the proofs. Their work relies on the following assumptions:

Assumption 5.8. *There exists $\kappa > 0$ such that for $\forall x \in \mathcal{X}$, $\left\|K(x,x)\right\|_{\text{op}} \leq \kappa^2$.*

Assumption 5.9. *For all $h_1, h_2 \in \mathcal{H}_{k_\Theta}$, the function*

$$(x_1, x_2) \in \mathcal{X} \times \mathcal{X} \mapsto \langle K(x_1, x_2)h_1, h_2 \rangle_{\mathcal{H}_{k_\Theta}} \in \mathbb{R}$$

is measurable.

We recall the choice of OVK we make: $K(x,z) = k_{\mathcal{X}}(x,z)I_{\mathcal{H}_{k_\Theta}}$ with $(x,z) \in \mathcal{X} \times \mathcal{X}$, $k_{\mathcal{X}}$ and k_Θ are bounded scalar-valued kernels; in other words there exist $(\kappa_{\mathcal{X}}, \kappa_\Theta) \in \mathbb{R}^2$ such that $\sup_{x \in \mathcal{X}} k_{\mathcal{X}}(x,x) \leq \kappa_{\mathcal{X}}^2$ and $\sup_{\theta \in \Theta} k_\Theta(\theta,\theta) \leq \kappa_\Theta^2$.

Remark 5.10. *Assumptions 5.8, 5.9 are satisfied for our choice of kernel.*

Assumption 5.11. *The application $(y,h,x) \mapsto L(y,h,x)$ is σ-admissible,* i.e. *convex with respect to h and Lipschitz continuous with respect to $f(x)$, with σ as its Lipschitz constant.*

We now state a proposition from Kadri et al. (2016) which gives β-stability guarantees for the output of the algorithm

$$\hat{h}_{\mathcal{S}} = \operatorname*{arg\,min}_{h \in \mathcal{H}_K} \frac{1}{n} \sum_{i=1}^{n} L(h, x_i, y_i) + \lambda \left\|h\right\|_{\mathcal{H}_K}^2 . \tag{5.12}$$

Proposition 5.12. *(Kadri et al., 2016) Under Assumption 5.8, Assumption 5.9, and Assumption 5.11, a learning algorithm that maps a training set $\mathcal{S}$ to the function $\hat{h}_{\mathcal{S}}$ defined in Equation (5.12) is β-stable with $\beta = \frac{\sigma^2\kappa^2}{2\lambda n}$.*

5.3.1 Generalization Bound for Non-Approximated Scheme

In this section, we derive generalization bounds based on Proposition 5.12 for the non-approximated ITL scheme, in the sense that we analyse the theoretical estimator without considering the approximations induced by the practical optimization

algorithm:

$$\hat{h}_{\mathcal{S}} = \underset{h \in \mathcal{H}_K}{\arg\min} \ \frac{1}{n} \sum_{i=1}^{n} L(h, x_i, y_i) + \lambda \left\| h \right\|_{\mathcal{H}_K}^2 \tag{5.13}$$

where

$$L(h, x, y) = \int_{\Theta} \ell(\theta, h(x)(\theta), y) \mathrm{d}\mu(\theta). \tag{5.14}$$

Quantile regression We recall that in this setting,

$$\ell(\theta, h(x)(\theta), y) = \max\left(\theta(y - h(x)(\theta)), (1-\theta)(y - h(x)(\theta))\right)$$

and consider the associated L from Equation (5.14). Moreover, we will assume that $|\mathsf{Y}|$ is bounded almost surely by $B \in \mathbb{R}$. We will therefore verify the hypothesis for $y \in [-B, B]$ and not $y \in \mathbb{R}$.

Lemma 5.13. *In the case of the QR, the loss L is σ-admissible with $\sigma = 2\kappa_\Theta$.*

Proof Let $h_1, h_2 \in \mathcal{H}_K$ and $\theta \in [0,1]$. $\forall x, y \in \mathcal{X} \times \mathbb{R}$, it holds that

$$\ell(\theta, h_1(x)(\theta), y) - \ell(\theta, h_2(x)(\theta), y) = (\theta - t)(h_2(x)(\theta) - h_1(x)(\theta)) + (t - s)(y - h_1(x)(\theta)),$$

where $s = \mathbb{1}_{y \leq h_1(x)(\theta)}$ and $t = \mathbb{1}_{y \leq h_2(x)(\theta)}$. We consider all possible cases for t and s :

- $t = s = 0$: $|(t-s)(y - h_1(x)(\theta))| \leq |h_2(x)(\theta) - h_1(x)(\theta)|$,
- $t = s = 1$: $|(t-s)(y - h_1(x)(\theta))| \leq |h_2(x)(\theta) - h_1(x)(\theta)|$,
- $s = 1$, $t = 0$: $|(t-s)(y - h_1(x)(\theta))| = |h_1(x)(\theta) - y| \leq |h_1(x)(\theta) - h_2(x)(\theta)|$,
- $s = 0$, $t = 1$: $|(t-s)(y - h_1(x)(\theta))| = |y - h_1(x)(\theta)| \leq |h_1(x)(\theta) - h_2(x)(\theta)|$ because of the conditions on t, s.

Thus

$$\begin{aligned}\left|\ell(\theta, h_1(x)(\theta), y) - \ell(\theta, h_2(x)(\theta), y)\right| &\leq (\theta+1)|h_1(x)(\theta) - h_2(x)(\theta)| \\ &\leq (\theta+1)\kappa_\Theta ||h_1(x) - h_2(x)||_{\mathcal{H}_{k_\Theta}},\end{aligned}$$

where we used the Cauchy-Schwartz inequality applied to $h_1(x) - h_2(x)$ and $k_\Theta(\cdot, \theta)$ in $\mathcal{H}_{k_\Theta}$ at the last line. By integrating this expression over the Θ space, we get that

$$\begin{aligned}\left|L(h_1, x, y) - L(h_2, x, y)\right| &\leq \int_0^1 (\theta+1)\kappa_\Theta ||h_1(x) - h_2(x)||_{\mathcal{H}_{k_\Theta}} \mathrm{d}\mu(\theta) \\ &\leq 2\kappa_\Theta ||h_1(x) - h_2(x)||_{\mathcal{H}_{k_\Theta}}\end{aligned}$$

and L is σ-admissible with $\sigma = 2\kappa_\Theta$. ■

Lemma 5.14. *Let $\mathcal{S} = ((x_1, y_1), \ldots, (x_n, y_n))$ be a training set and $\lambda > 0$. Then for all $(x, \theta) \in \mathcal{X} \times (0,1)$, it holds that $|\hat{h}_{\mathcal{S}}(x)(\theta)| \leq \kappa_{\mathcal{X}} \kappa_\Theta \sqrt{\frac{B}{\lambda}}$.*

Proof Since $\hat{h}_{\mathcal{S}}$ is the output of our algorithm and $0 \in \mathcal{H}_K$, it holds that

$$\lambda ||\hat{h}_{\mathcal{S}}||^2 \leq \frac{1}{n} \sum_{i=1}^{n} \int_0^1 \ell(\theta, 0, y_i) \mathrm{d}\mu(\theta) \leq \frac{1}{n} \sum_{i=1}^{n} \int_0^1 \max(\theta, 1-\theta)|y_i| \mathrm{d}\mu(\theta) \leq B.$$

Thus $||\hat{h}_{\mathcal{S}}|| \leq \sqrt{\frac{B}{\lambda}}$. Moreover, $\forall x, \theta \in \mathcal{X} \times (0,1)$,

$$\begin{aligned}\left|\hat{h}_{\mathcal{S}}(x)(\theta)\right| &= \left|\langle \hat{h}_{\mathcal{S}}(x), k_\Theta(\theta,\cdot)\rangle_{\mathcal{H}_{k_\Theta}}\right| \\ &\leq \kappa_\Theta \left\|\hat{h}_{\mathcal{S}}(x)\right\|_{\mathcal{H}_{k_\Theta}} && \text{by Cauchy-Schwartz inequality in } \mathcal{H}_{k_\Theta} \\ &\leq \kappa_\mathcal{X}\kappa_\Theta \left\|\hat{h}_{\mathcal{S}}\right\|_{\mathcal{H}_K} && \text{by the bounded operator norm of } K_x^\sharp\end{aligned}$$

which concludes the proof. ■

Lemma 5.15. *Assumption 5.5 is satisfied for* $\xi = 2\left(B + \kappa_\mathcal{X}\kappa_\Theta\sqrt{\frac{B}{\lambda}}\right)$.

Proof Let $\mathcal{S} = ((x_1, y_1), \ldots, (x_n, y_n))$ be a training set and $\hat{h}_{\mathcal{S}}$ be the output of our algorithm. For all $(x, y) \in \mathcal{X} \times [-B, B]$, it holds that

$$\begin{aligned}L(y, \hat{h}_{\mathcal{S}}, x) &= \int_0^1 \max\left(\theta(y - \hat{h}_{\mathcal{S}}(x)(\theta)), (\theta - 1)(y - \hat{h}_{\mathcal{S}}(x)(\theta))\right) \mathrm{d}\mu(\theta) \\ &\leq 2\int_0^1 \left|y - \hat{h}_{\mathcal{S}}(x)(\theta)\right| \mathrm{d}\mu(\theta) \leq 2\int_0^1 |y| + \left|\hat{h}_{\mathcal{S}}(x)(\theta)\right| \mathrm{d}\mu(\theta) \\ &\leq 2\left(B + \kappa_\mathcal{X}\kappa_\Theta\sqrt{\frac{B}{\lambda}}\right).\end{aligned}$$

■

Proposition 5.16. *The QR learning algorithm defined in Equation (5.13) is such that for all* $n \geq 1$ *and* $\delta \in (0,1)$*, with probability at least* $1-\delta$ *on the drawing of the samples, it holds that*

$$\mathcal{R}\left(\hat{h}_{\mathcal{S}}\right) \leq \mathcal{R}_{\mathcal{S}}\left(\hat{h}_{\mathcal{S}}\right) + \frac{4\kappa_\mathcal{X}^2\kappa_\Theta^2}{\lambda n} + \left[\frac{8\kappa_\mathcal{X}^2\kappa_\Theta^2}{\lambda} + 2\left(B + \kappa_\mathcal{X}\kappa_\Theta\sqrt{\frac{B}{\lambda}}\right)\right]\sqrt{\frac{\log(1/\delta)}{n}}. \quad (5.15)$$

Proof *This is a direct consequence of Proposition 5.12, Proposition 5.6, Lemma 5.13 and Lemma 5.15.* ■

Cost-sensitive classification In this setting, the loss ℓ is

$$\ell(\theta, h(x)(\theta), y) = \left|\frac{\theta+1}{2} - \mathbb{1}_{\{-1\}}(y)\right| \left|1 - yh(x)(\theta)\right|_+$$

and the associated L is given by Equation (5.14). One can verify (the way it was done for quantile regression) that the properties above still hold, but with constants

$$\sigma = \kappa_\Theta, \qquad \beta = \frac{\kappa_\mathcal{X}^2\kappa_\Theta^2}{2\lambda n}, \qquad \xi = 1 + \frac{\kappa_\mathcal{X}\kappa_\Theta}{\sqrt{\lambda}}.$$

This allows to formulate the following bound.

Proposition 5.17. *The CSC learning algorithm defined in Equation (5.13) is such that for all $n \geq 1$ and $\delta \in (0,1)$, with probability at least $1-\delta$ on the drawing of the samples, it holds that*

$$\mathcal{R}\left(\hat{h}_{\mathcal{S}}\right) \leq \mathcal{R}_{\mathcal{S}}\left(\hat{h}_{\mathcal{S}}\right) + \frac{\kappa_{\mathsf{X}}^2 \kappa_{\Theta}^2}{\lambda n} + \left(\frac{2\kappa_{\mathsf{X}}^2 \kappa_{\Theta}^2}{\lambda} + 1 + \frac{\kappa_{\mathsf{X}} \kappa_{\Theta}}{\sqrt{\lambda}}\right) \sqrt{\frac{\log(1/\delta)}{n}}.$$

5.3.2 Generalization Bound for Approximated Scheme

In this section, we derive generalization bounds based on Proposition 5.12 for the approximated ITL scheme

$$\hat{h}_{\mathcal{S}} = \operatorname*{arg\,min}_{h \in \mathcal{H}_K} \frac{1}{n} \sum_{i=1}^{n} L(h, x_i, y_i) + \lambda \left\|h\right\|_{\mathcal{H}_K}^2$$

where

$$L(h, x, y) = \frac{1}{m} \sum_{j=1}^{m} \ell(\theta_j, h(x)(\theta_j), y). \tag{5.16}$$

and $(\theta_j)_{j=1}^m$ is a quasi Monte-Carlo sequence used to approximate the loss function as proposed in Section 3.3.1. We remind the reader of the statistical quantities:

$$\mathcal{R}\left(\hat{h}_{\mathcal{S}}\right) = \mathbb{E}_{\mathsf{X},\mathsf{Y}}\left[I_\ell(\hat{h}_{\mathcal{S}}(\mathsf{X}), \mathsf{Y})\right], \qquad \mathcal{R}_{\mathcal{S}}\left(\hat{h}_{\mathcal{S}}\right) = \frac{1}{n}\sum_{i=1}^{n} I_\ell(\hat{h}_{\mathcal{S}}(x_i), y_i)$$

$$\widetilde{\mathcal{R}}\left(\hat{h}_{\mathcal{S}}\right) = \mathbb{E}_{\mathsf{X},\mathsf{Y}}\left[\frac{1}{m}\sum_{j=1}^{m} \ell(\theta_j, \hat{h}_{\mathcal{S}}(\mathsf{X})(\theta_j), \mathsf{Y})\right], \quad \widetilde{\mathcal{R}}_{\mathcal{S}}\left(\hat{h}_{\mathcal{S}}\right) = \frac{1}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m} \ell(\theta_j, \hat{h}_{\mathcal{S}}(x_i)(\theta_j), y_i).$$

The uniform stability framework then informs about the generalization gap between $\widetilde{\mathcal{R}}$ and $\widetilde{\mathcal{R}}_{\mathcal{S}}$, and additional work is needed to control the quantity $\mathcal{R} - \widetilde{\mathcal{R}}_{\mathcal{S}}$. In particular we make use of convergence properties of QMC integration which benefit from $\mathcal{O}\left(\frac{\log m}{m}\right)$ convergence rates for bounded variation functions.

Definition 5.18 (Hardy-Krause variation)**.** *Let Π be the set of subdivisions of the interval $\Theta = [0,1]$. A subdivision will be denoted by $\sigma = (\theta_1 = 0, \theta_2, \ldots, \theta_p = 1)$ and $f\colon \Theta \to \mathbb{R}$ be a function. Using these notations, the Hardy-Krause variation of the function f is defined as $V(f) = \sup_{\sigma \in \Pi} \sum_{i=1}^{p-1} \left|f(\theta_{i+1}) - f(\theta_i)\right|$.*

Remark 5.19. *If f is continuous, $V(f)$ is also the limit as the mesh of σ goes to zero of the above quantity.*

In the following, let $f\colon \theta \mapsto \mathbb{E}_{\mathsf{X},\mathsf{Y}}\left[\ell(\theta, \hat{h}_{\mathcal{S}}(\mathsf{X})(\theta), \mathsf{Y})\right]$. This function is of primary importance for our analysis, since in the quasi Monte-Carlo setting, the convergence properties of the integration scheme only hold if the function to integrate has finite Hardy-Krause variation.

Quantile regression The following lemma quantifies the Hardy-Krause variation of the function f defined above in the QR setting.

Lemma 5.20. *Assume the boundeness of both scalar-valued kernels $k_{\mathcal{X}}$ and k_Θ. Assume moreover that k_Θ is $\mathcal{C}^1$ and that its partial derivatives are uniformly bounded by some constant C. Finally, assume that Y is almost surely bounded by some constant B. Then*

$$V(f) \leq B + \kappa_{\mathcal{X}}\kappa_\Theta\sqrt{\frac{B}{\lambda}} + 2\kappa_{\mathcal{X}}\sqrt{\frac{2BC}{\lambda}}. \tag{5.17}$$

Proof It holds that

$$\begin{aligned}
V(f) &= \sup_{\sigma\in\Pi}\sum_{i=1}^{p-1}\left|f(\theta_{i+1}) - f(\theta_i)\right| \\
&= \sup_{\sigma\in\Pi}\sum_{i=1}^{p-1}\left|\mathbb{E}_{\mathsf{X},\mathsf{Y}}\left[\ell(\theta_{i+1}, \hat{h}_{\mathcal{S}}(\mathsf{X})(\theta_{i+1}), \mathsf{Y})\right] - \mathbb{E}_{\mathsf{X},\mathsf{Y}}\left[\ell(\theta_i, \hat{h}_{\mathcal{S}}(\mathsf{X})(\theta_i), \mathsf{Y})\right]\right| \\
&= \sup_{\sigma\in\Pi}\sum_{i=1}^{p-1}\left|\mathbb{E}_{\mathsf{X},\mathsf{Y}}\left[\ell(\theta_{i+1}, \hat{h}_{\mathcal{S}}(\mathsf{X})(\theta_{i+1}), \mathsf{Y}) - \ell(\theta_i, \hat{h}_{\mathcal{S}}(\mathsf{X})(\theta_i), \mathsf{Y})\right]\right| \\
&\leq \sup_{\sigma\in\Pi}\sum_{i=1}^{p-1}\mathbb{E}_{\mathsf{X},\mathsf{Y}}\left[\left|\ell(\theta_{i+1}, \hat{h}_{\mathcal{S}}(\mathsf{X})(\theta_{i+1}), \mathsf{Y}) - \ell(\theta_i, \hat{h}_{\mathcal{S}}(\mathsf{X})(\theta_i), \mathsf{Y})\right|\right] \\
&\leq \sup_{\sigma\in\Pi}\mathbb{E}_{\mathsf{X},\mathsf{Y}}\left[\sum_{i=1}^{p-1}\left|\ell(\theta_{i+1}, \hat{h}_{\mathcal{S}}(\mathsf{X})(\theta_{i+1}), \mathsf{Y}) - \ell(\theta_i, \hat{h}_{\mathcal{S}}(\mathsf{X})(\theta_i), \mathsf{Y})\right|\right].
\end{aligned}$$

The supremum of the expectation is smaller than the expectation of each supremum, hence

$$V(f) \leq \int V(f_{x,y})\mathrm{d}\mathbb{P}_{X,Y}, \tag{5.18}$$

where $f_{x,y}\colon \theta \mapsto \ell(\theta, y, \hat{h}_{\mathcal{S}}(x)(\theta))$ is the local counterpart of the function f at point (x, y). To bound this quantity, let us first bound locally $V(f_{x,y})$. To that extent, we fix some (x, y) in the following. Since $f_{x,y}$ is continuous (because k_Θ is $\mathcal{C}^1$), then using Choquet (1969, Theorem 24.6), it holds that

$$V(f_{x,y}) = \lim_{|\sigma|\to 0}\sum_{i=1}^{p-1}\left|f_{x,y}(\theta_{i+1}) - f_{x,y}(\theta_i)\right|.$$

Moreover since $k \in \mathcal{C}^1$ and $\partial k_\theta = (\partial_1 k)(\cdot, \theta)$ has a finite number of zeros for all $\theta \in \Theta$, one can assume that in the subdivision considered afterhand all the zeros (in θ) of the residuals $y - \hat{h}_{\mathcal{S}}(x)(\theta)$ are present, so $y - \hat{h}_{\mathcal{S}}(x)(\theta_{i+1})$ and $y - \hat{h}_{\mathcal{S}}(x)(\theta_i)$ are always of the same sign. Indeed, if not, create a new, finer subdivision with this property and work with this one. Let us begin the proper calculation: let $\sigma = (\theta_1, \theta_2, \ldots, \theta_p)$ be a subdivision of Θ, it holds that $\forall i \in \{1, \ldots, p-1\}$:

$$\begin{aligned}
\left|f_{x,y}(\theta_{i+1}) - f_{x,y}(\theta_i)\right| = \Big|&\max\left(\theta_{i+1}(y - \hat{h}_{\mathcal{S}}(x)(\theta_{i+1})), (1-\theta_{i+1})(y - \hat{h}_{\mathcal{S}}(x)(\theta_{i+1}))\right) \\
&- \max\left(\theta_i(y - \hat{h}_{\mathcal{S}}(x)(\theta_i)), (1-\theta_{i+1})(y - \hat{h}_{\mathcal{S}}(x)(\theta_i))\right)\Big|.
\end{aligned}$$

We now study the two possible outcomes for the residuals:

- If $y - h(x)(\theta_{i+1}) \geq 0$ and $y - h(x)(\theta_i) \geq 0$ then

$$\begin{aligned}\left|f_{x,y}(\theta_{i+1}) - f_{x,y}(\theta_i)\right| &= \left|\theta_{i+1}(y - \hat{h}_{\mathcal{S}}(x)(\theta_{i+1})) - \theta_i(y - \hat{h}_{\mathcal{S}}(x)(\theta_i))\right| \\ &= \Big|(\theta_{i+1} - \theta_i)y + (\theta_i - \theta_{i+1})\hat{h}_{\mathcal{S}}(x)(\theta_{i+1}) \\ &\qquad + \theta_i(\hat{h}_{\mathcal{S}}(x)(\theta_i) - \hat{h}_{\mathcal{S}}(x)(\theta_{i+1}))\Big| \\ &\leq \left|(\theta_{i+1} - \theta_i)y\right| + \left|(\theta_i - \theta_{i+1})\hat{h}_{\mathcal{S}}(x)(\theta_{i+1})\right| \\ &\qquad + \left|\theta_i(\hat{h}_{\mathcal{S}}(x)(\theta_i) - \hat{h}_{\mathcal{S}}(x)(\theta_{i+1}))\right|.\end{aligned}$$

From Lemma 5.14, it holds that $\hat{h}_{\mathcal{S}}(x)(\theta_{i+1}) \leq \kappa_{\mathcal{X}}\kappa_\Theta\sqrt{\frac{B}{\lambda}}$. Moreover,

$$\begin{aligned}\left|\hat{h}_{\mathcal{S}}(x)(\theta_i) - \hat{h}_{\mathcal{S}}(x)(\theta_{i+1})\right| &= \left|\left\langle \hat{h}_{\mathcal{S}}(x), k_\Theta(\theta_i, \cdot) - k_\Theta(\theta_{i+1}, \cdot)\right\rangle_{\mathcal{H}_{k_\Theta}}\right| \\ &\leq \left\|\hat{h}_{\mathcal{S}}(x)\right\|_{\mathcal{H}_{k_\Theta}} \left\|k_\Theta(\theta_i, \cdot) - k_\Theta(\theta_{i+1}, \cdot)\right\|_{\mathcal{H}_{k_\Theta}} \\ &\leq \kappa_{\mathcal{X}}\sqrt{\frac{B}{\lambda}}\sqrt{\left|k_\Theta(\theta_i, \theta_i) + k_\Theta(\theta_{i+1}, \theta_{i+1}) - 2k_\Theta(\theta_{i+1}, \theta_i)\right|} \\ &\leq \kappa_{\mathcal{X}}\sqrt{\frac{B}{\lambda}}\Bigg(\sqrt{\left|k_\Theta(\theta_{i+1}, \theta_{i+1}) - k_\Theta(\theta_{i+1}, \theta_i)\right|} \\ &\qquad + \sqrt{\left|k_\Theta(\theta_i, \theta_i) - k_\Theta(\theta_{i+1}, \theta_i)\right|}\Bigg).\end{aligned}$$

Since k_Θ is $\mathcal{C}^1$, with partial derivatives uniformly bounded by C, it holds that

$$\left|k_\Theta(\theta_{i+1}, \theta_{i+1}) - k_\Theta(\theta_{i+1}, \theta_i)\right| \leq C(\theta_{i+1} - \theta_i)$$

and

$$\left|k_\Theta(\theta_i, \theta_i) - k_\Theta(\theta_{i+1}, \theta_i)\right| \leq C(\theta_{i+1} - \theta_i)$$

so

$$\left|\hat{h}_{\mathcal{S}}(x)(\theta_i) - \hat{h}_{\mathcal{S}}(x)(\theta_{i+1})\right| \leq \kappa_{\mathcal{X}}\sqrt{\frac{2BC}{\lambda}}\sqrt{\theta_{i+1} - \theta_i}$$

and overall

$$\left|f_{x,y}(\theta_{i+1}) - f_{x,y}(\theta_i)\right| \leq \left(B + \kappa_{\mathcal{X}}\kappa_\Theta\sqrt{\frac{B}{\lambda}}\right)(\theta_{i+1} - \theta_i) + \kappa_{\mathcal{X}}\sqrt{\frac{2BC}{\lambda}}\sqrt{\theta_{i+1} - \theta_i}.$$

- If $y - h(x)(\theta_{i+1}) \leq 0$ and $y - h(x)(\theta_i) \leq 0$, then

$$\begin{aligned}\left|f_{x,y}(\theta_{i+1}) - f_{x,y}(\theta_i)\right| &= \left|(1 - \theta_{i+1})(y - \hat{h}_{\mathcal{S}}(x)(\theta_{i+1})) - (1 - \theta_i)(y - \hat{h}_{\mathcal{S}}(x)(\theta_i))\right| \\ &\leq \left|\hat{h}_{\mathcal{S}}(x)(\theta_i) - \hat{h}_{\mathcal{S}}(x)(\theta_{i+1})\right| + \left|(\theta_{i+1} - \theta_i)y\right| \\ &\quad + \left|(\theta_i - \theta_{i+1})\hat{h}_{\mathcal{S}}(x)(\theta_{i+1})\right| + \left|\theta_i(\hat{h}_{\mathcal{S}}(x)(\theta_i) - \hat{h}_{\mathcal{S}}(x)(\theta_{i+1}))\right|\end{aligned}$$

so with similar arguments one gets

$$\left|f_{x,y}(\theta_{i+1}) - f_{x,y}(\theta_i)\right| \leq \left(B + \kappa_\chi \kappa_\Theta \sqrt{\frac{B}{\lambda}}\right)(\theta_{i+1} - \theta_i) + 2\kappa_\chi \sqrt{\frac{2BC}{\lambda}}\sqrt{\theta_{i+1} - \theta_i}. \tag{5.19}$$

Therefore, regardless of the sign of the residuals $y - h(x)(\theta_{i+1})$ and $y - h(x)(\theta_i)$, one gets Equation (5.19). Since the square root function has Hardy-Kraus variation of 1 on the interval $\Theta = [0, 1]$, it holds that

$$\sup_{\sigma \in \Pi} \sum_{i=1}^{p-1} |f_{x,y}(\theta_{i+1}) - f_{x,y}(\theta_i)| \leq B + \kappa_\chi \kappa_\Theta \sqrt{\frac{B}{\lambda}} + 2\kappa_\chi \sqrt{\frac{2BC}{\lambda}}.$$

Combining this with Equation (5.18) finally gives

$$V(f) \leq B + \kappa_\chi \kappa_\Theta \sqrt{\frac{B}{\lambda}} + 2\kappa_\chi \sqrt{\frac{2BC}{\lambda}}.$$

■

We are now ready to compare the true risk $\mathcal{R}$ of the ITL estimator with its sampled version $\widetilde{\mathcal{R}}$ based on the sampled integral loss $\widetilde{I}_\ell$.

Lemma 5.21. *Let $\mathcal{R}$ be the risk defined in Equation (5.6) for the quantile regression problem. Assume that $(\theta)_{j=1}^m$ have been generated via the Sobol sequence and that k_Θ is $\mathcal{C}^1$ with its partial derivatives being uniformly bounded by some constant C. Then*

$$\left|\mathcal{R}\left(\hat{h}_\mathcal{S}\right) - \widetilde{\mathcal{R}}\left(\hat{h}_\mathcal{S}\right)\right| \leq \left(B + \kappa_\chi \kappa_\Theta \sqrt{\frac{B}{\lambda}} + 2\kappa_\chi \sqrt{\frac{2BC}{\lambda}}\right)\frac{\log(m)}{m}. \tag{5.20}$$

Proof Let $f \colon \theta \mapsto \mathbb{E}_{\mathsf{X},\mathsf{Y}}\left[\ell(\theta, \hat{h}_\mathcal{S}(\mathsf{X})(\theta), \mathsf{Y})\right]$. It holds that

$$|\mathcal{R}(\hat{h}_\mathcal{S}) - \widetilde{\mathcal{R}}(\hat{h}_\mathcal{S})| \leq V(f)\frac{\log(m)}{m}$$

according to the Koksma-Hlawka inequality (Morokoff and Caflisch, 1995), where $V(f)$ is the Hardy-Krause variation of f. Lemma 5.20 allows then to conclude. ■

When gathered together, Lemma 5.21 and Proposition 5.16 provide a bound on the true risk of the estimator *w.r.t.* its sampled empirical risk.

Proposition 5.22. *Let $\hat{h}_\mathcal{S}$ be the quantile estimator resulting from Equation (5.13) with sampled loss from Equation (5.16). Assume that Y is almost surely bounded, the kernels $k_\mathcal{X}$, k_Θ are bounded and k_Θ is $\mathcal{C}^1$ with bounded derivatives. Then it holds that*

$$\mathcal{R}\left(\hat{h}_\mathcal{S}\right) \leq \widetilde{\mathcal{R}}_\mathcal{S}\left(\hat{h}_\mathcal{S}\right) + \mathcal{O}_{\mathbb{P}_{\mathsf{X},\mathsf{Y}}}\left(\frac{1}{\lambda\sqrt{n}}\right) + \mathcal{O}\left(\frac{\log m}{\sqrt{\lambda}m}\right). \tag{5.21}$$

Proof This is a direct application of Lemma 5.21 and Proposition 5.16. ■

Cost-sensitive classification The bound for the CSC case follows the same reasoning, and the Hardy-Kraus variation of the function $f\colon \theta \mapsto \mathbb{E}_{\mathsf{X},\mathsf{Y}}\left[\ell(\theta, \hat{h}_{\mathcal{S}}(\mathsf{X})(\theta), \mathsf{Y})\right]$ is bounded by the following lemma.

Lemma 5.23. *Assume the boundeness of both scalar-valued kernels $k_{\mathcal{X}}$ and k_{Θ}. Assume moreover that k_{Θ} is $\mathcal{C}^1$ and that its partial derivatives are uniformly bounded by some constant C. Then*

$$V(f) \leq 1 + \kappa_{\mathcal{X}}\kappa_{\Theta}\sqrt{\frac{1}{\lambda}} + 2\kappa_{\mathcal{X}}\sqrt{\frac{2C}{\lambda}}. \tag{5.22}$$

Proof The proof is similar to the one of Lemma 5.20, replacing the residual $y - \hat{h}_{\mathcal{S}}(x)$ by $1 - y\hat{h}_{\mathcal{S}}(x)$ and performing the case distinction whether $1 - y\hat{h}_{\mathcal{S}}(x)$ is positive of negative. ■

5.4 Quantile Regression

This section is dedicated to quantile regression when using ITL to learn the *whole* quantile function. As a reminder, given an $\mathcal{X}$-valued random variable X and real-valued output random variable Y we define its conditional quantile function by

$$q(x)(\theta) = \inf_{u\in\mathbb{R}} \left\{t \,:\, \mathbb{P}[\mathsf{Y} \leq u|\mathsf{X} = x] = \theta\right\}, \quad (x,\theta) \in \mathcal{X} \times (0,1).$$

While the conditional quantile function is not defined for $\theta = 0$ and $\theta = 1$, it can be extended to these values by continuity whenever Y is a bounded random variable.

Remark 5.24. *By definition, for all $x \in \mathcal{X}$, the conditional quantile function $q(x)$ is a nondecreasing function. However, estimates based on the ITL method do not necessarily satisfy this property, and it can be challenging to impose such shape constraint for the estimator. This will be discussed in Section 5.4.1.*

It is well-known (Koenker and Bassett Jr, 1978) that for any quantile level $\theta \in (0,1)$ and point $x \in \mathcal{X}$, the quantile enjoys a variational formula

$$q(x)(\theta) = \operatorname*{arg\,min}_{u\in\mathbb{R}} \; \mathbb{E}_{\mathsf{Y}|\mathsf{X}=x}\left[\ell(\theta, u, \mathsf{Y})\right],$$

where the pinball loss is defined by

$$\ell(\theta, u, v) : (u, v) \in \mathbb{R}^2 \mapsto \max\left(\theta(v-u), (\theta-1)(v-u)\right) \in \mathbb{R}, \tag{5.23}$$

and it is illustrated in Figure 5.1. The following proposition validates the introduction of the integral pinball loss I_ℓ to tackle the estimation of the whole quantile function in one go.

Proposition 5.25. *Let X, Y be two r.v. respectively taking values in $\mathcal{X}$ and $\mathbb{R}$, with Y bounded almost surely. Let $q\colon \mathcal{X} \to \mathcal{F}([0,1],\mathbb{R})$ be the associated conditional quantile function and μ be a positive probability measure on $[0,1]$ such that $\mu(\{0\}) = \mu(\{1\}) = 0$. Then for $\forall h \in \mathcal{F}(\mathcal{X}, \mathcal{F}([0,1],\mathbb{R}))$*

$$\mathcal{R}(h) - \mathcal{R}(q) \geq 0,$$

where $\mathcal{R}$ is the risk defined in Equation (5.6).

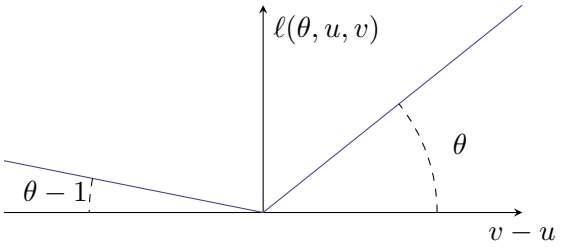


Figure 5.1 – Pinball loss for $\theta = 0.8$.

Proof The proof is based on the one given (Li et al., 2007) for a single quantile. Let $f \in \mathcal{F}(\mathcal{X}, \mathcal{F}([0,1], \mathbb{R}))$, $\theta \in (0,1)$ and $(x, y) \in \mathcal{X} \times \mathbb{R}$. Let also

$$s = \begin{cases} 1 \text{ if } y \leq f(x)(\theta) \\ 0 \text{ otherwise} \end{cases}, \qquad t = \begin{cases} 1 \text{ if } y \leq q(x)(\theta) \\ 0 \text{ otherwise} \end{cases}.$$

It holds that

$$\begin{aligned} &\ell(\theta, h(x)(\theta), y) - \ell(\theta, q(x)(\theta), y) = \\ &\quad = \theta(1-s)(y - h(x)(\theta)) + (\theta - 1)s(y - h(x)(\theta)) \\ &\qquad -\theta(1-t)(y - q(x)(\theta)) - (\theta - 1)t(y - q(x)(\theta)) \\ &\quad = \theta(1-t)(q(x)(\theta) - h(x)(\theta)) + \theta((1-t) - (1-s))h(x)(\theta) \\ &\qquad +(\theta - 1)t(q(x)(\theta - h(x)(\theta))) + (\theta - 1)(t-s)h(x)(\theta) + (t-s)y \\ &\quad = (\theta - t)(q(x)(\theta) - h(x)(\theta)) + (t-s)(y - h(x)(\theta)). \end{aligned}$$

Then, notice that

$$\begin{aligned} \mathbb{E}\Big[(\theta - t)(q(\mathsf{X})(\theta) - h(\mathsf{X})(\theta))\Big] &= \mathbb{E}\Big[\mathbb{E}\Big[(\theta - t)(q(\mathsf{X})(\theta) - h(\mathsf{X})(\theta))\Big]|\mathsf{X}\Big] \\ &= \mathbb{E}\Big[\mathbb{E}\Big[(\theta - t)|\mathsf{X}\Big](q(\mathsf{X})(\theta) - h(\mathsf{X})(\theta))\Big] \end{aligned}$$

by the tower rule for expectations, and since q is the true quantile function,

$$\mathbb{E}\Big[t|\mathsf{X}\Big] = \mathbb{E}\Big[\mathbb{1}_{\{\mathsf{Y} \leq q(\mathsf{X})(\theta)\}}|\mathsf{X}\Big] = \mathbb{P}\Big[\mathsf{Y} \leq q(\mathsf{X})(\theta)|\mathsf{X}\Big] = \theta,$$

so

$$\mathbb{E}\Big[(\theta - t)(q(\mathsf{X})(\theta) - h(\mathsf{X})(\theta))\Big] = 0.$$

Moreover, $(t-s)$ is negative when $q(x)(\theta) \leq y \leq h(x)(\theta)$, positive when $h(x)(\theta) \leq y \leq q(x)(\theta)$ and 0 otherwise, thus the quantity $(t-s)(y - h(x)(\theta))$ is always positive. As a consequence,

$$\mathcal{R}(h) - \mathcal{R}(q) = \int_0^1 \mathbb{E}\Big[\ell(\theta, h(\mathsf{X})(\theta), \mathsf{Y}) - \ell(\theta, q(\mathsf{X})(\theta), \mathsf{Y})\Big] \mathrm{d}\mu(\theta) \geq 0$$

which concludes the proof. ■

Proposition 5.25 allows us to derive conditions under which the minimization of the risk above yields the true quantile function. Under the assumption that (i) q is continuous

(as seen as a function of two variables), (ii) $\mathrm{supp}(\mu) = [0,1]$, then the minimization of the integrated pinball loss performed in the space of continuous functions yields the true quantile function on the support of $\mathbb{P}_{\mathsf{X},\mathsf{Y}}$.

The ITL framework proposes to learn this quantile function in a vv-RKHS $\mathcal{H}_K$ by solving a regularized empirical risk problem:

$$\hat{h} = \operatorname*{arg\,min}_{h \in \mathcal{H}_K} \frac{1}{n} \sum_{i=1}^{n} I_\ell(h(x_i), y_i) + \frac{\lambda}{2} \|h\|^2_{\mathcal{H}_K} . \tag{5.24}$$

This problem can then be solved using the optimization algorithm developed in Chapter 3, in particular by sampling the integral loss and benefiting from a *double representer theorem* (see Section 3.3.1) or by using dual algorithms (see Section 3.4.3). In the following, we further explore this problem: in Section 5.4.1 we study ways to enforce shape constraints for the estimate by means of a *soft penalty* added to the objective function in Problem 5.24. Later in Section 5.4.2 we propose to smooth the pinball loss, to the benefit of enjoying fast solvers exploiting the differentiability of the loss, then study in Section 5.4.3 the impact of the number of sampled locations m used to approximate the integral loss. Finally in Section 5.4.4 we apply the ITL framework to a quantile estimation problem where the input data are images, and propose to use a neural network to learn the input kernel.

5.4.1 Enforcing Shape Constraints

By definition, the quantile function is nondecreasing with respect to the parameter θ. It is desirable that an estimator $\hat{h}$ output by the ITL scheme possesses this property, so that it can be interpreted as a quantile function. Visually, this means that given two quantile levels $(\theta_1, \theta_2) \in \Theta^2$ the curves associated to $\hat{h}(\cdot)(\theta_1)$ and $\hat{h}(\cdot)(\theta_2)$ must not cross. This phenomenon, denoted as *crossing quantiles*, is a challenge to avoid in practice. While previous approaches in multi-output learning try to control the finite differences $\hat{h}(x)(\theta_{j+1}) - \hat{h}(x)(\theta_j)$ (Sangnier et al., 2016), having a functional model allows to consider the partial derivative of the estimator with respect to θ, which should be nonnegative to respect the nondecreasibility condition. The smoothness of the model can be ensured by the smoothness of the output kernel: following Ziemer (2012), whenever $k_\Theta \in \mathcal{C}^2(\Theta^2, \mathbb{R})$ then any $f \in \mathcal{H}_{k_\Theta} \in \mathcal{C}^1(\Theta, \mathbb{R})$. Denoting by $\partial_\Theta h$ the partial derivative operator in $\mathcal{H}_K$ with respect to θ, one would ideally want to solve

$$\begin{aligned} \hat{h} = \operatorname*{arg\,min}_{h \in \mathcal{H}_K} \; & \frac{1}{n} \sum_{i=1}^{n} I_\ell(h(x_i), y_i) + \frac{\lambda}{2} \|h\|^2_{\mathcal{H}_K} \\ & s.t.\ (\partial_\Theta h)(x)(\theta) \geq 0 \quad \forall (x, \theta) \in \mathcal{X} \times \Theta. \end{aligned} \tag{5.25}$$

Remark 5.26. *There is a slight notation overloading with the subdifferential operator ∂ introduced in Section 2.1. In this chapter we are dealing with differentiable functions and identify $\partial f(\theta) = \{f'(\theta)\}$ with $f'(\theta)$'?. For a kernel $k_\Theta(\theta_1, \theta_2)$ of two variables, assumed to be $\mathcal{C}^2$, we denote $\partial_1 k_\Theta$ the partial derivative of k with respect to θ_1 and $\partial_2 k_\Theta$ the partial derivative of k with respect to θ_2. From Zhou (2008), for all $\theta \in \Theta$, $(\partial_2 k_\Theta)(\cdot, \theta) \in \mathcal{H}_{k_\Theta}$ and the reproducing property in $\mathcal{H}_{k_\Theta}$ reads as*

$$(\partial f)(\theta) = \Big\langle f, (\partial_2 k_\Theta)(\cdot, \theta) \Big\rangle_{\mathcal{H}_{k_\Theta}}, \quad (f, \theta) \in \mathcal{H}_{k_\Theta} \times \Theta. \tag{5.26}$$

This translates in the vv-RKHS $\mathcal{H}_K$ to

$$(\partial_\Theta h)(x)(\theta) = \left\langle h, K_x\left((\partial_2 k_\Theta)(\cdot,\theta)\right)\right\rangle_{\mathcal{H}_K}, \quad (h,x,\theta) \in \mathcal{H}_K \times \mathcal{X} \times \Theta. \tag{5.27}$$

This type of constraint belongs to the family of *hard constraints* and in our case the functional constraint prevents a tractable optimization scheme. We can point to Aubin-Frankowski and Szabó 2020 for a recent work on enforcing hard shape constraints in kernel based learning. To mitigate this bottleneck, we propose to penalize the objective function in Problem 5.24 with a *soft constraint* based on the derivative of the model:

$$\Omega_{\mathrm{nc}}(h) := \lambda_{\mathrm{nc}} \int_{\mathcal{X}} \int_{\Theta} \left| -(\partial_\Theta h)(x)(\theta) \right|_+ \mathrm{d}\mathbb{P}_{\mathsf{X}}(x) \mathrm{d}\mu(\theta).$$

Given that such a penalty is not computable analytically, we approximate it and define

$$\widetilde{\Omega}_{\mathrm{nc}}(h) := \lambda_{\mathrm{nc}} \frac{1}{n_{\mathrm{nc}} m_{\mathrm{nc}}} \sum_{i=1}^{n_{\mathrm{nc}}} \sum_{j=1}^{m_{\mathrm{nc}}} \left| -(\partial_\Theta h)(\tilde{x}_i)(\tilde{\theta}_j) \right|_+ .$$

where $(\tilde{x}_i)_{i=1}^{n_{\mathrm{nc}}}$ and $(\tilde{\theta}_j)_{j=1}^{m_{\mathrm{nc}}}$ form a grid at which the penalty is computed. They can be chosen different from the real data $(x_i)_{i=1}^n$ and sampled locations $(\theta_j)_{j=1}^m$ used in the double representer theorem. We then arrive at the following optimization problem

$$\hat{h} = \operatorname*{arg\,min}_{h \in \mathcal{H}_K} \frac{1}{n} \sum_{i=1}^{n} I_\ell(h(x_i), y_i) + \frac{\lambda}{2} \|h\|^2_{\mathcal{H}_K} + \widetilde{\Omega}_{\mathrm{nc}}(h). \tag{5.28}$$

Is is to be noticed that the resulting estimator does not necessarily prevents crossing quantile, hence the term *soft constraints*. The double representer expression from Theorem 3.9 can then be adapted to encompass this type of penalty, as emphasized in the theorem below.

Theorem 5.27. *Problem 5.28 admits a unique solution $\hat{h} \in \mathcal{H}_K$, and there exist $(\hat{\alpha}_{ij})_{i,j=1}^{n,m} \in \mathbb{R}^{nm}$ and $(\hat{\beta}_{ij})_{i,j=1}^{n_{\mathrm{nc}},m_{\mathrm{nc}}} \in \mathbb{R}^{n_{\mathrm{nc}} m_{\mathrm{nc}}}$ such that for all $(x,\theta) \in \mathcal{X} \times \Theta$,*

$$\hat{h}(x)(\theta) = \sum_{i,j=1}^{n,m} \hat{\alpha}_{ij} k_{\mathcal{X}}(x, x_i) k_\Theta(\theta, \theta_j) + \sum_{i,j=1}^{n_{\mathrm{nc}},m_{\mathrm{nc}}} \hat{\beta}_{ij} k_{\mathcal{X}}(x, \tilde{x}_i) \partial_2 k_\Theta(\theta, \tilde{\theta}_j). \tag{5.29}$$

Proof The proof is similar to the one of Theorem 3.9. Define

$$E_1 = \left\{ \left(K_{x_i} k_\Theta(\cdot, \theta_j) \right)_{i,j=1}^{n,m} : (i,j) \in [n] \times [m] \right\},$$

$$E_2 = \left\{ \left(K_{\tilde{x}_i} \left(\partial_2 k_\Theta(\cdot, \tilde{\theta}_j) \right) \right)_{i,j=1}^{n_{\mathrm{nc}},m_{\mathrm{nc}}} : (i,j) \in [n_{\mathrm{nc}}] \times [m_{\mathrm{nc}}] \right\}$$

and

$$E = \mathrm{Span}\left\{ E_1 \cup E_2 \right\} \subset \mathcal{H}_K.$$

Let $\mathcal{J}(h) = \frac{1}{n} \sum_{i=1}^n \widetilde{I}_\ell(h(x_i), y_i) + \frac{\lambda}{2} \|h\|^2_{\mathcal{H}_K} + \widetilde{\Omega}(h)$. $\mathcal{J}$ is coercive and bounded below, so that there exists a minimizer $\hat{h} \in \mathcal{H}$ which is unique by strong convexity of $\mathcal{J}$. Since E is a finite-dimensional subspace of $\mathcal{H}_K$, it is closed in $\mathcal{H}_K$, and $E \oplus E^\perp = \mathcal{H}_K$.

By decomposing $\hat{h} = \hat{h}_E + \hat{h}_{E^\perp}$ and applying the reproducing property, we get that $\mathcal{J}(\hat{h}_E + \hat{h}_{E^\perp}) = \mathcal{J}(\hat{h}_E) + \frac{\lambda}{2}\left\|\hat{h}_{E^\perp}\right\|^2_{\mathcal{H}_K}$, which in turns implies that $\hat{h}_{E^\perp} = 0$ and validates the decomposition in Equation (5.29). ■

The benefit of Theorem 5.27 is to provide a workable finite-dimensional parameterization of $\hat{h}$. The size of the representation is $nm + n_{\text{nc}}m_{\text{nc}}$, in particular if we use the data points $(x_i)_{i=1}^n$ and sampled locations $(\theta_j)_{j=1}^m$ in the computation of $\widetilde{\Omega}$, then the size of the representation is $2nm$. The resulting finite dimensional problem can then be tackled by first order optimization methods such as subgradient descent.

The efficiency of the non-crossing penalty is illustrated in Figure 5.2 on a synthetic dataset from Sangnier et al. (2016). This dataset consists in a sine curve at $1Hz$ modulated by a sine envelope at $1/3Hz$ and mean 1, distorted with a Gaussian noise of mean 0 and a linearly decreasing standard deviation from 1.2 at $x = 0$ to 0.2 at $x = 1.5$. Here $n = 40$ data samples and $m = 20$ sampling locations have been generated, and we used the same anchors for the non-crossing penalty. Many crossings are visible on the right plot, while they are almost not noticeable on the left plot, using the non-crossing penalty.

Concerning our real-world examples, to study the efficiency of the proposed scheme in quantile regression the following experimental protocol was applied. We worked with 20 UCI datasets, each dataset was splitted randomly into a training set (70%) and a test set (30%). We optimized the hyperparameters by minimizing a 5-folds cross validation with a Bayesian optimizer[2] Once the hyperparameters were obtained, a new regressor was learned on the whole training set using the optimized hyperparameters. We report the value of the pinball loss and the crossing loss on the test set for three methods: our technique is called infinite quantile regression (IQR), we refer to Sangnier et al. (2016)'s approach as joint quantile regression (JQR), and independent learning (abbreviated as IND-QR) represents a further baseline. We repeated 20 simulations (different random training-test splits); the results are also compared using a Mann-Whitney-Wilcoxon test. A summary is provided in Table 5.1.

Notice that while JQR is tailored to predict finite many quantiles, our IQR method estimates the *whole quantile function* hence solves a more challenging task. Despite the more difficult problem solved, as Table 5.1 suggest the performance in terms of pinball loss of IQR is comparable to that of the state-of-the-art JQR on all the twenty studied benchmarks, except for the 'crabs' and 'cpus' datasets (pval $< 0.25\%$). In addition, when considering the non-crossing penaltycone can observe that IQR outperforms the IND-QR baseline on eleven datasets (pval $< 0.25\%$) and JQR on two datasets. This illustrates the efficiency of the constraint based on the continuum scheme.

5.4.2 Smoothing the Loss Function

There are several ways to solve the non-smooth optimization problems associated to the QR task. One could proceed for example by duality—as presented in Section 3.4.3—, or apply sub-gradient descent techniques (which often converge quite slowly). In our experiments we used the L-BFGS-B (Zhu et al., 1997) optimization scheme which is widely popular in machine learning, with non-smooth extensions (Skajaa, 2010; Keskar and Wächter, 2017). The technique requires only evaluation of objective function along

[2]We used a Gaussian process model and minimized the expected improvement. The optimizer was initialized using 27 samples from a Sobol sequence and ran for 50 iterations.

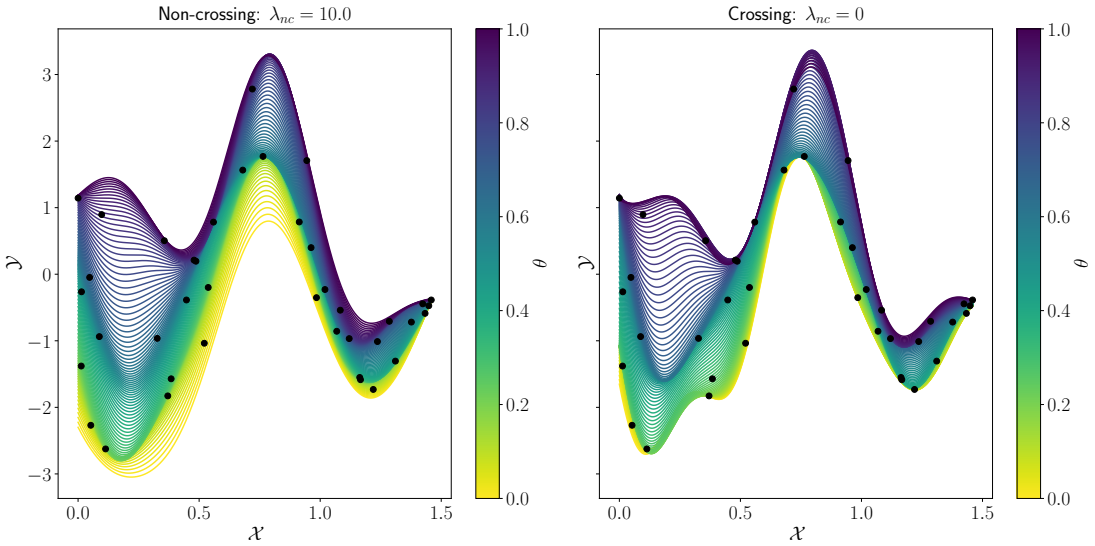


Figure 5.2 – Impact of crossing penalty on toy data. Left plot: strong non-crossing penalty ($\lambda_{\text{nc}} = 10$). Right plot: no non-crossing penalty ($\lambda_{\text{nc}} = 0$). The plots show 100 quantiles of the continuum learned, linearly spaced between 0 (blue) and 1 (red). Notice that the non-crossing penalty does not provide crossings to occur in the regions where there is no points to enforce the penalty (*e.g.* $x \in [0.13, 0.35]$). This phenomenon is alleviated by the regularity of the model.

Table 5.1 – Quantile regression on 20 UCI datasets. Reported: 100×value of the pinball loss, 100×crossing loss (smaller is better). pval: outcome of the Mann-Whitney-Wilcoxon test of JQR vs IQR and Independent vs IQR. Boldface: significant values *w.r.t.* IQR.

DATASET	JQR				IND-QR				IQR	
	(PINBALL	PVAL)	(CROSS	PVAL)	(PINBALL	PVAL)	(CROSS	PVAL)	PINBALL	CROSS
CobarOre	159 ± 24	$9 \cdot 10^{-01}$	0.1 ± 0.4	$6 \cdot 10^{-01}$	150 ± 21	$2 \cdot 10^{-01}$	0.3 ± 0.8	$7 \cdot 10^{-01}$	165 ± 36	2.0 ± 6.0
engel	175 ± 555	$6 \cdot 10^{-01}$	0.0 ± 0.2	$1 \cdot 10^{+00}$	63 ± 53	$8 \cdot 10^{-01}$	4.0 ± 12.8	$8 \cdot 10^{-01}$	47 ± 6	0.0 ± 0.1
BostonHousing	49 ± 4	$8 \cdot 10^{-01}$	0.7 ± 0.7	$2 \cdot 10^{-01}$	49 ± 4	$8 \cdot 10^{-01}$	$\mathbf{1.3 \pm 1.2}$	$1 \cdot 10^{-05}$	49 ± 4	0.3 ± 0.5
caution	88 ± 17	$6 \cdot 10^{-01}$	0.1 ± 0.2	$6 \cdot 10^{-01}$	89 ± 19	$4 \cdot 10^{-01}$	$\mathbf{0.3 \pm 0.4}$	$2 \cdot 10^{-04}$	85 ± 16	0.0 ± 0.1
ftcollinssnow	154 ± 16	$8 \cdot 10^{-01}$	0.0 ± 0.0	$6 \cdot 10^{-01}$	155 ± 13	$9 \cdot 10^{-01}$	0.2 ± 0.9	$8 \cdot 10^{-01}$	156 ± 17	0.1 ± 0.6
highway	103 ± 19	$4 \cdot 10^{-01}$	0.8 ± 1.4	$2 \cdot 10^{-02}$	99 ± 20	$9 \cdot 10^{-01}$	$\mathbf{6.2 \pm 4.1}$	$1 \cdot 10^{-07}$	105 ± 36	0.1 ± 0.4
heights	127 ± 3	$1 \cdot 10^{+00}$	0.0 ± 0.0	$1 \cdot 10^{+00}$	127 ± 3	$9 \cdot 10^{-01}$	0.0 ± 0.0	$1 \cdot 10^{+00}$	127 ± 3	0.0 ± 0.0
sniffer	43 ± 6	$8 \cdot 10^{-01}$	0.1 ± 0.3	$2 \cdot 10^{-01}$	44 ± 5	$7 \cdot 10^{-01}$	$\mathbf{1.4 \pm 1.2}$	$6 \cdot 10^{-07}$	44 ± 7	0.1 ± 0.1
snowgeese	55 ± 20	$7 \cdot 10^{-01}$	0.3 ± 0.8	$3 \cdot 10^{-01}$	53 ± 18	$6 \cdot 10^{-01}$	0.4 ± 1.0	$5 \cdot 10^{-02}$	57 ± 20	0.2 ± 0.6
ufc	81 ± 5	$6 \cdot 10^{-01}$	$\mathbf{0.0 \pm 0.0}$	$4 \cdot 10^{-04}$	82 ± 5	$7 \cdot 10^{-01}$	$\mathbf{1.0 \pm 1.4}$	$2 \cdot 10^{-04}$	82 ± 4	0.1 ± 0.3
BigMac2003	80 ± 21	$7 \cdot 10^{-01}$	$\mathbf{1.4 \pm 2.1}$	$4 \cdot 10^{-04}$	74 ± 24	$9 \cdot 10^{-02}$	$\mathbf{0.9 \pm 1.1}$	$7 \cdot 10^{-05}$	84 ± 24	0.2 ± 0.4
UN3	98 ± 9	$8 \cdot 10^{-01}$	0.0 ± 0.0	$1 \cdot 10^{-01}$	99 ± 9	$1 \cdot 10^{+00}$	$\mathbf{1.2 \pm 1.0}$	$1 \cdot 10^{-05}$	99 ± 10	0.1 ± 0.4
birthwt	141 ± 13	$1 \cdot 10^{+00}$	0.0 ± 0.0	$6 \cdot 10^{-01}$	140 ± 12	$9 \cdot 10^{-01}$	0.1 ± 0.2	$7 \cdot 10^{-02}$	141 ± 12	0.0 ± 0.0
crabs	$\mathbf{11 \pm 1}$	$4 \cdot 10^{-05}$	0.0 ± 0.0	$8 \cdot 10^{-01}$	$\mathbf{11 \pm 1}$	$2 \cdot 10^{-04}$	$\mathbf{0.0 \pm 0.0}$	$2 \cdot 10^{-05}$	13 ± 3	0.0 ± 0.0
GAGurine	61 ± 7	$4 \cdot 10^{-01}$	0.0 ± 0.1	$3 \cdot 10^{-03}$	62 ± 7	$5 \cdot 10^{-01}$	$\mathbf{0.1 \pm 0.2}$	$4 \cdot 10^{-04}$	62 ± 7	0.0 ± 0.0
geyser	105 ± 7	$9 \cdot 10^{-01}$	0.1 ± 0.3	$9 \cdot 10^{-01}$	105 ± 6	$9 \cdot 10^{-01}$	0.2 ± 0.3	$6 \cdot 10^{-01}$	104 ± 6	0.1 ± 0.2
gilgais	51 ± 6	$5 \cdot 10^{-01}$	0.1 ± 0.1	$1 \cdot 10^{-01}$	49 ± 6	$6 \cdot 10^{-01}$	$\mathbf{1.1 \pm 0.7}$	$2 \cdot 10^{-05}$	49 ± 7	0.3 ± 0.3
topo	69 ± 18	$1 \cdot 10^{+00}$	0.1 ± 0.5	$1 \cdot 10^{+00}$	71 ± 20	$1 \cdot 10^{+00}$	$\mathbf{1.7 \pm 1.4}$	$3 \cdot 10^{-07}$	70 ± 17	0.0 ± 0.0
mcycle	66 ± 9	$9 \cdot 10^{-01}$	0.2 ± 0.3	$7 \cdot 10^{-03}$	66 ± 8	$9 \cdot 10^{-01}$	$\mathbf{0.3 \pm 0.3}$	$7 \cdot 10^{-06}$	65 ± 9	0.0 ± 0.1
cpus	$\mathbf{7 \pm 4}$	$2 \cdot 10^{-04}$	$\mathbf{0.7 \pm 1.0}$	$5 \cdot 10^{-04}$	$\mathbf{7 \pm 5}$	$3 \cdot 10^{-04}$	$\mathbf{1.2 \pm 0.8}$	$6 \cdot 10^{-08}$	16 ± 10	0.0 ± 0.0

with its gradient, which can be computed automatically using reverse mode automatic differentiation (as in Abadi et al. (2016)). To benefit from the available fast smooth implementations (Jones et al., 2001; Fei et al., 2014), we smoothed the non-differentiable part of the loss function. To do that, remark that the pinball loss writes as

$$\ell(\theta, u, v) = \Big|\theta - \mathbb{1}_{\mathbb{R}_-}(v-u)\Big| \, |v-u|$$

and replace the absolute value by a smoothed approximation given by its Moreau enveloppe (Bauschke et al., 2011). The resulting κ-smoothed ($\kappa > 0$) absolute value is as follows:

$$\psi_1^\kappa(r) := \left(|\cdot| \,\square\, \frac{1}{2\kappa}|\cdot|^2\right)(r) = \begin{cases} \frac{1}{2\kappa} r^2 & \text{if } |r| \leq \kappa \\ |r| - \frac{\kappa}{2} & \text{otherwise,} \end{cases}$$

Minimizing the κ-smoothed pinball loss

$$\ell_\kappa(\theta, h(x), y) = \Big|\theta - \mathbb{1}_{\mathbb{R}_-}(y - h(x))\Big| \, \psi_1^\kappa(y - h(x))$$

yields the quantiles when $\kappa \to 0$, the expectiles as $\kappa \to +\infty$. The intermediate values are known as M-quantiles (Breckling and Chambers, 1988). The influence of the κ parameter is illustrated in Figure 5.3. For this experiment, 10000 samples have been generated from the sine wave dataset described in Section 5.4.1, and the model have been trained on 100 quantiles generated from a Gauss-Legendre quadrature. When κ is small the quantiles are recovered (the dashed lines on the right plot match the theoretical quantiles in plain lines). It took 225s (258 iteration, and 289 function evaluations) to train for $\kappa = 1 \cdot 10^1$, 1313s for $\kappa = 1 \cdot 10^{-1}$ (1438 iterations and 1571 function evaluations), 931s for $\kappa = 1e^{-3}$ (1169 iterations and 1271 function evaluations) and 879s for $\kappa = 0$ (1125 iterations and 1207 function evaluations). We used a GPU Tensorflow implementation and run the experiments in float64 on a computer equipped with a GTX 1070, and intel i7 7700 and 16GB of DRAM.

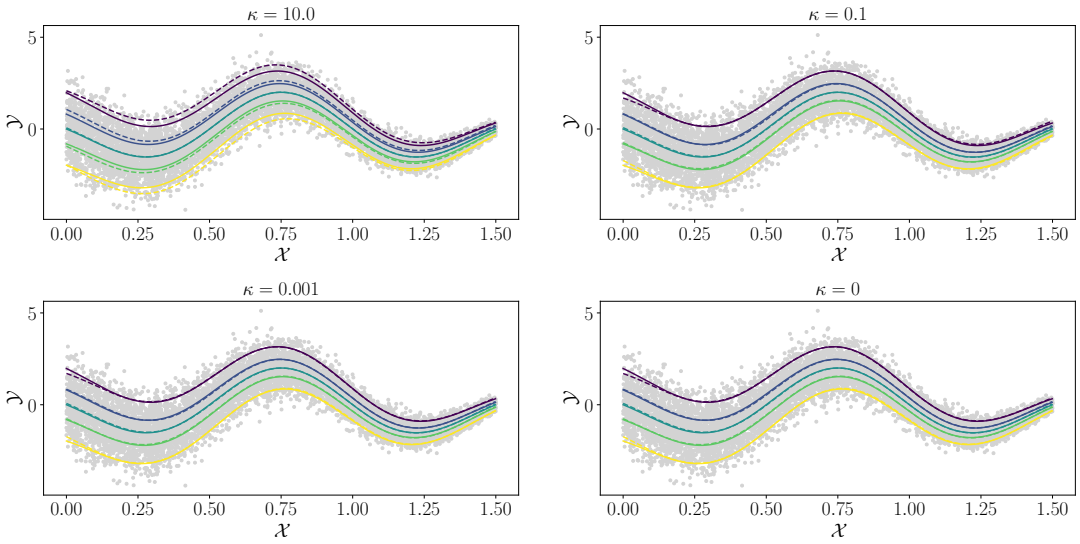


Figure 5.3 – Impact of the smoothing of the pinball loss for different values of κ.

5.4.3 Influence of the Number of Sampled Locations

In this section, we quickly present an additional experiment on the number of sampled locations m used to approximate the integral loss I_ℓ.

In the experiment presented in Figure 5.4, on the sine synthetic benchmark, we draw $n = 1000$ training points and study the impact of increasing m on the quality of the quantiles at $\theta \in \{0.05, 0.25, 0.5, 0.75, 0.95\}$. We notice that when $m \geq 34 \approx \sqrt{1000}$ there is little benefit to draw more m samples: the quantile curves do not change on the $n_{\text{test}} = 2000$ test points.

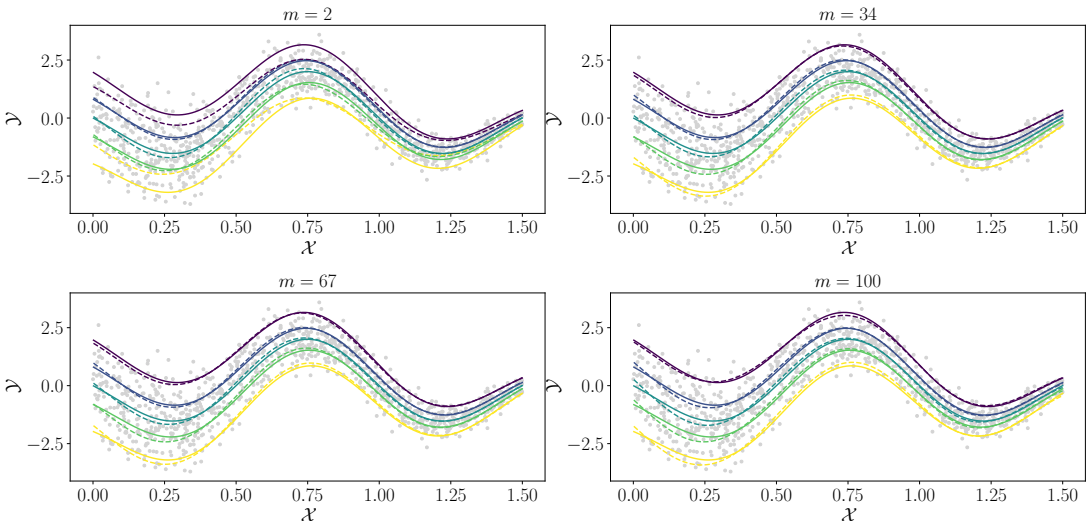


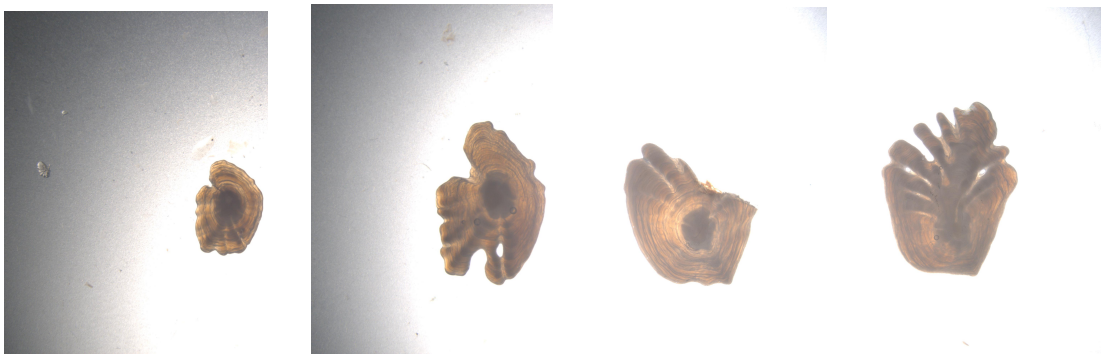
Figure 5.4 – Impact of the number of hyperparameters sampled.

5.4.4 Deep Kernel Learning for Quantile Regression on Images

In this section, we apply the IQR method in the problem of estimating the age of fishes (Y) from images (X). This task is critical in sustainable management of fish resources with the goal of avoiding overfishing. The age of a fish can be inferred from its *otholiths*, which are calcium carbonate structures growing behind their brains. As the fish gets older, the otholith becomes larger and more convoluted, as illustrated in Figure 5.5. The estimation problem is rather challenging, and typically involves experts who carefully examine the growth zones of the otholiths. This process is costly, especially when we realize that the number of otholiths analysed yearly around the world scales in millions (Campana and Thorrold, 2001). Automating the task is of fundamental interest, and has been addressed recently using *convolutional neural networks* (CNNs) in Moen et al. (2018), yielding satisfactory predictions and interpretable results (Ordonez et al., 2020).

Beyond the single real-valued prediction of the age of the fish, we propose to learn the conditional quantile function of Y (age of the fish) given X (photography of its otholith). In the ITL framework, this requires to define a kernel on the input space, which can be difficult. As CNN are known to learn relevant features stored in their last layer (Goodfellow et al., 2016), a first idea would be to use a pretrained CNN to extract features for X, and then apply the ITL framework with an input kernel on the resulting feature space, denoted $\mathcal{V}$ in what follows. This amounts to using a kernel

$$k_{\mathcal{X}}(x, y) = k_{\mathcal{V}}(\phi_\omega(x), \phi_\omega(y)), \tag{5.30}$$



(a) Age 3 (b) Age 7 (c) Age 10 (d) Age 13

Figure 5.5 – Fish bone images of otholiths at different ages. Dataset from the Norwegian Marine Data Center.

where $k_{\mathcal{V}}$ is a kernel on the feature space, and $\phi_\omega \colon \mathcal{X} \to \mathcal{V}$ is the neural architecture used for extracting features parameterized by its weights ω. In a deep kernel learning spirit, we propose here to learn jointly the parameters of the neural architecture and that of the estimator. To do so, we consider an input kernel of the form Equation (5.30), where ϕ_ω is an Inception v3 architecture (Szegedy et al., 2016), and $k_{\mathcal{V}}$ is a Gaussian random Fourier feature (RFF) kernel. We also choose k_Θ to be a Gaussian RFF kernel, which allows to exploit Algorithm 3.2 for the optimization. To train jointly on both parameters of the kernel and the estimator, at each epoch we make a gradient step on the kernel parameters, starting from pre-trained weights used in Ordonez et al. (2020) for predicting fish ages. Many tricks are needed to make the training smooth, such as rescaling images, feature augmentation, and so on; we refer to Ordonez et al. (2020) for details. While visualization is a bit hard for this problem, we illustrate in Figure 5.6 a few quantile functions associated to random samples from the dataset. We can see that the quantile functions are nondecreasing with respect to θ, which is at in agreement with theoretical properties of quantiles.

5.5 Cost-Sensitive Classification

In this section we present some numerical illustration to the CSC application of the ITL scheme. The optimization is performed using a L-BFGS solver on the finite dimensional problem provided by the double representer expression from Theorem 3.9, associated to a smoothing of the imbalanced Hinge loss obtained by its Moreau enveloppe, similarly to the quantile regression case developed in Section 5.4.2.

We used the Iris UCI dataset with 4 attributes and 150 samples, the two synthetic SCIKIT-LEARN (Pedregosa et al., 2011) datasets TWO-MOONS (noise=0.4) and CIRCLES (noise=0.1) with both 2 attributes and 1000 samples. As detailed in section 5.2.3, CSC on a continuum $\Theta = [-1, 1]$ that we call *infinite cost-sensitive classification* (ICSC) can be tackled by our proposed technique. In this case, the hyperparameter θ controls the tradeoff between the importance of the correct classification with labels -1 and $+1$. When $\theta = -1$, class -1 is emphasized; the probability of correctly classified instances with this label (called specificity) is desired to be 1. Similarly, for $\theta = +1$, the probability of correct classification of samples with label $+1$ (called sensitivity) is ideally 1.

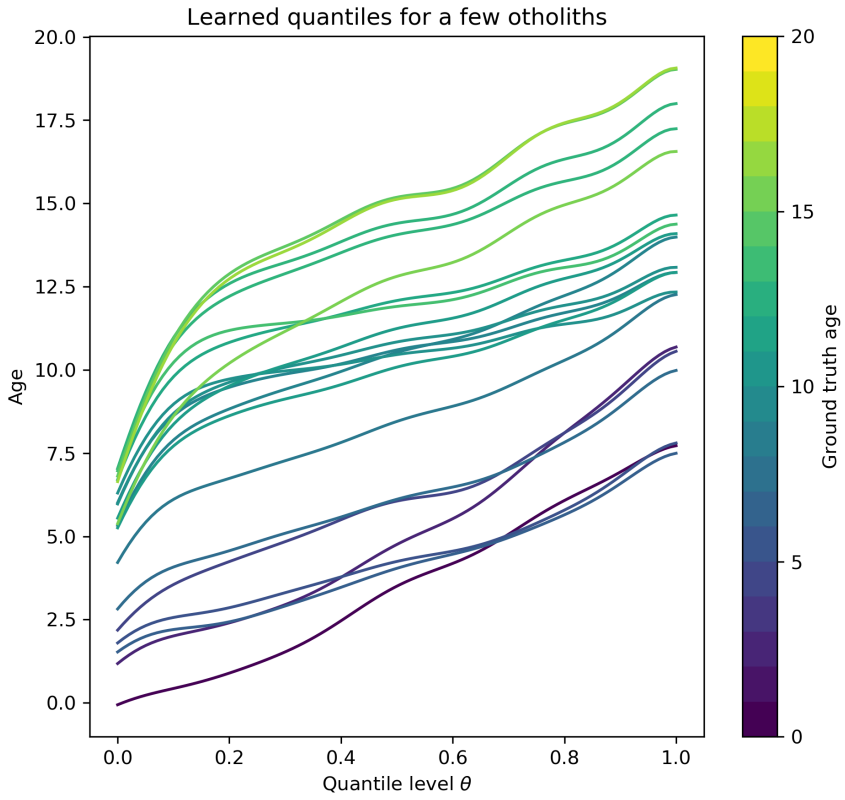


Figure 5.6 – Estimated quantiles of a few otholiths samples chosen randomly in the dataset. The colorbar on the right indicates the ground truth age of the fish, and on the left is plotted its estimated quantile as a function of θ. We can see that the quantiles are nondecreasing thanks to the non crossing penalty.

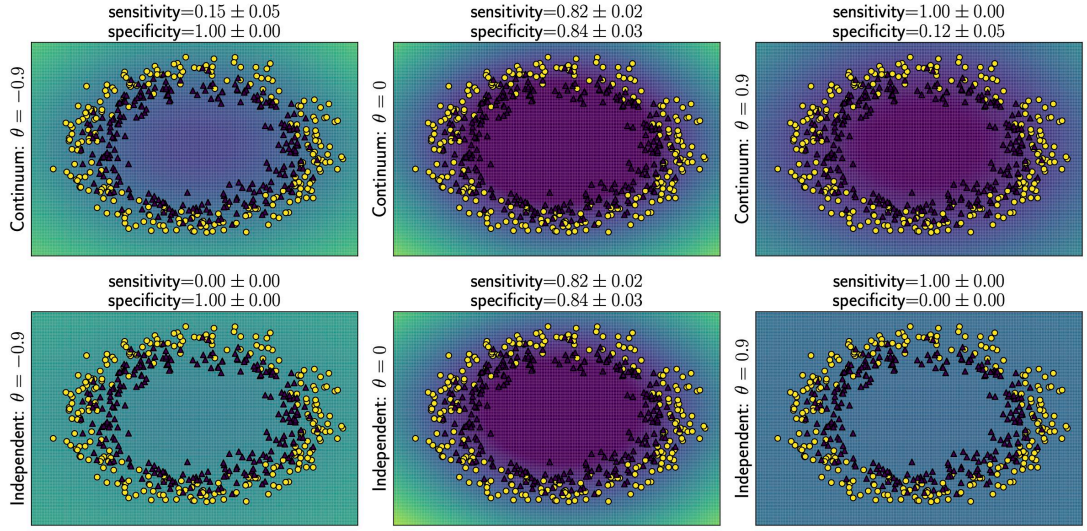
We chose $k_{\mathcal{X}}$ to be a Gaussian kernel with bandwidth $\sigma_{\mathcal{X}} = (2\gamma_{\mathcal{X}})^{(-1/2)}$ the median of the Euclidean pairwise distances of the input points (Jaakkola et al., 1999). The output kernel k_Θ is also a Gaussian kernel with bandwidth $\gamma_\Theta = 5$. We used $m = 20$ for all datasets in the double representer theorem associated to the sobol sequence. As a baseline we trained independently 3 CSC classifiers with $\theta \in \{-0.9, 0, 0.9\}$. We repeated 50 times a random $50 - 50\%$ train-test split of the dataset and report the average test error and standard deviation (in terms of sensitivity and specificity).

Our results are illustrated in Table 5.2. For $\theta = -0.9$, both independent and joint learners give the desired 100% specificity; the joint CSC scheme however has significantly higher sensitivity value (15% vs 0%) on the dataset Circles. Similar conclusion holds for the $\theta = +0.9$ extreme: the ideal sensitivity is reached by both techniques, but the joint learning scheme performs better in terms of specificity (0% vs 12%) on the dataset Circles.

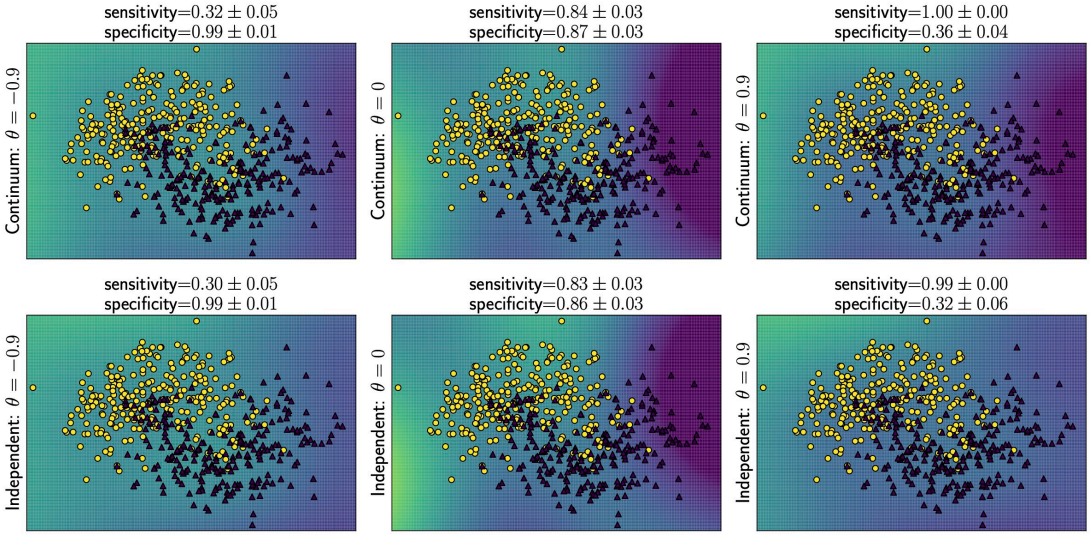
The results from Table 5.2 are highlighted in Figure 5.7 and Figure 5.8 where we present a vizualization of the independent vs continuum learning problems.

Table 5.2 – ICSC vs Independent (IND)-CSC. Higher is better.

Dataset	Method	$\theta = -0.9$		$\theta = 0$		$\theta = +0.9$	
		SENSITIVITY	SPECIFICITY	SENSITIVITY	SPECIFICITY	SENSITIVITY	SPECIFICITY
TWO-MOONS	IND	0.3 ± 0.05	0.99 ± 0.01	0.83 ± 0.03	0.86 ± 0.03	0.99 ± 0	0.32 ± 0.06
	ICSC	0.32 ± 0.05	0.99 ± 0.01	0.84 ± 0.03	0.87 ± 0.03	1 ± 0	0.36 ± 0.04
CIRCLES	IND	0 ± 0	1 ± 0	0.82 ± 0.02	0.84 ± 0.03	1 ± 0	0 ± 0
	ICSC	0.15 ± 0.05	1 ± 0	0.82 ± 0.02	0.84 ± 0.03	1 ± 0	0.12 ± 0.05
IRIS	IND	0.88 ± 0.08	0.94 ± 0.06	0.94 ± 0.05	0.92 ± 0.06	0.97 ± 0.05	0.87 ± 0.06
	ICSC	0.89 ± 0.08	0.94 ± 0.05	0.94 ± 0.06	0.92 ± 0.05	0.97 ± 0.04	0.90 ± 0.05



(a) CIRCLE dataset



(b) TWO-MOONS dataset

Figure 5.7 – Illustration of ICSC on toy datasets used in Table 5.2.

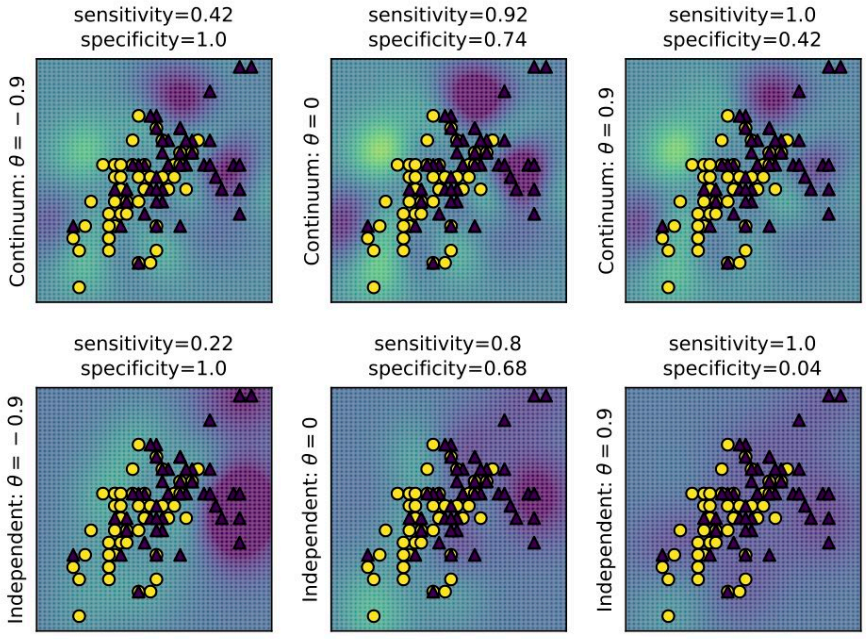


Figure 5.8 – Illustration of ICSC on the IRIS dataset used in Table 5.2.

5.6 Density Level Set Estimation

In this section, we adapt the ITL framework to an unsupervised learning problem, the DLSE carried out by means of OCSVM. As a reminder, given some data $(x_i)_{i=1}^n$ following an unknown law $\mathbb{P}_\mathcal{X}$ and a hyperparameter value $\theta \in (0, 1]$, the classical OCSVM in a scalar RKHS $\mathcal{H}_{k_\mathcal{X}}$ can be formulated as

$$\left(\hat{h}, \hat{t}\right) = \underset{h \in \mathcal{H}_{k_\mathcal{X}}, t \in \mathbb{R}}{\arg\min} \frac{1}{n\theta} \sum_{i=1}^{n} \max\left(0, t - h(x_i)\right) - t + \frac{1}{2} \left\|h\right\|^2_{\mathcal{H}_{k_\mathcal{X}}}. \tag{5.31}$$

Given the resulting estimator $\left(\hat{h}, \hat{t}\right) \in \mathcal{H}_{k_\mathcal{X}} \times \mathbb{R}$, the rule

$$d(x) := \operatorname{sign}\left(\hat{h}(x) - \hat{t}\right)$$

is then a reasonable choice for deciding whether a new point x is an inlier ($d(x) = 1$) or an outlier ($d(x) = -1$). In particular, it is known (Schölkopf et al., 2001b) that if the distribution $\mathbb{P}_\mathcal{X}$ does not contain discrete components, and the kernel $k_\mathcal{X}$ is analytic and non-constant, then this approach is theoretically grounded as the decision function d will separate the input space $\mathcal{X}$ in two sets (inliers and outliers) with respective mass $1 - \theta$ and θ, a property refered to as *the θ-property*.

Our objective is to generalize this algorithm to learning an OCSVM jointly for the continuum $\theta \in \Theta := (0, 1]$. To that end, we use a function-valued model $h \in \mathcal{H}_K$ where $K = k_\mathcal{X} \operatorname{Id}_{\mathcal{H}_{k_\Theta}}$, and model t as a function belonging to an RKHS $\mathcal{H}_{k_b}$ where $k_b : \Theta \times \Theta \to \mathbb{R}$ is a scalar-valued kernel on Θ. We want each $\left(\hat{h}(\cdot)(\theta), \hat{t}(\theta)\right)$ to be (close to) a solution of Problem 5.31. For this reason we introduce the L^2-RKHS mixed regularizer

$$\Omega(h) = \frac{1}{2} \int_\Theta \left\|h(\cdot)(\theta)\right\|^2_{\mathcal{H}_{k_\mathcal{X}}} \mathrm{d}\mu(\theta),$$

and consider the problem

$$\left(\hat{h}, \hat{t}\right) = \underset{h \in \mathcal{H}_K, t \in \mathcal{H}_{k_b}}{\arg\min} \frac{1}{n} \sum_{i=1}^{n} I_\ell(h(x_i), t) + \Omega(h) + \frac{\lambda}{2} \|t\|^2_{\mathcal{H}_{k_b}}, \tag{5.32}$$

where $\lambda > 0$ is a regularization parameter, and I_ℓ is the integral loss associated to

$$\ell(\theta, u, v) = \frac{1}{\theta} \max(0, v - u) - v$$

and to a probability measure μ. The decision function is then able to predict the outlierness of a new data point x for all $\theta \in (0, 1]$ by

$$d(x)(\theta) = \operatorname{sign}\left(\hat{h}(x)(\theta) - \hat{t}(\theta)\right).$$

Remark 5.28. *The regularizer* $\Omega(h)$ *is* uniformly weaker *than a vv-RKHS norm based regularizer* $\frac{1}{2}\|h\|^2_{\mathcal{H}_K}$*:*

$$\forall \theta \in \Theta, \ \left\|h(\cdot)(\theta)\right\|^2_{\mathcal{H}_{k_\mathcal{X}}} = \left\langle h(\cdot)(\theta), h(\cdot)(\theta)\right\rangle_{\mathcal{H}_{k_\mathcal{X}}} \leq k_\Theta(\theta, \theta)\langle h, h\rangle_{\mathcal{H}_K}$$

so that after integration

$$\Omega(h) \leq \frac{1}{2}\left(\int_\Theta k(\theta, \theta) \mathrm{d}\mu(\theta)\right) \|h\|^2_{\mathcal{H}_K}.$$

This prevents the direct use of a representer theorem, as the regularizer does not write as a nondecreasing function of $\|h\|_{\mathcal{H}_K}$*.*

Remark 5.29. *Due to the non-integrability of* $\frac{1}{\theta}$ *at a neighborhood of* 0*, the integral in* I_ℓ *is not assured to converge to a finite value. To alleviate this bottleneck, one can choose the measure* μ *appropriately and consider either a measure with support* $[\epsilon, 1]$ *for some small* $\epsilon > 0$ *or a measure with full support that ensures proper convergence of* I_ℓ *(*e.g. *the measure of density* $\theta \mapsto 2\theta$*).*

We now devote the following part to solving Problem 5.32, that we refer to as *infinite one-class SVM* (IOCSVM). In Section 5.6.1, we propose a representer theorem for a sampled version of Problem 5.32 that is amenable to optimization and emphasize the benefit of this approach with numerical experiments in Section 5.6.2.

5.6.1 Representer Theorem for Mixed Regularization

One difficulty of working with Problem 5.32 is that the classical representer theorem from Theorem 2.43 does not apply. To get a finite parametrization of the estimator, we will solve a sampled version of the problem, and prove a *double representer theorem* for $\hat{h}$. Given a sampling scheme $(\eta_j, \theta_j)_{j=1}^m \in (\mathbb{R} \times \Theta)^m$ using for instance quasi Monte-Carlo or quadratures (see Section 3.3.1), we introduce the notation

$$\widetilde{I}_\ell(h(x), t) := \sum_{j=1}^{m} \eta_j \ell(\theta_j, h(x_i)(\theta_j), t(\theta_j)), \qquad \widetilde{\Omega}(h) := \frac{1}{2}\sum_{j=1}^{m} \eta_j \left\|h(\cdot)(\theta_j)\right\|^2_{\mathcal{H}_{k_\mathcal{X}}} \tag{5.33}$$

and propose to solve

$$\left(\hat{h}, \hat{t}\right) = \underset{h \in \mathcal{H}_K, t \in \mathcal{H}_{k_b}}{\arg\min} \frac{1}{n} \sum_{i=1}^{n} \widetilde{I}_\ell(h(x_i), t) + \widetilde{\Omega}(h) + \frac{\lambda}{2} \|t\|^2_{\mathcal{H}_{k_b}}. \tag{5.34}$$

It turns out that the estimator resulting from Problem 5.34 enjoy a representer expression. To prove it, we need the following two lemmas. They make use of the equivalence of views between $\mathcal{H}_K$ and $\mathcal{H}_{k_\mathcal{X}} \otimes \mathcal{H}_{k_\Theta}$ (see Remark 2.41).

Lemma 5.30. *Let $k_\mathcal{X}$ and k_Θ be two scalar-valued kernels on $\mathcal{X}$ and Θ, and $K : (\theta, \theta') \mapsto k_\Theta(\theta, \theta') \operatorname{Id}_{\mathcal{H}_{k_\mathcal{X}}}$. Then for all $m \in \mathbb{N}^*$ and $(\theta_j)_{j=1}^m \in \Theta^m$,*

$$\left(+_{j=1}^m \operatorname{Im}(K_{\theta_j})\right) \oplus \left(\cap_{j=1}^m \operatorname{Ker}(K^\sharp_{\theta_j})\right) = \mathcal{H}_K, \tag{5.35}$$

where $+_{j=1}^m \operatorname{Im}(K_{\theta_j})$ denotes the Minkowski sum of the sets $\left(\operatorname{Im}(K_{\theta_j})\right)_{j \in [m]}$.

Proof The statement boils down to proving that $\mathcal{V} := \left(+_{j=1}^m \operatorname{Im}(K_{\theta_j})\right)$ is closed in $\mathcal{H}_K$, since it is straightforward by double inclusion that $\mathcal{V}^\perp = \left(\cap_{j=1}^m \operatorname{Ker}\left(K^\sharp_{\theta_j}\right)\right)$. Let $(e_j)_{j=1}^l$ be an orthonormal basis of $\operatorname{Span}\{(k_\Theta(\cdot, \theta_j))_{j=1}^m\} \subset \mathcal{H}_{k_\Theta}$, with $l \neq m$ potentially. Such basis can be obtained by applying the Gram-Schmidt orthonormalization method to $(k_\Theta(\cdot, \theta_j))_{j=1}^m$. Then, $\mathcal{V} = \operatorname{Span}\{e_j \otimes f, 1 \leq j \leq k, f \in \mathcal{H}_{k_\mathcal{X}}\}$. Notice also that for $\forall (j_1, j_2) \in [l] \times [l], \forall f, g \in \mathcal{H}_{k_\mathcal{X}}$,

$$\langle e_{j_1} \otimes f, e_{j_2} \otimes g \rangle_{\mathcal{H}_K} = \langle e_{j_1}, e_{j_2} \rangle_{\mathcal{H}_{k_\Theta}} \langle f, g \rangle_{\mathcal{H}_{k_\mathcal{X}}}. \tag{5.36}$$

Let $(h_n)_{n \in \mathbb{N}^*}$ be a sequence in $\mathcal{V}$ converging to some $h \in \mathcal{H}_K$. By definition, one can find sequences $(f_{1,n})_{n \in \mathbb{N}^*}, \ldots, (f_{l,n})_{n \in \mathbb{N}^*} \in \mathcal{H}_{k_\mathcal{X}}$ such that for $\forall n \in \mathbb{N}^*$, $h_n = \sum_{j=1}^l e_j \otimes f_{j,n}$. Let $p, q \in \mathbb{N}^*$. It holds that, using the orthonormal property of $(e_j)_{j=1}^l$ and Equation (5.36),

$$\left\|h_p - h_q\right\|^2_{\mathcal{H}_K} = \left\|\sum_{j=1}^l e_j (f_{j,p} - f_{j,q})\right\|^2_{\mathcal{H}_K} = \sum_{j=1}^l \left\|f_{j,p} - f_{j,q}\right\|^2_{\mathcal{H}_{k_\mathcal{X}}}.$$

Since $(h_n)_{n \in \mathbb{N}^*}$ converges in $\mathcal{H}_K$, it is a Cauchy sequence, thus so are the sequences $(f_{j,n})_{n \in \mathbb{N}^*}$. But $\mathcal{H}_{k_\mathcal{X}}$ is a complete space, so these sequences converge in $\mathcal{H}_{k_\mathcal{X}}$, and by denoting $f_j = \lim_{n \to \infty} f_{j,n}$, one gets $h = \sum_{j=1}^l e_j \otimes f_j$. Therefore $h \in \mathcal{V}$, $\mathcal{V}$ is closed and the orthogonal decomposition in Equation (5.35) holds. ■

Lemma 5.31. *Let $k_\mathcal{X}, k_\Theta$ be two scalar-valued kernels, with k_Θ a strictly positive definite kernel* (i.e. *any Gram matrix associated to k_Θ is strictly positive), and $K : (\theta, \theta') \mapsto k_\Theta(\theta, \theta') I_{\mathcal{H}_{k_\mathcal{X}}}$. Let also $m \in \mathbb{N}^*$, $(\theta_j)_{j=1}^m \in \Theta^m$, and $\mathcal{V} = \left(+_{j=1}^m \operatorname{Im}(K_{\theta_j})\right)$. Then $\mathcal{I} : \mathcal{V} \to \mathbb{R}$ defined as $\mathcal{I}(h) := \sum_{j=1}^m \left\|h(\theta_j)\right\|^2_{\mathcal{H}_{k_\mathcal{X}}}$ is γ-strongly convex where γ is the smallest eigenvalue of the Gram matrix $\mathbf{K}_\Theta$ associated to $(\theta_j)_{j=1}^m$ and k_Θ. In particular, $\mathcal{I}$ is coercive.*

Proof Notice that $\mathcal{I}$ is the quadratic form on $\mathcal{V}$ associated to the linear mapping $L \in \mathcal{L}(\mathcal{H}_K)$ defined by $L := \sum_{j=1}^m K_{\theta_j} K_{\theta_j}^\sharp$, in the sense that for all $h \in \mathcal{V}, \mathcal{I}(h) = \langle h, Lh \rangle_{\mathcal{H}_K}$. Indeed,

$$\forall h \in \mathcal{V},\ \mathcal{I}(h) = \sum_{j=1}^m \left\langle K_{\theta_j}^\sharp h, K_{\theta_j}^\sharp h \right\rangle_{\mathcal{H}_{k_\mathcal{X}}} = \sum_{j=1}^m \left\langle h, K_{\theta_j} K_{\theta_j}^\sharp h \right\rangle_{\mathcal{H}_K} = \left\langle h, Lh \right\rangle_{\mathcal{H}_K}.$$

Let $h \in \mathcal{V}$, by definition there exist $(f_j)_{j=1}^m \in \mathcal{H}_{k_\mathcal{X}}^m$ such that $h = \sum_{j=1}^m K_{\theta_j} f_j$. We begin by computing $\|h\|_{\mathcal{H}_K}^2$:

$$\begin{aligned}
\langle h, h \rangle_{\mathcal{H}_K} &= \left\langle \sum_{i=1}^m K_{\theta_i} f_i, \sum_{j=1}^m K_{\theta_j} f_j \right\rangle_{\mathcal{H}_K} = \sum_{i=1}^m \sum_{j=1}^m \left\langle K_{\theta_i} f_i, K_{\theta_j} f_j \right\rangle_{\mathcal{H}_K} \\
&= \sum_{i=1}^m \sum_{j=1}^m \left\langle K_{\theta_j}^\sharp K_{\theta_i} f_i, f_j \right\rangle_{\mathcal{H}_{k_\mathcal{X}}} = \sum_{i=1}^m \sum_{j=1}^m k_\Theta(\theta_i, \theta_j) \left\langle f_i, f_j \right\rangle_{\mathcal{H}_{k_\mathcal{X}}} \\
&= \left\langle \mathbf{K}_\Theta, \mathbf{K}_f \right\rangle_{\mathrm{F}}, = \mathrm{Tr}\left(\mathbf{K}_\Theta \mathbf{K}_f \right),
\end{aligned}$$

where we introduced the notation $\mathbf{K}_\Theta \in \mathcal{M}_m(\mathbb{R})$ for the Gram matrix over the points $(\theta_j)_{j=1}^m$ with the kernel k_Θ and $\mathbf{K}_f \in \mathcal{M}_m(\mathbb{R})$ for the Gram matrix over the $(f_j)_{j=1}^m$ with the linear kernel. We compute $\mathcal{I}(h)$ similarly, starting by

$$Lh = \sum_{j=1}^m K_{\theta_j} K_{\theta_j}^\sharp \left(\sum_{i=1}^m K_{\theta_i} f_i \right) = \sum_{i=1}^m \sum_{j=1}^m k_\Theta(\theta_i, \theta_j) K_{\theta_j} f_i,$$

and following by

$$\begin{aligned}
\langle h, Lh \rangle_{\mathcal{H}_K} &= \left\langle \sum_{l=1}^m K_{\theta_l} f_l, \sum_{i=1}^m \sum_{j=1}^m k_\Theta(\theta_i, \theta_j) K_{\theta_j} f_i \right\rangle_{\mathcal{H}_K} \\
&= \sum_{l=1}^m \sum_{i=1}^m \sum_{j=1}^m k_\Theta(\theta_i, \theta_j) \left\langle K_{\theta_l} f_l, K_{\theta_j} f_i \right\rangle_{\mathcal{H}_K} = \sum_{l=1}^m \sum_{i=1}^m \sum_{j=1}^m k_\Theta(\theta_i, \theta_j) \left\langle K_{\theta_j}^\sharp K_{\theta_l} f_l, f_i \right\rangle_{\mathcal{H}_{k_\mathcal{X}}} \\
&= \sum_{l=1}^m \sum_{i=1}^m \sum_{j=1}^m k_\Theta(\theta_i, \theta_j) k_\Theta(\theta_j, \theta_l) \left\langle f_l, f_i \right\rangle_{\mathcal{H}_{k_\mathcal{X}}} = \left\langle \mathbf{K}_\Theta, \mathbf{K}_f \mathbf{K}_\Theta \right\rangle_{\mathrm{F}} = \mathrm{Tr}\left(\mathbf{K}_\Theta \mathbf{K}_\Theta \mathbf{K}_f \right).
\end{aligned}$$

By the strictly positive assumption on k_Θ, $\mathbf{K}_\Theta$ has a minimum eigenvalue $\gamma > 0$. Then $\mathbf{K}_\Theta - \gamma \, \mathrm{Id}_m$ and $\mathbf{K}_\Theta \mathbf{K}_f$ are positive symetric matrices and

$$\mathrm{Tr}\left(\left(\mathbf{K}_\Theta - \gamma \, \mathrm{Id}_m \right) \mathbf{K}_\Theta \mathbf{K}_f \right) \geq 0 \tag{5.37}$$

which can be rewritten as

$$\mathrm{Tr}\left(\mathbf{K}_\Theta \mathbf{K}_\Theta \mathbf{K}_f \right) \geq \gamma \, \mathrm{Tr}\left(\mathbf{K}_\Theta \mathbf{K}_f \right)$$

or

$$\mathcal{I}(h) \geq \gamma \left\| h \right\|_{\mathcal{H}_K}^2.$$

Finally, we recognize in the left term of Equation (5.37) a convex function of h, so $\mathcal{I} - \gamma \left\| \cdot \right\|_{\mathcal{H}_K}^2$ is convex which proves the γ-strong convexity of $\mathcal{I}$ on $\mathcal{V}$. ■

Remark 5.32. *Lemma 5.31 highlights the difference between the regularizers $\Omega(h)$, $\widetilde{\Omega}(h)$ and $\frac{1}{2}\|h\|^2_{\mathcal{H}_K}$. While the latter is uniformly stronger than the two others, on the specific subspace $\mathcal{V}$, $\widetilde{\Omega}(h)$ is γ-strongly convex. As m grows, the strong convexity constant γ goes to zero, and when $m = +\infty$, we recover $\Omega(h)$ which does not bring strong convexity to the problem and is thus strictly weaker than $\frac{1}{2}\|h\|^2_{\mathcal{H}_K}$.*

We are now ready to prove a double representer theorem for the solution of Problem 5.34.

Theorem 5.33. *Let $k_{\mathcal{X}}$, k_Θ, k_b be three scalar-valued kernels, with k_Θ a strictly positive definite kernel, k_b bounded, and $K = k_{\mathcal{X}} \operatorname{Id}_{\mathcal{H}_{k_\Theta}}$. Then Problem 5.34 admits a solution $\left(\hat{h}, \hat{t}\right) \in \mathcal{H}_K \times \mathcal{H}_{k_b}$ which writes for all $(x, \theta) \in \mathcal{X} \times \Theta$*

$$\hat{h}(x)(\theta) = \sum_{i=1}^{n}\sum_{j=1}^{m} \hat{\alpha}_{ij} k_{\mathcal{X}}(x, x_i) k_\Theta(\theta, \theta_j), \qquad \hat{t}(\theta) = \sum_{j=1}^{m} \hat{\beta}_j k_b(\theta, \theta_j)$$

for some $(\hat{\alpha}_{ij})_{i,j=1}^{n,m} \in \mathbb{R}^{n\times m}$ and $(\hat{\beta}_j)_{j=1}^m \in \mathbb{R}^m$. Moreover the solution is unique on $\mathcal{V} \times \mathcal{H}_{k_b}$ with $\mathcal{V}$ defined in Lemma 5.31.

Proof Introduce $K' : (\theta, \theta') \in \Theta \times \Theta \mapsto k_\Theta(\theta, \theta') I_{\mathcal{H}_{k_{\mathcal{X}}}} \in \mathcal{L}(\mathcal{H}_{k_{\mathcal{X}}})$. The spaces $\mathcal{H}_K$ and $\mathcal{H}_{K'}$ are unitarily equivalent by means of the feature operator $W \colon \mathcal{H}_{K'} \to \mathcal{H}_K$ such that $\forall (f, x, \theta) \in \mathcal{H}_{K'} \times \mathcal{X} \times \Theta, Wf(x)(\theta) = f(\theta)(x)$. This means that they are essentially the same spaces of functions, whose elements are seen either as mappings $\mathcal{X} \to (\Theta \to \mathbb{R})$ or $\Theta \to (\mathcal{X} \to \mathbb{R})$. Define

$$\mathcal{J} \colon \begin{pmatrix} \mathcal{H}_K \times \mathcal{H}_{k_b} & \to & \mathbb{R} \\ (h, t) & \mapsto & \frac{1}{n}\sum_{i=1}^n \widetilde{l}_\ell(h(x_i), t) + \widetilde{\Omega}(h) + \frac{\lambda}{2}\|t\|^2_{\mathcal{H}_{k_b}} \end{pmatrix}.$$

Let $\mathcal{V} = W\left(+_{j=1}^m \operatorname{Im}(K'_{\theta_j})\right)$. Since W is an isometry, thanks to Equation (5.35), it holds that $\mathcal{V} \oplus \mathcal{V}^\perp = \mathcal{H}_K$. Let $(h, t) \in \mathcal{H}_K \times \mathcal{H}_{k_b}$, there exists unique $h_{\mathcal{V}^\perp} \in \mathcal{V}^\perp$, $h_{\mathcal{V}} \in \mathcal{V}$ such that $h = h_{\mathcal{V}} + h_{\mathcal{V}^\perp}$. Notice that $\mathcal{J}(h, t) = \mathcal{J}(h_{\mathcal{V}} + h_{\mathcal{V}^\perp}, t) = \mathcal{J}(h_{\mathcal{V}}, t)$ since $\forall j \in [m], \forall x \in \mathcal{X}$, $h_{\mathcal{V}^\perp}(x)(\theta_j) = W^{-1}h_{\mathcal{V}^\perp}(\theta_j)(x) = 0$. Moreover, $\mathcal{J}$ is bounded by below so its infimum is well-defined, and $\inf_{(h,t)\in\mathcal{H}_K\times\mathcal{H}_{k_b}} \mathcal{J}(h, t) = \inf_{(h,t)\in\mathcal{V}\times\mathcal{H}_{k_b}} \mathcal{J}(h, t)$. Finally, notice that J is coercive on $\mathcal{V} \times \mathcal{H}_{k_b}$ endowed with the sum of the norms in $\mathcal{V}$ and $\mathcal{H}_{k_b}$ (which makes it a Hilbert space): if $(h_n, t_n)_{n\in\mathbb{N}^*} \in \mathcal{V} \times \mathcal{H}_{k_b}$ is such that $\|h_n\|_{\mathcal{H}_K} + \|t_n\|_{\mathcal{H}_{k_b}} \underset{n\to\infty}{\to} +\infty$, then either $(\|h_n\|_{\mathcal{H}_K})_{n\in\mathbb{N}}$ or $(\|t_n\|_{\mathcal{H}_{k_b}})_{n\in\mathbb{N}}$ has to diverge :

- If $\|t_n\|_{\mathcal{H}_{k_b}} \underset{n\to\infty}{\to} +\infty$, since

$$t_n(\theta_j) = \langle t_n, k_b(\cdot, \theta_j)\rangle_{\mathcal{H}_{k_b}} \leq k_b(\theta_j, \theta_j)\|t_n\|_{\mathcal{H}_{k_b}} \leq \kappa_b \|t_n\|_{\mathcal{H}_{k_b}} \ (\forall j \in [m]),$$

then

$$\mathcal{J}(h_n, t_n) \geq \frac{\lambda}{2}\|t_n\|^2_{\mathcal{H}_{k_b}} - \sum_{j=1}^m w_j t(\theta_j) \underset{n\to\infty}{\to} +\infty.$$

- If $\|h_n\|_{\mathcal{H}_K} \underset{n\to\infty}{\to} +\infty$, according to Lemma 5.31, $\mathcal{J}(h_n, t_n) \underset{n\to\infty}{\to} +\infty$ as long as all η_j are strictly positive, which is verified for the quasi Monte-Carlo scheme.

Thus $\mathcal{J}$ is coercive, so Bauschke et al. (2011, Proposition 11.15) allows to conclude that $\mathcal{J}$ has a minimizer $(\hat{h}, \hat{t})$ on $\mathcal{V} \times \mathcal{H}_{k_b}$. Then, in the same fashion as Theorem 3.9, define $\mathcal{U}_1 = \operatorname{Span}\{(K(\cdot, x_i)k_\Theta(\cdot, \theta_j))_{i,j=1}^{n,m}\} \subset \mathcal{V}$ and $\mathcal{U}_2 = \operatorname{Span}\{(k_b(\cdot, \theta_j))_{j=1}^{m}\} \subset \mathcal{H}_{k_b}$, and use the reproducing property to show that $(\hat{h}, \hat{t}) \in \mathcal{U}_1 \times \mathcal{U}_2$, so there exist $(\alpha_{ij})_{i,j=1}^{n,m}$ and $(\beta_j)_{j=1}^{m}$ such that for $\forall (x, \theta) \in \mathcal{X} \times \Theta$,

$$\hat{h}(x)(\theta) = \sum_{i=1}^{n}\sum_{j=1}^{m} \hat{\alpha}_{ij} k_\mathcal{X}(x, x_i) k_\Theta(\theta, \theta_j), \qquad \hat{t}(\theta) = \sum_{j=1}^{m} \hat{\beta}_j k_b(\theta, \theta_j).$$

Finally, using the γ notation from Lemma 5.31, $\mathcal{J}$ is $\min(\lambda, \gamma)$-strongly convex on $\mathcal{V} \times \mathcal{H}_{k_b}$ which ensures that the resulting $(\hat{h}, \hat{t})$ is unique. ■

Theorem 5.33 guarantees a finite-dimensional expansion for the estimator $(\hat{h}, \hat{t})$, which can be used as a plug-in in Problem 5.34. The next section highlights the performance of this approach on some synthetic and real-world datasets.

5.6.2 Numerical Experiments

To assess the quality of the estimated model by IOCSVM, we illustrate the θ-property (Schölkopf et al., 2000): the proportion of inliers has to be approximately $1 - \theta$ ($\forall \theta \in (0, 1)$). For the studied datasets (Wilt, Spambase) we used the raw inputs without applying any preprocessing. After experimenting with a few input kernels, we settled for the exponentiated χ^2 kernel $k_\mathcal{X}(x, z) := \exp\left(-\gamma_\mathcal{X} \sum_{k=1}^{d} (x_k - z_k)^2/(x_k + z_k)\right)$ with bandwidth $\gamma_\mathcal{X} = 0.25$. A Gauss-Legendre quadrature rule provided the integral approximation in Equation (5.33), with $m = 100$ samples. We chose the Gaussian kernel for k_Θ; its bandwidth parameter γ_Θ was the $0.2-$quantile of the pairwise Euclidean distances between the θ_j's obtained via the quadrature rule. The margin (bias) kernel was $k_b = k_\Theta$. As it can be seen in Fig. 5.9, the θ-property holds for the estimate which illustrates the efficiency of the proposed continuum approach for density level-set estimation. In the scalar case, the OCSVM algorithm with the Gaussian kernel in input estimates the level sets of the density of X. Combined with the θ-property, this means that the IOCSVM estimator should be able to estimate these level sets as a function of θ. We show in Figure 5.10 that this is the case on a simple Gaussian mixture underlying probability $\mathbb{P}_\mathsf{X}$. The strength of the approach is then to be able to provide a score of outlierness for a new point $x \in \mathcal{X}$:

$$s(x) := \sup_{\theta \in \Theta} \left\{\theta \,:\, d(x)(\theta) \geq 0\right\}.$$

This is well-suited to anomaly detection problems as it indicates whether a point $x \in \mathcal{X}$ is a rare event (tail of the distribution, $s(x)$ close to 0) or a common event ($s(x)$ close to 1).

5.7 Conclusion

In this chapter we proposed infinite task learning (ITL), a novel nonparametric framework aiming at jointly solving parametrized tasks for a continuum of hyperparameters.

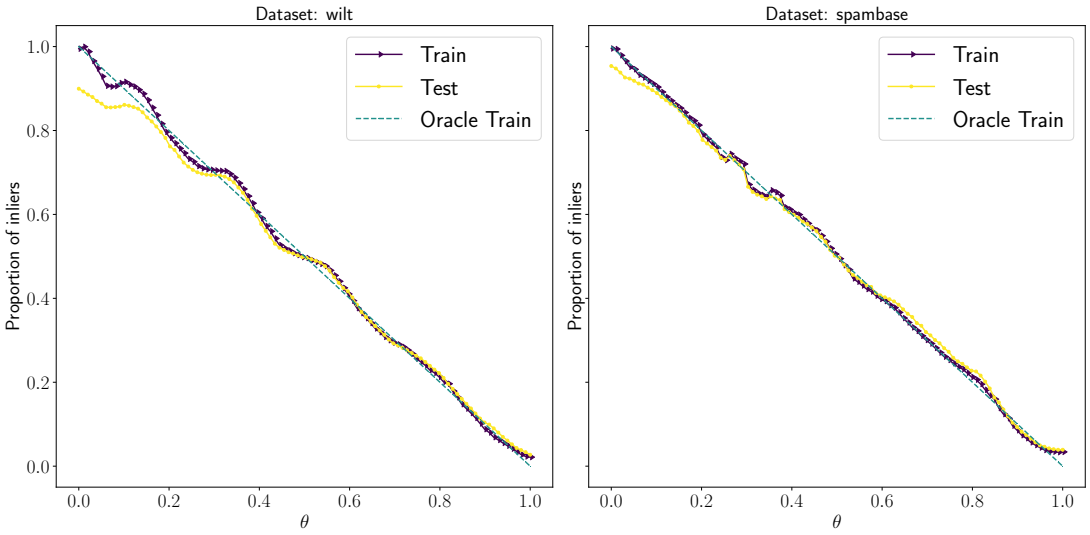
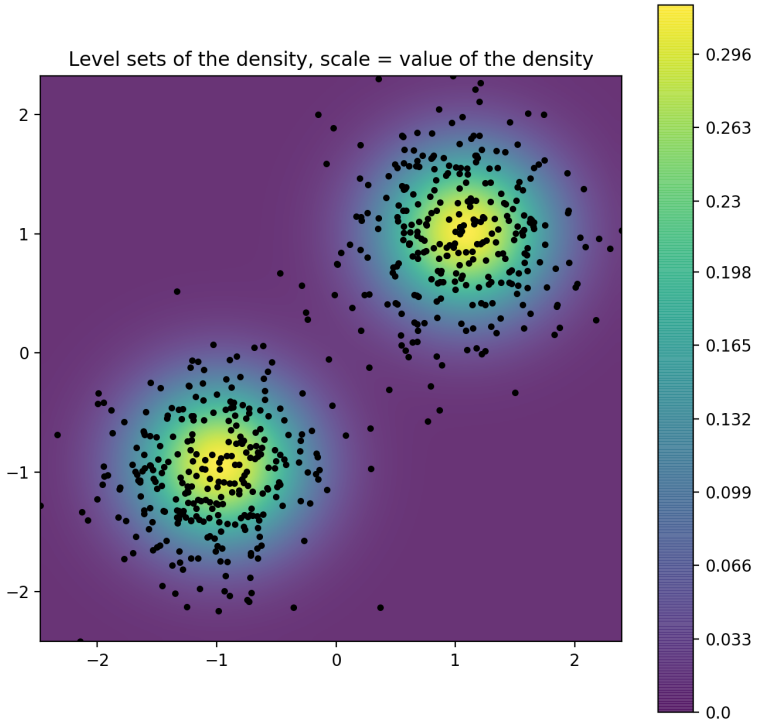


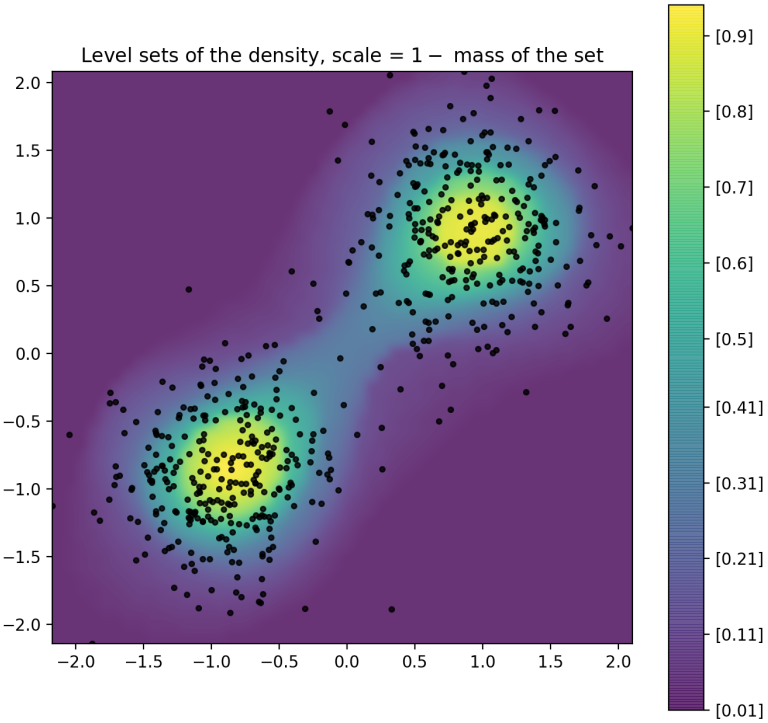
Figure 5.9 – DLSE: the θ-property is approximately satisfied.

We provided excess risk guarantees for the studied ITL scheme, and demonstrated its practical efficiency and flexibility in various tasks including cost-sensitive classification, quantile regression and density level set estimation.

While in all aforementioned tasks Θ and Y are one-dimensional, the ITL framework can be adapted to more complex tasks such as emotion transfer tasks as emphasized in Chapter 6.



(a) Theoretical level sets of the Gaussian mixture, scaled by the value of the density.



(b) Estimated level sets of the Gaussian mixture, scaled by 1− mass of the sets

Figure 5.10 – Proof of concept of ∞-OCSVM for DLSE on a simple Gaussian mixture.

6
Emotion Transfer

Contents

6.1 Introduction

Recent years have witnessed an increasing attention around style transfer problems (Gatys et al., 2016; Wynen et al., 2018; Jing et al., 2020) in machine learning. In a nutshell, style transfer refers to the transformation of an object according to a target style. It has found numerous applications in computer vision (Ulyanov et al., 2016; Choi et al., 2018; Puy and Pérez, 2019; Yao et al., 2020), natural language processing (Fu et al., 2018) as well as audio signal processing (Grinstein et al., 2018) where objects at hand are contents in which style is inherently part of their perception. Style transfer is one of the key components of data augmentation (Mikołajczyk and Grochowski, 2018) as a means to artificially generate meaningful additional data for the training of deep neural networks. Besides, it has also been shown to be useful for counterbalancing bias in data by producing stylized contents with a well-chosen style (see for instance Geirhos et al. 2019) in image recognition. More broadly, style transfer fits into the wide paradigm of parametric modeling, where a system, a process or a signal can be controlled by its parameter value. Adopting this perspective, style transfer-like applications can also be found in digital twinning (Tao et al., 2019; Barricelli et al., 2019; Lim et al., 2020), a field of growing interest in health and industry.

In this chapter, we propose a novel principled approach for style transfer, exemplified in the context of emotion transfer of face images. Given a set of emotions, classical emotion transfer refers to the task of transforming face images according to these target emotions. The pioneering works in emotion transfer include that of Blanz and Vetter (1999) who proposed a morphable 3D face model whose parameters could be modified

for facial attribute editing. Susskind et al. (2008) designed a deep belief net for facial expression generation using action unit (AU) annotations.

More recently, extensions of generative adversarial networks (GANs, Goodfellow et al. 2014) have proven to be particularly powerful for tackling image-to-image translation problems (Zhu et al., 2017). Several works have addressed emotion transfer for facial images by conditioning GANs on a variety of guiding information ranging from discrete emotion labels to photos and videos. In particular, StarGAN (Choi et al., 2018) is conditioned on discrete expression labels for face synthesis. ExprGAN (Ding et al., 2018) proposes synthesis with the ability to control expression intensity through a controller module conditioned on discrete labels. Other GAN-based approaches make use of additional information such as AU labels (Pumarola et al., 2018), target landmarks (Qiao et al., 2018), fiducial points (Song et al., 2018) and photos/videos (Geng et al., 2018). While GANs have achieved high quality image synthesis, they come with some pitfalls: they are particularly difficult to train and require large amounts of training data.

Unlike previous approaches, we adopt a functional point of view: given some person, we assume that the full range of the emotional faces can be modelled as a continuous function from emotions to images. This view exploits the geometry of the representation of emotions (Russell, 1980), assuming that one can pass a facial image “continuously" from one emotion to an other. We then propose to address the problem of emotion transfer by learning an image-to-function model able to predict for a given facial input image represented by its landmarks (Tautkute et al., 2018), the continuous function that maps an emotion to the image transformed by this emotion.

This function-valued regression approach relies on a the infinite task learning (ITL) technique developed in Chapter 5. ITL enlarges the scope of multi-task learning (Evgeniou and Pontil, 2004; Evgeniou et al., 2005) by learning to solve simultaneously a set of tasks parametrized by a continuous parameter. While strongly linked to other parametric learning methods such the one proposed by Takeuchi et al. (2006), the approach differs from previous works by leveraging the use of operator-valued kernels and vector-valued reproducing kernel Hilbert spaces (vv-RKHS; Pedrick 1957; Micchelli and Pontil 2005; Carmeli et al. 2006) to model function-valued functions with vectorial outputs. vv-RKHSs have proven to be relevant in solving supervised learning tasks such as multiple quantile regression (Sangnier et al., 2016) or unsupervised problems like anomaly detection (Schölkopf et al., 2001b). A common property of these works is that the output to be predicted is a real-valued function of a real parameter.

To solve the emotion transfer problem, we present an extension of ITL, vector ITL (or shortly vITL) which involves functional outputs with vectorial representation of the faces and the emotions, showing that the approach is still easily controllable by the choice of appropriate kernels guaranteeing continuity and smoothness. In particular, the functional point of view by the inherent regularization induced by the kernel makes the approach suitable even for limited and partially observed emotional images. We demonstrate the efficiency of the vITL approach in a series of numerical experiments showing that it can achieve state-of-the-art performance on two benchmark datasets.

The chapter is structured as follows. We formulate the problem and introduce the vITL framework in Section 6.2. Section 6.3 is dedicated to the underlying optimization problem. Numerical experiments conducted on two benchmarks of the domain are presented in Section 6.4. Discussion and future work conclude the chapter in Section 6.5.

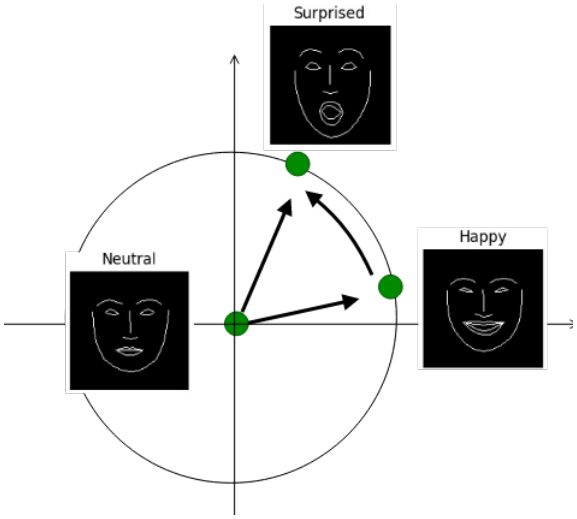


Figure 6.1 – Illustration of emotion transfer.

6.2 Problem Setting

In this section we define our problem. Our aim is to design a system capable of transferring emotions: having access to the face image of a given person our goal is to convert his/her face to a specified target emotion. In other words, the system should implement a mapping of the form

$$(\text{face, emotion}) \mapsto \text{face}. \tag{6.1}$$

In order to tackle this task, one requires a representation of the emotions, and similarly that of the faces. The classical categorical description of emotions deals with the classes 'happy', 'sad', 'angry', 'surprised', 'disgusted', 'fearful'. The valence-arousal model (Russell, 1980) embeds these categories into the 2-dimensional space. The resulting representation of the emotions are points $\theta \in \mathbb{R}^2$, each coordinate of these vectors encoding the valence (pleasure to displeasure) and arousal (high to low) associated to the emotions. This is the emotion representation we use while noting that there are alternative encodings in higher dimension ($\Theta \subset \mathbb{R}^p$, $p \geq 2$; Vemulapalli and Agarwala 2019) to which the presented framework can be naturally adapted. Throughout this work faces are represented by landmark points. Landmarks have been proved to be a useful representation in facial recognition (Saragih et al., 2009; Scherhag et al., 2018; Zhang et al., 2015), 3D facial reconstruction and sentiment analysis. Tautkute et al. (2018) have shown that emotions can be accurately recognized by detecting changes in the localization of the landmarks. Given M number of landmarks on the face, this means a description $x \in \mathcal{X} := \mathbb{R}^{2M}$, and hence a $d := 2M$-dimensional representation. The resulting mapping from Equation (6.1) is illustrated in Figure 6.1: starting from a neutral face and the target happy one can traverse to the happy face; from the happy face, given the target emotion surprise one can get to the surprised face.

In an ideal world, for each person, one would have access to a trajectory z mapping each emotion $\theta \in \Theta$ to the corresponding landmark locations $x \in \mathcal{X}$; this function $z : \Theta \mapsto \mathcal{X}$ can be taken for instance to be the element of $L^2[\Theta, \mu; \mathcal{X}]$, the space of $\mathbb{R}^d$-valued square-integrable function w.r.t. to a measure μ. The probability measure μ allows capturing the frequency of the individual emotions. In practice, one has realizations $(z_i)_{i\in[n]}$, each z_i corresponds to a single person possible appearing multiple times. The

trajectories are observable at finite many emotions $\left(\tilde{\theta}_{i,j}\right)_{j\in[m]}$.[1] In order to capture the relation from Equation (6.1) one can rely on a hypothesis space $\mathcal{H}$ with elements

$$h : \mathcal{X} \mapsto (\Theta \mapsto \mathcal{X}). \tag{6.2}$$

The value $h(x)(\theta)$ represents the landmark prediction from face x and target emotion θ.

We consider **two tasks** for emotion transfer:

- **Single emotional input:** In the first problem, the assumption is that all the faces appearing as the input in Equation (6.1) come from a fixed emotion $\tilde{\theta}_0 \in \Theta$. The data which can be used to learn the mapping h consists of $t = n$ triplets[2]

$$x_i = z_i\left(\tilde{\theta}_0\right) \in \mathcal{X}, \qquad \mathbf{Y}_i = \Big(\underbrace{z_i\left(\tilde{\theta}_{i,j}\right)}_{=:y_{i,j}}\Big)_{j\in[m]} \in \mathcal{X}^m,$$

$$(\theta_{i,j})_{j\in[m]} = \left(\tilde{\theta}_{i,j}\right)_{j\in[m]} \in \Theta^m,\ i \in [t].$$

  To measure the quality of the reconstruction using a function h, one can consider a convex loss $\ell\colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ on the landmark space. The resulting objective function to minimize is

$$\mathcal{R}_{\mathcal{S}}(h) := \frac{1}{tm} \sum_{i\in[t]} \sum_{j\in[m]} \ell(h(x_i)(\theta_{i,j}), y_{i,j}). \tag{6.3}$$

  The risk $\mathcal{R}_{\mathcal{S}}(h)$ captures how well the function h reconstructs on average the landmarks $y_{i,j}$ when applied to the input landmark locations x_i.

- **Joint emotional input:** In this problem, the faces appearing as input in Equation (6.1) can arise from any emotion. The observations consist of triplets

$$x_{m(i-1)+l} = z_i\left(\tilde{\theta}_{i,l}\right) \in \mathcal{X}, \qquad \mathbf{Y}_{m(i-1)+l} = \Big(\underbrace{z_i\left(\tilde{\theta}_{i,j}\right)}_{=:y_{m(i-1)+l,j}}\Big)_{j\in[m]} \in \mathcal{X}^m$$

$$(\theta_{m(i-1)+l,j})_{j\in[m]} = \left(\tilde{\theta}_{i,j}\right)_{j\in[m]} \in \Theta^m,$$

  where $(i, l) \in [n] \times [m]$ and the number of pairs is $t = nm$. Having defined this dataset one can optimize the same objective in Equation (6.3) as before. Particularly, this means that the pair (i, l) plays the role of index i of the previous case. The $(\theta_{i,j})_{i,j\in[t]\times[m]}$ is an extended version of the $\left(\tilde{\theta}_{i,j}\right)_{i,j\in[t]\times[m]}$ to match the indices going from 1 to t in Equation (6.3).

We leverage the flexible class of vector-valued reproducing kernel Hilbert spaces (vv-RKHS; Carmeli et al. 2010) for the hypothesis class schematically illustrated in Equation (6.2). Learning within vv-RKHS has been shown to be relevant for tackling

[1] To keep the notation simple, we assume that m is the same for all the z_i-s.

[2] In this case $\theta_{i,j}$ is a literal copy of $\tilde{\theta}_{i,j}$ which helps to get a unified formulation with the joint emotional input setting.

function-valued regression (Kadri et al., 2016). The construction follows the structure

$$h : \mathcal{X} \mapsto \underbrace{\underbrace{(\Theta \mapsto \mathcal{X})}_{\in \mathcal{H}_G}}_{\in \mathcal{H}_K} \tag{6.4}$$

which we detail below. A general reminder on these spaces of functions is presented in Chapter 2. The vector ($\mathbb{R}^d$)-valued capability is beneficial to handle the $\Theta \mapsto \mathcal{X} = \mathbb{R}^d$ mapping; the associated $\mathbb{R}^d$-valued RKHS $\mathcal{H}_G$ is uniquely determined by a matrix-valued kernel $G : \Theta \times \Theta \to \mathbb{R}^{d\times d} = \mathcal{L}(\mathcal{X})$. Similarly, in Equation (6.4) the $\mathcal{X} \to \mathcal{H}_G$ mapping is modelled by a vv-RKHS $\mathcal{H}_K$ corresponding to an operator-valued kernel $K : \mathcal{X} \times \mathcal{X} \to \mathcal{L}(\mathcal{H}_G)$. We chose K and G to be separable kernels of the form

$$G(\theta, \theta') = k_\Theta(\theta, \theta')\mathbf{A}, \qquad K(x, x') = k_\mathcal{X}(x, x')\,\mathrm{Id}_{\mathcal{H}_G} \tag{6.5}$$

with a scalar-valued kernel $k_\mathcal{X} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ and $k_\Theta : \Theta \times \Theta \to \mathbb{R}$, and symmetric, positive definite matrix $\mathbf{A} \in \mathbb{R}^{d\times d}$. This choice corresponds to the intuition that for similar input landmarks and target emotions, the predicted output landmarks should also be similar, as measured by $k_\mathcal{X}$, k_Θ and $\mathbf{A}$, respectively. More precisely, smoothness (analytic property) of the emotion-to-landmark output function can be induced for instance by choosing a Gaussian kernel $k_\Theta(\theta, \theta') = \exp(-\gamma\|\theta - \theta'\|_2^2)$ with $\gamma > 0$. The matrix $\mathbf{A}$ when chosen as $\mathbf{A} = \mathrm{Id}_d$ corresponds to independent landmarks coordinates while other choices encode prior knowledge about the dependency among the landmarks coordinates (Álvarez et al., 2012). Similarly, the smoothness of function h can be driven by the choice of a Gaussian kernel over $\mathcal{X}$ while the identity operator on $\mathcal{H}_G$ is the simplest choice to cope with functional outputs. By denoting the norm in $\mathcal{H}_K$ as $\|\cdot\|_{\mathcal{H}_K}$, the final objective function is

$$\min_{h\in\mathcal{H}_K} \; \mathcal{R}_\mathcal{S}(h) + \frac{\lambda}{2}\left\|h\right\|_{\mathcal{H}_K}^2 \tag{6.6}$$

with a regularization parameter $\lambda > 0$ which balances between the data-fitting term ($\mathcal{R}_\mathcal{S}(h)$) and smoothness ($\left\|h\right\|_{\mathcal{H}_K}^2$). We refer to Problem 6.6 as vector-valued infinite task learning (vITL).

Remark: This problem is a natural adaptation of the ITL framework (Brault et al., 2019) learning with operator-valued kernels mappings of the form $\mathcal{X} \mapsto (\Theta \mapsto \mathcal{Y})$ where $\mathcal{Y}$ is a subset of $\mathbb{R}$; here $\mathcal{Y} = \mathcal{X}$. An other difference is μ: in ITL this probability measure is designed to approximate integrals via quadrature rule, in vITL it captures the observation mechanism.

6.3 Optimization

This section is dedicated to the solution of Problem 6.6 which is an optimization problem over functions ($h \in \mathcal{H}_K$). General tools to handle problems such as Problem 6.6 have been proposed in Chapter 3, in particular throughout this chapter we consider the squared loss $\ell(x, x') = \frac{1}{2}\|x - x'\|_2^2$; in this case we recognize the partially observed scenario with square loss from Section 3.2.2. Thus, Lemma 3.7 resumes solving Problem 6.6 to solving a *Sylvester equation*, as it holds that the solution $\hat{h} \in \mathcal{H}_K$ writes

$$\hat{h}(x)(\theta) = \sum_{i=1}^{t}\sum_{j=1}^{m} k_\mathcal{X}(x, x_i)k_\Theta(\theta, \theta_{i,j})\mathbf{A}\hat{\alpha}_{i,j}, \quad \forall (x, \theta) \in \mathcal{X} \times \Theta \tag{6.7}$$

for some $(\hat{\alpha}_{i,j})_{i,j\in[t]\times[m]} \in \mathcal{U}^{nt}$ which can be gathered in a matrix $\hat{\boldsymbol{\alpha}} \in \mathcal{M}_{nt,d}(\mathbb{R})$ so that $\hat{\boldsymbol{\alpha}}$ is solution to

$$\mathbf{K}\hat{\boldsymbol{\alpha}}\mathbf{A} + tm\lambda\hat{\boldsymbol{\alpha}} = \mathbf{Y} \tag{6.8}$$

where the Gram matrix $\mathbf{K} = [k_{i,j}]_{i,j\in[tm]} \in \mathcal{M}_{tm}(\mathcal{R})$, and the matrix consisting of all the observations $\mathbf{Y} = [\mathbf{Y}_i]_{i\in[tm]} \in \mathcal{M}_{tm,d}(\mathbb{R})$ are defined as

$$\begin{aligned} k_{m(i_1-1)+j_1,m(i_2-1)+j_2} &:= k_{\mathcal{X}}(x_{i_1}, x_{i_2})k_{\Theta}(\theta_{i_1,j_1}, \theta_{i_2,j_2}),\ (i_1,j_1),(i_2,j_2) \in [t]\times[m], \\ \mathbf{Y}_{m(i-1)+j} &:= y_{i,j}^{\top},\ (i,j) \in [t]\times[m]. \end{aligned}$$

Remarks:

- Computational complexity: In case of $\mathbf{A} = \mathrm{Id}_d$, the complexity of the closed form solution is $\mathcal{O}\left((tm)^3\right)$. If all the samples are observed at the same locations $(\theta_{i,j})_{i,j\in[t]\times[n]}$, i.e. $\theta_{i,j} = \theta_{l,j}$ for $\forall(i,l,j) \in [t]\times[t]\times[m]$, then the Gram matrix $\mathbf{K}$ has a tensorial structure $\mathbf{K} = \mathbf{K}_{\mathcal{X}} \otimes \mathbf{K}_{\Theta}$ with $\mathbf{K}_{\mathcal{X}} = [k_{\mathcal{X}}(x_i, x_j)]_{i,j\in[t]} \in \mathbb{R}^{t\times t}$ and $\mathbf{K}_{\Theta} = [k_{\Theta}(\theta_{1,i}, \theta_{1,j})]_{i,j\in[m]} \in \mathbb{R}^{m\times m}$. In this case, the computational complexity reduces to $\mathcal{O}\left(t^3 + m^3\right)$. If additional scaling is required one can leverage recent dedicated kernel ridge regression solvers (Rudi et al., 2017; Meanti et al., 2020). If $\mathbf{A}$ is not identity but invertible, then multiplying Equation (6.8) with $\mathbf{A}^{-1}$ gives $\mathbf{K}\hat{\boldsymbol{\alpha}} + tm\lambda\hat{\boldsymbol{\alpha}}\mathbf{A}^{-1} = \mathbf{Y}\mathbf{A}^{-1}$ which is a Sylvester equation for which efficient custom solvers exist (El Guennouni et al., 2002). If $\mathbf{A}$ is not invertible, one can use Singular Value Decomposition to reduce the dimensionality of $\hat{\boldsymbol{\alpha}}$ and fall back on the invertible case.

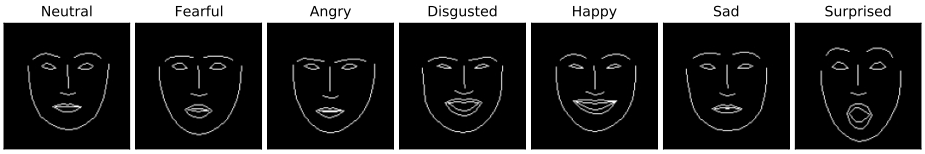
- Regularization in vv-RKHS: Using the notations above, for any $h \in \mathcal{H}_K$ parameterized by a matrix $\boldsymbol{\alpha}$, it holds that $\|h\|^2_{\mathcal{H}_K} = \mathrm{Tr}\left(\mathbf{K}\boldsymbol{\alpha}\mathbf{A}\boldsymbol{\alpha}^{\top}\right)$. Given two matrices $\mathbf{A}_1, \mathbf{A}_2$ and associated vv-RKHSs $\mathcal{H}_{K_1}$ and $\mathcal{H}_{K_2}$, if $\mathbf{A}_1$ and $\mathbf{A}_2$ are invertible then any function in $\mathcal{H}_{K_1}$ parameterized by $\boldsymbol{\alpha}$ also belongs to $\mathcal{H}_{K_2}$ (and vice versa), within which it is parameterized by $\boldsymbol{\alpha}\mathbf{A}_2^{-1}\mathbf{A}_1$. This means that the two spaces contain the same functions, but their norms are different.

6.4 Experiments

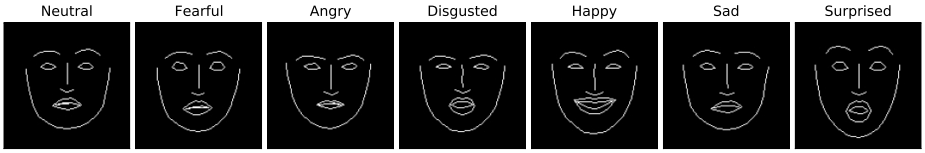
In this section we demonstrate the efficiency of the proposed vITL technique in emotion transfer. We first introduce the two benchmark datasets we used in our experiments and give details about data representation and choice of the hypothesis space in Section 6.4.1. Then, in Section 6.4.2, we provide a quantitative performance assessment of the vITL approach (in mean squared error and classification accuracy sense) with a comparison to the state-of-the-art StarGAN method. Section 6.4.3 is dedicated to investigation of the role of $\mathbf{A}$ (see Equation (6.5)) and the robustness of the approach w.r.t. missing obervation. These two sets of experiments (Section 6.4.2 and Section 6.4.3) are augmented with a qualitative analysis (Section 6.4.4). The code written for all these experiments is available on GitHub.

6.4.1 Experimental Setup

We used the following two popular face datasets for evaluation.



(a) KDEF



(b) RaFD

Figure 6.2 – Illustration of the landmark edge maps for different emotions and both datasets.

- Karolinska Directed Emotional Faces (KDEF; Lundqvist et al. 1998): This dataset contains facial emotion pictures from 70 actors (35 females and 35 males) recorded over two sessions which give rise to a total of 140 samples per emotion. In addition to neutral, the captured facial emotions include afraid, angry, disgusted, happy, sad and surprised.

- Radboud Faces Database (RaFD; Langner et al. 2010): This benchmark contains emotional pictures of 67 unique identities (including Caucasian males and females, Caucasian children, and Moroccan Dutch males). Each subject was trained to show the following expressions: anger, disgust, fear, happiness, sadness, surprise, contempt, and neutral according to the facial action coding system (FACS; Ekman et al. 2002).

In our experiments, we used frontal images and seven emotions from each of these datasets. An edge map illustration of landmarks for different emotions is shown in Figure 6.2.

At this point, it is worth recalling that we are learning a function-valued function, $h : \mathcal{X} \mapsto (\Theta \mapsto \mathcal{X})$ using a vv-RKHS as our hypothesis class (see Section 6.2). In the following we detail the choices made concerning the representation of the landmarks in $\mathcal{X}$, that of the emotions in Θ, and in the kernel design $k_{\mathcal{X}}, k_{\Theta}$ and $\mathbf{A}$.

Landmark representation, pre-processing: We applied the following pre-processing steps to get the landmark representations which form the input of the algorithms. To extract 68 landmark points for all the facial images, we used the standard `dlib` library. The estimator is based on `dlib`'s implementation of Kazemi and Sullivan (2014), trained on the iBUG 300-W face landmark dataset. Each landmark is represented by its 2D location. The alignment of the faces was carried out by the Python library `imutils`. The method ensures that faces across all identities and emotions are vertical, centered and of similar sizes. In essence, this is implemented through an affine transformation computed after drawing a line segment between the estimated eye centers. Each image was resized to the size 128×128. The landmark points computed in the step above were transformed through the same affine transformation. These two preprocessing steps gave rise to the aligned, scaled and vectorized landmarks $x \in \mathbb{R}^{136=2\times 68}$.

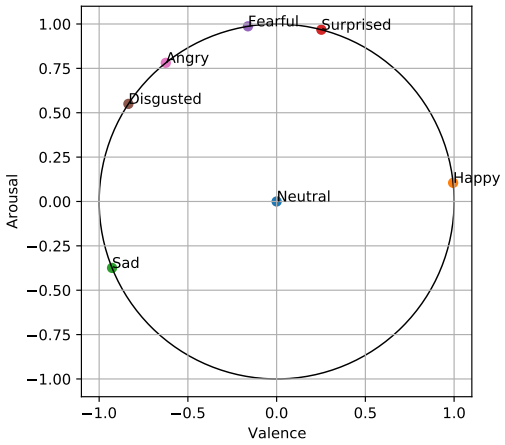


Figure 6.3 – Extracted ℓ_2-normalized valence-arousal centroids for each emotion from the manually annotated train set of the AffectNet database.

Emotion representation: We represented emotion labels as points in the 2D valence-arousal space (VA, Russell 1980). Particularly, we used a manually annotated part of the large-scale AffectNet database (Mollahosseini et al., 2017). For all samples of a particular emotion in the AffectNet data, we computed the centroid (data mean) of the valence and arousal values. The resulting ℓ_2-normalized 2D vectors constituted our emotion representation as depicted in Figure 6.3. The normalization is akin to assuming that the modeled emotions are of the same intensity. In our experiments, the emotion 'neutral' was represented by the origin. Such an emotion embedding allowed us to take into account prior knowledge about the angular proximity of emotions in the VA space, while keeping the representation simple and interpretable for post-hoc manipulations.

Kernel design: We took the kernels $k_{\mathcal{X}}$, k_{Θ} to be Gaussian on the landmark representation space and the emotion representation space, with respective bandwidth $\gamma_{\mathcal{X}}$ and γ_{Θ}. $\mathbf{A}$ was assumed to be Id_d unless specified otherwise.

6.4.2 Quantitative Performance Assessment

In this section we provide a quantitative assessment of the proposed vITL approach.

Performance measures: We applied two metrics to quantify the performance of the compared systems, namely the test mean squared error (MSE) and emotion classification accuracy. The classification accuracy can be thought of as an indirect evaluation. To compute this measure, for each dataset we trained a ResNet-18 classifier to recognize emotions from ground-truth landmark edge maps (as depicted in Figure 6.2). The trained network was then used to compute classification accuracy over the predictions at test time. To rigorously evaluate outputs for each split of the data, we used a classifier trained on RaFD to evaluate KDEF predictions and vice-versa; this also allowed us to make the problem more challenging. The ResNet-18 network was appropriately modified to take grayscale images as input. During training, we used random horizontal flipping and cropping between 90-100% of the original image size to augment the data. All the images were finally resized to 224×224 and fed to the network. The network was trained from scratch using the stochastic gradient descent optimizer with learning rate and momentum set to 0.001 and 0.9, respectively. The training was carried out

for 10 epochs with a batch size of 16.

We report the mean and standard deviation of the aforementioned metrics over ten 90%-10% train-test splits of the data. The test set for each split is constructed by removing 10% of the identities from the data. For each split, the best $\gamma_{\mathcal{X}}, \gamma_{\Theta}$ and λ values were determined by 6-fold and 10-fold cross-validation on KDEF and RaFD, respectively.

Baseline: We used the popular StarGAN (Choi et al., 2018) system as our baseline. Other GAN-based studies use additional information and are not directly comparable to our setting. For fair comparison, the generator G and discriminator D were modified to be fully-connected networks that take vectorized landmarks as input. In particular, G was an encoder-decoder architecture where the target emotion, represented as a 2D emotion encoding as for our case, was appended at the bottleneck layer. It contained approximately one million parameters, which was chosen to be comparable with the number of coefficients in vITL ($839,664 = 126 \times 7 \times 7 \times 136$ for KDEF). ReLU activation function was used in all layers except before bottleneck in G and before penultimate layers of both G and D. We used their default parameter values in the code.[3] Experiments over each split of KDEF and RaFD were run for 50K and 25K iterations, respectively.

MSE results: The test MSE for the compared systems is summarized in Table 6.1. As the table shows, the vITL technique outperforms StarGAN on both datasets. One can observe low reconstruction cost for vITL in both the single and the joint emotional input case. Interestingly, a performance gain is obtained with vITL-joint on the RaFD data in MSE sense. We hypothesize that this is due to the joint model benefiting from input landmarks for other emotions in the small data regime (only 67 samples per emotion for RaFD). Despite our best efforts, we found it quite difficult to train StarGAN reliably and the diversity of its outputs was low.

Classification results: The emotion classification accuracies are available in Table 6.2. The classification results clearly demonstrate the improved performance and the higher quality of the generated emotion of vITL over StarGAN; the latter also produces predictions with visible face distortions as it is illustrated in Section 6.4.4. To provide further insight into the classification performance we also show the confusion matrices for the joint vITL model on a particular split of KDEF and RaFD datasets in Figure 6.4. For both the datasets, the classes 'happy' and 'surprised' are easiest to detect. Some confusions arise between the classes 'neutral' vs 'sad' and 'fearful' vs 'surprised'. Such mistakes are expected when only using landmark locations for recognizing emotions.

6.4.3 Analysis of Additional Properties of vITL

This section is dedicated to the effect of the choice of $\mathbf{A}$ (in kernel G) and to the robustness of vITL w.r.t. missing obervation.

Influence of A in the matrix-valued kernel G: Here, we illustrate the effect of matrix $\mathbf{A}$ (see Equation (6.5)) on the vITL estimator and show that a good choice of $\mathbf{A}$ can lead to lower dimensional models, while preserving the quality of the prediction. The choice of $\mathbf{A}$ is built on the knowledge that the empirical covariance matrices of the output training data contains structural information that can be exploited with vv-RKHS (Kadri et al., 2013). In order to investigate this possibility, we performed the

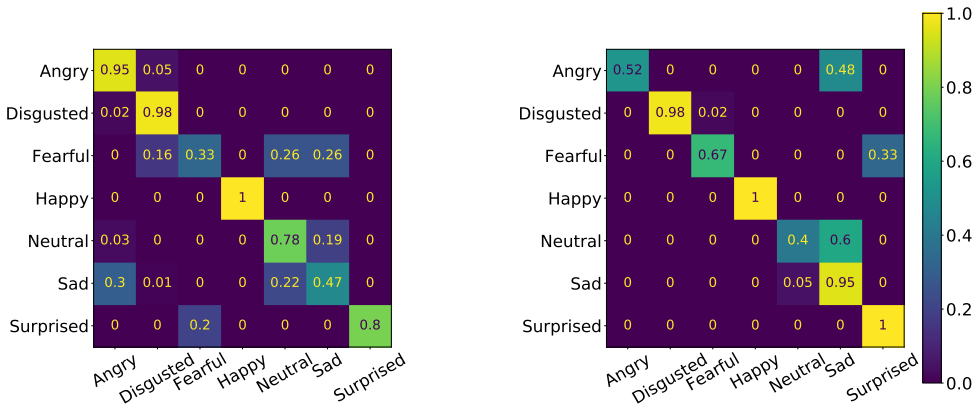
[3]The code is available at https://github.com/yunjey/stargan.

Table 6.1 – MSE error (mean ± std) on test data for the vITL-single (top), the vITL-joint and the StarGAN system (bottom). Lower is better.

Methods	KDEF frontal	RaFD frontal
vITL: θ_0 = neutral	0.010 ± 0.001	0.009 ± 0.004
vITL: θ_0 = fearful	0.010 ± 0.001	0.010 ± 0.005
vITL: θ_0 = angry	0.012 ± 0.002	0.010 ± 0.005
vITL: θ_0 = disgusted	0.012 ± 0.001	0.010 ± 0.004
vITL: θ_0 = happy	0.011 ± 0.001	0.010 ± 0.004
vITL: θ_0 = sad	0.011 ± 0.001	0.009 ± 0.004
vITL: θ_0 = surprised	0.010 ± 0.001	0.011 ± 0.006
vITL: Joint	0.011 ± 0.001	0.007 ± 0.001
StarGAN	0.029 ± 0.003	0.024 ± 0.007

Table 6.2 – Emotion classification accuracy (mean ± std) for the vITL-single (top), the vITL-joint (middle) and the StarGAN system (bottom). Higher is better.

Methods	KDEF frontal	RaFD frontal
vITL: θ_0 = neutral	76.12 ± 4.57	79.76 ± 7.88
vITL: θ_0 = fearful	76.22 ± 4.91	78.81 ± 8.36
vITL: θ_0 = angry	74.49 ± 2.31	78.10 ± 7.51
vITL: θ_0 = disgusted	74.18 ± 4.22	78.33 ± 4.12
vITL: θ_0 = happy	73.57 ± 2.74	80.48 ± 5.70
vITL: θ_0 = sad	75.82 ± 4.11	77.62 ± 5.17
vITL: θ_0 = surprised	74.69 ± 2.25	80.71 ± 5.99
vITL: Joint	74.81 ± 3.10	77.11 ± 3.97
StarGAN	70.69 ± 8.46	65.88 ± 8.92



(a) KDEF (b) RaFD

Figure 6.4 – Confusion matrices for classification accuracy of vITL-joint model. Left: dataset KDEF. Right: dataset RaFD. The y axis represents the true labels, the x axis stands for the predicted labels. More diagonal is better.

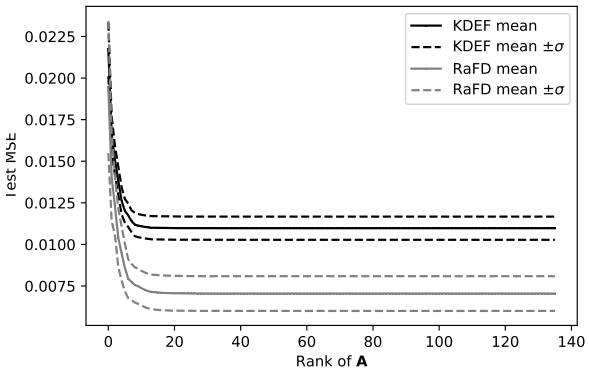


Figure 6.5 – Test MSE (mean ± std) as a function of the rank of the matrix $\mathbf{A}$. Smaller MSE is better.

singular value decomposition of $\mathbf{Y}^\top\mathbf{Y}$ which gives the eigenvectors collected in matrix $\mathbf{V} \in \mathbb{R}^{d\times d}$. For a fixed rank $r \leq d$, define $\mathbf{J}_r = \mathrm{diag}(\underbrace{1,\ldots,1}_{r},\underbrace{0,\ldots,0}_{d-r})$, set $\mathbf{A} = \mathbf{V}\,\mathbf{J}_r\,\mathbf{V}^\top$ and train a vITL system with the resulting $\mathbf{A}$. While in this case $\mathbf{A}$ is no more invertible, each coefficient $\hat{\alpha}_{i,j}$ from Equation (6.7) belongs to the r-dimensional subspace of $\mathbb{R}^d$ generated by the eigenvectors associated to the r largest eigenvalues of $\mathbf{Y}^\top\mathbf{Y}$. This makes a reparameterization possible and leads to a decrease in the size of the model, going from $t \times m \times d$ parameters to $t \times m \times r$. We report in Figure 6.5 the resulting test MSE performance (mean ± standard deviation) obtained from 10 different splits, and empirically observe that $r = 20$ suffices to preserve the optimal performances of the model.

Learning under a missing obervation regime: To assess the robustness of vITL w.r.t. missing data, we considered a random mask $(\eta_{i,j})_{i\in[n],j\in[m]} \in \{0,1\}^{n\times m}$; a sample $z_i(\theta_{i,j})$ was used for learning only when $\eta_{i,j} = 1$. Thus, the percentage of missing data was $p := \frac{1}{nm}\sum_{i,j\in[n]\times[m]}\eta_{i,j}$. The experiment was repeated for 10 splits of the dataset, and on each split we averaged the results using 4 different random masks $(\eta_{i,j})_{i\in[n],j\in[m]}$. The resulting test MSE of the predictor as a function of p is summarized in Figure 6.6. As it can be seen, the vITL approach is quite stable in the presence of missing data on both datasets.

6.4.4 Qualitative Analysis

In this section we show example outputs produced by vITL in the context of discrete and continuous emotion generation. While the former is the classical task of synthesis given input landmarks and target emotion label, the latter serves to demonstrate a key benefit of our approach, which is the ability to synthesize meaningful outputs while continuously traversing the emotion embedding space.

Discrete emotion generation: In Figure 6.7 and Figure 6.8 we show qualitative results for generating landmarks using discrete emotion labels present in the datasets. For vITL, not only are the emotions recognizable, but landmarks on the face boundary are reasonably well-synthesized and other parts of the face visibly less distorted when compared to StarGAN. The identity in terms of the face shape is also better preserved.

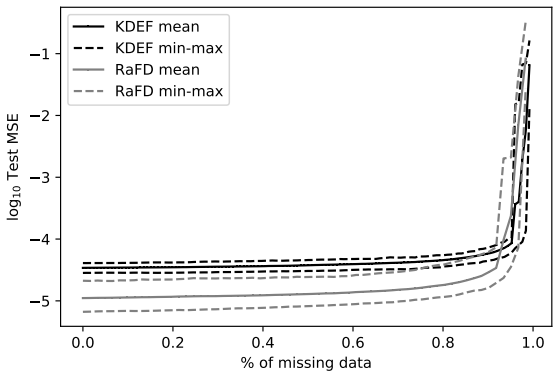


Figure 6.6 – Logarithm of the test MSE (min-mean-max) as a function of the percentage of missing data. Solid line: mean; dashed line: min-max. Smaller MSE is better.

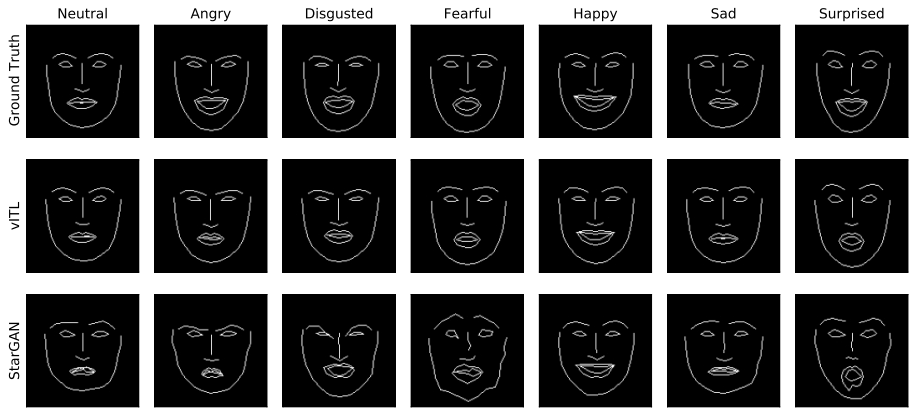


Figure 6.7 – Discrete expression synthesis results on the KDEF dataset with ground-truth neutral landmarks as input.

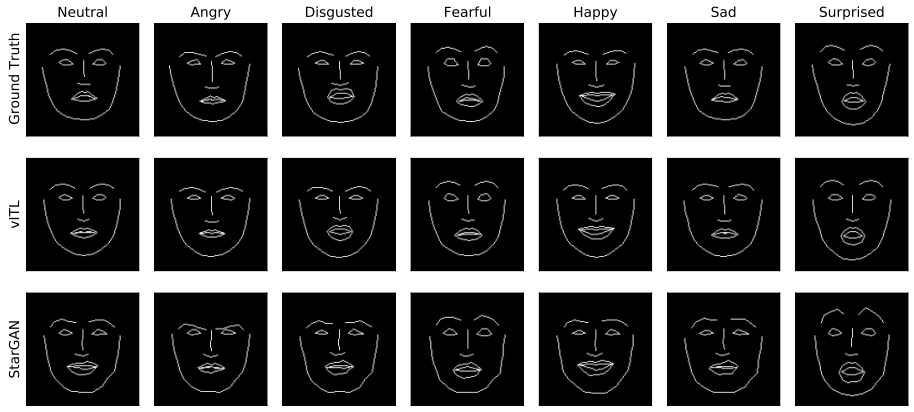


Figure 6.8 – Discrete expression synthesis results on the RaFD dataset with ground-truth neutral landmarks as input.

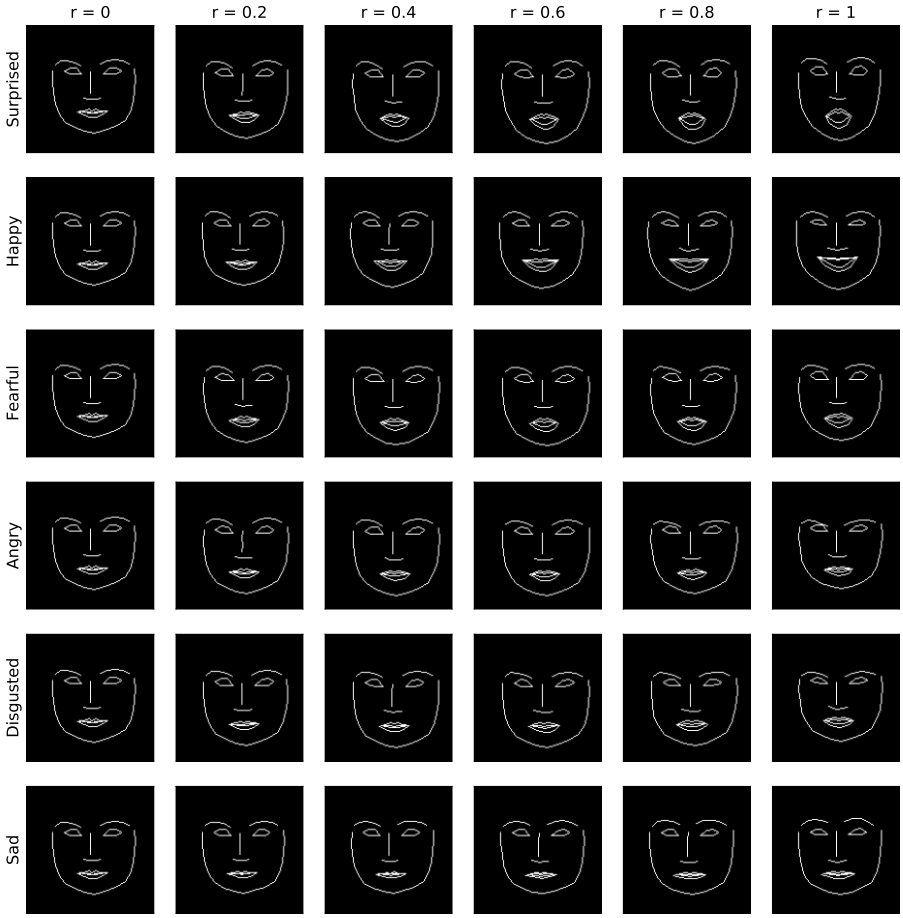


Figure 6.9 – Continuous expression synthesis results with vITL on the KDEF dataset, with ground-truth neutral landmarks. The generation is starting from neutral and proceeds in the radial direction towards an emotion with increasing radii r.

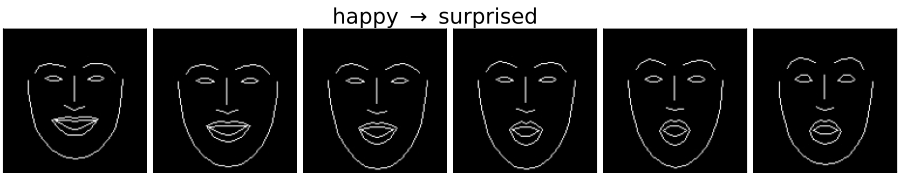


Figure 6.10 – Continuous expression synthesis with vITL technique on the RaFD dataset, with ground-truth neutral landmarks. The generation is starting from 'happy' and proceeds by changing angular position towards 'surprised'. For a more fine-grained video illustration traversing from 'happy' to 'sad' along the circle, see the demo on GitHub.

Continuous emotion generation: Starting from neutral emotion, continuous generation in the radial direction is illustrated in Figure 6.9. The landmarks vary smoothly and conform to the expected intensity variation in each emotion on increasing the radius of the vector in VA space. We also show in Figure 6.10 the capability to generate intermediate emotions by changing the angular position, in this case from 'happy' to 'surprised'. For a more fine-grained video illustration traversing from 'happy' to 'sad' along the circle, see the GitHub repository.

These experiments and qualitative results demonstrate the efficiency of the vITL approach in emotion transfer.

6.5 Conclusion

In this chapter we introduced a novel approach to style transfer based on function-valued regression, and exemplified it on the problem of emotion transfer. The proposed vector-valued infinite task learning (vITL) framework relies on operator-valued kernels. vITL (i) is capable of encoding and controlling continuous style spaces, (ii) benefit from a representer theorem for efficient computation, and (iii) facilitates regularity control via the choice of the underlying kernels. The framework can be extended in several directions. Other losses can be leveraged to produce outlier-robust or sparse models in the spirit of Chapter 4. Instead of being chosen prior to learning, the input kernel could be learned using deep architectures (Mehrkanoon and Suykens, 2018; Liu et al., 2020) opening the door to a wide range of applications.

Conclusion and Perspectives

In this thesis, we have introduced a general learning framework based on vector-valued reproducible kernel Hilbert spaces and integral losses, akin to estimate predictive models with functional outputs. Due to the complexity of the task, certain approximations are required for the resulting optimization problems to be tractable. These can be applied either on the loss side, gaining a workable representation of the solution by means of a *double representer theorem* or on the hypothesis space by restricting the coefficients to live in a finite-dimensional subspace of the vv-RKHS, chosen to be well-suited to the problem. These approximations enable primal or dual algorithm using variations of gradient descent. Among them, the use of *random Fourier features* allows to tackle large-scale learning settings. This learning framework is proved to be useful to extend functional output regression to more involved losses such as robust losses. Moreover, scenarii where the observed training outputs are not functional but for which the task to be solved is parameterized can also benefit from the functional point of view, resulting in the *infinite task learning* framework.

We exemplified the proposed framework in quantile regression, cost-sensitive classification, density level set estimation and emotion transfer with non scalar outputs and parameters. In these applications, the space of parameters Θ is relatively simple; future work could include learning scenarii where Θ is a complex structured set, over which the design of the output kernel is crucial. One could extend for instance quantile regression to Hilbert-valued output random variables that result in Θ being the unit ball of a certain Hilbert space.

Functional output regression in vv-RKHSs was tackled by leverage the duality view, allowing to design losses through infimal convolution whose associated estimator shows robustness or sparsity. This idea could be pushed further by considering different norms to convolute with the loss function, giving rise to a large family of dual problems that might influence the behavior of the estimator in interesting ways.

From a statistical learning point of view, many questions are still to be investigated. In particular, while we have studied the generalization capabilities of the ITL estimator, the convergence of the risk associated to the estimator towards the Bayes risk remains an open question. The analysis of such risk could be performed in the context of random Fourier features, possibly allowing for the derivation of optimality conditions on the sampling measure to achieve good performance with the fewer features possible.

From a modeling point of view, we have evocated the combination of neural networks with kernels to deal with objects over which choosing a good kernel is notoriously hard like images or documents. These hybrid approaches seem promising and could be further developed, as learning the kernel brings nonconvexity to the optimization problems and requires dedicated solvers. Moreover, the class of vv-RKHSs used in this thesis rely on *separable kernels* which can be less appropriate in certain applications as they induce a tensorial structure in the Gram matrix. Going beyond such kernels requests to start anew from the problem formulation as we would lose the computability advantages brought by the separability.

Concerning further applications, we are particularly interested in applying the developed framework to the study of dynamical systems. Imagining applications where the output variable is a trajectory having to satisfy a certain differential equation, choosing appropriate output vv-RKHS would ensure by design a control over the properties of the model, and using the reproducing property for derivatives in a RKHS paves the way to tractable optimization problems.

Finally, we want to mention the potential of ITL for meta-modeling and meta-learning (Ton et al., 2021), which could be deepened in further works.

Bibliography

M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. In *USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, pages 265–283, 2016. page 103

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. pages 16, 19, 38, 86, 121

G. Aneiros, F. Ferraty, and P. Vieu. Variable selection in semi-functional regression models. In *Recent Advances in Functional Data Analysis and Related Topics*. Physica-Verlag HD, 2011. page 15

G. Aneiros-Pérez and P. Vieu. Semi-functional partial linear regression. *Statistics & Probability Letters*, 76(11):1102–1110, 2006. page 15

A. Argyriou, T. Evgeniou, and M. Pontil. Convex multi-task feature learning. *Machine learning*, 73(3):243–272, 2008a. page 19

A. Argyriou, M. Pontil, Y. Ying, and C. Micchelli. A spectral regularization framework for multi-task structure learning. In *Advances in Neural Information Processing Systems (NIPS)*, 2008b. page 19

N. Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, pages 337–404, 1950. page 29

P.-C. Aubin-Frankowski and Z. Szabó. Hard shape-constrained kernel machines. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 384–395, 2020. page 100

J. Audiffren and H. Kadri. Stability of multi-task kernel regression algorithms. In *Asian Conference on Machine Learning (ACML)*, pages 1–16, 2013. page 90

N. Azzedine, A. Laksaci, and E. Ould-Saïd. On robust nonparametric regression estimation for a functional regressor. *Statistics & Probability Letters*, 78(18):3216–3221, 2008. page 15

F. Bach. On the equivalence between kernel quadrature rules and random feature expansions. *Journal of Machine Learning Research*, 18:1–38, 2017. page 32

F. Bach, R. Jenatton, J. Mairal, and G. Obozinski. Optimization with sparsity-inducing penalties. *Foundations and Trends in Machine Learning*, 4(1):1–106, 2012. page 84

F. R. Bach, D. Heckerman, and E. Horvitz. Considering cost asymmetry in learning classifiers. *Journal of Machine Learning Research*, 7(Aug):1713–1741, 2006. pages 18, 43

L. Baldassarre, L. Rosasco, A. Barla, and A. Verri. Multi-output learning via spectral filtering. *Machine Learning*, 87(3):259–301, 2012. page 19

B. R. Barricelli, E. Casiraghi, and D. Fogli. A survey on digital twin: Definitions, characteristics, applications, and design implications. *IEEE Access*, 7:167653–167671, 2019. page 117

P. L. Bartlett and S. Mendelson. Rademacher and Gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3:463–482, 2002. page 88

H. H. Bauschke, P. L. Combettes, et al. *Convex analysis and monotone operator theory in Hilbert spaces*. Springer, 2011. pages 25, 26, 46, 103, 113

S. Ben-David, N. Eiron, and P. M. Long. On the difficulty of approximately maximizing agreements. *Journal of Computer and System Sciences*, 66(3):496–514, 2003. page 17

A. Berlinet and C. Thomas-Agnan. *Reproducing Kernel Hilbert Spaces in Probability and Statistics*. Kluwer, 2004. page 29

V. Blanz and T. Vetter. A morphable model for the synthesis of 3D faces. In *Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 187–194, 1999. page 117

G. Boente, M. Salibian-Barrera, and P. Vena. Robust estimation for semi-functional linear regression models. *Computational Statistics & Data Analysis*, 152:107041, 2020. page 66

L. Bottou. Stochastic gradient learning in neural networks. *Proceedings of Neuro-Nımes*, 91(8):12, 1991. page 55

L. Bottou, F. E. Curtis, and J. Nocedal. Optimization methods for large-scale machine learning. *SIAM Review*, 60(2):223–311, 2018. page 55

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. pages 66, 77

O. Bousquet and A. Elisseeff. Stability and generalization. *Journal of Machine Learning Research*, 2:499–526, 2002. pages 88, 89

O. Bousquet, Y. Klochkov, and N. Zhivotovskiy. Sharper bounds for uniformly stable algorithms. In *Conference on Learning Theory (COLT)*, pages 610–626, 2020. pages 89, 90

S. Boyd, S. P. Boyd, and L. Vandenberghe. *Convex optimization*. Cambridge University Press, 2004. page 25

R. Brault, M. Heinonen, and F. d'Alché Buc. Random Fourier features for operator-valued kernels. In *Asian Conference on Machine Learning (ACML)*, pages 110–125, 2016. pages 39, 54

R. Brault, A. Lambert, Z. Szabó, M. Sangnier, and F. d'Alché-Buc. Infinite task learning in RKHSs. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 1294–1302, 2019. pages 39, 42, 43, 121

J. Breckling and R. Chambers. M-quantiles. *Biometrika*, 75(4):761–771, 1988. page 103

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 38

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 38, 39

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 66

S. E. Campana and S. R. Thorrold. Otoliths, increments, and elements: keys to a comprehensive understanding of fish populations? *Canadian Journal of Fisheries and Aquatic Sciences*, 58(1):30–38, 2001. page 104

H. Cardot, F. Ferraty, and P. Sarda. Spline estimators for the functional linear model. *Statistica Sinica*, pages 571–591, 2003. page 15

H. Cardot, C. Crambes, and P. Sarda. Quantile regression when the covariates are functions. *Nonparametric Statistics*, 17(7):841–856, 2005. page 15

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. pages 35, 118

C. Carmeli, E. De Vito, A. Toigo, and V. Umanitá. Vector valued reproducing kernel Hilbert spaces and universality. *Analysis and Applications*, 8(01):19–61, 2010. pages 32, 35, 37, 38, 58, 120

Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo. StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8789–8797, 2018. pages 117, 118, 125

G. Choquet. *Cours d'analyse: Tome II. Topologie*. Masson et Cie., 1969. page 94

C. Ciliberto, A. Rudi, L. Rosasco, and M. Pontil. Consistent multitask learning with nonlinear output relations. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1986–1996, 2017. page 86

C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995. pages 17, 33

C. Crambes, L. Delsol, and A. Laksaci. Robust nonparametric estimation for functional data. *Journal of Nonparametric Statistics*, 20(7):573–598, 2008. pages 15, 66

M. Cuturi, K. Fukumizu, and J.-P. Vert. Semigroup kernels on measures. *Journal of Machine Learning Research*, 6:1169–1198, 2005. page 29

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 29

B. Dai, B. Xie, N. He, Y. Liang, A. Raj, M.-F. F. Balcan, and L. Song. Scalable kernel methods via doubly stochastic gradients. In *Advances in Neural Information Processing Systems (NIPS)*, 2014. page 56

H. Ding, K. Sricharan, and R. Chellappa. ExprGAN: Facial expression editing with controllable expression intensity. In *Conference on Artificial Intelligence (AAAI)*, pages 6781–6788, 2018. page 118

H. Drucker, C. J. Burges, L. Kaufman, A. J. Smola, and V. Vapnik. Support vector regression machines. In *Advances in Neural Information Processing Systems (NIPS)*, pages 155–161, 1997. page 84

R. M. Dudley. *Real Analysis and Probability*. Cambridge Studies in Advanced Mathematics. Cambridge University Press, 2 edition, 2002. page 67

P. Ekman, W. Friesen, and J. Hager. Facial action coding system: The manual. *Salt LakeCity, UT: Research Nexus.*, 2002. page 123

A. El Guennouni, K. Jbilou, and A. Riquet. Block Krylov subspace methods for solving large Sylvester equations. *Numerical Algorithms*, 29(1):75–96, 2002. pages 50, 122

C. Elkan. The foundations of cost-sensitive learning. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 973–978, 2001. pages 16, 85

T. Evgeniou and M. Pontil. Regularized multi-task learning. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 109–117, 2004. pages 16, 84, 118

T. Evgeniou, C. Micchelli, and M. Pontil. Learning multiple tasks with kernel methods. *Journal of Machine Learning Research*, 6:615–637, 2005. pages 19, 86, 118

Y. Fei, G. Rong, B. Wang, and W. Wang. Parallel L-BFGS-B algorithm on GPU. *Computers & Graphics*, 40:1–9, 2014. page 103

V. Feldman and J. Vondrak. Generalization bounds for uniformly stable algorithms. Technical report, 2018. (https://arxiv.org/abs/1812.09859). page 89

F. Ferraty and P. Vieu. *Nonparametric functional data analysis: theory and practice*. Springer Science & Business Media, 2006. page 15

F. Ferraty, A. Laksaci, A. Tadj, P. Vieu, et al. Kernel regression with functional response. *Electronic Journal of Statistics*, 5:159–171, 2011. pages 15, 66

Z. Fu, X. Tan, N. Peng, D. Zhao, and R. Yan. Style transfer in text: Exploration and evaluation. In *Conference on Artificial Intelligence (AAAI)*, pages 663–670, 2018. page 117

K. Fukumizu, A. Gretton, X. Sun, and B. Schölkopf. Kernel measures of conditional dependence. In *Advances in Neural Information Processing Systems (NIPS)*, pages 498–496, 2008. page 32

J. J. Gatys, A. A., and F.-F. Li. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision (ECCV)*, pages 694–711, 2016. page 117

R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann, and W. Brendel. ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations (ICLR)*, 2019. page 117

J. Geng, T. Shao, Y. Zheng, Y. Weng, and K. Zhou. Warp-guided GANs for single-photo facial animation. *ACM Transactions on Graphics*, 37(6):1–12, 2018. page 118

F. Girosi, M. J. Jones, and T. A. Poggio. Regularization theory and neural networks architectures. *Neural Computation*, 7(2):219–269, 1995. page 17

A. Glazer, M. Lindenbaum, and S. Markovitch. q-OCSVM: A q-quantile estimator for high-dimensional distributions. In *Advances in Neural Information Processing Systems (NIPS)*, pages 503–511, 2013. pages 20, 84, 86

I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016. page 104

I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2672–2680, 2014. page 118

A. Gretton, K. Fukumizu, C. H. Teo, L. Song, B. Schölkopf, and A. Smola. A kernel statistical test of independence. In *Advances in Neural Information Processing Systems (NIPS)*, pages 585–592, 2008. page 32

A. Gretton, K. Borgwardt, M. Rasch, B. Schölkopf, and A. Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13:723–773, 2012. page 32

E. Grinstein, N. Q. Duong, A. Ozerov, and P. Pérez. Audio style transfer. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 586–590, 2018. page 117

T. Hastie, S. Rosset, R. Tibshirani, and J. Zhu. The entire regularization path for the support vector machine. *Journal of Machine Learning Research*, 5(Oct):1391–1415, 2004. page 84

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 31

X. Huang, A. Maier, J. Hornegger, and J. A. Suykens. Indefinite kernels in least squares support vector machines and principal component analysis. *Applied and Computational Harmonic Analysis*, 43(1):162–172, 2017. page 29

P. J. Huber. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, pages 73–101, 1964. pages 66, 68, 69

R. Huusari and H. Kadri. Entangled kernels - beyond separability. *Journal of Machine Learning Research*, 22(24):1–40, 2021. page 37

R. Huusari, H. Kadri, and C. Capponi. Multi-view metric learning in vector-valued kernel spaces. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 415–424, 2018. page 37

T. Jaakkola, M. Diekhans, and D. Haussler. Using the Fisher kernel method to detect remote protein homologies. In *International Conference on Intelligent Systems for Molecular Biology (ISMB)*, volume 99, pages 149–158, 1999. page 106

N. Japkowicz and S. Stephen. The class imbalance problem: a systematic study. *Intelligent Data Analysis*, 6(5):429–449, 2002. page 85

N. Jean, S. M. Xie, and S. Ermon. Semi-supervised deep kernel learning: Regression with unlabeled data by minimizing predictive variance. *arXiv preprint arXiv:1805.10407*, 2018. page 35

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 29

Y. Jing, Y. Yang, Z. Feng, J. Ye, Y. Yu, and M. Song. Neural style transfer: A review. *IEEE Transactions on Visualization and Computer Graphics*, 26(11):3365–3385, 2020. page 117

E. Jones, T. Oliphant, P. Peterson, et al. SciPy: Open source scientific tools for Python, 2001. page 103

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)*, pages 374–380, 2010. pages 15, 20, 46, 50, 66

H. Kadri, A. Rakotomamonjy, F. Bach, and P. Preux. Multiple operator-valued kernel learning. *arXiv preprint arXiv:1203.1596*, 2012. page 37

H. Kadri, M. Ghavamzadeh, and P. Preux. A generalized kernel approach to structured output learning. *International Conference on Machine Learning (ICML)*, pages 471–479, 2013. pages 38, 125

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(20):1–54, 2016. pages 16, 20, 31, 38, 46, 47, 66, 67, 71, 79, 90, 121

I. Kalogridis and S. Van Aelst. Robust functional regression based on principal components. *Journal of Multivariate Analysis*, 173:393–415, 2019. page 66

M. Kanagawa, B. K. Sriperumbudur, and K. Fukumizu. Convergence analysis of deterministic kernel-based quadrature rules in misspecified settings. *Foundations of Computational Mathematics*, 20(1):155–194, 2020. page 32

J. Kandola, J. Shawe-Taylor, and N. Cristianini. Optimizing kernel alignment over combinations of kernel. 2002. page 35

V. Kazemi and J. Sullivan. One millisecond face alignment with an ensemble of regression trees. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1867–1874, 2014. page 123

N. Keskar and A. Wächter. A limited-memory quasi-Newton algorithm for bound-constrained non-smooth optimization. *Optimization Methods and Software*, pages 1–22, 2017. page 101

G. Kimeldorf and G. Wahba. Some results on Tchebycheffian spline functions. *Journal of mathematical analysis and applications*, 33(1):82–95, 1971. page 33

S. Klus, I. Schuster, and K. Muandet. Eigendecompositions of transfer operators in reproducing kernel Hilbert spaces. *Journal of Nonlinear Science*, 30(1):283–315, 2020. page 31

R. Koenker and G. Bassett Jr. Regression quantiles. *Econometrica: Journal of the Econometric Society*, pages 33–50, 1978. pages 16, 18, 42, 85, 97

P. Laforgue, A. Lambert, L. Brogat-Motte, and F. dAlché Buc. Duality in RKHSs with infinite dimensional outputs: Application to robust losses. In *International Conference on Machine Learning (ICML)*, pages 5598–5607, 2020. pages 66, 68, 73

O. Langner, R. Dotsch, G. Bijlstra, D. H. Wigboldus, S. T. Hawk, and A. Van Knippenberg. Presentation and validation of the Radboud faces database. *Cognition and emotion*, 24(8):1377–1388, 2010. page 123

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. pages 34, 35

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. pages 66, 68

C.-L. Li, W.-C. Chang, Y. Cheng, Y. Yang, and B. Póczos. Mmd gan: Towards deeper understanding of moment matching network. *arXiv preprint arXiv:1705.08584*, 2017. page 35

C.-L. Li, W.-C. Chang, Y. Mroueh, Y. Yang, and B. Póczos. Implicit kernel learning. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 2007–2016. PMLR, 2019a. page 35

Y. Li, Y. Liu, and J. Zhu. Quantile regression in reproducing kernel Hilbert spaces. *Journal of the American Statistical Association*, 102(477):255–268, 2007. page 98

Z. Li, J.-F. Ton, D. Oglic, and D. Sejdinovic. A unified analysis of random Fourier features. In *International Conference on Machine Learning (ICML)*, volume 97, pages 3905–3914, 2019b. page 34

H. Lian. Nonlinear functional models for functional responses in reproducing kernel Hilbert spaces. *Canadian Journal of Statistics*, 35(4):597–606, 2007. pages 15, 46, 50, 66, 79

K. Y. H. Lim, P. Zheng, and C.-H. Che. A state-of-the-art survey of digital twin: techniques, engineering product lifecycle management and business innovation perspectives. *Journal of Intelligent Manufacturing*, 31:1313–1337, 2020. page 117

X. Lin, H.-L. Zhen, Z. Li, Q.-F. Zhang, and S. Kwong. Pareto multi-task learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. page 20

F. Liu, W. Xu, J. Lu, G. Zhang, A. Gretton, and D. J. Sutherland. Learning deep kernels for non-parametric two-sample tests. *arXiv preprint arXiv:2002.09116*, 2020. pages 35, 130

D. Lundqvist, A. Flykt, and A. Öhman. The Karolinska directed emotional faces (KDEF). *CD ROM from Department of Clinical Neuroscience, Psychology section, Karolinska Institutet*, 91(630):2–2, 1998. page 123

I. Macedo and R. Castro. Learning div-free and curl-free vector fields by matrix-valued kernels. 2010. page 37

P. Mahé and J.-P. Vert. Graph kernels based on tree patterns for molecules. *Machine learning*, 75(1):3–35, 2009. page 29

R. A. Maronna and V. J. Yohai. Robust functional linear regression based on splines. *Computational Statistics & Data Analysis*, 65:46–55, 2013. page 66

A. Maurer and M. Pontil. Bounds for vector-valued function estimation. Technical report, 2016. (http://arxiv.org/abs/1606.01487). page 88

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)*, 2020. pages 33, 122

S. Mehrkanoon and J. A. K. Suykens. Deep hybrid neural-kernel networks using random Fourier features. *Neurocomputing*, 298:46–54, 2018. pages 35, 130

S. Mehrkanoon, A. Zell, and J. A. Suykens. Scalable hybrid deep neural kernel networks. In *ESANN*, 2017. page 35

C. Micchelli and M. Pontil. On learning vector-valued functions. *Neural Computation*, 17:177–204, 2005. pages 16, 19, 38, 83, 118

C. Micchelli, Y. Xu, and H. Zhang. Universal kernels. *Journal of Machine Learning Research*, 7:2651–2667, 2006. page 32

A. Mikołajczyk and M. Grochowski. Data augmentation for improving deep learning in image classification problem. In *International Interdisciplinary PhD Workshop (IIPhDW)*, pages 117–122, 2018. page 117

E. Moen, N. O. Handegard, V. Allken, O. T. Albert, A. Harbitz, and K. Malde. Automatic interpretation of otoliths using deep learning. *PLoS One*, 13(12):e0204713, 2018. page 104

A. Mollahosseini, B. Hasani, and M. H. Mahoor. AffectNet: A database for facial expression, valence, and arousal computing in the wild. *IEEE Transactions on Affective Computing*, 10(1):18–31, 2017. page 124

J. J. Moreau. Proximité et dualité dans un espace hilbertien. Technical report, 1965. (https://hal.archives-ouvertes.fr/hal-01740635). page 27

W. J. Morokoff and R. E. Caflisch. Quasi-Monte Carlo integration. *Journal of computational physics*, 122(2):218–230, 1995. page 96

J. S. Morris. Functional regression. *Annual Review of Statistics and Its Application*, 2: 321–359, 2015. pages 15, 66

Y. E. Nesterov. A method for solving the convex programming problem with convergence rate $\mathcal{O}(1/k^2)$. In *Dokl. akad. nauk Sssr*, volume 269, pages 543–547, 1983. pages 54, 55

E. Novak. *Deterministic and stochastic error bounds in numerical analysis*. Springer, 2006. page 32

J. Oliva, W. Neiswanger, B. Póczos, E. Xing, H. Trac, S. Ho, and J. Schneider. Fast function to function regression. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 717–725, 2015. pages 16, 66

J. B. Oliva, A. Dubey, A. G. Wilson, B. Póczos, J. Schneider, and E. P. Xing. Bayesian nonparametric kernel-learning. In *Artificial Intelligence and Statistics*, pages 1078–1086. PMLR, 2016. page 35

C. S. Ong, X. Mary, S. Canu, and A. J. Smola. Learning with non-positive kernels. In *Proceedings of the twenty-first international conference on Machine learning*, page 81, 2004. page 29

A. Ordonez, L. Eikvil, A.-B. Salberg, A. Harbitz, S. M. Murray, and M. C. Kampffmeyer. Explaining decisions of deep neural networks used for fish age prediction. *PloS one*, 15(6):e0235013, 2020. pages 104, 105

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12(Oct):2825–2830, 2011. page 105

G. Pedrick. Theory of reproducing kernels for Hilbert spaces of vector-valued functions. Technical report, University of Kansas, Department of Mathematics, 1957. pages 16, 35, 118

W. Polonik. Minimum volume sets and generalized quantile processes. *Stochastic processes and their applications*, 69(1):1–24, 1997. page 19

A. Pumarola, A. Agudo, A. M. Martinez, A. Sanfeliu, and F. Moreno-Noguer. GANimation: Anatomically-aware facial animation from a single image. In *European Conference on Computer Vision (ECCV)*, pages 818–833, 2018. page 118

G. Puy and P. Pérez. A flexible convolutional solver for fast style transfers. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8963–8972, 2019. page 117

F. Qiao, N. Yao, Z. Jiao, Z. Li, H. Chen, and H. Wang. Geometry-contrastive GAN for facial expression transfer. Technical report, 2018. (https://arxiv.org/abs/1802.01822). page 118

T. Qingguo. M-estimation for functional linear regression. *Communications in Statistics-Theory and Methods*, 46(8):3782–3800, 2017. page 66

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 33, 54

A. Rahimi and B. Recht. Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1313–1320, 2008. page 33

J. Ramsay and B. Silverman. Functional data analysis, 1997. pages 15, 65

J. O. Ramsay. Functional data analysis. *Encyclopedia of Statistical Sciences*, 4, 2004. page 77

J. O. Ramsay and B. W. Silverman. *Applied functional data analysis: methods and case studies*. Springer, 2007. page 20

M. Reimherr, B. Sriperumbudur, B. Taoufik, et al. Optimal prediction for additive function-on-function regression. *Electronic Journal of Statistics*, 12(2):4571–4601, 2018. pages 16, 66

R. T. Rockafellar. *Convex analysis*, volume 36. Princeton university press, 1970. page 25

R. T. Rockafellar. *Conjugate duality and optimization*, volume 16. Siam, 1974. page 57

R. T. Rockafellar. Integral functionals, normal integrands and measurable selections. In *Nonlinear operators and the calculus of variations*, pages 157–207. Springer, 1976. page 57

A. Rudi and L. Rosasco. Generalization properties of learning with random features. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 3218–3228, 2017. pages 34, 88

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. pages 33, 122

W. Rudin. *Fourier Analysis on Groups*. Wiley-Interscience, 1990. page 33

J. A. Russell. A circumplex model of affect. *Journal of Personality and Social Psychology*, 39(6):1161–1178, 1980. pages 118, 119, 124

R. Salakhutdinov and G. E. Hinton. Using deep belief nets to learn covariance kernels for gaussian processes. In *NIPS*, volume 7, pages 1249–1256. Citeseer, 2007. page 35

M. Sangnier, O. Fercoq, and F. d'Alché Buc. Joint quantile regression in vector-valued RKHSs. *Advances in Neural Information Processing Systems (NIPS)*, pages 3693–3701, 2016. pages 42, 84, 86, 99, 101, 118

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. pages 20, 66, 68, 73

J. M. Saragih, S. Lucey, and J. F. Cohn. Face alignment through subspace constrained mean-shifts. In *International Conference on Computer Vision (ICCV)*, pages 1034–1041, 2009. page 119

U. Scherhag, D. Budhrani, M. Gomez-Barrero, and C. Busch. Detecting morphed face images using facial landmarks. In *International Conference on Image and Signal Processing (ICISP)*, pages 444–452, 2018. page 119

B. Schölkopf and A. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, 2002. page 29

B. Schölkopf, A. J. Smola, R. C. Williamson, and P. L. Bartlett. New support vector algorithms. *Neural computation*, 12(5):1207–1245, 2000. pages 85, 113

B. Schölkopf, R. Herbrich, and A. J. Smola. A generalized representer theorem. In *International Conference on Computational Learning Theory (COLT)*, pages 416–426. Springer, 2001a. page 33

B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson. Estimating the support of a high-dimensional distribution. *Neural computation*, 13(7): 1443–1471, 2001b. pages 16, 19, 108, 118

D. Sejdinovic, B. Sriperumbudur, A. Gretton, and K. Fukumizu. Equivalence of distance-based and RKHS-based statistics in hypothesis testing. *Annals of Statistics*, 41:2263–2291, 2013. page 32

H. Shin and S. Lee. An RKHS approach to robust functional linear regression. *Statistica Sinica*, pages 255–272, 2016. page 66

S. Singh, S. M. Richards, V. Sindhwani, J.-J. E. Slotine, and M. Pavone. Learning stabilizable nonlinear dynamics with contraction-based regularization. *The International Journal of Robotics Research*, 2020. page 39

A. Skajaa. Limited memory BFGS for nonsmooth optimization. *Master's thesis*, 2010. page 101

A. Smola, A. Gretton, L. Song, and B. Schölkopf. A Hilbert space embedding for distributions. In *Algorithmic Learning Theory (ALT)*, pages 13–31, 2007. page 32

I. M. Sobol. On the distribution of points in a cube and the approximate evaluation of integrals. *Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki*, 7(4):784–802, 1967. page 52

L. Song, Z. Lu, R. He, Z. Sun, and T. Tan. Geometry guided adversarial facial expression synthesis. In *International Conference on Multimedia (MM)*, pages 627–635, 2018. page 118

S. Sonnenburg, G. Rätsch, C. Schäfer, and B. Schölkopf. Large scale multiple kernel learning. *The Journal of Machine Learning Research*, 7:1531–1565, 2006. page 35

B. Sriperumbudur and Z. Szabó. Optimal rates for random Fourier features. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1144–1152, 2015. page 34

B. Sriperumbudur, A. Gretton, K. Fukumizu, B. Schölkopf, and G. Lanckriet. Hilbert space embeddings and metrics on probability measures. *Journal of Machine Learning Research*, 11:1517–1561, 2010. page 32

B. Sriperumbudur, K. Fukumizu, and G. Lanckriet. Universality, characteristic kernels and RKHS embedding of measures. *Journal of Machine Learning Research*, 12:2389–2410, 2011. page 32

O. Stein. How to solve a semi-infinite optimization problem. *European Journal of Operational Research*, 223(2):312–320, 2012. page 84

I. Steinwart and A. Christmann. *Support vector machines*. Springer Science & Business Media, 2008. pages 29, 33, 88

I. Steinwart and C. Scovel. Mercer's theorem on general domains: on the interaction between measures, kernels, and RKHSs. *Constructive Approximation*, 35(3):363–417, 2012. page 31

I. Steinwart, A. Christmann, et al. Estimating conditional quantiles with the help of the pinball loss. *Bernoulli*, 17(1):211–225, 2011. pages 18, 33

M. H. Stone. *Linear transformations in Hilbert space and their applications to analysis*, volume 15. American Mathematical Soc., 1932. page 31

J. M. Susskind, G. E. Hinton, J. R. Movellan, and A. K. Anderson. Generating facial expressions with deep belief nets. In *Affective Computing*, chapter 23. IntechOpen, 2008. page 118

J. A. Suykens and J. Vandewalle. Least squares support vector machine classifiers. *Neural processing letters*, 9(3):293–300, 1999. page 33

Z. Szabó and B. Sriperumbudur. Characteristic and universal tensor product kernels. *Journal of Machine Learning Research*, 18:233, 2018. page 32

C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016. page 105

I. Takeuchi, Q. V. Le, T. D. Sears, and A. J. Smola. Nonparametric quantile estimation. *Journal of Machine Learning Research*, 7:1231–1264, 2006. pages 84, 118

I. Takeuchi, T. Hongo, M. Sugiyama, and S. Nakajima. Parametric task learning. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1358–1366, 2013. pages 16, 20, 84, 87

F. Tao, H. Zhang, A. Liu, and A. Y. C. Nee. Digital twin in industry: State-of-the-art. *IEEE Transactions on Industrial Informatics*, 15(4):2405 – 2415, 2019. page 117

I. Tautkute, T. Trzciski, and A. Bielski. I know how you feel: Emotion recognition with facial landmarks. *Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1959–19592, 2018. pages 118, 119

R. Tibshirani. Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996. page 84

A. N. Tikhonov and V. Y. Arsenin. *Solution of Ill-posed Problems*. Winston & Sons, 1977. pages 17, 84

J. Ton, L. Chan, Y. W. Teh, and D. Sejdinovic. Noise contrastive meta-learning for conditional density estimation using kernel mean embeddings. In A. Banerjee and K. Fukumizu, editors, *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 1099–1107, 2021. page 132

S. Ullah and C. F. Finch. Applications of functional data analysis: A systematic review. *BMC medical research methodology*, 13(1):1–12, 2013. page 15

D. Ulyanov, V. Lebedev, A. Vedaldi, and V. Lempitsky. Texture networks: Feed-forward synthesis of textures and stylized images. In *International Conference on Machine Learning (ICML)*, pages 1349–1357, 2016. page 117

V. Vapnik. Principles of risk minimization for learning theory. In *Advances in Neural Information Processing Systems (NIPS)*, pages 831–838, 1992. page 88

V. Vapnik. An overview of statistical learning theory. *IEEE Trans. Neural Networks*, 10(5):988–999, 1999. page 16

V. N. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, 1998. pages 17, 18, 33

R. Vemulapalli and A. Agarwala. A compact embedding for facial expression similarity. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5683–5692, 2019. page 119

R. Vert and J.-P. Vert. Consistency and convergence rates of one-class SVMs and related algorithms. *Journal of Machine Learning Research*, 7:817–854, 2006. page 86

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 15, 65

U.-P. Wen and S.-T. Hsu. Linear bi-level programming problemsa review. *Journal of the Operational Research Society*, 42(2):125–133, 1991. page 84

L. Wenliang, D. J. Sutherland, H. Strathmann, and A. Gretton. Learning deep kernels for exponential family densities. *arXiv preprint arXiv:1811.08357*, 2018. page 35

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. page 33

A. G. Wilson, Z. Hu, R. Salakhutdinov, and E. P. Xing. Deep kernel learning. In *Artificial Intelligence and Statistics*, pages 370–378, 2016. page 35

D. Wynen, C. Schmid, and J. Mairal. Unsupervised learning of artistic styles with archetypal style analysis. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 6584–6593, 2018. page 117

J. Yang, V. Sindhwani, H. Avron, and M. Mahoney. Quasi-Monte Carlo feature maps for shift-invariant kernels. *International Conference on Machine Learning (ICML)*, pages 485–493, 2014. page 34

Z. Yang, M. Moczulski, M. Denil, N. De Freitas, A. Smola, L. Song, and Z. Wang. Deep fried convnets. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1476–1483, 2015. page 35

X. Yao, G. Puy, A. Newson, Y. Gousseau, and P. Hellier. High resolution face age editing. In *International Conference on Pattern Recognition (ICPR)*, 2020. page 117

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 34

Z. Zhang, L. Wang, Q. Zhu, S.-K. Chen, and Y. Chen. Pose-invariant face recognition using facial landmarks and Weber local descriptor. *Knowledge-Based Systems*, 84: 78–88, 2015. page 119

D.-X. Zhou. The covering number in learning theory. *Journal of Complexity*, 18(3): 739–767, 2002. page 88

D.-X. Zhou. Derivative reproducing properties for kernel methods in learning theory. *Journal of Computational and Applied Mathematics*, 220:456–463, 2008. page 99

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 (TOMS)*, 23(4):550–560, 1997. pages 54, 101

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 66

J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *International Conference on Computer Vision (ICCV)*, pages 2223–2232, 2017. page 118

W. P. Ziemer. *Weakly differentiable functions: Sobolev spaces and functions of bounded variation*, volume 120. Springer Science & Business Media, 2012. page 99

ECOLE DOCTORALE

Titre : Apprentissage de Fonctions à Valeurs Fonctionnelles dans des Espaces de Hilbert à Noyaux Auto-reproduisants avec Pertes Intégrales : Application à l'Apprentissage d'un Continuum de Tâches

Mots clés : noyaux à valeurs opérateurs, pertes intégrales, dualité lagrangienne, regression fonctionnelle, regression quantile, transfert d'émotion

Résumé : Les méthodes à noyaux sont au coeur de l'apprentissage statistique. Elles permettent de modéliser des fonctions à valeurs réelles dans des espaces de fonctions à fort potentiel représentatif, sur lesquels la minimisation de risques empiriques régularisés est possible et produit des estimateurs dont le comportement statistique est largement étudié. Lorsque les sorties ne sont plus réelles mais de plus grande dimension, les Espaces de Hilbert à Noyaux Reproduisants à valeurs vectorielles (vv-RKHSs) basés sur des Noyaux à Valeurs Opérateurs (OVKs) fournissent des espaces de fonctions similaires et permettent de traiter des problèmes tels que l'apprentissage multi-tâche, la prédiction structurée ou la regression à valeurs fonctionnelles.
Dans cette thèse, nous introduisons une extension fonctionnelle originale du cadre multi-tâche appelée *Apprentissage d'un Continuum de Tâches* (ITL), qui permet de résoudre conjointement un continuum de tâches paramétrées, parmi lesquelles la régression quantile, la classification à coût assymétrique, ou l'estimation de niveaux de densité.
Nous proposons un cadre d'apprentissage basé sur des fonctions de pertes intégrales qui comprend à la fois l'ITL et la régression à valeurs fonctionnelles, ainsi que des méthodes d'optimisation pour résoudre les problèmes de minimisation de risque empirique régularisé résultants. Par un échantillonnage des pertes intégrales, nous obtenons une représentation de dimension finie des solutions pour différents choix de régularisation ou pénalités liées à la forme des fonctions, tout en gardant un contrôle théorique sur les capacités en généralisation des estimateurs. L'usage de la dualité lagrangienne vient approfondir ces méthodes, en apportant en particulier les moyens d'imposer des estimateurs parcimonieux ou robustes à l'aide de pertes convoluées. Les problèmes de passages à l'échelle sont traités par l'utilisation noyaux approchés, dont les vv-RKHSs associés sont de dimension finie. Nous proposons aussi une architecture composée d'un réseau de neurone et d'une dernière couche à noyaux, qui permet l'apprentissage de représentations appropriées aux noyaux utiles dans les applications avec des données complexes comme les images.
Ces techniques sont appliquées à plusieurs problèmes d'ITL, ainsi qu'au problème de regression fonction-à-fonction robuste en présence de valeurs aberrantes. Enfin, nous revisitons les problemes de transfert de style sous l'angle ITL, avec une application au transfert d'émotion.

Title : Learning Function-Valued Functions in Reproducible Kernel Hilbert Spaces with Integral Losses : Application to Infinite Task Learning

Keywords : operator-valued kernels, integral losses, lagrangian duality, functional regression, quantile regression, emotion transfer

Abstract : Kernel methods are regarded as a cornerstone of machine learning. They allow to model real-valued functions in expressive functional spaces, over which regularized empirical risk minimization problems are amenable to optimization and yield estimators whose statistical behavior is well studied. When the outputs are not reals but higher dimensional, vector-valued Reproducible Kernel Hilbert Spaces (vv-RKHSs) based on Operator-Valued Kernels (OVKs) provide similarly powerful spaces of functions, and have proven useful to tackle problems such as multi-task learning, structured prediction, or function-valued regression.
In this thesis, we introduce an original functional extension of multi-output learning called *Infinite Task Learning* (ITL), that allows to jointly solve an infinite number of parameterized tasks, including for instance quantile regression, cost-sensitive classification and density level set estimation.
We propose a learning framework based on convex integral losses that encompasses the ITL problem and function-valued regression. Optimization schemes dedicated to solving the associated regularized empirical risk minimization problems are designed. By sampling the integral losses, we derive finite-dimensional representation of the solution under several choices of regularizers or shape constraints penalties, while keeping theoretical guarantees over their generalization capabilities. We also employ dualization techniques with the benefit of bringing desirable properties such as robustness or sparsity to the estimators thanks to the use of convoluted losses. Scalability issues are addressed by deriving optimization algorithms in the the context of approximated OVKs whose corresponding vv-RKHSs are of finite dimension. The use of trainable deep architectures composed by a neural network followed by a shallow kernel layer is also investigated as a way to learn the kernel used in practice on complex data such as images.
We apply these techniques to various ITL problems and to robust function-to-function regression, that are tackled in the presence of outliers. We also cast style transfer problems as a vectorial output ITL problem and demonstrate its efficiency in emotion transfer.

Institut Polytechnique de Paris
91120 Palaiseau, France