INSTITUT POLYTECHNIQUE DE PARIS

NNT : 2020IPPAT006

Thèse de doctorat

Deep Kernel Representation Learning for Complex Data and Reliability Issues

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

École Doctorale de l'Institut Polytechnique de Paris
École doctorale n°626 (ED IP Paris)
Spécialité de doctorat : Mathématiques et Informatique

Thèse présentée et soutenue à Palaiseau, le 26 Juin 2020, par

PIERRE LAFORGUE

Composition du Jury :

Thomas Bonald Professeur, Télécom Paris	Président
Gábor Lugosi Professeur, Université Pompeu-Fabra	Rapporteur
Jean-Philippe Vert Professeur/Chercheur, Mines ParisTech/Google Brain	Rapporteur
Hachem Kadri Professeur, Université Aix-Marseille	Examinateur
Julien Mairal Chercheur, Inria Grenoble	Examinateur
Florence d'Alché-Buc Professeure, Télécom Paris	Directrice de thèse
Stephan Clémençon Professeur, Télécom Paris	Co-directeur de thèse

Contents

Remerciements

Je tiens ici à remercier toutes les personnes sans qui ce manuscrit n'aurait pu voir le jour, du moins sous sa forme actuelle.

Mes premiers remerciements vont à mes deux directeurs, à Florence, pour m'avoir guidé tout au long de ces trois (et quelques) années à travers un univers aussi riche que passionnant, et dont explorer les idées aura toujours été un plaisir, et à Stephan, pour sa complémentarité qui m'aura permis d'étudier bien plus de sujets que je n'aurai osé l'imaginer en débutant cette thèse. J'aurai beaucoup appris durant ces années de doctorat, et je vous le dois en grande partie.

I also thank Gábor Lugosi and Jean-Philippe Vert for having reviewed my manuscript, Thomas Bonald for having accepted to serve as a physically present president of the jury in these complex sanitary times, as well as Hachem Kadri and Julien Mairal for being be part of my jury.

Merci à Alex et Lucho, cela a été un réel plaisir de travailler avec vous, et j'ose penser que ce fut réciproque. Je vous souhaite tout le meilleur pour la suite. Merci également à Olivier le sorcier pour sa disponibilité et son aide précieuse.

Merci à Mathurin, sans qui ma thèse n'aurait assurément pas eu la même saveur, de Barceloneta en meilleure pizza, de Kamikaze en cordon d'hôtel *emprunté*, les années passent trop vite quand on est bien entouré, à Pierre, jamais à court de salive lorsqu'il s'agit du passage à l'échelle des méthodes à noyaux, et à la Gazelle Gana: c'est vraiment très appréciable un co-bureau qui choisit de lui même la pire chaise, le plus petit bureau, et la place avec le soleil dans les yeux!

Mreci à Kveni le Bon Marcheur, dont les problèmes de mémoire trouvenront une solutino j'en suis sûr, au de bon air Quentin et à son pendant maléfique, le plein de contentieux Charles, à Simon en souvenir de notre petite virée australe, à Alexandre et sa bonne foi au baby, à Mastane et Robin, avec qui je partage mes débuts, et dont j'attends dorénavant les soutenances avec impatience.

Merci aux vieux grognards: Albert (continue, ça fait toujours plaisir de se faire appeler *les jeunes*), Guillaume, si vieux qu'il prétend avoir vu de son vivant son Olympique triompher de la capitale, Anna, dont pour la première fois depuis bien longtemps je ne devrais pas suivre les pas, Adil et ses coupes de cheveux fantaisistes, Raymondinho le plombier de ces dames, et Paul de Télécom avant l'heure, mon sauveteur de M2.

Merci et bon courage à la nouvelle vague: Rico de Porto Novo, Hamid, Guillaume, Anas, Emile, Pierre, Nidham, Kimia, Dimitri, Rémi, allez, Palaiseau c'est pas si loin!

Merci à tous d'avoir fait de chaque jour à Télécom un moment agréable à partager à vos côtés.

À mes parents, qui m'auront toujours encouragé à faire ce que j'aimais, sans jamais me soucier du reste.

À Maud, dont la patience face aux (trop nombreuses) nuits de travail saura je l'espère trouver une juste récompense à travers ce manuscrit et cette soutenance.

Abstract

The first part of this thesis aims at exploring deep kernel architectures for complex data. One of the known keys to the success of deep learning algorithms is the ability of neural networks to extract meaningful internal representations. However, the theoretical understanding of why these compositional architectures are so successful remains limited, and deep approaches are almost restricted to vectorial data. On the other hand, kernel methods provide with functional spaces whose geometry are well studied and understood. Their complexity can be easily controlled, by the choice of kernel or penalization. In addition, vector-valued kernel methods can be used to predict kernelized data. It then allows to make predictions in complex structured spaces, as soon as a kernel can be defined on it.

The deep kernel architecture we propose consists in replacing the basic neural mappings by functions from vector-valued Reproducing Kernel Hilbert Spaces (vv-RKHSs). Although very different at first glance, the two functional spaces are actually very similar, and differ only by the order in which linear/nonlinear functions are applied. Apart from gaining understanding and theoretical control on layers, considering kernel mappings allows for dealing with structured data, both in input and output, broadening the applicability scope of networks. We finally expose works that ensure a finite dimensional parametrization of the model, opening the door to efficient optimization procedures for a wide range of losses.

The second part of this thesis investigates alternatives to the sample mean as substitutes to the expectation in the Empirical Risk Minimization (ERM) paradigm. Indeed, ERM implicitly assumes that the empirical mean is a good estimate of the expectation. However, in many practical use cases (e.g. heavy-tailed distribution, presence of outliers, biased training data), this is not the case.

The Median-of-Means (MoM) is a robust mean estimator constructed as follows: the original dataset is split into disjoint blocks, empirical means on each block are computed, and the median of these means is finally returned. We propose two extensions of MoM, both to randomized blocks and/or U-statistics, with provable guarantees. By construction, MoM-like estimators exhibit interesting robustness properties. This is further exploited by the design of robust learning strategies. The (randomized) MoM minimizers are shown to be robust to outliers, while MoM tournament procedure are extended to the pairwise setting.

We close this thesis by proposing an ERM procedure tailored to the sample bias issue. If training data comes from several biased samples, computing blindly the empirical mean yields a biased estimate of the risk. Alternatively, from the knowledge of the biasing functions, it is possible to reweight observations so as to build an unbiased estimate of the test distribution. We have then derived non-asymptotic guarantees for the minimizers of the debiased risk estimate thus created. The soundness of the approach is also empirically endorsed.

Résumé

Cette thèse débute par l'étude d'architectures profondes à noyaux pour les données complexes. L'une des clefs du succès des algorithmes d'apprentissage profond est la capacité des réseaux de neurones à extraire des représentations pertinentes. Cependant, les raisons théoriques de ce succès nous sont encore largement inconnues, et ces approches sont presque exclusivement réservées aux données vectorielles. D'autre part, les méthodes à noyaux engendrent des espaces fonctionnels étudiés de longue date, les Espaces de Hilbert à Noyau Reproduisant (Reproducing Kernel Hilbert Spaces, RKHSs), dont la complexité est facilement contrôlée par le noyau ou la pénalisation, tout en autorisant les prédictions dans les espaces structurés complexes via les RKHSs à valeurs vectorielles (vv-RKHSs).

L'architecture proposée consiste à remplacer les blocs élémentaires des réseaux usuels par des fonctions appartenant à des vv-RKHSs. Bien que très différents à première vue, les espaces fonctionnels ainsi définis sont en réalité très similaires, ne différant que par l'ordre dans lequel les fonctions linéaires/non-linéaires sont appliquées. En plus du contrôle théorique sur les couches, considérer des fonctions à noyau permet de traiter des données structurées, en entrée comme en sortie, étendant le champ d'application des réseaux aux données complexes. Nous conclurons cette partie en montrant que ces architectures admettent la plupart du temps une paramétrisation finie-dimensionnelle, ouvrant la voie à des méthodes d'optimisation efficaces pour une large gamme de fonctions de perte.

La seconde partie de cette thèse étudie des alternatives à la moyenne empirique comme substitut de l'espérance dans le cadre de la Minimisation du Risque Empirique (Empirical Risk Minimization, ERM). En effet, l'ERM suppose de manière implicite que la moyenne empirique est un bon estimateur. Cependant, dans de nombreux cas pratiques (e.g. données à queue lourde, présence d'anomalies, biais de sélection), ce n'est pas le cas.

La Médiane-des-Moyennes (Median-of-Means, MoM) est un estimateur robuste de l'espérance construit comme suit: des moyennes empiriques sont calculées sur des sous-échantillons disjoints de l'échantillon initial, puis est choisie la médiane de ces moyennes. Nous proposons et analysons deux extensions de MoM, via des sous-échantillons aléatoires et/ou pour les U-statistiques. Par construction, les estimateurs MoM présentent des propriétés de robustesse, qui sont exploitées plus avant pour la construction de méthodes d'apprentissage robustes. Il est ainsi prouvé que la minimisation d'un estimateur MoM (aléatoire) est robuste aux anomalies, tandis que les méthodes de tournoi MoM sont étendues au cas de l'apprentissage sur les paires.

Enfin, nous proposons une méthode d'apprentissage permettant de résister au biais de sélection. Si les données d'entraînement proviennent d'échantillons biaisés, la connaissance des fonctions de biais permet une repondération non-triviale des observations, afin de construire un estimateur non biaisé du risque. Nous avons alors démontré des garanties non-asymptotiques vérifiées par les minimiseurs de ce dernier, tout en supportant empiriquement l'analyse.

Notation

$\mathcal{X}$	Input space
$\mathcal{Y}$	Output space
$\mathcal{Y}^{\mathcal{X}} = \mathcal{F}(\mathcal{X}, \mathcal{Y})$	Set of applications from $\mathcal{X}$ to $\mathcal{Y}$
$\mathcal{L}(\mathcal{Y})$	Bounded linear operators on $\mathcal{Y}$
$k \colon \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$	Output scalar kernel
$\mathcal{K} \colon \mathcal{X} \times \mathcal{X} \to \mathcal{L}(\mathcal{Y})$	Operator-valued kernel
$\mathcal{H}_{\mathcal{K}} \subset \mathcal{Y}^{\mathcal{X}}$	Vector-valued RKHS associated to $\mathcal{K}$
$\mathbf{I}_{\mathcal{Y}}$	Identity operator on $\mathcal{Y}$
$\mathbf{I}_n$	Identity matrix of size n
$\mathbb{R}^{n \times p}$	Set of matrices of size n by p
$\mathbf{Tr}(A)$	Trace of operator or matrix A
A^*	Adjoint of operator $A \in \mathcal{L}(\mathcal{Y})$
$A^\top$	Transpose of matrix A
$A_{i:}$	ith line of matrix A
$\|A\|_{p,q}$	$\ell_{p,q}$ row wise mixed norm: ℓ_q norm of the lines ℓ_p norms
$f^\star$	Fenchel-Legendre conjugate of function f
$f \,\square\, g$	Infimal convolution of functions f and g
χ_S	Characteristic function of set S: null on S, $+\infty$ otherwise
$\mathbb{1}\{\cdot\}$	Indicator of an event
$\mathbb{P}\{\cdot\}$	Probability of an event
$\mathbb{E}[\cdot]$	Expectation of an event

For any loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$, any predictor $h \in \mathcal{Y}^{\mathcal{X}}$ and any labeled observation $z = (x, y) \in \mathcal{Z} = \mathcal{X} \times \mathcal{Y}$, the shortcut notation $\ell(h, z)$ may be used instead of $\ell(h(x), y)$.

1
Motivation and Contribution

Contents

In its generic form, supervised Machine Learning can be seen as the task of inferring, from a set of examples, the relationship that might exist between some explanatory variables, also called *features*, and a target output, often referred to as *label*. Algorithms are computational procedures that take as input a sample of observations, the *training dataset*, and return a relationship, or *prediction function*. If a distinction could be made, Machine Learning would be an algorithm-oriented study, while Statistical Learning focuses on theoretical guarantees and statistical aspects.

From an algorithmic point of view, the first dominant approach in Machine Learning, popularized during the 1990s by Support Vector Machines (Cortes and Vapnik, 1995), has been that of kernel methods. Apart from providing the best empirical results at the time (see *e.g.* MNIST database), kernel methods are grounded on solid mathematical foundations (Aronszajn, 1950), and supported by strong arguments such as margin theory (Vapnik, 1998). Another advantage of kernel methods is their ability to deal with complex structured data, ranging from graphs (Mahé and Vert, 2009), to trees and time series (Cuturi et al., 2007). This is made possible by the use of the *kernel trick* in the input space, ensuring that only the knowledge of the kernel evaluations is required to learn kernel machines. Notice that recent works by Brouard et al. (2016b) have introduced the use of the kernel trick in the *output* space. Up to an inverse problem resolution, it allows to handle structured data in output as well. This remark turns out to be crucial in the next chapters. But the popularity of kernel methods may also come from the natural tuning parameters the practitioner can leverage to control the class complexity: choices of kernel and penalization. In particular, choosing the kernel results in the choice of a (possibly implicit) data representation, completely decoupled from the learning stage. If this was considered as a strength until recently, the advances made by Deep Learning approaches tend to question this idea.

Indeed, with the recent developments of computational power, neural networks produce nowadays state-of-the-art algorithms for numerous tasks (*e.g.* ImageNet challenge). In opposition to kernel methods, our theoretical understanding of why these architectures are so successful remains nonetheless limited. One known, if not fully understood, reason to this success is the capacity of neural networks to automatically extract meaningful representations, along the training process itself (Erhan et al., 2009). This paradigm breaks with the kernel vision, and makes Representation Learning a central issue. Its

scope is however constrained by the intrinsic nature of neural nets, as the operations they perform on inputs mainly apply to vectorial data (*e.g.* convolution). An important part of this manuscript thus investigates the data representation question, by adopting an intermediary position that leverages advantages from both deep and kernel methods.

From a theoretical perspective, a vast majority of algorithms select a prediction function by minimizing the errors its predictions incur on the training dataset. If sufficiently many examples are available, one may hope that good prediction functions during the training stage will also provide accurate predictions on new test datapoints following the same law. Assessing this *generalization capacity* is one of the major issues Statistical Learning tries to address. In that respect, solutions to Empirical Risk Minimization (ERM, *e.g.* Devroye et al. (1996b)) are usually studied under suitable class complexity assumptions by means of concentration inequalities for empirical processes (Boucheron et al., 2013). Nevertheless, the more and more complex tasks addressed by Machine Learning make vanilla analyses often inadequate. The second common thread of this manuscript can be seen as adapting these guarantees to three unfriendly situations encountered in practice: compositional architectures with infinite dimensional outputs, outliers in the training set and heavy-tailed data, presence of sample bias.

As a first go, we shall make the above discussion a bit more formal, and briefly recall in Section 1.1 the theoretical framework of Statistical Learning this manuscript builds upon. Despite tremendous successes, using off-the-shelf Machine Learning algorithms as black boxes, what is now possible thanks to libraries such as scikit-learn (Pedregosa et al., 2011), often yields deceptive results in practice. Indeed, in many real-world applications obstacles arise, that jeopardize standard approaches and analyses. Three of them, acting as motivations, are exposed in Section 1.2. Section 1.3 is then devoted to the contributions we have developed to address these issues, with a list of publications resulting from this work. Section 1.4 finally details the organization of the manuscript.

1.1 Statistical Learning

Let $Z = (X, Y)$ be a random variable valued in a space $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ with unknown probability distribution P. Here, Y represents some target (*e.g.* a class, a real value), and X some features supposedly useful to predict Y. The general goal of supervised Machine Learning is to recover from realizations of Z the relationship that might exist between X and Y. Let $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ be some loss function defining a discrepancy on space $\mathcal{Y}$. The generic ideal supervised problem then consists in finding

$$h^* \in \underset{h \text{ measurable}}{\operatorname{argmin}} \quad \mathcal{R}(h) = \mathbb{E}_P \left[\ell \Big(h(X), Y \Big) \right].$$

In the particular case of *binary classification*, $\mathcal{Y} = \{-1, +1\}$, $\ell(y, y') = \mathbb{1}\{y \neq y'\}$ and h^* is trivially given by the so called *Bayes classifier*

$$h^*(x) = 2 \cdot \mathbb{1}\Big\{ \eta(x) \geq 1/2 \Big\} - 1,$$

with $\eta(x) = \mathbb{P}\{Y = 1 \mid X = x\}$ the posterior probability. However, the latter requires the knowledge of P, that we cannot access in practice. Empirical Risk Minimization (ERM) consists in replacing the unknown expectation by an empirical average computed on a sample $\mathcal{S}_n = \{z_i = (x_i, y_i)\}_{i \leq n}$ independent identically distributed (i.i.d.) as Z. Moreover, optimization on the whole set of measurable functions is often impossible in practice, and one has to restrict the search domain to a so called *hypothesis set* $\mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$.

The problem then consists in finding

$$\hat{h}_n \in \operatorname*{argmin}_{h \in \mathcal{H}} \ \widehat{\mathcal{R}}_n(h) = \frac{1}{n} \sum_{i=1}^{n} \ell\left(h(x_i), y_i\right).$$

While the first part of this manuscript focuses on a specific hypothesis set (namely the composition of functions from vector-valued Reproducing Kernel Hilbert Spaces), the second part explores alternatives to the empirical mean as substitutes to the intractable expectation.

The performance of any predictor h is often assessed through the *excess risk*

$$\mathcal{E}(h) = \mathcal{R}(h) - \mathcal{R}(h^*).$$

The excess risk of $\hat{h}_n$ can be decomposed into an *approximation error*, characterizing how far the best solution is from the hypothesis set $\mathcal{H}$, and an *estimation error*, due to the difference between the empirical average minimized and the true expected value:

$$\mathcal{E}(\hat{h}_n) = \underbrace{\mathcal{R}(\hat{h}_n) - \mathcal{R}(h^*_{\mathcal{H}})}_{\text{estimation error}} + \underbrace{\mathcal{R}(h^*_{\mathcal{H}}) - \mathcal{R}(h^*)}_{\text{approximation error}},$$

with $h^*_{\mathcal{H}}$ the minimizer of $\mathcal{R}(h)$ on $\mathcal{H}$. Once $\mathcal{H}$ is fixed, the approximation error is usually considered as given and part of the problem. From now on, we thus drop the notation $h^*_{\mathcal{H}}$ for h^*. More focus is put on the estimation error, that can be bounded as follows

$$\begin{aligned}\mathcal{R}(\hat{h}_n) - \mathcal{R}(h^*) &= \mathcal{R}(\hat{h}_n) - \widehat{\mathcal{R}}_n(\hat{h}_n) \ + \ \widehat{\mathcal{R}}_n(\hat{h}_n) - \widehat{\mathcal{R}}_n(h^*) \ + \ \widehat{\mathcal{R}}_n(h^*) - \mathcal{R}(h^*), \\ &\leq 2 \sup_{h \in \mathcal{H}} \left| \widehat{\mathcal{R}}_n(h) - \mathcal{R}(h) \right|,\end{aligned}$$

using that $\hat{h}_n$ minimizes $\widehat{\mathcal{R}}_n(h)$ on $\mathcal{H}$. It is then enough to control the empirical process $\sup_{h \in \mathcal{H}} \ \widehat{\mathcal{R}}_n(h) - \mathcal{R}(h)$. Assuming that $\sup_{y,y'} \ell(y, y') \leq 1$, the Bounded Differences inequality (Appendix A) yields that it holds with probability $1 - \delta$

$$\sup_{h \in \mathcal{H}} \ \widehat{\mathcal{R}}_n(h) - \mathcal{R}(h) \leq \mathbb{E}_{\mathcal{S}_n} \left[\sup_{h \in \mathcal{H}} \ \widehat{\mathcal{R}}_n(h) - \mathcal{R}(h) \right] + \sqrt{\frac{\ln \frac{1}{\delta}}{2n}}.$$

Classical symmetrization arguments then allows to control the right hand side in terms of *Rademacher averages* (Bartlett and Mendelson, 2002):

$$\begin{aligned}\mathbb{E}_{\mathcal{S}_n} \left[\sup_{h \in \mathcal{H}} \ \widehat{\mathcal{R}}_n(h) - \mathcal{R}(h) \right] &= \mathbb{E}_{\mathcal{S}_n} \left[\sup_{h \in \mathcal{H}} \ \widehat{\mathcal{R}}_n(h) - \mathbb{E}_{\mathcal{S}'_n} \left[\widehat{\mathcal{R}}'_n(h) \right] \right], \\ &\leq \mathbb{E}_{\mathcal{S}_n, \mathcal{S}'_n} \left[\sup_{h \in \mathcal{H}} \ \frac{1}{n} \sum_{i=1}^{n} \ell(h, z_i) - \ell(h, z'_i) \right], \\ &\leq \mathbb{E}_{\mathcal{S}_n, \mathcal{S}'_n, \boldsymbol{\sigma}} \left[\sup_{h \in \mathcal{H}} \ \frac{1}{n} \sum_{i=1}^{n} \sigma_i \left(\ell(h, z_i) - \ell(h, z'_i) \right) \right], \\ &\leq 2 \mathbb{E}_{\mathcal{S}_n, \boldsymbol{\sigma}} \left[\sup_{h \in \mathcal{H}} \ \frac{1}{n} \sum_{i=1}^{n} \sigma_i \ell(h, z_i) \right],\end{aligned}$$

where we have used successively the introduction of a phantom sample $\mathcal{S}_n'$ independent from $\mathcal{S}_n$ and identically distributed, Jensen's inequality, the introduction of n i.i.d. Rademacher random variable $(\sigma_i)_{i\leq n}$ such that $\mathbb{P}\left\{\sigma_i = 1\right\} = \mathbb{P}\left\{\sigma_i = -1\right\} = 1/2$, and the notation abuse $\ell(h, z)$ for $\ell(h(x), y)$, that is utilized throughout this manuscript.

Definition 1.1. *Let $\mathcal{H} \subset \mathbb{R}^{\mathcal{Z}}$ be a hypothesis set, and $\mathcal{S}_n = \{z_i = (x_i, y_i)\}_{i\leq n} \in \mathcal{Z}^n$ a fixed sample of size n. The* empirical Rademacher complexity *of class $\mathcal{H}$ is defined as*

$$\widehat{\mathscr{R}}(\mathcal{H}, \mathcal{S}_n) = \mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{h\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^{n} \sigma_i h(z_i)\right],$$

with $(\sigma_i)_{i\leq n}$ n i.i.d. Rademacher random variables. The Rademacher complexity *of class $\mathcal{H}$ of size n and according to distribution P is defined as*

$$\mathscr{R}_n(\mathcal{H}) = \mathbb{E}\left[\widehat{\mathscr{R}}(\mathcal{H}, \mathcal{S}_n)\right] = \mathbb{E}_{\mathcal{S}_n, \boldsymbol{\sigma}}\left[\sup_{h\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^{n} \sigma_i h(z_i)\right].$$

Intuitively, Rademacher averages can be seen as an evaluation of the class capacity to fit random noise. Notice that using the Bounded Differences inequality again allows to switch from Rademacher complexities to their empirical versions.

Rademacher averages are standard complexity measures, and the analyses produced in the present manuscript heavily build upon them. In Part I, they are extended to infinite dimensional outputs for a specific choice of function class. In Part II, they are adapted to the presence of outliers, and help controlling a Median-of-Means empirical process.

Remark 1.2. *It seems important to recall that the framework described so far is that of* supervised *Statistical Learning.* Unsupervised *Learning happens when the random variable of interest does not contain any target. One may be interested then in building homogeneous groups of observations (clustering), or simply inferring the underlying distribution. However, the lack of possible comparison often makes it complex to assess the quality of answers given to an Unsupervised Learning problem.*

We next move to three concrete problems encountered in practice, which motivated the subsequent works.

1.2 Motivations

In practice, many problems arise, that may downgrade the performance of Machine Learning approaches if they are not addressed correctly. We now illustrate three of them on practical examples

Diabetes Occurrence Prediction. Assume a binary classification setting, where one has to predict if an individual will develop diabetes or not. Without prior work, the observations (*i.e.* the patients) are described by a large number of variables, also called *features*. This may include physiological data, census data, or any other descriptor. Learning a predictive function from these raw data is very unlikely to work, as the potentially explanatory variables are hidden among a large number of irrelevant ones. One alternative consists in asking a physician for insights. He or she will help selecting *good* features, such as age, gender, or family background, that are known from clinical

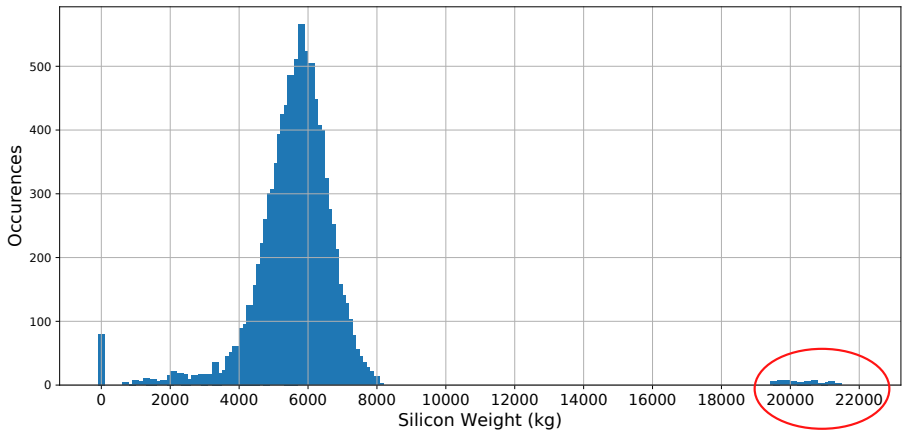


Figure 1.1 – Distribution of the Silicon Daily Production

observation to influence diabetes occurrence. Even more critical, he or she will probably advise to consider a new feature, the *Body Mass Index* (BMI = *weight* / *size*^{2}), that has been shown empirically to be highly correlated to the development of type 2 diabetes (Dubois-Laforgue et al., 2000).

This example is very instructive, as both discarding irrelevant features, also known as *feature selection*, or creating new ones (*feature engineering*) are two of the key goals of Representation Learning. This recent Machine Learning domain (see Chapter 3, Section 3.1.1) indeed aims at extracting relevant representations from raw data in an unsupervised fashion (supervision might though be added to tailor the representation to a specific purpose). If the difficulty of no supervision has been overcome through the design of *self-supervised* criteria such as Autoencoders (Chapter 3, Section 3.1.2), the latter model only applies to vectorial data. On the other hand, there is a critical need in chemoinformatics to learn molecule representations (Matsuda et al., 2005). Indeed, currently available feature vectors are either low-informative (long and sparse fingerprints) or too complex (labeled graphs mimicking the molecule structure). These data are thus traditionally handled through Kernel Methods (Chapter 2), and do not benefit from the recent algorithmic advances, mostly devoted to vectorial inputs. The first part of the present manuscript is then dedicated to bridging this gap, and tries to offer an alternative benefiting both from the relevance of Kernel Methods and deep architectures to solve the *Structured Representation Learning* problem.

Silicon Production. The second example is taken from my personal monitoring experience as a scientific advisor for master's students. These students were working jointly with Ferroglobe, an electrometallurgical company specialized in silicon metal production. Their goal was to infer the working process of a silicon furnace, in order to maximize the silicon production while minimizing electricity consumption. The data at disposal featured the daily silicon production, reproduced in Figure 1.1. Knowing that productions were manually entered in the database, observations around 20 tons a day may reasonably be considered as outliers. Yet, the furnace works in a cyclical fashion, and removing these points would result in removing the whole cycles they pertain to, potentially harming the predictive functions learned on a drastically reduced dataset.

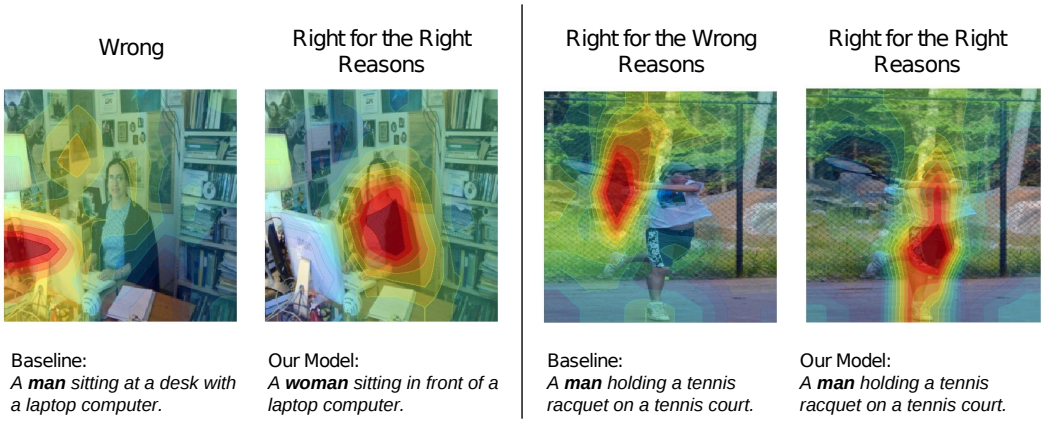


Figure 1.2 – Gender-Specific Image Captioning (from Burns et al. (2018))

Annotation mistakes, and more generally outliers, constitute a crucial issue for Machine Learning practitioners. Ignoring them, and computing standard empirical means as in Section 1.1, is likely to generate many mistakes, as a small number of very atypical values like that of Figure 1.1 heavily shifts optimal decision functions. Discarding them seems not a viable option neither, due to the loss of information induced. This is all the more true for U-statistics (Chapter 6), that compare tuples of observations: one corrupted observation contaminates all tuples it appears in. While Median-of-Means approaches have been developed to address these issues for learning criteria writing as simple means (Lecué et al., 2018), fewer efforts have been made to design robust and computationally efficient pairwise learning strategies. Part II of this manuscript thus presents an attempt to do so, via the randomization of the original Median-of-Means (MoM) estimator.

Women also Snowboard. The third and last motivating example is taken from Burns et al. (2018). The goal pursued here is to predict the gender of a person from a picture of him/her in context. What has been observed empirically is that parts of the image motivating the algorithm's decision do not describe intrinsically the person, but rather rely on objects in the scene (see Figure 1.2). This results in predicting no "woman" caption for snowboard scenes, as a vast majority of training images containing a snowboard are labeled as "man". In the context of decision making and socially impacting algorithms, it is central to deal with such bias issues.

The fact that the test distribution may differ, to some extent, from the training one (*e.g.* no woman seen snowboarding during the train phase) is known as *dataset shift* (Quionero-Candela et al., 2009). Its study is of considerable interest for the application of Machine Learning, as assuming the distribution to be invariable is often unrealistic in practice. Dataset shift can be due to a wide variety of causes (Heckman, 1990), and previous contributions in the literature usually propose *ad hoc* solutions for specific problems, or concentrate on *covariate shift*, a simple case of dataset shift, where the conditional distribution remains unchanged. The work presented in Chapter 9 rather aims at proposing a general ERM framework to address the more global *sample bias* issue with theoretical guarantees.

1.3 Contributions and Publications

We now list the contributions we have developed to address the issues previously raised.

About *structured representation learning*:

- Introduction of Kernel Autoencoders (KAEs) that combine deep architectures and operator-valued kernels (OVKs) to allow autoencoding on complex structured data
- Kernel Autoencoders as deep versions of Kernel Principal Component Analysis
- Generalization bound tailored to infinite dimensional outputs
- Representer Theorem devoted to Kernel Autoencoders' compositional architecture
- Optimization strategy coupling Gradient Descent (GD) and Kernel Ridge updates
- Python package for Deep Input Output Kernel Regression (encompasses KAEs)

About *operator-valued kernel machines*:

- Double Representer Theorem to solve OVK dual problems under mild assumptions
- New loss functions unlocked for OVK machines with infinite outputs

About *robustness to heavy-tailed data and outliers*:

- Extension of the MoM estimator to randomized blocks, with guarantees
- Extension of the MoM estimator to (randomized) U-statistics, with guarantees
- Adaptation of MoM minimizers and MoM GD to randomized / U-statistic versions
- Extension of tournament procedures to the case of pairwise learning

About *robustness to selection bias*:

- Asymptotic guarantees for the density estimation problem in a general sampling bias framework made non-asymptotic
- Plugging density estimation into a risk minimization problem to derive a general debiased ERM framework, with guarantees extended to the minimizers
- Python package to compute the debiasing weights from the biasing functions

These contributions have resulted in several accepted publications and preprints, that are presented here in chronological order (* indicates equal contribution).

- **P. Laforgue**, S. Clémençon, F. d'Alché-Buc. Autoencoding any data through kernel autoencoders. In *Proceedings of Artificial Intelligence and Statistics*, 2019.
- **P. Laforgue**, S. Clémençon, P. Bertail. On medians of (Randomized) pairwise means. In *Proceedings of International Conference on Machine Learning*, 2019.
- **P. Laforgue**, S. Clémençon. Statistical learning from biased training samples. *arXiv preprint arXiv:1906.12304*, 2019.
- **P. Laforgue***, A. Lambert*, L. Brogat-Motte, F. d'Alché-Buc. On the dualization of operator-valued kernel machines. *arXiv preprint arXiv:1910.04621*, 2019.

1.4 Manuscript Organization

The rest of the manuscript is organized as follows. Chapters 2 and 6 recall basic notions related to Parts I and II, while contributions are exposed in Chapters 3 to 5 and 7 to 9.

Part I is devoted to the analysis of deep kernel architectures for complex data.

- Chapter 2 gathers reminders about scalar kernel methods and their vector-valued extensions. The latter indeed provide functional spaces whose geometry is well understood, and that can be controlled either by the choice of kernel or the norm penalization. More importantly, they allow for predicting complex structured objects through the kernelization of the output. Functions from vector-valued Reproducing Kernel Hilbert Spaces (vv-RKHSs) are the building blocks for the architectures developed in the next chapter.
- Chapter 3 is dedicated to the introduction and analysis of Kernel Autoencoders. Inspired from standard Autoencoders, Kernel Autoencoders are compositions of functions from vv-RKHSs, that compress and reconstruct the inputs. The ability of functions from vv-RKHSs to handle infinite dimensional inputs and outputs allows to extend the autoencoding scheme to all types of data, by first mapping them through a canonical feature map. A clear connection to Kernel Principal Component Analysis is established, as well as a generalization bound in terms of reconstruction error. This architecture however shall not be limited to the task of autoencoding. Deep kernel machines, for which inputs and outputs differ, benefit from the same guarantees, opening many applications ranging from structured prediction to the learning of output embeddings.
- Chapter 4 also deals with Kernel Autoencoders, but from an optimization point of view. Despite the non-convexity of the criterion, a Representer Theorem dedicated to the composition architecture ensures a finite expansion for every layer. When all internal spaces are finite dimensional, the problem is fully characterized by a finite dimensional parameter and a Gradient Descent strategy can be applied in a straightforward manner. When the output space is infinite dimensional, it is proved that the gradient may still easily propagate through the last layer, allowing for internal coefficients updates. The last layer's infinite dimensional coefficients are alternatively updated using the Kernel Ridge Regression closed form. Finally, numerical experiments are presented, both on synthetic and biological datasets.
- Chapter 5 investigates a duality approach to vv-RKHSs with values in infinite dimensional Hilbert spaces. A *Double Representer Theorem*, that expresses the optimal coefficients as linear combinations of the outputs, allows to tackle many loss functions, so far unused within infinite dimensional vv-RKHSs. The analysis of the dual problems also provides interesting insights on the assumptions needed on the operator-valued kernel to make the infinite dimensional problem (easily) computable. The particular cases of ϵ-insensitive Ridge Regression and Huber Regression are thoroughly studied, from their dual problems derivation to the undeniable empirical improvements they yield in surrogate approaches. Of course, these new losses can be plugged on the last layer of the deep kernel machines described in precedent chapters. Interestingly, the results established here indicate that a finite dimensional parametrization is possible, even for infinite dimensional outputs. This suggests that faster and better optimization procedures than the alternated Kernel Ridge Regression previously recommended are achievable.

Part II focuses on reliable alternatives to standard ERM in presence of outliers or bias.

- Chapter 6 recalls some basic notions about U-statistics. These quantities appear naturally in Machine Learning when the criterion of interest involves pairs of observations (*e.g.* in *metric learning*, *ranking*). More surprisingly, U-statistics also arise during the analysis of the next chapter's randomized estimators. Their strong concentration properties are then crucial to derive sharp bounds for these newly introduced estimators.

- Chapter 7 explores robust mean estimators inspired from the Median-of-Means. The standard Median-of-Means estimator is built as follows: first partition the dataset into groups of equal size, then compute the empirical mean on each block, and finally take the median of the computed means. This estimator introduced in the 1980s is particularly well suited to outliers and heavy-tailed distributions. Indeed, one atypical data may contaminate one block only, but is less likely to affect the final estimator, as the median should not select the mean of a corrupted block. This estimator is further extended to the case of randomized blocks, and similar guarantees are derived despite the created dependence between blocks. The estimator is then tailored to U-statistics, both standard and randomized. The computationally attractive Median-of-Incomplete-U-Statistics is also considered. Unfortunately, proof techniques used so far happen to be inadequate to derive satisfactory guarantees for this last estimator.

- Chapter 8 then exploits the previously introduced robust estimators to perform learning. The minimizers of a Median-of-Means estimator of the risk have been shown to exhibit good properties in presence of outliers. These guarantees are extended to minimizers of the randomized and U-statistics versions developed in the previous chapter. The Median-of-Means Gradient Descent algorithm is also adapted to all settings. The randomized version even shows desirable properties, as it naturally avoids local minima, without requiring any additional and artificial shuffling at each iteration. Another way to use Median-of-Means estimators in learning consists in computing tournament procedures. This approach compares the performances of pair of candidates, and finally selects a decision function with provably low excess risk, under mild assumption on the distribution. The tournament technique is adapted to the pairwise setting.

- Chapter 9 addresses the sample bias issue. In this setting, the training data at disposal does not follow the test distribution. Instead, several datasets are available, generated from biased distributions, absolutely continuous with respect to the test one. From the knowledge of the biasing functions, and under mild identifiability assumptions, it is then possible to compute a debiased estimate of the test distribution. When plugged into the empirical risk, it yields a reweighted ERM problem, whose weights are nontrivial solutions to a complex system of equations. The asymptotic guarantees about the debiased distribution estimate are first made non-asymptotic. These non-asymptotic guarantees then translate into guarantees about the debiased risk estimate, and finally to its minimizers. The generality of this approach (it totally encompasses the covariate shift scenario) makes it useful in many practical situations, and its soundness is finally endorsed by conclusive numerical experiments.

Part I

Deep Kernel Architectures for Complex Data

The first part of the present manuscript aims at exploring deep kernel architectures for complex data.

One of the known keys to the success of deep learning algorithms is the ability of neural networks to extract meaningful internal representations (Erhan et al., 2009). However, the theoretical understanding of why these compositional architectures are relevant and so successful remains limited. Furthermore, aside from recent advances on graph neural networks (Kipf and Welling, 2016a), deep approaches are almost restricted to vectorial data, by the nature itself of the operations they perform on inputs (*e.g.* convolution).

On the other hand, kernel methods provide with functional spaces whose geometry are well studied and understood. Their complexity can be easily controlled, either by the choice of kernel or penalization (Chapter 2). In addition, vector-valued kernel methods can be used to predict kernelized data. It then allows to make predictions in complex structured spaces, as soon as a kernel can be defined on it (Section 2.2.2).

The deep kernel architecture proposed in Chapter 3 consists in replacing the neural mappings of the form $\sigma(Wx+b)$ generally used in standard neural networks by functions from vector-valued Reproducing Kernel Hilbert Spaces. Although very different at first glance, the two functional spaces are actually very similar, and differ only by the order in which linear/nonlinear functions are applied (Remark 3.1).

Apart from gaining understanding and theoretical control on layers, considering kernel mappings allows for dealing with structured data, both in input and output. Hence, the main purpose of deep kernel architectures is not to challenge neural networks on tasks they have been optimized for during decades, such as image recognition. Alternatively, experiments presented in Chapter 4 highlight their ability to handle complex objects like molecules.

Finally, recent works exposed in Chapter 5 ensure a finite dimensional parametrization of the model, even when outputs are infinite dimensional or kernelized. These results open the door to efficient optimization procedures, for a wide range of losses and kernels.

2

Reminders on Operator-Valued Kernels

Contents

Kernel methods were at the core of Machine Learning's development during the 1990s. Based on well-understood mathematical concepts (Aronszajn, 1950), and combined with careful theoretical approaches (*e.g.* margin theory, Cortes and Vapnik (1995); Vapnik (1998)), they provided state-of-the-art algorithms on tasks such as digits recognition (LeCun et al., 1998). If Deep Learning approaches now globally outperform them for image or speech recognition, their relevance today still comes from their intrinsic ability to handle non-vectorial data, both in input and output. In particular, kernel methods remain among the most popular approaches to deal with biological sequences, making them a key asset in the field of bioinformatics and computational biology (Schölkopf et al., 2004; Saigo et al., 2004; Brouard et al., 2016b).

In this introductory chapter, we first recall important notions about scalar kernels (Section 2.1). In Section 2.2, we next focus on a major extension of the latter, widely used in the present manuscript: operator-valued kernels (OVKs) and their associated vector-valued Reproducing Kernel Hilbert Spaces (vv-RKHSs). Finally, in Section 2.2.2 are detailed several important applications of OVKs and vv-RKHSs, which the works presented in Chapters 3 and 5 build upon.

2.1 Reminders on Scalar Kernels

If one considers the standard regularized-ERM supervised learning criterion

$$\min_{h \in \mathcal{H}} \quad \frac{1}{n} \sum_{i=1}^{n} \ell \left(h(x_i), y_i \right) + \mathrm{Reg}(h), \tag{2.1}$$

kernel methods can be seen as a specific choice of functional space $\mathcal{H}$ in which the optimal solution is searched, namely Reproducing Kernel Hilbert Spaces (RKHSs in short). Incidentally, the assumption made on $\mathcal{H}$ may be summarized as the continuity of its functions evaluations.

Definition 2.1. *The Hilbert space $\mathcal{H} \subset \mathbb{R}^{\mathcal{X}}$ is a* Reproducing Kernel Hilbert Space *if and only if for any $x \in \mathcal{X}$ the following mapping is continuous*

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

However, another popular definition of RKHSs is based on their associated *reproducing kernel*. This alternative construction is detailed in the next section, as well as the equivalence with Definition 2.1.

2.1.1 Kernels and RKHSs

Another way to define a RKHS, the space in which we restrict our search for the optimal regression function h in Problem (2.1), is to build them from *reproducing kernels*. As a first go, let us recall the definition of *positive definite* kernels.

Definition 2.2. *Let $\mathcal{X}$ be any set. A* (scalar) positive definite kernel *on $\mathcal{X}$ is an application $k \colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ that satisfies the following two properties:*

1. $\forall x, x' \in \mathcal{X}^2, \qquad k(x, x') = k(x', x),$
2. $\forall (x_i)_{i \leq n} \in \mathcal{X}^n, \boldsymbol{\alpha} \in \mathbb{R}^n, \qquad \sum_{i,j=1}^n \alpha_i k(x_i, x_j) \alpha_j \geq 0.$

One may already notice that $k : (x, x') \mapsto \langle x, x' \rangle_{\mathcal{X}}$ is trivially a positive definite kernel on any Hilbert space $\mathcal{X}$. Similarly, if there exist a Hilbert $\mathcal{H}$, and a application $\phi : \mathcal{X} \to \mathcal{H}$, $\mathcal{X}$ denoting here any set, then one may easily verify that $k : (x, x') \mapsto \langle \phi(x), \phi(x') \rangle_{\mathcal{H}}$ is also a positive definite kernel. Interestingly, the converse is also true, as revealed by the following theorem.

Theorem 2.3. *Let $\mathcal{X}$ be any set. An application $k \colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a positive definite kernel on $\mathcal{X}$ if and only if there exist a Hilbert space $\mathcal{H}$ and an application $\phi \colon \mathcal{X} \to \mathcal{H}$ such that*

$$\forall x, x' \in \mathcal{X}^2, \qquad k(x, x') = \Big\langle \phi(x), \phi(x') \Big\rangle_{\mathcal{H}}.$$

The proof of Theorem 2.3 has been established in Aronszajn (1950) for the general formulation stated here, while previous partial proofs for $\mathcal{X}$ compact and k continuous or $\mathcal{X}$ countable may be found in Mercer (1909) and Kolmogorov (1941) respectively.

The next theorem now links positive definite kernels and Reproducing Kernel Hilbert Spaces.

Theorem 2.4. *Let $\mathcal{X}$ be any set, and $k \colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ a positive definite kernel on $\mathcal{X}$. Then there exists a unique Hilbert space $\mathcal{H}_k \subset \mathbb{R}^{\mathcal{X}}$ such that*

- $\forall x \in \mathcal{X}, \qquad k(\cdot, x) \in \mathcal{H}_k,$
- $\forall x \in \mathcal{X}, \forall h \in \mathcal{H}_k, \qquad h(x) = \langle h, k(\cdot, x) \rangle_{\mathcal{H}_k}.$

The positive definite kernel k is then called a reproducing kernel*, and $\mathcal{H}_k$ its associated RKHS coincides with $\mathcal{H}$ of Theorem 2.3.*

If any RKHS as defined in Theorem 2.4 directly satisfies Definition 2.1 by the use of Cauchy-Schwarz inequality, it can be easily shown that the converse holds true by virtue of Riesz representation theorem.

Hence, after having set the search space as a RKHS, this rich mathematical background helps analyzing and understanding the *kernel machines*, *i.e.* the algorithms induced by different choices of loss function ℓ in Problem (2.1).

2.1.2 Kernel Machines

Come back now to Problem (2.1), with the classical specific choice of regularization $\mathrm{Reg}(h) = (\Lambda/2)\|h\|^2_{\mathcal{H}_k}$, for some penalization parameter $\Lambda > 0$:

$$\min_{h\in\mathcal{H}_k} \quad \frac{1}{n}\sum_{i=1}^{n} \ell\left(h(x_i), y_i\right) + \frac{\Lambda}{2}\|h\|^2_{\mathcal{H}_k}. \tag{2.2}$$

Another crucial tool in kernel methods, that leverages the Hilbert structure of the search space, is the *Representer Theorem*, that ensures the optimal solution actually lies in a finite dimensional subspace of $\mathcal{H}_k$. Formally, it is stated as follows.

Theorem 2.5. *Let $\mathcal{X}$ be any set, endowed with a positive definite kernel k, $\mathcal{H}_k \subset \mathbb{R}^{\mathcal{X}}$ its associated RKHS, and $(x_i)_{i\leq n} \in \mathcal{X}^n$. Let $V\colon \mathbb{R}^{n+1} \to \mathbb{R}$ be a functional, strictly increasing with respect to its last argument. Then, if $\hat{h}$ denotes the solution to problem*

$$\min_{h\in\mathcal{H}_k} \quad V\left(h(x_1), \ldots, h(x_n), \|h\|_{\mathcal{H}_k}\right),$$

there exist $(\hat{\alpha}_i)_{i\leq n} \in \mathbb{R}^n$ such that $\hat{h}$ writes

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

Proof. Let E be $\mathrm{Span}\{k(\cdot, x_i), i \leq n\}$. It is a final dimensional subspace of $\mathcal{H}_k$, so that h can be decomposed into $\bar{h} + h^\perp$, with $\bar{h}, h^\perp \in E \times E^\perp$. Using the reproducing property, $h(x_i) = \bar{h}(x_i)$ for all $i \leq n$, while $\|h\|_{\mathcal{H}_k} = \|\bar{h} + h^\perp\|_{\mathcal{H}_k} \geq \|\bar{h}\|_{\mathcal{H}_k}$ by Pythagoras' theorem. Therefore, the orthogonal component necessarily makes the overall criterion increase: it is null, and h admits a decomposition as stated in Theorem 2.5. □

This theorem, that applies to all problems expressed as Problem (2.2), together with the observation that $\langle k(\cdot, x_i), k(\cdot, x_j)\rangle_{\mathcal{H}_k} = k(x_i, x_j)$, often referred to as *kernel trick*, has important consequences. Indeed, it makes most kernel machines computable, as long as only dot products are involved in the criterion, and from the knowledge of the gram matrix $K \in \mathbb{R}^{n\times n}$ such that $K_{ij} = k(x_i, x_j)$ only. Consider for instance the Kernel Ridge Regression problem:

$$\min_{h\in\mathcal{H}_k} \quad \frac{1}{2n}\sum_{i=1}^{n} \left(y_i - h(x_i)\right)^2 + \frac{\Lambda}{2}\|h\|^2_{\mathcal{H}_k}.$$

Theorem 2.5 applies: plugging the expansion and differentiating with respect to the $(\alpha_i)_{i\leq n}$ gives that (with bold letter referring to the $\mathbb{R}^n$ vectors concatenating the α_i's or y_i's scalar values):

$$\hat{\boldsymbol{\alpha}} = (K + n\Lambda \boldsymbol{I}_n)^{-1}\boldsymbol{y}.$$

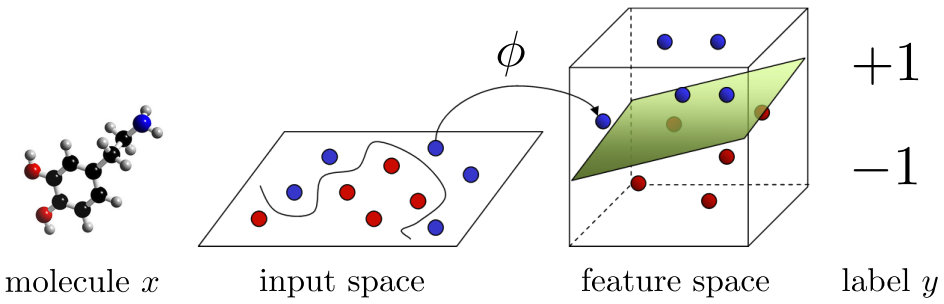


Figure 2.1 – Linear Classification after mapping by ϕ (adapted from Shehzadex (2017)).

A bit more involved are the Support Vector Machines (SVMs, Cortes and Vapnik (1995); Vert and Vert (2006)) that feature the hinge loss (the y_i's are assumed here to be labels in $\{-1, +1\}$):

$$\min_{h \in \mathcal{H}_k} \quad \frac{1}{2n} \sum_{i=1}^{n} \max\Big(0, 1 - y_i h(x_i)\Big) + \frac{\Lambda}{2} \|h\|^2_{\mathcal{H}_k}.$$

SVMs are generally computed using duality (see Chapter 5). But rewriting $h(x_i)$ as $\langle k(\cdot, x_i), h \rangle_{\mathcal{H}_k} = \langle \phi(x_i), h \rangle_{\mathcal{H}_k}$ gives an interesting intuition. If the datapoints are not linearly separable in the original input space, they might be after the mapping through a high dimensional feature map ϕ. Kernel methods then just consist in linear techniques in the high dimensional feature space, where a separating hyperplane is more likely to exist. This is summarized by Figure 2.1.

Three important remarks can be made at this point. First, since the algorithm only relies on the $\phi(x_i)$'s, the nature of the original inputs x_i's has no incidence. This observation makes kernel methods of particular interest when inputs are complex objects (Gärtner, 2008). One of the key steps in kernel learning then lies in the design of meaningful and expressive kernels. Structured objects dealt with include for instance time series (Cuturi et al., 2007), graphs (Mahé and Vert, 2009), strings (Saigo et al., 2004) or trees (Vert, 2002). Second, it is important to notice that the (potentially infinite dimensional) feature representations $\phi(x_i)$'s may never be computed explicitly. Indeed, as long as only dot products (and consequently squared norms) are involved in the optimized criterion, only the $\langle \phi(x_i), \phi(x_j) \rangle_{\mathcal{H}_k} = k(x_i, x_j)$ need to be computed. This leads to the third remark. One can actually proceed the other way around. If one algorithm's criterion only depends on dot products, then replacing the x_i's by some $\phi(x_i)$'s, one creates a *kernelized* version of the algorithm. This version is generally not harder to compute, but can benefit from the new structure of the data in the high dimensional space. This can be applied in unsupervised learning problematics also, with celebrated adaptations such as Kernel k-means (Dhillon et al., 2004), Kernel Principal Component Analysis (Schölkopf et al., 1997, 1998), Kernel Independent Component Analysis (Bach and Jordan, 2002), or Kernel Canonical Component Analysis (Lai and Fyfe, 2000; Yamanishi et al., 2003; Hardoon et al., 2004).

Before focusing on the operator-valued extension of scalar kernels, we conclude this section by giving an upper bound on the Rademacher complexity of (scalar) RKHS balls. This bound is classical, and similar techniques are used in Chapter 3 to analyze the complexity of vector-valued extensions of RKHSs.

Proposition 2.6. *Let* $\mathcal{H}_{k,\Lambda} = \{h \in \mathcal{H} : \|h\|_{\mathcal{H}_k} \leq \Lambda\}$, $\mathcal{S}_n = \{x_1, \ldots, x_n\}$, *and* K *the Gram matrix associated to sample* $\mathcal{S}_n$. *Then it holds*

$$\widehat{\mathscr{R}}(\mathcal{H}_{k,\Lambda}, \mathcal{S}_n) = \frac{1}{n}\mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{h\in\mathcal{H}_{k,\Lambda}} \sum_{i=1}^{n} \sigma_i h(x_i)\right] \leq \frac{\Lambda\sqrt{\mathbf{Tr}(K)}}{n}.$$

Proof.

$$\mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{h\in\mathcal{H}_{k,\Lambda}} \sum_{i=1}^{n} \sigma_i h(x_i)\right] = \mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{h\in\mathcal{H}_{k,\Lambda}} \left\langle \sum_{i=1}^{n} \sigma_i k(\cdot, x_i), h \right\rangle\right] \leq \Lambda\mathbb{E}_{\boldsymbol{\sigma}}\left\|\sum_{i=1}^{n} \sigma_i k(\cdot, x_i)\right\|,$$

$$\leq \Lambda\sqrt{\mathbb{E}_{\boldsymbol{\sigma}}\left\|\sum_{i=1}^{n} \sigma_i k(\cdot, x_i)\right\|^2} = \Lambda\sqrt{\sum_{i=1}^{n} k(x_i, x_i)} = \Lambda\sqrt{\mathbf{Tr}(K)}.$$

□

Kernel methods have been extensively studied in the Machine Learning literature. The interested reader may refer to the overview proposed in Hofmann et al. (2008), or to the excellent monographs by Schölkopf et al. (2002), Schölkopf et al. (2004), Shawe-Taylor et al. (2004), Steinwart and Christmann (2008) and Berlinet and Thomas-Agnan (2011).

We shall now move to operator-valued extensions of scalar kernels, that are crucial tools in the following of this manuscript

2.2 Reminders on Operator-Valued Kernels

Assume now that the target outputs are not scalar anymore, but rather valued in $\mathbb{R}^p$, no further assumption being made on the input space $\mathcal{X}$. One straightforward way to extend scalar kernel methods to this setting simply consists in stacking p independent functions from a scalar RKHS. This, actually corresponds to the particular case of an identity decomposable matrix-valued kernel (see Álvarez et al. (2012)). Operator-valued kernels can themselves be seen as extensions of matrix-valued kernels to the case of any output Hilbert space $\mathcal{Y}$, and not necessarily $\mathbb{R}^p$. In Section 2.2.1, we detail the construction of vector-valued Reproducing Kernel Hilbert Spaces (vv-RKHSs) from Operator-Valued Kernels (OVKs), similarly to what has been done in Section 2.1.1. In Section 2.2.2 are finally detailed important applications unlocked by the possibility to predict outputs in infinite dimensional spaces.

Vector-Valued RKHSs also benefit from an important theoretical literature, starting from the work by Senkene and Tempel'man (1973), or by Micchelli and Pontil (2005), that the next section is largely inspired from. More recent important contributions include for instance Caponnetto et al. (2008) and Carmeli et al. (2006, 2010).

2.2.1 Operator-Valued Kernels and Vector-Valued RKHSs

Similarly to scalar kernels, a vv-RKHS can be primarily defined by the continuity of its functions evaluations.

Definition 2.7. *A Hilbert space $\mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$ is a* (vector-valued) Reproducing Kernel Hilbert Space *if and only if for any $x \in \mathcal{X}$ and any $y \in \mathcal{Y}$, the following mapping is continuous*

$$F_{x,y}\colon \begin{pmatrix} \mathcal{H} & \to & \mathbb{R} \\ h & \mapsto & \Big\langle h(x), y \Big\rangle \end{pmatrix}.$$

And it can also be constructed through operator-valued kernels.

Definition 2.8. *Let $\mathcal{X}$ be any set and $\mathcal{Y}$ a Hilbert space. A* positive definite operator-valued kernel *on $\mathcal{X}$ and $\mathcal{Y}$ is an application $\mathcal{K}\colon \mathcal{X}\times\mathcal{X} \to \mathcal{L}(\mathcal{Y})$ that satisfies the following two properties:*

1. $\forall x, x' \in \mathcal{X}^2, \qquad \mathcal{K}(x,x') = \mathcal{K}(x',x)^*,$
2. $\forall (x_i)_{i\le n} \in \mathcal{X}^n, (y_i)_{i\le n} \in \mathcal{Y}^n, \qquad \sum_{i,j=1}^{n} \Big\langle y_i, \mathcal{K}(x_i,x_j)y_j \Big\rangle_{\mathcal{Y}} \ge 0,$

where A^* denotes the adjoint of any operator A, and $\mathcal{L}(E)$ the set of bounded linear operators of any vector space E.

A simple example of OVK is the *separable kernel*.

Definition 2.9. *Let $\mathcal{X}$ be any set and $\mathcal{Y}$ a Hilbert space. The OVK $\mathcal{K} : \mathcal{X} \times \mathcal{X} \to \mathcal{L}(\mathcal{Y})$ is a* separable kernel *if and only if there exist a positive definite scalar kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, and a positive semi-definite operator A on $\mathcal{Y}$ such that:*

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

If furthermore $A = \mathbf{I}_{\mathcal{Y}}$, $\mathcal{K}$ is said identity decomposable.

The counterpart of Theorem 2.3 is as follows.

Theorem 2.10. *Let $\mathcal{X}$ be any set and $\mathcal{Y}$ a Hilbert space. An application $\mathcal{K}\colon \mathcal{X} \times \mathcal{X} \to \mathcal{L}(\mathcal{Y})$ is a positive definite OVK if and only if there exist a Hilbert space $\mathcal{H}$ and an application $\Phi\colon \mathcal{X} \to \mathcal{L}(\mathcal{H}, \mathcal{Y})$ such that*

$$\forall x, x' \in \mathcal{X}^2, \qquad \mathcal{K}(x,x') = \Phi(x)\Phi^*(x').$$

Just as for standard scalar-valued kernels, an OVK can be uniquely associated to a functional space (its vv-RKHS), as detailed by the next definition.

Definition 2.11. *Let $\mathcal{K}\colon \mathcal{X} \times \mathcal{X} \to \mathcal{L}(\mathcal{Y})$ be a (positive definite) OVK, and for $x \in \mathcal{X}$, let $\mathcal{K}_x\colon y \mapsto \mathcal{K}_x y \in \mathcal{F}(\mathcal{X}, \mathcal{Y})$ the linear operator such that:*

$$\forall x' \in \mathcal{X}, \ (\mathcal{K}_x y)(x') = \mathcal{K}(x',x)y.$$

There is a unique Hilbert space $\mathcal{H}_{\mathcal{K}} \subset \mathcal{F}(\mathcal{X}, \mathcal{Y})$ called the vv-RKHS associated to $\mathcal{K}$ such that $\forall x \in \mathcal{X}$:

- *$\mathcal{K}_x$ spans the space $\mathcal{H}_{\mathcal{K}}$ ($\forall y \in \mathcal{Y}\colon \mathcal{K}_x y \in \mathcal{H}_{\mathcal{K}}$)*
- *$\mathcal{K}_x$ is bounded for the uniform norm*
- *$\forall f \in \mathcal{H}_{\mathcal{K}}, \ f(x) = \mathcal{K}_x^* f$ (reproducing property)*

Learning within vv-RKHSs also relies on Representer Theorems, that are derived from the Minimal Norm Interpolation principle.

Theorem 2.12. *Let $\mathcal{X}$ be any set, $\mathcal{Y}$ a Hilbert space, and $(x_i, y_i)_{i\leq n} \in (\mathcal{X}\times\mathcal{Y})^n$. For $i \leq n$, let L_{x_i} be the linear functional such that $L_{x_i}h = h(x_i)$ for $f \in \mathcal{H}_\mathcal{K}$. If the L_{x_i}'s are linearly independent, then unique solution to the variational problem*

$$\begin{aligned} \min_{h\in\mathcal{H}_\mathcal{K}} \quad & \|h\|_{\mathcal{H}_\mathcal{K}}, \\ s.t. \quad & h(x_i) = y_i, \qquad i \leq n, \end{aligned}$$

is given by

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

with $(\hat{\alpha}_i)_{i\leq n} \in \mathcal{Y}^n$ the unique solution to the linear system of equations

$$\sum_{i=1}^{n} \mathcal{K}(x_j, x_i)\hat{\alpha}_i = y_j, \qquad j \leq n.$$

Proof. Let h be any element of $\mathcal{H}_\mathcal{K}$ such that $h(x_i) = y_i$ for all $i \leq n$. Let $h^\perp = h - \hat{h}$. It holds that $\|h\|^2_{\mathcal{H}_\mathcal{K}} = \|\hat{h}\|^2_{\mathcal{H}_\mathcal{K}} + \|h^\perp\|^2_{\mathcal{H}_\mathcal{K}} \geq \|\hat{h}\|^2_{\mathcal{H}_\mathcal{K}}$, so that $\hat{h}$ is the unique solution. □

The next section is now devoted to important learning applications, typically tackled by the use of vv-RKHSs, and their inherent capacity to handle infinite dimensional outputs.

2.2.2 Important Applications of Vector-Valued RKHSs

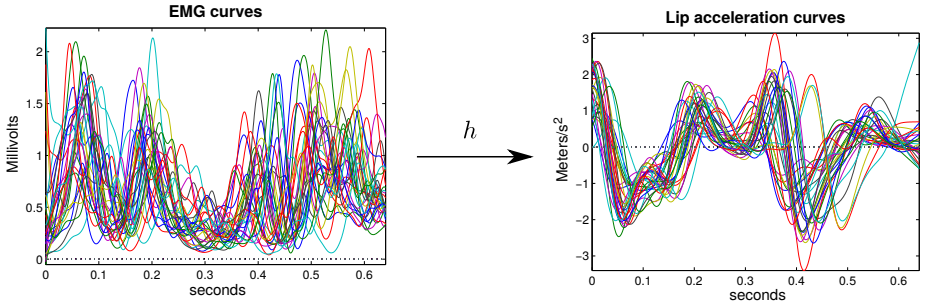
First used in the finite dimensional case ($\mathcal{Y} = \mathbb{R}^p$) to solve multi-task regression problems (Micchelli and Pontil, 2005) and multiple class classification (Dinuzzo et al., 2011), OVK methods also stand out for their ability to cope with infinite dimensional and functional outputs. One of the most straightforward application thus made possible is functional regression. This ranges from the minimization of the L^2 norms of square integrable functions, to the more involved case of learning the whole conditional quantile function. Furthermore, through the use of a kernel embedding, similar to that done in Figure 2.1, but for the outputs this time, the OVK framework provides an interesting theoretically grounded strategy to address structured output prediction.

Leveraging the Functional Nature of the Outputs

A learning scenario, far from being unusual, when OVKs can be of great help is that of functional regression. For instance, consider a function-to-function problem. Within this setting, each input x_i is a function, that must be mapped to an output function y_i. A very representative example is that of lip acceleration prediction, taken from Ramsay and Silverman (2007). The training input data consists in 32 electromyograms (EMG) recording the nervous activity of the lip muscles during 690 milliseconds during which the patient pronounces the syllable *bob*. The output data gathers the corresponding lower lip acceleration curves, on the same period of time. The goal here is to learn a predictive function h to map each EMG function to the acceleration curve, as illustrated in Figure 2.2.

A Ridge Regression then corresponds to minimizing over $\mathcal{H}_K \subset \mathcal{F}(L^2[0, 690], L^2[0, 690])$:

$$\frac{1}{2n}\sum_{i=1}^{n} \|y_i - h(x_i)\|^2_{L^2} + \frac{\Lambda}{2}\|h\|^2_{\mathcal{H}_\mathcal{K}} = \frac{1}{2n}\sum_{i=1}^{n} \int_{\theta=0}^{690} \Big(y_i(\theta) - (h(x_i))(\theta)\Big)^2 + \frac{\Lambda}{2}\|h\|^2_{\mathcal{H}_\mathcal{K}}.$$



Input functions $\{x_i\}_{i\leq n}$ Output functions $\{y_i\}_{i\leq n}$

Figure 2.2 – Function-to-Function Regression (taken from Kadri et al. (2016)).

This examples has been studied at length in Kadri et al. (2016). The criterion written above can however be considerably enlarged. Indeed, leveraging the functional nature of the outputs, on can generalize this integral loss to many more case than the square of the difference between the targeted and predicted functions.

For some compact set Θ, and a scalar loss function $l : \Theta \times \mathbb{R}^2 \to \mathbb{R}$, define the *integral loss function*:

$$\ell\colon \begin{pmatrix} L^2[\Theta,\mu] \times L^2[\Theta,\mu] & \to & \mathbb{R} \\ (f,g) & \mapsto & \int_\Theta l\Big(\theta, f(\theta), g(\theta)\Big)\mathrm{d}\mu(\theta) \end{pmatrix},$$

where μ is a probability measure over Θ.

If $l(\theta, x, y) = \frac{1}{2}(x-y)^2$, one recovers the minimizations of the L^2 norm of Kadri et al. (2016). The following two loss functions lead to other interesting problems.

- $l(\theta, x, y) = \max\left(\theta(y-x), (\theta-1)(y-x)\right)$. This loss function, referred to as the pinball loss (Koenker, 2005), is used at fixed θ to perform conditional quantile regression of some random variables $X, Y \in \mathbb{R}^d \times \mathbb{R}$ based on i.i.d. samples $(x_i, y_i)_{i=1}^n$. The minimization of its integrated counterpart yields an estimate of the whole conditional quantile function when applied to $(x_i, y_i)_{i=1}^n$, the $(y_i)_{i=1}^n$ being considered as constant functions in $L^2[\Theta, \mu]$. Learning the whole quantile function at the same time yields multiple benefits, among which the possibility to introduce suitable non-crossing constraints. A complete study of this learning problem can be found in Brault et al. (2019).

- $l(\theta, x, y) = |\theta - \mathbb{1}_{\{-1\}}(y)| \max(0, 1-yx)$. Given some fixed $\theta \in [0,1]$, this binary classification loss function is used in cost-sensitive classification (Zadrozny and Elkan, 2001). The coefficient $|\theta - \mathbb{1}_{\{-1\}}(y)|$ is asymmetric with respect to the two classes $y \in \{-1, 1\}$, which models a different impact for mistakes committed on one class or another. Minimizing the integrated loss lifts the need to choose the asymmetric coefficient (which is almost never known in practice), and allows a practitioner to evaluate the effect of this asymmetry posterior to the learning phase, since the algorithm outputs a maximum-margin classifier as a function of θ. Brault et al. (2019) also provides examples of the minimization of such an integral criterion by means of vv-RKHSs.

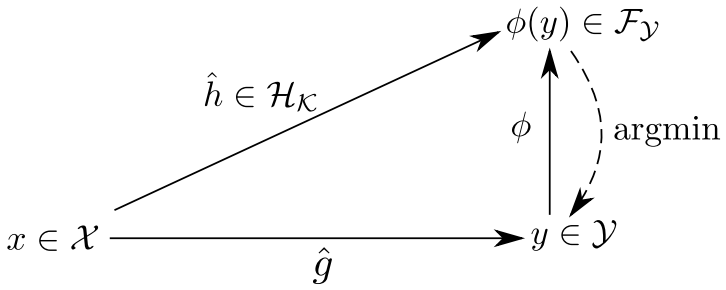


Figure 2.3 – Structured Output Prediction.

In the next example, outputs are no longer functions, but structured outputs. However, a simple mapping through a kernel feature map may transform them into functions in a Hilbert space. The vv-RKHS machinery next allows to learn a predictive function for the embedded outputs.

Structured Output Prediction

This denomination refers to a general supervised learning task from an input space $\mathcal{X}$ to an output space $\mathcal{Y}$, such that $\mathcal{Y}$ is a finite set of structured objects (Nowozin and Lampert, 2011). This includes biological sequences, trees, graphs, and more generally objects composed of several sub-objects. The major difficulty in learning a function with structured outputs $g : \mathcal{X} \to \mathcal{Y}$ lays in the fact that the set $\mathcal{Y}$ is not equipped with standard tools such as norms and dot products. Therefore, designing a meaningful loss and learning algorithms requires additional efforts compared to standard regression frameworks.

While Structural SVMs and variants only cope with discrete structures (Joachims et al., 2009), another solution is to embed output datapoints into an output feature space $\mathcal{F}_\mathcal{Y}$ through a feature map ϕ that possesses the desirable properties. Note that embeddings can be defined either explicitly within the finite dimensional Euclidean space $\mathcal{F}_\mathcal{Y} = \mathbb{R}^p$ (see for instance, semantic embeddings and SELF approaches in Ciliberto et al. (2016)), or implicitly with the help of kernels (Cortes et al., 2005; Brouard et al., 2011). Indeed, if only dots products are involved in the surrogate criterion that links the inputs x_i's to the embedded output $\phi(y_i)$'s, the use of the kernel trick for the inputs may prevent from the explicit computation of the feature representations. In that case, ϕ can be seen as the canonical feature map associated to a scalar kernel $k\colon \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$, and $\mathcal{F}_\mathcal{Y} = \mathcal{H}_k$ is its associated RKHS.

By definition of a kernel feature map, the outputs are sent in a Hilbert space, on which the vv-RKHS methodology applies. After having learnt the surrogate regression function in the new output space $\mathcal{F}_\mathcal{Y}$, solving a pre-image problem to provide a predicted output in the original space $\mathcal{Y}$ is however necessary. The whole procedure is recapped in Figure 2.3, and by the following two equations:

$$(1) \qquad \hat{h} = \operatorname*{argmin}_{h \in \mathcal{H}_K} \sum_{i=1}^{n} \left\| \phi(y_i) - h(x_i) \right\|_\mathcal{Y}^2 + \frac{\Lambda}{2} \|h\|_{\mathcal{H}_K}^2,$$

$$(2) \qquad \forall x \in \mathcal{X}, \qquad \hat{g}(x) = \operatorname*{argmin}_{y \in \mathcal{Y}} \left\| \phi(y) - \hat{h}(x) \right\|_\mathcal{Y}^2.$$

Compared to finite dimensional embeddings, kernels enable to compute more complex similarity functions between structured data. For instance, a Gaussian kernel on finite embeddings implicitly results in using an infinite feature map (Brouard et al., 2016a). Using infinite dimensional implicit feature maps leads to state-of-the-art methods in metabolite identification, where molecules are to be predicted from their mass spectra (Brouard et al., 2016b).

2.3 Conclusion

Kernel methods and RKHSs provide classes of functions for practicing ERM, whose geometrical structures are well analyzed, and thoroughly studied. Their vector-valued extensions yield similar understanding, together with the possibility to handle infinite dimensional outputs. The kernel embedding of the outputs, and the use of the kernel trick, then allows to perform regression tasks on any structured data. Functions from vv-RKHSs are the building blocks of the larger functional class which is introduced in Chapter 3. Their capacity to handle complex data is crucial in their utilization, especially for Representation Learning (Section 3.1.1). Extending the vector-valued Representer Theorem to composition of functions is then the key ingredient of Chapter 4 to make the designed model computable. Finally, Chapter 5 considers OVK machines through the angle of duality. In particular, this approach enables the use of loss functions that are hardly computable in the primal, but empirically meaningful.

3

Kernel Autoencoders for Complex Data

Contents

As seen in Chapter 2, OVKs and vv-RKHSs provide an elegant theoretical framework to handle infinite dimensional and structured outputs. However, the revolution of Deep Learning (Ian J. Goodfellow et al., 2016) has shown that models based on the successive composition of elementary mappings nowadays produce state-of-the-art performances.

The present chapter aims at combining both approaches, and proposes a novel model to automatically extract vectorial representations from complex structured objects. As a first go, the model is thoroughly detailed in Section 3.2, building on Operator-Valued Kernels (OVKs) and vector-valued Reproducing Kernel Hilbert Spaces (vv-RKHSs) studied in Chapter 2. It is then theoretically investigated, through its link to Kernel Principal Component Analysis (KPCA) in Section 3.3, or by means of Rademacher averages extended to infinite dimensional outputs and compositions of functions (see Section 3.4). In Section 3.5 are listed interesting extensions and applications of the proposed model. This chapter corresponds to the theoretical contribution of:

▶ **P. Laforgue**, S. Clémençon, F. d'Alché-Buc. Autoencoding any data through kernel autoencoders. In *Proceedings of Artificial Intelligence and Statistics*, 2019.

3.1 Introduction

One of the keys to the success of deep learning approaches is the ability of neural networks to extract meaningful internal representations (Erhan et al., 2009; Zeiler and Fergus, 2014). This concern has attracted a lot of interest lately, with now a dedicated world known conference (International Conference on Learning Representations, ICLR). We start this chapter by a brief overview of the domain, with a particular focus on Autoencoders, that inspired the proposed model. The context is also recalled, by the discussion of related works.

3.1.1 Learning Representations

As experienced by any practitioner, data representation is critical to the application of Machine Learning, whatever the targeted task, supervised or unsupervised. A first answer to this issue consists in *feature engineering*, that uses domain knowledge to create relevant descriptive variables. However, this step requires numerous interactions with domain experts, and it is often time-consuming. To overcome these limitations, Representation Learning (*e.g.* Bengio et al. (2013a)) aims at building automatically new features in an unsupervised fashion. With a growing concern in the community about the relevance of data representations, Representation Learning has now become a proper research field.

One of the most important motivation for Representation Learning is leveraging the unlabeled data. Indeed, in many applications, annotating the data is expensive and the unlabeled dataset is often much larger than the labeled one. This is the case, for instance, of the metabolite identification problem (Brouard et al., 2016b), for which labeled molecules represent approximately $10,000$ datapoints, compared to the several millions of unlabeled ones. If one could benefit from this huge amount of unlabeled data to learn good representations in a purely unsupervised fashion, this should improve the performance on potential supervised task. This advantage has been made particularly clear by *e.g.* Mesnil et al. (2011) and Goodfellow et al. (2012).

The central, and unsolved, question of Representation Learning could be expressed as follows: "*What makes a representation better than another?*". In the survey by Bengio et al. (2013a), authors try to list the suitable properties a good representation should have. Among them can be found smoothness, hierarchical organization of explanatory factors, shared factors across tasks, natural clustering (observations with different values for categorical variables should be separate), sparsity. These properties are however somewhat unmeasurable, and the mathematical understanding of why Representation Learning may help is still limited, so are the quantitative criteria to assess the goodness of representations.

Nevertheless, one common consequence can be found among all desirable properties above listed: that of disentangling causal factors. This is a key feature to perform transfer learning and domain adaptation (Ben-David et al., 2010), *i.e.* continue learning despite the change of the data distribution. In one-shot learning (Fei-Fei et al., 2006), only one labeled example is available for the targeted supervised task. The rationale behind is that the unsupervised representation learning phase was sufficiently powerful to have clearly isolated the classes and disentangled the causal/invariant factors among classes, so that one labeled observation is enough to predict the label of many others. In zero-shot learning (Larochelle and Bengio, 2008; Palatucci et al., 2009; Socher et al., 2013), no labeled data is given for the targeted class. Zero-shot learning is only made

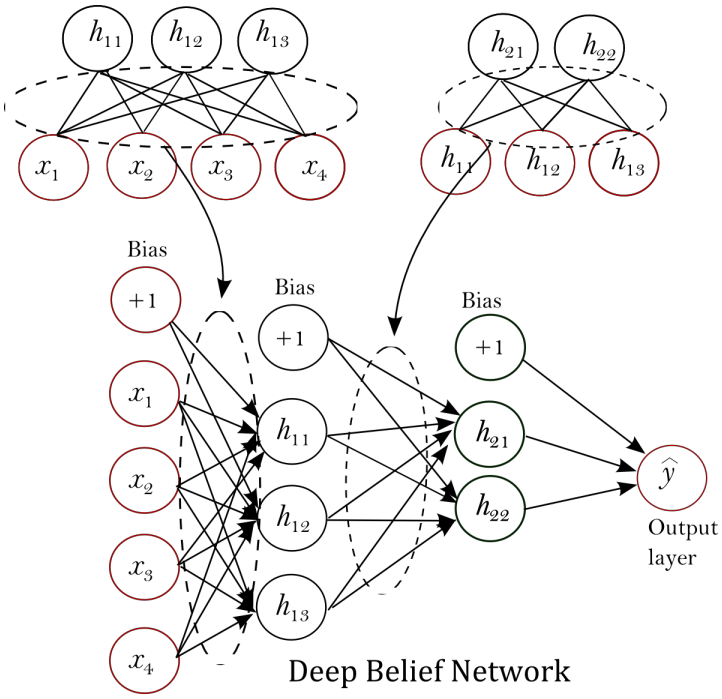
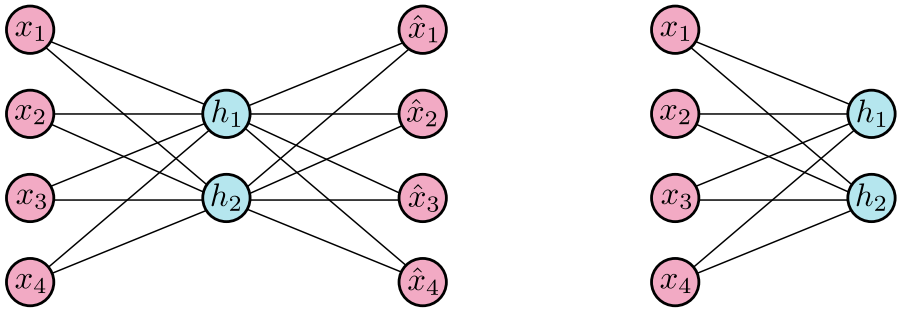


Figure 3.1 – Greedy Layer-wise Unsupervised Pretraining

possible by a strong underlying structure and the learning of adapted representations, as may be the case in machine translation for instance (see *e.g.* Mikolov et al. (2013)), or in sentiment analysis (Glorot et al., 2011). Nevertheless, it highlights how critical the choice of representation may be, by allowing to make correct predictions on unseen examples. The problem of metabolite identification can also be seen as a zero-shot learning problem. Indeed, every observation is a different molecule, so that it may be considered as a multiclass classification task, with as many classes as datapoints, and consequently no labeled training data for any of the classes.

But the primary successes of Representation Learning are to be found in *pretraining*. It consists in using unsupervised criteria to learn weights between two successive layers, that will serve as an initialization for the final *fine tuning* stage, learnt by minimizing the supervised criterion, see Figure 3.1. This procedure was very popular during the mid-2000s (Hinton et al., 2006; Hinton and Salakhutdinov, 2006; Bengio et al., 2007; Ranzato et al., 2007), and the work by Erhan et al. (2010) performed many experiments to explain the success of these approaches. However, regularized training techniques such as dropout (Srivastava et al., 2014) have progressively outperformed unsupervised pretraining, that is nowadays mostly abandoned except in the field of natural language processing. Yet, this paradigm has witnessed the resurgence, and promoted the use, of an interesting unsupervised architecture: Autoencoders. As the model proposed in this chapter is largely inspired from this so called *self-supervised* approach, the next section of this introduction focuses more precisely on this architecture, as well as their generative variants: Restricted Boltzmann Machines.



(a) 2-Layer Autoencoder (b) Restricted Boltzmann Machine

Figure 3.2 – Unsupervised Architectures

3.1.2 Autoencoders

Among successful unsupervised Representation Learning methods used in pretraining, mention has to be made of Autoencoders. According to Ian J. Goodfellow et al. (2016) (chapter 14 therein): "*An autoencoder is a neural network that is trained to attempt to copy its input to its output*". The difficulty of the reconstruction comes from the fact that the architecture often contains a bottleneck (Figure 3.2a). The underlying rationale is that if observations can be accurately reconstructed from the internal compressed codes, then the latter should have captured the most important properties of the data.

The idea of Autoencoders has a long history in neural networks (Bourlard and Kamp (1988) for a comparison to Principal Component Analysis, Hinton and Zemel (1994)). Their generative variants are called Restricted Boltzman Machines, and are depicted in Figure 3.2b. They are generative neural networks that learn the probability distribution of inputs by minimizing an energy-based criterion involving hidden units. Re-introduced by Salakhutdinov and Hinton (2009a), they already existed as *Harmoniums* in the book by McClelland et al. (1987). The focus of this chapter being not generative models, although bridging both approaches would constitute an interesting research direction, we shall now concentrate on Autoencoders exclusively.

Structural constraints and penalizations can of course be added to the data-fitting term in order to promote particular architectures. Sparse Autoencoders (Ranzato et al., 2007, 2008), that features an ℓ_1 penalty of the hidden representation, Contractive Autoencoders (Rifai et al., 2011), that penalizes the derivatives of the encoding function, and more generally all Regularized Autoencoders (Alain and Bengio, 2014), combining several types of constraints and penalizations, are further examples of sophistications around the standard autoencoder criterion.

Apart from bigger networks pretraining, a popular utilization of Autoencoders is thus dimensionality reduction (*e.g.* Hinton and Salakhutdinov (2006)). This yields important applications in information retrieval, and more specifically in semantic hashing, both applied to textual data (Salakhutdinov and Hinton, 2009b) and images (Weiss et al., 2009; Krizhevsky and Hinton, 2011). One last interesting application of Autoencoders can be found in denoising. The idea of Denoising Autoencoders (Vincent et al., 2010) is to reconstruct the input from a perturbed version of it. Bengio et al. (2013b) have shown that this method implicitly forces the encoder/decoder pair to learn the structure of the (non-corrupted) inputs distribution.

If they have mostly been studied under the angle of neural networks (Baldi, 2012) and deep architectures (Vincent et al., 2010), the concepts underlying Autoencoders are very general and go beyond neural implementations. In this chapter, we develop a general framework inspired from Autoencoders, but with elementary mappings being functions from vv-RKHSs (Chapter 2). This novel architecture, proposed in **Laforgue** et al. (2019a) and referred to as *Kernel Autoencoder*, allows in particular to autoencode all data on which a (scalar) kernel can be defined. Before a precise description of the model in Section 3.2, we close this introduction by recalling a few related works.

3.1.3 Related Works

Kernelizing an Autoencoder criterion has also been proposed by Gholami and Hajisami (2016). But their approach differs from ours in many key aspects. First, their model is very restrictive, as it is limited to Autoencoders with two layers, and composed of linear maps only. Second, its training crucially relies on semi-supervised information, while our approach is purely unsupervised. Third, it comes with no theoretical analysis, and within a hashing perspective solely.

Despite a similar denomination, the work by Kampffmeyer et al. (2017) has no direct connection with that exposed in this chapter. It uses standard Autoencoders, and just regularize the learning procedure by aligning the latent code with some predetermined kernel. In the experimental section for instance (Section 4.5 in Chapter 4), we implement our approach on molecules, seen a labeled graphs: each atom corresponds to a node, and edges link chemically bonded atoms. This cannot be done by means of standard Autoencoders, nor by using the work of Kampffmeyer et al. (2017).

As molecule autoencoding is one of the main applications made possible by Kernel Autoencoders, a word about Graph Neural Networks (Gori et al., 2005; Bruna et al., 2013; Li et al., 2015; Kipf and Welling, 2016a; Wu et al., 2019) seems appropriate. Indeed, this domain has gained a lot of attention lately. However, the angle taken is completely different from that of Kernel Autoencoders. The latter first transform the graphs through an implicit feature map associated to a kernel, and then practice autoencoding on this potentially infinite dimensional representation of the graph in the feature space. In opposition, Graph Neural Networks try to adapt the convolution operation in standard Neural Networks to the graph's structure. The goal is to pass messages along the edges of the graph, to agglomerate them and to transform them, so as to create higher level representations of each node. Each aggregation and transformation step is analogous to one layer in a standard Neural Network, and the practitioner finally gets the graph structure of the beginning, but with new features on the nodes. These new features can then be used to perform node classification (Kipf and Welling, 2016a), graph classification (Duvenaud et al., 2015) or link prediction (Kipf and Welling, 2016b). This last example is called "*Graph Autoencoder*", but it has almost nothing to share with our approach. It does not really autoencode graphs, but rather feature vectors of nodes, with the help of an additive graph characterizing the data structure. One approach that could be linked to ours in its ability to generate graphs is that of Valsesia et al. (2018). But it is rather inspired by Generative Adversarial Networks, and specifically tailored to 3D point clouds.

In its will to bridge kernel methods and deep architectures, this work can be linked to that by Mairal et al. (2014); Mairal (2016). However, these contributions are dedicated to image processing. They aim at replacing standard image low-level descriptors such

as Scale-Invariant Feature Transforms and Histograms of Oriented Gradients by kernel feature maps to gain theoretical control. The use of the image structure is critical here, to define convolutional kernels on patches, that is not assumed in our work for instance. We use vv-RKHSs instead, with aim to reconstruct the kernelized representation.

The next section now details at length the model we introduce. Its link with Kernel Principal Component Analysis and a generalization bound are further presented in Sections 3.3 and 3.4 respectively.

3.2 The Kernel Autoencoder

We start from the simplest formulation in which a Kernel Autoencoder is a pair of encoding/decoding functions lying in two different vv-RKHSs, and whose composition approximates the identity function (Section 3.2.1). This approach is further extended to a general framework involving the composition of an arbitrary number of mappings, defined and valued on Hilbert spaces (Section 3.2.2). A crucial application of Kernel Autoencoders arises if the input/output space is itself a RKHS: it allows to perform autoencoding on any type of data, by first mapping it to the RKHS, and then applying a Kernel Autoencoder (Section 3.2.3). The solutions computation, even in infinite dimensional spaces, is made possible by a Representer Theorem and the use of the kernel trick in the output space. These aspects are addressed in Chapter 4.

3.2.1 The 2-layer Kernel Autoencoder

Let $\mathcal{S}_n = (x_1, \ldots, x_n)$ denote a sample of n independent realizations of a random vector X, valued in a separable Hilbert space $(\mathcal{X}_0, \|\cdot\|_{\mathcal{X}_0})$, with unknown distribution P, and such that there exists $M < +\infty, \|X\|_{\mathcal{X}_0} \leq M$ almost surely. On the basis of the training sample $\mathcal{S}_n$, we are interested in constructing a pair of encoding/decoding mappings $\big(f_1 : \mathcal{X}_0 \to \mathcal{X}_1,\ f_2 : \mathcal{X}_1 \to \mathcal{X}_0\big)$, where $(\mathcal{X}_1, \|\cdot\|_{\mathcal{X}_1})$ is the (Hilbert) *representation space*. Just as for standard Autoencoders, we regard as good internal representations the ones that allow for an accurate recovery of the original information in expectation. The problem to be solved states as follows:

$$\min_{\substack{(f_1,f_2)\in\mathcal{H}_1\times\mathcal{H}_2 \\ \|f_1\|_{\mathcal{H}_1}\leq t_1,\ \|f_2\|_{\mathcal{H}_2}\leq t_2}} \epsilon(f_1, f_2) \coloneqq \frac{1}{2}\, \mathbb{E}_{X\sim P} \left\| X - f_2 \circ f_1(X) \right\|_{\mathcal{X}_0}^2 , \tag{3.1}$$

where $\mathcal{H}_1$ and $\mathcal{H}_2$ are two vv-RKHSs, and t_1 and t_2 two positive constants. The vv-RKHS $\mathcal{H}_1$ is associated to an OVK $\mathcal{K}_1 : \mathcal{X}_0 \times \mathcal{X}_0 \to \mathcal{L}(\mathcal{X}_1)$, while vv-RKHS $\mathcal{H}_2$ is associated to $\mathcal{K}_2 : \mathcal{X}_1 \times \mathcal{X}_1 \to \mathcal{L}(\mathcal{X}_0)$.

Figure 3.3 and Remark 3.1 illustrate the parallel and differences between standard and kernel 2-layer Autoencoders. Apart from the difference of functional spaces on which the criterion is optimized, one can already notice that Kernel Autoencoders encompasses standard ones by their applicability scope. Indeed, when Autoencoders are restricted to finite dimensional latent spaces ($\mathbb{R}^4$ or $\mathbb{R}^2$ here), the Kernel Autoencoder only needs Hilbert spaces. They may be finite dimensional (as the internal representation space $\mathcal{X}_1 = \mathbb{R}^2$), or infinite dimensional ($\mathcal{X}_0$). For computational issues (see Chapter 4), this possibility is however limited to the input/output space. Nevertheless, it enlarges the scope of standard Autoencoders, by allowing for the encoding of infinite dimensional objects (*e.g.* functions).

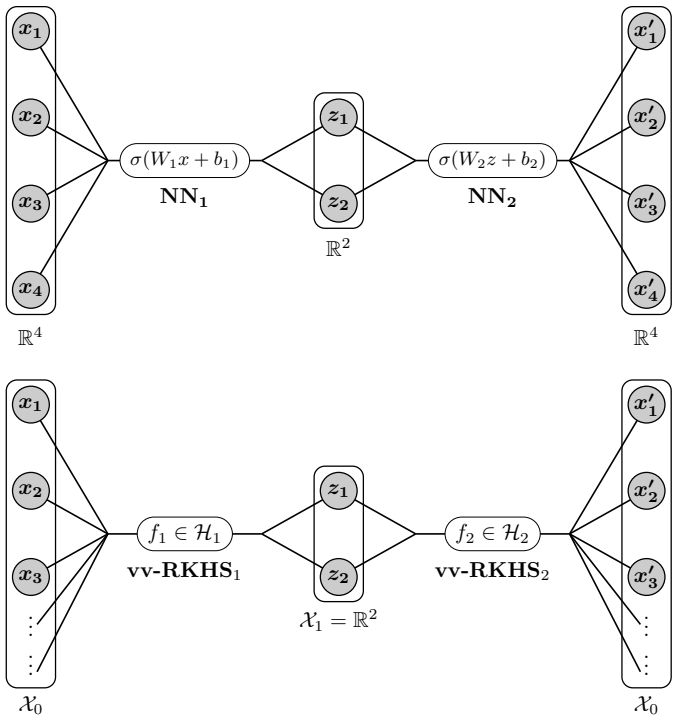


Figure 3.3 – Standard (top) and Kernel (bottom) 2-layer Autoencoders

Following the regularized Empirical Risk Minimization (ERM) paradigm, the expected risk in Problem (3.1) is replaced by its empirical version

$$\hat{\epsilon}_n(f_1, f_2) \coloneqq \frac{1}{2n} \sum_{i=1}^{n} \left\| x_i - f_2 \circ f_1(x_i) \right\|_{\mathcal{X}_0}^2 ,$$

and a penalty term $\Omega(f_1, f_2) \coloneqq (\Lambda_1/2)\|f_1\|_{\mathcal{H}_1}^2 + (\Lambda_2/2)\|f_2\|_{\mathcal{H}_2}^2$ is added instead of the norm constraints (see Theorem 3.4). Solutions to the following regularized ERM problem shall be referred to as *2-layer Kernel Autoencoder*:

$$\min_{(f_1, f_2) \in \mathcal{H}_1 \times \mathcal{H}_2} \hat{\epsilon}_n \left(f_1, f_2\right) + \Omega \left(f_1, f_2\right) . \tag{3.2}$$

Remark 3.1. *A nice remark can be made about the difference induced by the change of elementary mappings, from neural ones to functions in vv-RKHSs. Consider some intermediate representation $x \in \mathbb{R}^d$, that must be mapped to the next intermediate space of dimension p. The neural mapping f_{NN} is parametrized by a matrix $A \in \mathbb{R}^{p \times d}$, whose lines are denoted $(a_j)_{j \leq p}$, and an activation function σ. The kernel mapping $f_{\mathrm{vv-RKHS}}$ is associated to a decomposable OVK $\mathcal{K} = k\boldsymbol{I}_p$. This is equivalent to stacking p independent functions $(f^j)_{j \leq p}$ from the scalar RKHS $\mathcal{H}_k$, associated to the feature map ϕ_k. The next level representations then write*

$$f_{\mathrm{NN}}(x) = \begin{pmatrix} \sigma\left(\langle a_1, x \rangle\right) \\ \vdots \\ \sigma\left(\langle a_p, x \rangle\right) \end{pmatrix}, \quad \textit{and} \quad f_{\mathrm{vv-RKHS}}(x) = \begin{pmatrix} f^1(x) \\ \vdots \\ f^p(x) \end{pmatrix} = \begin{pmatrix} \left\langle f^1, \phi_k(x) \right\rangle \\ \vdots \\ \left\langle f^p, \phi_k(x) \right\rangle \end{pmatrix} .$$

This writing is particularly interesting as it reveals that both mappings are composed of linear transformations and termwise nonlinearities. The neural mapping first uses the linear dot products with the matrix lines, and then applies the nonlinearity σ. The kernel mapping does the opposite, by first embedding the input through the nonlinear feature map ϕ_k, and then apply p linear operations. Which function (linear or nonlinear) is applied first is made even less important as these elementary mappings are meant to be composed several times. Both global functional spaces are then successions of linear mappings and nonlinearities. The functional space introduced by composing functions from vv-RKHSs is thus much closer to the standard Neural Network architecture than it may seem at first glance.

3.2.2 The Multi-layer Kernel Autoencoder

Like for standard Autoencoders, the model described in Section 3.2.1 can be directly extended to more than 2 layers. Let $L \geq 3$, and consider a collection of Hilbert spaces $\mathcal{X}_0, \ldots, \mathcal{X}_L$, with $\mathcal{X}_L = \mathcal{X}_0$. For $0 \leq l \leq L-1$, the space $\mathcal{X}_l$ is supposed to be endowed with an OVK $\mathcal{K}_{l+1} : \mathcal{X}_l \times \mathcal{X}_l \to \mathcal{L}(\mathcal{X}_{l+1})$, associated to a vv-RKHS $\mathcal{H}_{l+1} \subset \mathcal{F}(\mathcal{X}_l, \mathcal{X}_{l+1})$. We then want to minimize $\epsilon(f_1, \ldots, f_L)$ over $\prod_{l=1}^{L} \mathcal{H}_l$. Setting $\Omega(f_1, \ldots, f_L) \coloneqq \sum_{l=1}^{L} (\Lambda_l/2) \|f_l\|^2_{\mathcal{H}_L}$ allows for a direct extension of Problem (3.2):

$$\min_{f_l \in \mathcal{H}_l,\ l \leq L} \quad \frac{1}{2n} \sum_{i=1}^{n} \left\| x_i - f_L \circ \ldots \circ f_1(x_i) \right\|^2_{\mathcal{X}_0} + \sum_{l=1}^{L} \frac{\Lambda_l}{2} \|f_l\|^2_{\mathcal{H}_l}. \tag{3.3}$$

3.2.3 The Kernel Autoencoder Applied to Kernelized Data

So far, and up to the regularization term, the main difference between standard and kernel Autoencoders is the functional space on which the reconstruction criterion is optimized: respectively neural functions or vv-RKHS ones. But what should also be highlighted is that vv-RKHS functions are valued in general Hilbert spaces, while neural functions are restricted to $\mathbb{R}^d$. This enables Kernel Autoencoders to handle data from infinite dimensional Hilbert spaces (*e.g.* function spaces), what standard Autoencoders are unable to do. To our knowledge, this first extension of the autoencoding scheme is novel. However, a specific choice of functional space in the input/output, namely (scalar RKHSs) yields even more interesting applications.

Indeed, assume now that inputs are valued in some space X, without any assumption on its structure. If a scalar kernel k can be defined on X, then we know the existence of a feature map ϕ, and a RKHS $\mathcal{H}_k$ such that $k(x, x') = \langle \phi(x), \phi(x') \rangle_{\mathcal{H}_k}$. So once the inputs have been mapped through ϕ, we are left with an autoencoding problem on points necessarily valued in a Hilbert space. The Kernel Autoencoder of Sections 3.2.1 and 3.2.2 applies, with $\mathcal{X}_0 = \mathcal{H}_k$. This way, one is totally blind to the (real) nature of the inputs, as autoencoding is practiced on the $\phi(x_i)$'s, and it enlarges the applicability scope of Autoencoders to any space X on which a scalar kernel can be defined. Finite dimensional representations can thus be extracted from all types of data, which, to our knowledge, is again a novel extension. Figure 3.4 depicts the procedure, whose associated criterion reads:

$$\min_{f_l \in \mathcal{H}_l,\ l \leq L} \quad \frac{1}{2n} \sum_{i=1}^{n} \left\| \phi(x_i) - f_L \circ \ldots \circ f_1(\phi(x_i)) \right\|^2_{\mathcal{H}_k = \mathcal{X}_0} + \sum_{l=1}^{L} \frac{\Lambda_l}{2} \|f_l\|^2_{\mathcal{H}_l}. \tag{3.4}$$

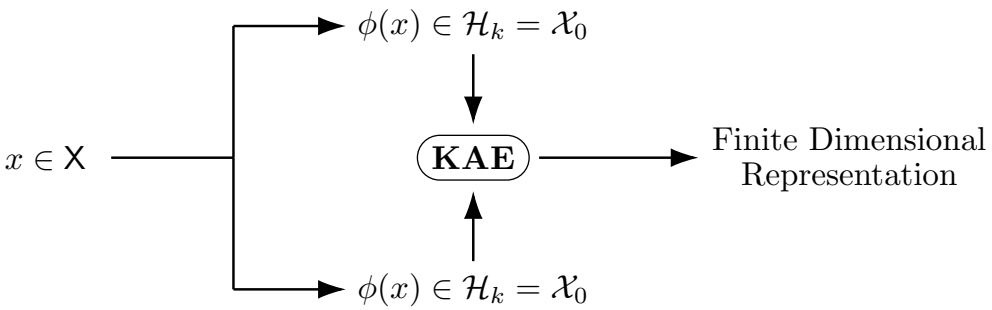


Figure 3.4 – Kernel Autoencoder (KAE) on Kernelized Data

Remark 3.2. *The Kernel Autoencoder on non-kernelized data can be seen as a specific case of Problem (3.4), with $\phi = id$. Therefore, to avoid too may overlapping names, the* Kernel Autoencoder *denomination refers now to the most general case of Section 3.2.3.*

Remark 3.3. *In order to preserve the Autoencoder-like criterion, Problem (3.4) is presented with f_1 element of any vv-RKHS $\mathcal{H}_1$ applied to $\phi(x_i)$. In practice, and for computational purposes, $\mathcal{H}_1$ is often assumed to be associated to a linear decomposable OVK $\mathcal{K}_1$. This is equivalent to write*

$$\min_{f_l \in \mathcal{H}_l,\ l \leq L} \quad \frac{1}{2n} \sum_{i=1}^{n} \left\| \phi(x_i) - f_L \circ \ldots \circ f_1(x_i) \right\|_{\mathcal{H}}^2 + \sum_{l=1}^{L} \frac{\Lambda_l}{2} \|f_l\|_{\mathcal{H}_l}^2,$$

with the only condition on $\mathcal{K}_1$ that it must be decomposable (and ϕ being the canonical feature map associated to k_1). This writing however completely misses the reconstruction intent, or at least does not make it explicit. This is why formulation of Problem (3.4) has been preferred.

The next section now draws an interesting connection between the introduced model, when applied to kernelized data, and Kernel Principal Component Analysis (Schölkopf et al., 1997, 1998).

3.3 Connection to Kernel Principal Component Analysis

Just as Bourlard and Kamp (1988) have shown a mere equivalence between Principal Component Analysis (PCA) and standard 2-layer Autoencoders, a similar link can be established between 2-layer Kernel Autoencoders and Kernel PCA. Throughout this section's analysis, a 2-layer Kernel Autoencoder is considered, applied on data $\phi(x_i)$'s, and with decomposable kernels made of linear scalar kernels and identity operators. Also, there is no penalization (*i.e.* $\Lambda_1 = \Lambda_2 = 0$). We thus want to autoencode data into $\mathbb{R}^p$, after the first embedding through the feature map ϕ.

After recalling the principle of Kernel PCA (Section 3.3.1), we show the equivalence in the simple case where ϕ is valued in a finite dimensional space (Section 3.3.2). Then, arguments on compact operators allow to extend the proof to infinite-valued feature maps (Section 3.3.3).

3.3.1 Kernel Principal Component Analysis

Kernel Principal Component Analysis is an extension of standard PCA to kernelized data introduced by Schölkopf et al. (1997, 1998). If standard PCA boils down to finding the eigenvalues and eigenvectors of the empirical covariance matrix

$$\frac{1}{n}\sum_{i=1}^{n} x_i x_i^\top,$$

Kernel PCA aims at performing so on the empirical covariance operator of the kernelized data (for a centered feature map ϕ):

$$\frac{1}{n}\sum_{i=1}^{n} \phi(x_i)\phi(x_i)^*.$$

The potential infinite dimensionality of the $\phi(x_i)$'s is avoided by noticing that the eigenvectors of the above operator are necessarily linear combinations of the $\phi(x_i)$'s. For a solution (λ_k, v_k), after the reparametrization $v_k = \sum_{j=1}^{n} \alpha_{kj}\phi(x_j)$ for $k, j \leq n$, the vectors $\alpha_{k:} \in \mathbb{R}^n$ are solutions to the eigenproblems

$$K\alpha_k = n\lambda_k \alpha_k,$$

with K the $n \times n$ Gram matrix such that $K_{ij} = \langle \phi(x_i), \phi(x_j) \rangle$.

We now show how the solution to the two layer Kernel Autoencoder with linear and decomposable OVKs links with that of Kernel PCA, assuming first that ϕ is valued in a finite dimensional space.

3.3.2 Finite Dimensional Feature Map

Assume that ϕ is valued in $\mathbb{R}^d$, with $p < d < n$, reminding that p is the dimension of the internal layer. Let $\Phi = (\phi(x_1), \ldots, \phi(x_n))^\top \in \mathbb{R}^{n \times d}$ denote the matrix storing the $\phi(x_i)^\top$ to autoencode in rows. Note that $K_\phi = \Phi\Phi^\top \in \mathbb{R}^{n \times n}$ corresponds to the Gram matrix associated to ϕ. As shall be seen in Section 4.1, the optimal encoder f_1 and decoder f_2 have a specific form, so that they only depend on two coefficient matrices, $C_1 \in \mathbb{R}^{n \times p}$ and $C_2 \in \mathbb{R}^{n \times d}$ respectively. Equipped with this notation, one has: $Y = f_1(\Phi) = \Phi\Phi^\top C_1 \in \mathbb{R}^{n \times p}$, and $\widetilde{\Phi} = f_2(Y) = YY^\top C_2 \in \mathbb{R}^{n \times d}$. Without penalization, the goal is then to minimize in C_1 and C_2:

$$\left\| \Phi - \widetilde{\Phi} \right\|_{\mathrm{Fr}}^2.$$

Reconstructed matrix $\widetilde{\Phi}$ being at most of rank p, we know from Eckart-Young Theorem that the best possible $\widetilde{\Phi}$ is given by

$$\Phi^* = U\Sigma_p V^\top,$$

where $U \in \mathbb{R}^{n \times d}$, $\Sigma \in \mathbb{R}^{d \times d}$, and $V^\top \in \mathbb{R}^{d \times d}$ are the thin Singular Value Decomposition (SVD) of Φ such that $\Phi = U\Sigma V^\top$, and Σ_p is equal to Σ, but with the $d - p$ smallest singular values zeroed.

It suffices now to prove that there exists a couple of coefficient matrices (C_1^*, C_2^*) such that $f_{C_2^*} \circ f_{C_1^*}(\Phi) = \Phi^*$. One can verify that

$$\begin{cases} C_1^* = U_p \overline{\Sigma}_p^{-3/2}, \\ C_2^* = UV^\top, \end{cases}$$

with $U_p \in \mathbb{R}^{n \times p}$ storing only the p largest eigenvectors of K_ϕ, and $\overline{\Sigma}_p \in \mathbb{R}^{p \times p}$ the $p \times p$ top left block of Σ_p, satisfy it. Finally, the optimal encoding returned is

$$Y^* = f_{C_1^*}(\Phi) = \left(\sqrt{\sigma_1} u_1, \ldots, \sqrt{\sigma_p} u_p\right)$$

with $u_1, \ldots, u_p$ the p largest eigenvectors of K_ϕ, and $(\sigma_i)_{i \leq p}$ the diagonal entries of $\overline{\Sigma}_p$. It must be compared to Kernel PCA's new representations:

$$Y^*_{\text{KPCA}} = \left(\sigma_1 u_1, \ldots, \sigma_p u_p\right).$$

We have thus shown that a specific instance of Kernel Autoencoder can be solved explicitly using a SVD, and that the optimal coding returned is close to the one output by Kernel PCA.

3.3.3 Infinite Dimensional Feature Map

Assume now that ϕ is valued in a general Hilbert space $\mathcal{H}$. Φ is now seen as the linear operator from $\mathcal{H}$ to $\mathbb{R}^n$ such that for all $\alpha \in \mathcal{H}$

$$\Phi\alpha = \left(\left\langle \alpha, \phi(x_1) \right\rangle_{\mathcal{H}}, \ldots, \left\langle \alpha, \phi(x_n) \right\rangle_{\mathcal{H}}\right) \in \mathbb{R}^n.$$

Since Theorem 4.1 makes no assumption on the dimensionality, everything stated in the finite dimensional scenario applies, except that $C_2 \in \mathcal{L}(\mathcal{H}, \mathbb{R}^n)$, and that we minimize instead of the Frobenius norm the Hilbert-Schmidt one

$$\left\| \Phi - \widetilde{\Phi} \right\|_{\text{HS}}^2 = \sum_{i=1}^{n} \left\| \phi(x_i) - \tilde{\phi}_i \right\|_{\mathcal{H}}^2,$$

with $\tilde{\phi}_i \in \mathcal{H}$ such that $(\widetilde{\Phi}\alpha)_i = \langle \tilde{\phi}_i, \alpha \rangle_{\mathcal{H}}$ for $i \leq n$. We then need an equivalent of Eckart-Young Theorem. It still holds since its proof only requires the existence of a SVD for any operator, which is granted in our case since we deal with compact operators (they have finite rank lower or equal than n). The end of the proof is analogous to the finite dimensional case.

We have thus shown that the Kernel Autoencoder model we have introduced can be regarded as a deep version of Kernel PCA. With two layers only, and specific choices of kernels, they are equivalent. Adding layers have shown in the standard Autoencoder case to improve the performances compared to standard PCA (Hinton and Salakhutdinov, 2006). We can hope the same mechanism to work in our setting.

The next section is dedicated to the derivation of a generalization bound by means of Rademacher averages. The two main difficulties addressed are: 1) the compositional nature of the functional space on which the reconstruction criterion is optimized, and 2) the potential infinite dimensionality of the inputs/outputs.

3.4 Theoretical Guarantees

In this section, we establish a generalization bound for the Kernel Autoencoders in terms of reconstruction error. The difficulty of having compositions of vv-RKHS functions is addressed by using the work of Maurer and Pontil (2016), adapted to our infinite dimensional setting. The analysis starts with a classical theorem, stating the equivalence between constrained and penalized problems. The bound is then stated at the end of Section 3.4.1, the technical elements of the proof being deferred to Section 3.4.2.

3.4.1 Generalization Bound

While the algorithmic formulation aims at minimizing the regularized Problem (3.3), the subsequent theoretical analysis focuses on the constrained Problem (3.1). Theorem 3.4 relates the solutions to both problems, so that bounds derived in the latter setting also apply to numerical solutions of the first one.

Theorem 3.4. *Let $V : \mathcal{H}_1 \times \ldots \times \mathcal{H}_L \to \mathbb{R}$ be an arbitrary function. Consider the two problems:*

$$\min_{f_l \in \mathcal{H}_l} \left\{ V(f_1, \ldots, f_L) + \sum_{l=1}^{L} \frac{\Lambda_l}{2} \|f_l\|_{\mathcal{H}_l}^2 \right\}, \tag{3.5}$$

and

$$\min_{\substack{f_l \in \mathcal{H}_l \\ \|f_l\|_{\mathcal{H}_l} \leq t_l}} V(f_1, \ldots, f_L). \tag{3.6}$$

Then, for any $(\Lambda_l)_{l \leq L} \in \mathbb{R}_+^L$, there exists $(t_l)_{l \leq L} \in \mathbb{R}_+^L$ such that any (respectively, local) solution to Problem (3.5) is also a (respectively, local) solution to Problem (3.6).

Proof. Let $(\Lambda_l)_{l \leq L} \in \mathbb{R}_+^L$, and $(f_l^*)_{l \leq L}$ a solution to Problem (3.5) for this choice of regularizers. For $l \leq L$, let $t_l = \|f_l^*\|_{\mathcal{H}_l}$. We shall prove that $(f_l^*)_{l \leq L}$ is also a solution to Problem (3.6) for this choice of constraints.

Consider $(f_l)_{l \leq L}$ satisfying Problem (3.6)'s constraints. For $l \leq L$, it holds $\|f_l\|_{\mathcal{H}_l} \leq t_l = \|f_l^*\|_{\mathcal{H}_l}$. Hence, we have $\sum_l \Lambda_l \|f_l\|_{\mathcal{H}_l}^2 \leq \sum_l \Lambda_l \|f_l^*\|_{\mathcal{H}_l}^2$. On the other hand, by definition of the f_l^*'s, it holds

$$V(f_1, \ldots, f_L) + \sum_{l=1}^{L} \frac{\Lambda_l}{2} \|f_l\|_{\mathcal{H}_l}^2 \;\geq\; V(f_1^*, \ldots, f_L^*) + \sum_{l=1}^{L} \frac{\Lambda_l}{2} \|f_l^*\|_{\mathcal{H}_l}^2.$$

Thus, we necessarily have: $V(f_1, \ldots, f_L) \;\geq\; V(f_1^*, \ldots, f_L^*)$.

A similar argument can be used for local solutions, details are left to the reader.

Although this result may appear rather simple, we thought it was worth mentioning as our setting is particularly unfriendly: the objective function V is not assumed convex, and the variables f_l are infinite dimensional. As a consequence, in absence of additional assumptions, the converse statement (that solutions to Problem (3.6) are also solutions to Problem (3.5) for a suitable choice of Λ_l's) is not guaranteed. The proof indeed relies on the existence of Lagrangian multipliers, which has been shown when the variables are finite dimensional (KKT conditions), or when the objective function is convex (Bauschke et al., 2011), but is not ensured in our case. □

In order to establish generalization bound results for empirical minimizers in the present setting, we now define two key quantities involved in the proof, *i.e.* Rademacher and Gaussian averages for classes of Hilbert-valued functions.

Definition 3.5. *Let $\mathcal{X}$ be any measurable space, and H a separable Hilbert space. Let $\mathcal{C}$ be a class of measurable functions $h : \mathcal{X} \to H$. Recall that $\mathcal{S}_n = \{x_1, \dots, x_n\} \in \mathcal{X}^n$ is our sample of interest. Let $\boldsymbol{\sigma}_1, \dots, \boldsymbol{\sigma}_n$ be $n \geq 1$ independent H-valued Rademacher variables and define:*

$$\widehat{\mathscr{R}}(\mathcal{C}, \mathcal{S}_n) = \mathbb{E}_{\boldsymbol{\sigma}} \left[\sup_{h \in \mathcal{C}} \frac{1}{n} \sum_{i=1}^{n} \left\langle \boldsymbol{\sigma}_i, h(x_i) \right\rangle_H \right].$$

If $H = \mathbb{R}$, it is the classical Rademacher average (see *e.g.* Chapter 1, or Mohri et al. (2012) p.34), while, when $H = \mathbb{R}^p$, it corresponds to the expectation of the supremum of the sum of the Rademacher averages over the p components of h (see Definition 2.1 in Maurer and Pontil (2016)). If H is an infinite dimensional Hilbert space with countable orthonormal basis $(e_k)_{k \in \mathbb{N}}$, we have:

$$\widehat{\mathscr{R}}(\mathcal{C}, \mathcal{S}_n) = \mathbb{E}_{\boldsymbol{\sigma}} \left[\sup_{h \in \mathcal{C}} \frac{1}{n} \sum_{i=1}^{n} \sum_{k=1}^{\infty} \sigma_{i,k} \left\langle h(x_i), e_k \right\rangle_H \right].$$

The Gaussian counterpart of $\widehat{\mathscr{R}}(\mathcal{C}, \mathcal{S}_n)$, obtained by replacing Rademacher random variables/processes with standard H-valued Gaussian ones, is denoted by $\widehat{\mathscr{G}}(\mathcal{C}, \mathcal{S}_n)$ throughout the paper.

For the sake of simplicity, results in the rest of the subsection are derived in the 2-layer case solely, with $\mathcal{X}_1$ finite dimensional (*i.e.* $\mathcal{X}_1 = \mathbb{R}^p$), although the approach remains valid for deeper architectures.

A few notation more is needed (recall that the norm of random variable X is almost surely bounded by M, see Section 3.2.1). Let

$$\mathcal{H}_{1,t_1} \coloneqq \left\{ f_1 \in \mathcal{H}_1 : \|f_1\|_{\mathcal{H}_1} \leq t_1 \right\},$$

and similarly

$$\mathcal{H}_{2,t_2} \coloneqq \left\{ f_2 \in \mathcal{H}_2 : \|f_2\|_{\mathcal{H}_2} \leq t_2, \ \sup_{y \in \mathbb{R}^p} \|f_2(y)\|_{\mathcal{X}_0} \leq M \right\}.$$

We also use the notation

$$\mathcal{H}_{t_1,t_2} = \mathcal{H}_{1,t_1} \circ \mathcal{H}_{2,t_2} = \left\{ h \in \mathcal{F}(\mathcal{X}_0, \mathcal{X}_0) : \exists (f_1, f_2) \in \mathcal{H}_{1,t_1} \times \mathcal{H}_{2,t_2}, \ h = f_2 \circ f_1 \right\}.$$

To simplify notation, ϵ (and $\hat{\epsilon}_n$) may be abusively considered as a functional with one or two arguments:

$$\epsilon(f_1, f_2) = \epsilon(f_2 \circ f_1) = \frac{1}{2} \, \mathbb{E}_{X \sim P} \left\| X - f_2 \circ f_1(X) \right\|_{\mathcal{X}_0}^2.$$

Finally, let $\hat{h}_n$ denote the minimizer of $\hat{\epsilon}_n$ over $\mathcal{H}_{t_1,t_2}$, and ϵ^* the infimum of ϵ on the same functional space.

The following assumptions on $\mathcal{K}_1$ and $\mathcal{K}_2$ are needed to establish the bound stated below.

Assumption 3.6. *There exists $K < +\infty$ such that:*

$$\forall x \in \mathcal{X}_0, \ \ \mathbf{Tr}\Big(\mathcal{K}_1(x,x)\Big) \leq Kp.$$

Assumption 3.7. *There exists $L < +\infty$ such that for all y, y' in $\mathbb{R}^p$:*

$$\mathbf{Tr}\Big(\mathcal{K}_2(y,y) - 2\mathcal{K}_2(y,y') + \mathcal{K}_2(y',y')\Big) \leq L^2 \ \|y - y'\|^2_{\mathbb{R}^p}.$$

The generalization bound then states as follows.

Theorem 3.8. *Let $\mathcal{K}_1$ and $\mathcal{K}_2$ be OVKs satisfying Assumptions 3.6 and 3.7 respectively. Then, there exists a universal constant $C_0 < +\infty$ such that, for any $0 < \delta < 1$, we have with probability at least $1 - \delta$:*

$$\epsilon(\hat{h}_n) - \epsilon^* \leq C_0 LMt_1t_2\sqrt{\frac{Kp}{n}} + 24M^2\sqrt{\frac{\log(2)/\delta}{2n}}.$$

Remark 3.9. *Attention should be paid to the fact that constants in Theorem 3.8 appear in a very interpretable fashion: the less spread the input (the smaller the constant M), the more restrictive the constraints on the functions (the smaller K, L, t_1 and t_2), and the smaller the internal dimension p, the sharper the bound.*

The technical details of Theorem 3.8's proof are now to be given.

3.4.2 Technical Proof

The proof sketch is as follows:

- first use standard arguments to bound the excess risk by Rademacher averages (see Chapter 1, Section 1.1 therein) and turn to Gaussian averages,
- then extend Theorem 2 in Maurer (2014) to the infinite dimensional output case,
- finally bound each term appeared in the extension of the above theorem.

Standard Rademacher Generalization Bound.

Let loss ℓ denote the squared norm on $\mathcal{X}_0$: $\forall x \in \mathcal{X}_0, \ell(x) = \|x\|^2_{\mathcal{X}_0}$. Notice that, on the set considered, the mapping ℓ is $2M$-Lipschitz, and: $\ell(x_i - h(x_i)) - \ell(x_{i'} - h(x_{i'})) \leq 4M^2$. Hence, by applying McDiarmid's inequality, together with standard arguments in the statistical learning literature (symmetrization/randomization tricks, see *e.g.* Theorem 3.1 in Mohri et al. (2012), or again Section 1.1), one may show that, for any $\delta \in]0,1[$, it holds with probability at least $1 - \delta$:

$$\frac{1}{2}\left(\epsilon(\hat{h}_n) - \epsilon^*\right) \leq \sup_{h \in \mathcal{H}_{t_1,t_2}} |\epsilon(h) - \hat{\epsilon}_n(h)| \leq 2\widehat{\mathscr{R}}\Big(\ell \circ (\mathrm{id} - \mathcal{H}_{t_1,t_2}), \mathcal{S}_n\Big) + 12M^2\sqrt{\frac{\ln\frac{2}{\delta}}{2n}}. \quad (3.7)$$

The subsequent results shall provide tools to properly bound the quantity

$$\widehat{\mathscr{R}}\Big(\ell \circ (\mathrm{id} - \mathcal{H}_{t_1,t_2}), \mathcal{S}_n\Big).$$

Operations on the Rademacher Average.

As a first go, we state a preliminary lemma that establishes a comparison between Rademacher and Gaussian averages.

Lemma 3.10. *We have:* $\forall n \geq 1$,

$$\widehat{\mathscr{R}}(\mathcal{C}, \mathcal{S}_n) \leq \sqrt{\frac{\pi}{2}}\, \widehat{\mathscr{G}}(\mathcal{C}, \mathcal{S}_n).$$

Proof. The proof is based on the fact that $\gamma_{i,k}$ and $\sigma_{i,k}\left|\gamma_{i,k}\right|$ have the same distribution, combined with Jensen's inequality. See also Lemma 4.5 in Ledoux and Talagrand (1991). □

Hence, the application of the lemma above yields:

$$\begin{aligned}
\widehat{\mathscr{R}}\Big(\ell \circ (\mathrm{id} - \mathcal{H}_{t_1,t_2}), \mathcal{S}_n\Big) &\leq 2\sqrt{2}M\ \widehat{\mathscr{R}}\Big(\mathrm{id} - \mathcal{H}_{t_1,t_2}, \mathcal{S}_n\Big), \qquad (3.8)\\
&\leq 2\sqrt{2}M\ \left[\widehat{\mathscr{R}}\Big(\{\mathrm{id}\}, \mathcal{S}_n\Big) + \widehat{\mathscr{R}}\Big(\mathcal{H}_{t_1,t_2}, \mathcal{S}_n\Big)\right],\\
&\leq 2\sqrt{2}M\ \widehat{\mathscr{R}}\Big(\mathcal{H}_{t_1,t_2}, \mathcal{S}_n\Big),\\
&\leq 2\sqrt{\pi}M\ \widehat{\mathscr{G}}\Big(\mathcal{H}_{t_1,t_2}, \mathcal{S}_n\Big), \qquad (3.9)
\end{aligned}$$

where Equation (3.8) directly results from Corollary 4 in Maurer (2016) (observing that, even if they do not take their values in $\ell_2(\mathbb{N})$ but in the separable Hilbert space $\mathcal{X}_0$, the functions $h(x)$ can replaced by the square-summable sequence $(\langle h(x), e_k\rangle)_{k\in\mathbb{N}}$), and Equation (3.9) is a consequence of Lemma 3.10.

It now remains to bound $\widehat{\mathscr{G}}(\mathcal{H}_{t_1,t_2}, \mathcal{S}_n)$ using an extension of a result established in Maurer (2014), that applies to classes of functions valued in $\mathbb{R}^m$ only, while functions in $\mathcal{H}_{t_1,t_2}$ are Hilbert-valued.

Extension of Maurer's Chain Rule.

The result stated below extends Theorem 2 in Maurer (2014) to the Hilbert-valued situation.

Theorem 3.11. *Let H be a Hilbert space, X a H-valued standard Gaussian random vector, and $f : H \to \mathbb{R}$ a L-Lipschitz mapping. We have:*

$$\forall t > 0, \qquad \mathbb{P}\left(\left|f(X) - \mathbb{E}f(X)\right| > t\right) \leq \exp\left(-\frac{2t^2}{\pi^2 L^2}\right).$$

Proof. It is a direct extension of Corollary 2.3 in Pisier (1986), which states the result for $H = \mathbb{R}^N$ only, observing that the proof given therein actually makes no use of the assumption of finite dimensionality of H, and thus remains valid in our case. The reason why authors did not establish this general version in their work is probably because they only needed the $\mathbb{R}^N$ version for their purpose. Up to constants, it can also be viewed as an extension of Theorem 4 in Maurer (2014). □

We now introduce quantities involved in the rest of the analysis, see Definition 1 in Maurer (2014).

Definition 3.12. *Let $Y \subset \mathbb{R}^n$, H be a Hilbert space, $Z \subset H$, and γ be a H-valued standard Gaussian variable/process. We set:*

$$D(Y) = \sup_{y,y' \in Y} \|y - y'\|_{\mathbb{R}^n},$$

$$G(Z) = \sup_{z \in Z} \mathbb{E}_\gamma \left[\langle \gamma, z \rangle_H \right].$$

If $\mathcal{H}$ a class of functions from Y to H, we set:

$$L(\mathcal{H}, Y) = \sup_{h \in \mathcal{H}} \sup_{y,y' \in Y,\ y \neq y'} \frac{\|h(y) - h(y')\|_H}{\|y - y'\|_{\mathbb{R}^n}},$$

$$R(\mathcal{H}, Y) = \sup_{y,y' \in Y,\ y \neq y'} \mathbb{E}_{\boldsymbol{\gamma}} \left[\sup_{h \in \mathcal{H}} \frac{\left\langle \boldsymbol{\gamma}, h(y) - h(y') \right\rangle_H}{\|y - y'\|_{\mathbb{R}^n}} \right].$$

The next result establishes useful relationships between the quantities introduced above.

Theorem 3.13. *Let $Y \subset \mathbb{R}^n$ be a finite set, H a Hilbert space and $\mathcal{H}$ a finite class of functions $h : Y \to H$. Then, there are universal constants C_1 and C_2 such that, for any $y_0 \in Y$:*

$$G(\mathcal{H}(Y)) \leq C_1 L(\mathcal{H}, Y) G(Y) + C_2 R(\mathcal{H}, Y) D(Y) + G(\mathcal{H}(y_0)).$$

Proof. This result is a direct extension of Theorem 2 in Maurer (2014) for H-valued functions. The only part in the proof depending on the dimensionality of H is Theorem 4 in the same paper, whose extension to any Hilbert space is proved in Theorem 3.11. Indeed, considering $X_y = (\sqrt{2}/\pi L(F, Y)) \sup_{f \in F} \langle \gamma, f(y) \rangle$ (using the same notation as in Maurer (2014)) allows to finish the proof like in the finite dimensional case. □

Let $\mathcal{H}'_{1,t_1}$ be the set of functions from $(\mathcal{X}_0)^n$ to $\mathbb{R}^{np}$ that take as input $\mathcal{S}_n = (x_1, \ldots, x_n)$ and return $(f(x_1), \ldots, f(x_n))$, $f \in \mathcal{H}_{1,t_1}$. Let $Y = \mathcal{H}'_{1,t_1}(\mathcal{S}_n) \subset \mathbb{R}^{np}$, and $H = (\mathcal{X}_0)^n$, which is a Hilbert space. Let $\mathcal{H} = \mathcal{H}'_{2,t_2}$ be the set of functions from $\mathbb{R}^{np}$ to $(\mathcal{X}_0)^n$ that take as input $(y_1, \ldots, y_n)$ and return $(f_2(y_1), \ldots, f_2(y_n))$, $f_2 \in \mathcal{H}_{2,t_2}$. Finally, let $y_0 = (0_{\mathbb{R}^p}, \ldots, 0_{\mathbb{R}^p})$ (it actually belongs to $\mathcal{H}'_{1,t_1}(\mathcal{S}_n)$ since the null function is in $\mathcal{H}'_{1,t_1}$). Theorem 3.13 entails that:

$$\begin{aligned} G\Big(\mathcal{H}'_{2,t_2}\Big(\mathcal{H}'_{1,t_1}(\mathcal{S}_n)\Big)\Big) \leq\ & C_1 L\Big(\mathcal{H}'_{2,t_2}, \mathcal{H}'_{1,t_1}(\mathcal{S}_n)\Big) G\Big(\mathcal{H}'_{1,t_1}(\mathcal{S}_n)\Big) \\ & + C_2 R\Big(\mathcal{H}'_{2,t_2}, \mathcal{H}'_{1,t_1}(\mathcal{S}_n)\Big) D\Big(\mathcal{H}'_{1,t_1}(\mathcal{S}_n)\Big) + G\Big(\mathcal{H}'_{2,t_2}(0)\Big), \end{aligned}$$

and

$$\begin{aligned} \widehat{\mathscr{G}}\Big(\mathcal{H}_{t_1,t_2}, \mathcal{S}_n\Big) \leq\ & C_1 L\Big(\mathcal{H}'_{2,t_2}, \mathcal{H}'_{1,t_1}(\mathcal{S}_n)\Big) \widehat{\mathscr{G}}\Big(\mathcal{H}_{1,t_1}, \mathcal{S}_n\Big) \\ & + \frac{C_2}{n} R\Big(\mathcal{H}'_{2,t_2}, \mathcal{H}'_{1,t_1}(\mathcal{S}_n)\Big) D\Big(\mathcal{H}'_{1,t_1}(\mathcal{S}_n)\Big) + \frac{1}{n} G\Big(\mathcal{H}'_{2,t_2}(0)\Big). \end{aligned} \tag{3.10}$$

We now bound each term appearing on the right-hand side.

Bounding each Term in Equation (3.10).

Bounding $L(\mathcal{H}'_{2,t_2}, \mathcal{H}'_{1,t_1}(\mathcal{S}_n))$. Consider the following assumption, denoting by $\|.\|_*$ the operator norm of any bounded linear operator.

Assumption 3.14. *There exists a constant $L < +\infty$ such that: $\forall (y, y') \in \mathbb{R}^p$,*

$$\left\| \mathcal{K}_2(y,y) - 2\mathcal{K}_2(y,y') + \mathcal{K}_2(y',y') \right\|_* \leq L^2 \ \|y - y'\|^2_{\mathbb{R}^p}.$$

This assumption is not very compelling since it is enough for $\mathcal{K}_2$ to be the sum of T decomposable kernels $k_t(\cdot,\cdot)A_t$ such that the scalar feature maps ϕ_t are L_t-Lipschitz (the feature map of the Gaussian kernel with bandwidth $1/(2\sigma^2)$ has Lipschitz constant $1/\sigma$ for instance), and the A_t operators have finite operator norms σ_t.

Indeed, we would have then: $\forall z \in \mathcal{X}_0$,

$$\begin{aligned}
\left\| \Big(\mathcal{K}_2(y,y) - 2\mathcal{K}_2(y,y') + \mathcal{K}_2(y',y') \Big) z \right\|_{\mathcal{X}_0} &= \left\| \left(\sum_{t=1}^{T} \|\phi_t(y) - \phi_t(y')\|^2 A_t \right) z \right\|_{\mathcal{X}_0}, \\
&\leq \sum_{t=1}^{T} \|\phi_t(y) - \phi_t(y')\|^2 \sigma_t \ \|z\|_{\mathcal{X}_0}, \\
\left\| \Big(\mathcal{K}_2(y,y) - 2\mathcal{K}_2(y,y') + \mathcal{K}_2(y',y') \Big) z \right\|_{\mathcal{X}_0} &\leq \left(\sum_{t=1}^{T} L_t^2 \sigma_t \right) \|y - y'\|^2_{\mathbb{R}^p} \ \|z\|_{\mathcal{X}_0}, \\
\left\| \mathcal{K}_2(y,y) - 2\mathcal{K}_2(y,y') + \mathcal{K}_2(y',y') \right\|_* &\leq \left(\sum_{t=1}^{T} L_t^2 \sigma_t \right) \|y - y'\|^2_{\mathbb{R}^p}.
\end{aligned}$$

Let $\mathcal{K}_2$ satisfying Assumption 3.14, $g \in \mathcal{H}'_{2,t_2}$ and $(\boldsymbol{y}, \boldsymbol{y'}) \in \mathbb{R}^{np}$. We have:

$$\begin{aligned}
&\left\| g(\boldsymbol{y}) - g(\boldsymbol{y'}) \right\|^2_{(\mathcal{X}_0)^n} \\
&= \sum_{i=1}^{n} \left\| g(y_i) - g(y'_i) \right\|^2_{\mathcal{X}_0}, \\
&= \sum_{i=1}^{n} \Big\langle g(y_i) - g(y'_i), g(y_i) - g(y'_i) \Big\rangle_{\mathcal{X}_0}, \\
&= \sum_{i=1}^{n} \Big\langle \mathcal{K}_{2y_i}(g(y_i) - g(y'_i)), g \Big\rangle_{\mathcal{H}_2} - \Big\langle \mathcal{K}_{2y'_i}(g(y_i) - g(y'_i)), g \Big\rangle_{\mathcal{H}_2}, && (3.11) \\
&\leq \|g\|_{\mathcal{H}_2} \sum_{i=1}^{n} \left\| \mathcal{K}_{2y_i}(g(y_i) - g(y'_i)) - \mathcal{K}_{2y'_i}(g(y_i) - g(y'_i)) \right\|_{\mathcal{H}_2}, && (3.12) \\
&\leq t_2 \sum_{i=1}^{n} \sqrt{\Big\langle g(y_i) - g(y'_i), \Big(\mathcal{K}_2(y_i,y_i) - 2\mathcal{K}_2(y_i,y'_i) + \mathcal{K}_2(y'_i,y'_i) \Big)(g(y_i) - g(y'_i)) \Big\rangle_{\mathcal{X}_0}}, && (3.13) \\
&\leq L t_2 \sum_{i=1}^{n} \left\| g(y_i) - g(y'_i) \right\|_{\mathcal{X}_0} \|y_i - y'_i\|_{\mathbb{R}^p}, && (3.14) \\
&\leq L t_2 \left\| g(\boldsymbol{y}) - g(\boldsymbol{y'}) \right\|_{(\mathcal{X}_0)^n} \left\| \boldsymbol{y} - \boldsymbol{y'} \right\|_{\mathbb{R}^{np}}, && (3.15)
\end{aligned}$$

where Equation (3.11) results from the reproducing property in vv-RKHSs (Equation (2.1) in Micchelli and Pontil (2005)), Equation (3.12) follows from Cauchy-Schwarz inequality, Equation (3.13) is again a consequence of the reproducing property (*e.g.* Equation (2.3) in Micchelli and Pontil (2005)), Equation (3.14) can be deduced from Assumption 3.14 and Equation (3.15) is a consequence of Cauchy-Schwarz inequality as well. Hence, we finally have:

$$\left\|g(\boldsymbol{y}) - g(\boldsymbol{y'})\right\|_{(\mathcal{X}_0)^n} \leq Lt_2 \ \left\|\boldsymbol{y} - \boldsymbol{y'}\right\|_{\mathbb{R}^{np}}$$

and consequently

$$L\Big(\mathcal{H}'_{2,t_2}, \mathcal{H}'_{1,t_1}(\mathcal{S}_n)\Big) \leq L\Big(\mathcal{H}'_{2,t_2}, \mathbb{R}^{np}\Big) \leq Lt_2. \tag{3.16}$$

Bounding $\widehat{\mathscr{G}}\Big(\mathcal{H}_{1,t_1}, \mathcal{S}_n\Big)$. Consider the assumption below.

Assumption 3.15. *There exists a constant $K < +\infty$ such that: $\forall x \in \mathcal{X}_0$,*

$$\mathbf{Tr}\Big(\mathcal{K}_1(x,x)\Big) \leq Kp.$$

This assumption is mild as well, since it is satisfied fro instance by the sum of T decomposable kernels $k_t(\cdot,\cdot)A_t$ such that the scalar kernels are bounded by κ_t (as X is supposed to be bounded, any continuous kernel is valid). Indeed, we have: $\forall x \in \mathcal{X}_0$,

$$\mathbf{Tr}\Big(\mathcal{K}_1(x,x)\Big) = \sum_{t=1}^{T} k_t(x,x)\ \mathbf{Tr}(A_t) \leq \left(\sum_{t=1}^{T} \kappa_t \|A_t\|_\infty\right) p.$$

Let $\mathcal{K}_1$ an OVK satisfying Assumption 3.15 and be such that $\mathcal{H}_1$ is separable. We then know that there exists $\Phi \in \mathcal{L}(\ell_2(\mathbb{N}), \mathbb{R}^p)$ such that: $\forall (x,x') \in \mathcal{X}_0$, $\mathcal{K}_1(x,x') = \Phi(x)\Phi^*(x')$ and $\forall f_1 \in \mathcal{H}_1, \exists u \in \ell_2(\mathbb{N})$ such that $f_1(\cdot) = \Phi(\cdot)u$, $\|f_1\|_{\mathcal{H}_1} = \|u\|_{\ell_2}$ (see Micchelli and Pontil (2005)). We have:

$$n\ \widehat{\mathscr{G}}\Big(\mathcal{H}'_{1,t_1}, \mathcal{S}_n\Big) \tag{3.17}$$

$$= \mathbb{E}_{\boldsymbol{\gamma}}\left[\sup_{f_1 \in \mathcal{H}_{1,t_1}} \sum_{i=1}^{n} \Big\langle \boldsymbol{\gamma}_i, f_1(x_i)\Big\rangle_{\mathbb{R}^p}\right] = \mathbb{E}_{\boldsymbol{\gamma}}\left[\sup_{\|u\|_{\ell_2} \leq t_1} \sum_{i=1}^{n}\sum_{k=1}^{p} \gamma_{i,k}, \Big\langle \Phi(x_i)u, e_k\Big\rangle_{\mathbb{R}^p}\right],$$

$$= \mathbb{E}_{\boldsymbol{\gamma}}\left[\sup_{\|u\|_{\ell_2} \leq t_1} \left\langle u, \sum_{i=1}^{n}\sum_{k=1}^{p} \gamma_{i,k}\Phi^*(x_i)e_k \right\rangle_{\ell_2}\right] \leq t_1\ \mathbb{E}_{\boldsymbol{\gamma}}\left[\left\|\sum_{i=1}^{n}\sum_{k=1}^{p} \gamma_{i,k}\Phi^*(x_i)e_k\right\|_{\ell_2}\right], \tag{3.18}$$

$$\leq t_1 \sqrt{\mathbb{E}_{\boldsymbol{\gamma}}\left[\left\|\sum_{i=1}^{n}\sum_{k=1}^{p} \gamma_{i,k}\Phi^*(x_i)e_k\right\|_{\ell_2}^2\right]} \leq t_1 \sqrt{\sum_{i=1}^{n}\sum_{k=1}^{p}\Big\langle \mathcal{K}(x_i,x_i)e_k, e_k\Big\rangle_{\mathbb{R}^p}}, \tag{3.19}$$

$$\leq t_1 \sqrt{\sum_{i=1}^{n} \mathbf{Tr}\Big(\mathcal{K}_1(x_i,x_i)\Big)}, \tag{3.20}$$

$$\leq t_1\sqrt{nKp}, \tag{3.21}$$

where Equation (3.18) follows from Cauchy-Schwarz inequality, Equation (3.19) from Jensen's inequality and the orthogonality of the Gaussian variables introduced, and Equation (3.21) from Assumption 3.15. Finally, we have:

$$\widehat{\mathscr{G}}\Big(\mathcal{H}_{1,t_1}, \mathcal{S}_n\Big) \leq t_1 \sqrt{\frac{Kp}{n}}. \tag{3.22}$$

Bounding $R(\mathcal{H}'_{2,t}, \mathcal{H}'_{1,s}(\mathcal{S}_n))$. Consider the following assumption.

Assumption 3.16. *There exists a constant $L < +\infty$ such that: $\forall (y, y') \in \mathbb{R}^p$,*

$$\mathbf{Tr}\Big(\mathcal{K}_2(y,y) - 2\mathcal{K}_2(y,y') + \mathcal{K}_2(y',y')\Big) \leq L^2 \ \|y - y'\|^2_{\mathbb{R}^p}.$$

Suppose that the OVK $\mathcal{K}_2$ is the sum of T decomposable kernels $k_t(\cdot,\cdot)A_t$ such that the scalar feature maps ϕ_t are L_t-Lipschitz and the A_t operators are trace class. Then, we have: $\forall (y, y') \in \mathbb{R}^p$,

$$\begin{aligned}\mathbf{Tr}\Big(\mathcal{K}_2(y,y) - 2\mathcal{K}_2(y,y') + \mathcal{K}_2(y',y')\Big) &= \sum_{t=1}^{T} \|\phi_t(y) - \phi_t(y')\|^2 \ \mathbf{Tr}(A_t) \\ &\leq \left(\sum_{t=1}^{T} L_t^2 \mathbf{Tr}(A_t)\right) \|y - y'\|^2_{\mathbb{R}^p}.\end{aligned}$$

Note also that Assumption 3.16 is stronger than Assumption 3.14, since $\|A\|_* \leq \mathbf{Tr}(A)$ for any trace class operator A.

Let OVK $\mathcal{K}_2$ satisfying Assumption 3.16, and be such that $\mathcal{H}_2$ is separable. We then know that there exists $\Psi \in \mathcal{L}(\ell_2(\mathbb{N}), \mathcal{X}_0)$ such that $\forall (y, y') \in \mathbb{R}^p$, $\mathcal{K}_2(y, y') = \Psi(y)\Psi^*(y')$ and $\forall f_2 \in \mathcal{H}_2, \exists v \in \ell_2(\mathbb{N})$ such that $f_2(\cdot) = \Psi(\cdot)v$, and $\|f_2\|_{\mathcal{H}_2} = \|v\|_{\ell_2}$. We have:

$$\begin{aligned}&\mathbb{E}_{\boldsymbol{\gamma}}\left[\sup_{f_2 \in \mathcal{H}_{2,t_2}} \Big\langle \boldsymbol{\gamma}_i, f_2(\boldsymbol{y} - f_2(\boldsymbol{y'})\Big\rangle_{\mathcal{X}_0^n}\right] \\ &\qquad = \mathbb{E}_{\boldsymbol{\gamma}}\left[\sup_{f_2 \in \mathcal{H}_{2,t_2}} \sum_{i=1}^{n}\sum_{k=1}^{\infty} \gamma_{i,k} \Big\langle (\Psi(y_i) - \Psi(y'_i))v, e_k\Big\rangle_{\mathcal{X}_0}\right], \\ &\qquad = \mathbb{E}_{\boldsymbol{\gamma}}\left[\sup_{f_2 \in \mathcal{H}_{2,t_2}} \left\langle \sum_{i=1}^{n}\sum_{k=1}^{\infty} \gamma_{i,k}(\Psi^*(y_i) - \Psi^*(y'_i))e_k, v\right\rangle_{\ell_2}\right], \\ &\qquad \leq t_2 \sqrt{\mathbb{E}_{\boldsymbol{\gamma}}\left\|\sum_{i=1}^{n}\sum_{k=1}^{\infty} \gamma_{i,k}(\Psi^*(y_i) - \Psi^*(y'_i))e_k\right\|^2_{\ell_2}}, \\ &\qquad \leq t_2 \sqrt{\sum_{i=1}^{n} \mathbf{Tr}\Big(\mathcal{K}_2(y_i,y_i) - 2\mathcal{K}_2(y_i,y'_i) + \mathcal{K}_2(y'_i,y'_i)\Big)}, \\ &\qquad \leq t_2 L \ \|\boldsymbol{y} - \boldsymbol{y'}\|_{\mathbb{R}^{np}},\end{aligned}$$

where only Assumption 3.16 and arguments previously involved have been used. Finally, we get:

$$R\Big(\mathcal{H}'_{2,t_2}, \mathcal{H}'_{1,t_1}(\mathcal{S}_n)\Big) \leq R\Big(\mathcal{H}'_{2,t_2}, \mathbb{R}^{np}\Big) \leq t_2 L. \tag{3.23}$$

Bounding $D\Big(\mathcal{H}'_{1,t_1}(\mathcal{S}_n)\Big)$. Consider the assumption below.

Assumption 3.17. *There exists $\kappa < +\infty$ such that: $\forall x \in \mathcal{S}_n$,*

$$\Big\|\mathcal{K}_1(x,x)\Big\|_* \leq \kappa^2.$$

This assumption is easily fulfilled, since X is almost surely bounded. Therefore, any OVK that writes as the (finite) sum of decomposable kernels with continuous scalar kernels fulfills it. Note also that it is a weaker assumption than Assumption 3.15, since one could choose $\kappa = \sqrt{Kp}$.

Let $\mathcal{K}_1$ that satisfies Assumption 3.17 and $(\boldsymbol{y}, \boldsymbol{y'}) \in \mathcal{H}'_{1,t_1}(\mathcal{S}_n)$. There exists functions $(f_1, f'_1) \in \mathcal{H}^2_{1,t_1}$ such that $\boldsymbol{y} = (f_1(x_1), \ldots, f_1(x_n))$ and $\boldsymbol{y'} = (f'_1(x_1), \ldots, f'_1(x_n))$. It holds then:

$$\begin{aligned}
\Big\|\boldsymbol{y} - \boldsymbol{y'}\Big\|^2_{\mathbb{R}^{np}} &= \sum_{i=1}^{n} \Big\|f_1(x_i) - f'_1(x_i)\Big\|^2_{\mathbb{R}^p}, \\
&\leq \sum_{i=1}^{n} \left(\Big\|f_1(x_i)\Big\|_{\mathbb{R}^p} + \Big\|f'_1(x_i)\Big\|_{\mathbb{R}^p}\right)^2, \\
&\leq \sum_{i=1}^{n} \left(\|f_1\|_{\mathcal{H}_1} \Big\|\mathcal{K}_1(x_i,x_i)\Big\|^{1/2}_* + \Big\|f'_1\Big\|_{\mathcal{H}_1} \Big\|\mathcal{K}_1(x_i,x_i)\Big\|^{1/2}_*\right)^2, \qquad (3.24) \\
&\leq 4\kappa^2 t_1^2 n,
\end{aligned}$$

where Equation (3.24) follows from Equation (f) of Proposition 2.1 in Micchelli and Pontil (2005). Finally, we get:

$$D\Big(\mathcal{H}'_{1,t_1}, \mathcal{S}_n\Big) \leq 2\kappa t_1 \sqrt{n}. \tag{3.25}$$

Bounding $G\Big(\mathcal{H}'_{2,t_2}(0)\Big)$. We introduce the following assumption.

Assumption 3.18. *$\mathcal{K}_2(0,0)$ is trace class.*

Then, using the same arguments as for Equation (3.20), we get:

$$n\, G\Big(\mathcal{H}'_{2,t_2}(0)\Big) \leq t_2 \sqrt{n\, \mathbf{Tr}\Big(\mathcal{K}_2(0,0)\Big)}, \quad \text{or} \quad G\Big(\mathcal{H}'_{2,t_2}(0)\Big) \leq t_2 \sqrt{\frac{\mathbf{Tr}\Big(\mathcal{K}_2(0,0)\Big)}{n}}.$$

Rather than shifting the kernel $\widetilde{\mathcal{K}}_2(y,y') = \mathcal{K}_2(y,y') - \mathcal{K}_2(0,0)$, one could consider that Assumption 3.18 is always satisfied. In addition, we have $\mathbf{Tr}(\widetilde{\mathcal{K}}_2(0,0)) = 0$ and consequently $G(\mathcal{H}'_{2,t_2}(0)) \leq 0$.

Final Argument

Now, combining Equations (3.7), (3.9), (3.10), (3.16), (3.22), (3.23) and (3.25) and defining $C_0 := 8\sqrt{\pi}(C_1 + 2C_2)$, for any $\delta \in]0,1[$, we have with probability at least $1-\delta$:

$$\epsilon(\hat{h}_n) - \epsilon^* \leq C_0 LMt_1t_2\sqrt{\frac{Kp}{n}} + 24M^2\sqrt{\frac{\ln\frac{2}{\delta}}{2n}}.$$

This finishes the proof of Theorem 3.8. □

Remark 3.19. *The trace class assumption (Assumptions 3.7, 3.15 and 3.16) is hard to avoid when considering Rademacher averages for vv-RKHSs. However, the often used in practice identity decomposable kernel does not satisfy it. Therefore, another strategy to derive generalization bounds that does not make this assumption is to use stability tools (Bousquet and Elisseeff, 2002). Indeed, they have been extended to vv-RKHSs in Audiffren and Kadri (2013), and to OVK Ridge Regression in particular. Gradient Descent, even on non-convex objective functions, also fulfills some stability requirements (Hardt et al., 2015). As shall be seen in Chapter 4, the algorithm to optimize Kernel Autoencoders consists in an alternation of Gradient Descent steps and of OVK Ridge Regressions for the last layer optimization. Nevertheless, the stability properties of each algorithm cannot be transferred to their alternation (due to initialization problems in particular). Thus, a stability analysis seems not well suited to Kernel Autoencoders.*

We shall now discuss important questions that have been set aside during the exposure of our model, among which the possibility to adapt the Kernel Autoencoder to supervised problems, and potential hybrid architectures.

3.5 Extensions

Several important extensions and applications of Kernel Autoencoders have yet not been addressed. It is the purpose of the present section to expose and discuss them, as they represent interesting future research directions.

3.5.1 Supervised Extension

Along this chapter, we have assumed an Autoencoder-like framework, and one key hypothesis is that inputs, whether kernelized or not, are equal to the outputs. This assumption can of course be removed, without harming any aspect of the conducted analysis.

If inputs are kernelized, but not outputs, our architecture would act as a multi-layer standard kernel machine. Notice that this idea was already evoked in Schölkopf et al. (1998), where it was considered giving to a Support Vector Machine the new points representations extracted by Kernel PCA.

However, the main strength of the designed model is to handle infinite dimensional objects both in inputs and in outputs. One can thus imagine having both kernelized inputs $\phi(x_i)$'s and kernelized outputs $\psi(y_i)$'s. This is the case for instance in Brouard et al. (2016b). The goal is to identify metabolites from their mass spectra. The latter being complex structured inputs, a natural way to process them is to use kernel methods, or again to work on the $\phi(x_i)$'s. As explained at length in Chapter 2 (Section 2.2.2 therein), the absence of geometrical structure in the output space is tackled by first

mapping the molecules through a feature map ψ, and learning a regression function on the $\psi(y_i)$'s (the predictions in the original space being recovered by solving inverse problems). In this context, the model proposed could be interpreted as a deep version of Input Output Kernel Regression (IOKR).

This promising research direction is currently under investigation, with the hope that adding layers would yield numerical improvements on IOKR, so far the sate-of-the-art method regarding metabolite identification.

3.5.2 Hybrid Architecture

Another point that could be raised is the necessity of having functions from vv-RKHSs all along the architecture. If having a vv-RKHS at the last layer is absolutely necessary to be able to predict infinite dimensional objects (standard neural mappings with finite coefficient matrices and bias vectors are unable to do so), the internal functions may perfectly pertain to other functional spaces.

Hybrid architectures sound as the perfect mixes between kernel methods and neural implementations. They benefit both from the capacity of vv-RKHS functions to predict kernelized data (and subsequently any structured data), and that of well studied neural architectures that have empirically shown good performances.

Finally, notice that if the input data is an image, one can also use the work developed in Mairal et al. (2014); Mairal (2016) for the first layers. One would end up with an architecture specifically tailored to the needs at each step: expressive first descriptors, powerful middle architecture, and infinite-valued last layer.

3.5.3 Learning Output Embeddings

So far, representations extracted by Kernel Autoencoders are computed in an agnostic fashion, meaning that what the representation is used for after the extraction is never taken into account. It is natural to consider a Kernel Autoencoder criterion with an additive supervised criterion to make the learned representation suited to the next task.

This could be for instance a term to make representation of inputs of the same class similar:

$$\frac{1}{2n}\sum_{i=1}^{n}\|x_i - f_2 \circ f_1(x_i)\|^2 + \frac{1}{n(n-1)}\sum_{i<j}\mathbb{1}\{y_i = y_j\}\|f_1(x_i) - f_1(x_j)\|^2.$$

This second term writes as a U-statistic, which are extensively studied in the second part of this manuscript.

But more classical semi-supervised criterion could also be considered:

$$\frac{1}{2n}\sum_{i=1}^{n}\|x_i - f_2 \circ f_1(x_i)\|^2 + \frac{1}{n}\sum_{i=1}^{n}\ell\left(y_i, h \circ f_1(x_i)\right).$$

The representation is encouraged to minimize both the reconstruction error and the supervised empirical risk.

However, the main strength of Kernel Autoencoders is to deal with infinite dimensional inputs/outputs. One could imagine performing exactly the same algorithm as above, but with aim to learn an output embedding, the outputs being potentially kernelized at the beginning.

The IOKR regression criterion

$$\frac{1}{n}\sum_{i=1}^{n}\|h(x_i)-\psi(y_i)\|+\frac{\Lambda}{2}\|h\|^2$$

could thus be replaced with

$$\sum_{i=1}^{n}\|h(x_i)-f_1\circ\psi(y_i)\|+\frac{\Lambda}{2}\|h\|^2+\frac{1}{2n}\sum_{i=1}^{n}\|\psi(y_i)-f_2\circ f_1\circ\psi(y_i)\|^2+\mathrm{Reg}(f_1,f_2).$$

The alternate approach developed in Chapter 2 to learn f_1 and f_2 could be readily adapted to incorporate the learning of h into the procedure. Learning a good output embedding is critical for the task of structured prediction, and yet not very studied. This autoencoder-like criterion is another interesting research direction to be investigated.

3.6 Conclusion

In this chapter, we have introduced a new framework for Autoencoders, based on vv-RKHSs and OVKs. The use of RKHS functions enables Kernel Autoencoders to handle data from possibly infinite dimensional Hilbert spaces, and then to extend the autoencoding scheme to any kind of data through the use of the kernel trick in the output space. A generalization bound in terms of reconstruction error is provided, and a connection to Kernel PCA is established. But far from being restricted to a deep version of Kernel PCA, we have shown that Kernel Autoencoders pave the way for many interesting applications, ranging from fully supervised vv-RKHSs networks to the learning of output embeddings. The optimization of this model is now to be studied at length in Chapter 4.

4

Optimization of Deep Kernel Architectures

Contents

In Chapter 3, we have detailed at length the Kernel Autoencoder model. In this chapter, we focus on its optimization process. As noticed in Section 3.5, having outputs different from the inputs does not change a single line in the analysis. The same goes for the optimization procedure. As a consequence, for universality purposes, the subsequent chapter is presented in the most general case where outputs differ from inputs, although numerical experiments mainly focus on autoencoding problems.

The optimization process, as for many kernel methods, crucially relies on a Representer Theorem. As the main tool of this chapter, this Representer Theorem, that is tailored to composition of functions in vv-RKHSs, is stated and proved in Section 4.1. Next, Section 4.2 focuses on the non-convexity of our problem, before and after the use of the Representer Theorem. This essential observation has important consequences, on the non global optimality of the solutions that may be found. The complete optimization process, that builds on Gradient Descent, is detailed in Section 4.3 in the case where all internal spaces are finite dimensional. When the output space is infinite dimensional, the Gradient Descent scheme has to be alternated with Kernel Ridge Regressions to update the last layer infinite dimensional coefficients (Section 4.4). Numerical experiments are exposed in Section 4.5. This chapter covers the implementation contribution of:

▶ **P. Laforgue**, S. Clémençon, F. d'Alché-Buc. Autoencoding any data through kernel autoencoders. In *Proceedings of Artificial Intelligence and Statistics*, 2019.

4.1 A Representer Theorem for Composed Criteria

Representer Theorems (see Theorems 2.5 and 2.12 in Chapter 2 for instance) are critical to the learning of kernel machines. They exhibit particular expansions of the solutions, so that the search space is reduced from the entire (vv-)RKHS to a smaller subspace. After a recall of the Deep Kernel Machine model in Section 4.1.1 (outputs are not anymore assumed to be equal to outputs), the theorem dedicated to composition of functions in vv-RKHSs is sated in Section 4.1.2.

4.1.1 The Deep Kernel Machine

As a reminder, (X, Y) is a random variable valued in $\mathcal{X}\times\mathcal{Y}$ with unknown distribution P. The sample $\mathcal{S}_n = \{(x_i, y_i)\}_{i\leq n} \in (\mathcal{X}\times\mathcal{Y})^n$ is composed of n i.i.d. realizations of (X, Y). Outputs may be embedded into a Hilbert space $\mathcal{H}$ through a feature map $\phi : \mathcal{Y} \to \mathcal{H}$. If $\mathcal{Y}$ is already a Hilbert space, one may have $\phi = id$ and $\mathcal{H} = \mathcal{Y}$. The important thing is that dot products must be easily computable in $\mathcal{H}$. If ϕ is the canonical feature map of a scalar kernel k defined on $\mathcal{Y}$, it holds for instance $\langle\phi(y_i), \phi(y_j)\rangle_{\mathcal{H}} = k(y_i, y_j)$ by virtue of the kernel trick.

Let $L \in \mathbb{N}^*$ and assume that there exists a collection of Hilbert spaces $(\mathcal{X}_l)_{1\leq l\leq L}$, such that $\mathcal{X}_L = \mathcal{H}$. For $l \leq L-1$, the space $\mathcal{X}_l$ is supposed to be endowed with an OVK $\mathcal{K}_{l+1} : \mathcal{X}_l \times \mathcal{X}_l \to \mathcal{L}(\mathcal{X}_{l+1})$, associated to a vv-RKHS $\mathcal{H}_{l+1} \subset \mathcal{F}(\mathcal{X}_l, \mathcal{X}_{l+1})$. Finally, $\mathcal{H}_1 \subset \mathcal{F}(\mathcal{X}, \mathcal{X}_1)$ is a vv-RKHS associated to an OVK $\mathcal{K}_1$. The Deep Kernel Machine problem then writes

$$\min_{f_l\in\mathcal{H}_l,\ l\leq L} \quad \frac{1}{2n}\sum_{i=1}^{n}\Big\|\phi(y_i) - f_L\circ\ldots\circ f_1(x_i)\Big\|_{\mathcal{H}}^2 + \sum_{l=1}^{L}\frac{\Lambda_l}{2}\|f_l\|_{\mathcal{H}_l}^2, \tag{4.1}$$

with $(\Lambda_l)_{l\leq L} \in \mathbb{R}_+^L$ a collection of regularization parameters.

We now states a Representer Theorem, that applies to the composition framework of Problem (4.1).

4.1.2 Theorem Statement

This theorem, like the standard vv-Representer Theorem, relies on Minimum Norm Interpolation results, see Theorem 2.12 in Chapter 2. It exhibits a very specific structure for the minimizers, as each layer's support vectors are the images of the original points by the previous one.

Theorem 4.1. *Let $L_0 \leq L$, and $V : \mathcal{X}_{L_0}^n \times \mathbb{R}_+^{L_0} \to \mathbb{R}$ a function of $n + L_0$ variables, strictly increasing in each of its L_0 last arguments. Suppose that $(f_1^*, \ldots, f_{L_0}^*)$ is a solution to the optimization problem:*

$$\min_{f_l\in\mathcal{H}_l} \quad V\Big((f_{L_0}\circ\ldots\circ f_1)(x_1), \ldots, (f_{L_0}\circ\ldots\circ f_1)(x_n), \|f_1\|_{\mathcal{H}_1}, \ldots, \|f_{L_0}\|_{\mathcal{H}_{L_0}}\Big).$$

For $i \leq n$ and $l \leq L$, let $x_i^{*(l)} = f_l^*\circ\ldots\circ f_1^*(x_i)$, with the notation convention $x_i^{*(0)} = x_i$. Then, there exist $\left(\varphi_{1,1}^*, \ldots, \varphi_{1,n}^*, \ldots, \varphi_{L_0,n}^*\right) \in \mathcal{X}_1^n \times \ldots \times \mathcal{X}_{L_0}^n$ such that

$$\forall\ l \leq L_0, \quad f_l^*(\cdot) = \sum_{i=1}^{n}\mathcal{K}_l\left(\ \cdot\ , x_i^{*(l-1)}\right)\varphi_{l,i}^*.$$

Proof. For simplicity, we shall use the following shortcut notation:

$$\xi(f_1^*, \dots, f_{L_0}^*, \mathcal{S}_n) = V\Big((f_{L_0} \circ \dots \circ f_1)(x_1), \dots, (f_{L_0} \circ \dots \circ f_1)(x_n), \|f_1\|_{\mathcal{H}_1}, \dots, \|f_{L_0}\|_{\mathcal{H}_{L_0}}\Big).$$

Let $l_0 \leq L_0$. Let $g_{l_0} \in \mathcal{H}_{l_0}$ such that :

$$g_{l_0}\left(x_i^{*(l_0-1)}\right) = f_{l_0}^*\left(x_i^{*(l_0-1)}\right), \quad \text{for all } i \leq n.$$

By definition, we have :

$$\xi(f_1^*, \dots, f_{l_0}^*, \dots, f_{L_0}^*, \mathcal{S}_n) \leq \xi(f_1^*, \dots, g_{l_0}, \dots, f_{L_0}^*, \mathcal{S}_n),$$

thus we necessarily have :

$$\|f_{l_0}^*\|_{\mathcal{H}_{l_0}}^2 \leq \|g_{l_0}\|_{\mathcal{H}_{l_0}}^2.$$

Therefore $f_{l_0}^*$ is a solution to the problem :

$$\begin{aligned} \min_{f \in \mathcal{H}_{l_0}} \quad & \|f\|_{\mathcal{H}_{l_0}}, \\ \text{s.t.} \quad & f\left(x_i^{*(l_0-1)}\right) = f_{l_0}^*\left(x_i^{*(l_0-1)}\right), \quad i \leq n, \end{aligned}$$

By Theorem 2.12, there exists $\left(\varphi_{l_0,1}^*, \dots, \varphi_{l_0,n}^*\right) \in \mathcal{X}_{l_0}^n$, such that :

$$f_{l_0}^*(\cdot) = \sum_{i=1}^{n} \mathcal{K}_{l_0}\left(\cdot\,, x_i^{*(l_0-1)}\right) \varphi_{l_0,i}^*.$$

□

Remark 4.2. *An important remark to make is that conditions on V are very loose. Indeed, no convexity assumption is made on V for instance, what perfectly matches Problem (4.1) (see in particular Section 4.2). The criterion should only depend on the composition of the vv-RKHS functions evaluations, and increasingly with respect to the norms, exactly as in Problem (4.1). This generality makes it possible to encompass all Deep Kernel Machines problems. Notice that simultaneously to the statement in* **Laforgue** *et al. (2019a), a similar result has been established in Bohn et al. (2019) (see Theorem 1 therein).*

Remark 4.3. *A second important thing to notice is that in general L_0 may be different from L. Although surprising at first sight, this remark becomes crucial when addressing optimization issues. Indeed, to address the infinite dimensionality of the last layer's coefficients, one is encouraged to proceed using an alternate descent that sometimes freezes part of the layers. With the flexibility over L_0, the Representer Theorem stated in Theorem 4.1 remains valid all the time, no matter the layers frozen.*

The Representer Theorem 4.1 thus transforms the optimization over the collection of Hilbert spaces $(\mathcal{H}_l)_{l \leq L}$ into an optimization over the intermediate spaces $(\mathcal{X}_l^n)_{l \leq L}$. This opens the door to a Gradient Descent resolution, with parameters the set of coefficients $(\varphi_{l,i})$ for $l \leq L$ and $i \leq n$. However, the compositions of functions in Problem (4.1) make the objective function highly non-convex. As a consequence, the solution found by a Gradient Descent strategy may always be locally optimal only. The non-convexity of the criterion is further addressed in the next section.

4.2 Non-Convexity of the Problem

Gradient Descent strategies are among the most used optimization methods in Machine Learning. However, they find a minimizer of the objective by finding a point canceling its gradient. When the objective is not convex, this implies that the solution approximately computed by the algorithm may be only locally optimal. Therefore, checking whether the objective function minimized is convex or not is one of the first check needed before proceeding with the algorithm *per se* in Sections 4.3 and 4.4.

4.2.1 Functional Setting

We show the non-convexity of Problem (4.1) in general by showing its non-convexity in a very specific case. Let $L = 2$, $\mathcal{X} = \mathcal{X}_1 = \mathcal{Y} = \mathcal{H} = \mathbb{R}$, $n = 1$, and $x_1 = y_1 = 1$. Further assume that both kernels are linear : $\mathcal{K}_1(x, x') = xx'$, $\mathcal{K}_2(z, z') = zz'$. By Theorem 4.1, we have the existence of φ_1 and φ_2 such that

$$f_1 : x \mapsto \mathcal{K}_1(x, x_1)\varphi_1 = \varphi_1 x,$$

$$f_2 : z \mapsto \mathcal{K}_2(z, f(x_1))\varphi_2 = \varphi_2 f_1(x_1) z.$$

Let $\mathcal{P}\colon \mathcal{H}_1 \times \mathcal{H}_2 \to \mathbb{R}$ the application mapping two candidate functions to the value of Problem (4.1)'s objective with the specific choices stated above. Functions f_1 and f_2 depending only on φ_1 and φ_2, let $\mathcal{Q}\colon \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ the application mapping to the same values as $\mathcal{P}$, but with inputs φ_1 and φ_2. In the following subsection we show that convexity of $\mathcal{Q}$ does not even hold, that allows to conclude that convexity of $\mathcal{P}$ does not hold either.

4.2.2 Parametric Setting

As a reminder, we have :

$$f_1(x) = \mathcal{K}_1(x, x_1)\varphi_1 = \varphi_1 x, \qquad f(x_1) = \varphi_1,$$
$$f_2(z) = \mathcal{K}_2(z, f_1(x_1))\varphi_2 = \varphi_1\varphi_2 z, \qquad f_2(f_1(x_1)) = \varphi_1^2\varphi_2.$$

Our problem reads :

$$\min_{\varphi_1,\varphi_2\in\mathbb{R}^2} \quad \mathcal{Q}(\varphi_1, \varphi_2) = \frac{1}{2}\left(1 - \varphi_1^2\varphi_2\right)^2 + \frac{\Lambda_1}{2}\varphi_1^2 + \frac{\Lambda_2}{2}\varphi_2^2,$$

or equivalently :

$$\min_{\varphi_1,\varphi_2\in\mathbb{R}^2} \quad 1 + \Lambda_1\varphi_1^2 + \Lambda_2\varphi_2^2 - 2\varphi_1^2\varphi_2 + \varphi_1^4\varphi_2^2.$$

Let us find the critical points and analyze them. We have :

$$\begin{aligned}
\frac{\partial\mathcal{Q}}{\partial\varphi_1}(\varphi_1, \varphi_2) &= 2\Lambda_1\varphi_1 - 4\varphi_1\varphi_2 + 4\varphi_1^3\varphi_2^2, \\
\frac{\partial\mathcal{Q}}{\partial^2\varphi_1}(\varphi_1, \varphi_2) &= 2\Lambda_1 - 4\varphi_2 + 12\varphi_1^2\varphi_2^2, \\
\frac{\partial\mathcal{Q}}{\partial\varphi_2}(\varphi_1, \varphi_2) &= 2\Lambda_2\varphi_2 - 2\varphi_1^2 + 2\varphi_1^4\varphi_2, \\
\frac{\partial\mathcal{Q}}{\partial^2\varphi_2}(\varphi_1, \varphi_2) &= 2\Lambda_2 + 2\varphi_1^4, \\
\frac{\partial\mathcal{Q}}{\partial\varphi_1\partial\varphi_2}(\varphi_1, \varphi_2) &= -4\varphi_1 + 8\varphi_1^3\varphi_2.
\end{aligned}$$

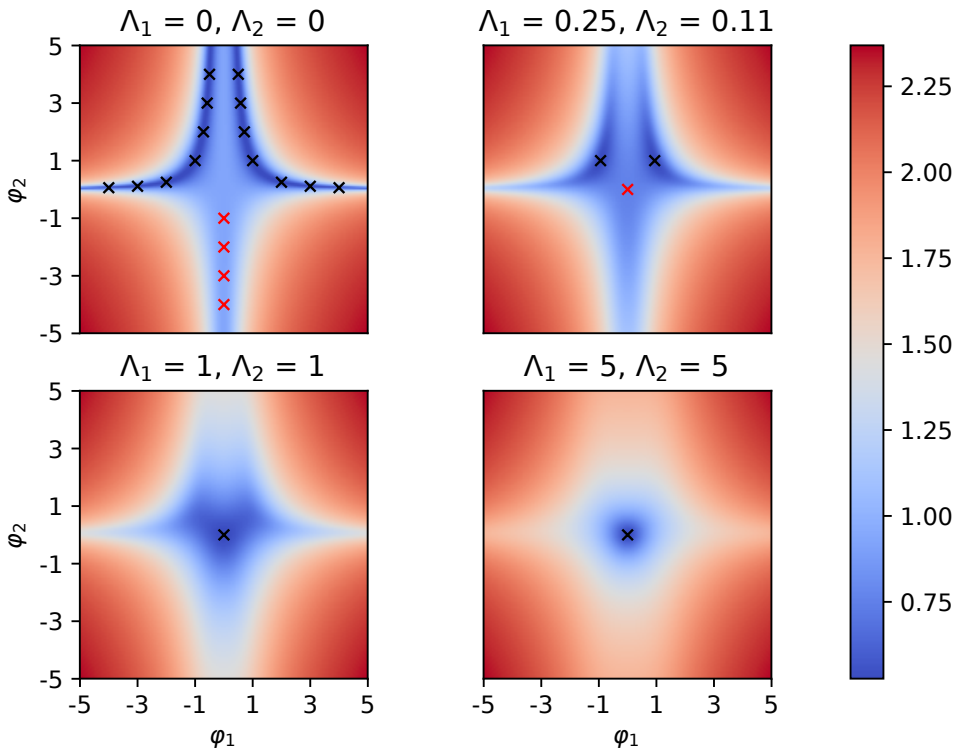


Figure 4.1 – Heatmaps of $\mathcal{Q}$ for different values of Λ_1 and Λ_2

The two following equivalence relationships hold true:

$$\frac{\partial \mathcal{Q}}{\partial \varphi_1}(\varphi_1^*, \varphi_2^*) = \left(2\Lambda_1 - 4\varphi_2^* + 4\varphi_1^{*2}\varphi_2^{*2}\right)\varphi_1^* = 0 \quad \Leftrightarrow \quad \varphi_1^* = 0 \;\text{ or }\; \varphi_1^{*2} = \frac{2\varphi_2^* - \Lambda_1}{2\varphi_2^{*2}},$$

$$\frac{\partial \mathcal{Q}}{\partial \varphi_2}(\varphi_1^*, \varphi_2^*) = 2\Lambda_2\varphi_2^* - 2\varphi_1^{*2} + 2\varphi_1^{*4}\varphi_2^* = 0 \quad \Leftrightarrow \quad \varphi_2^* = \frac{\varphi_1^{*2}}{\varphi_1^{*4} + \Lambda_2}.$$

Obviously, the point $(\varphi_1^*, \varphi_2^*) = (0, 0)$ is always critical. Notice that :

$$\mathbf{Hess}_{(0,0)}\mathcal{Q} = \begin{pmatrix} 2\Lambda_1 & 0 \\ 0 & 2\Lambda_2 \end{pmatrix} \succ 0.$$

Thus $(0, 0)$ is a local minimum with value $\mathcal{Q}(0, 0) = 1$. To prove that it is not a global minimizer, it is enough to find a couple (φ_1, φ_2) such that $\mathcal{Q}(\varphi_1, \varphi_2) < 1$. For example $\mathcal{Q}(1, 1) = (\Lambda_1 + \Lambda_2)/2$. As soon as $\Lambda_1 + \Lambda_2 < 2$, the objective $\mathcal{Q}$ is not invex, and a fortiori non-convex.

Figure 4.1 shows the heatmaps of $\mathcal{Q}$ with respect to φ_1 and φ_2 for different regularization settings. Note that in the non-regularized setting ($\Lambda_1 = \Lambda_2 = 0$), every point $(0, \varphi_2)$ with $\varphi_2 < 0$ is a local minimizer but not a global one. They are represented by red crosses. On the other hand, we have also an infinite number of global minima, namely every couple satisfying $\varphi_1^2\varphi_2 = 1$. See the black crosses on the top left figure. When the regularization parameters remain small enough, $(0, 0)$ is a local minimizer but not a global one (top right figure). Finally, the higher the regularization, the smoother the objective, even if convexity can never be verified (bottom figures).

4.3 Finite Dimensional Gradient Descent

Although Section 4.2 shows the non-convexity of Problem (4.1), Theorem 4.1 encourages to proceed with a Gradient Descent scheme, as functions are now fully parametrized by the coefficients $(\varphi_{l,i})$, $l \leq L$ and $i \leq n$. As a reminder, coefficients $\varphi_{l,i}$ are valued in $\mathcal{X}_l$ for $i \leq n$. Therefore, a condition for Gradient Descent to be computable (and that was not necessary to the statement of Theorem 4.1) is that all intermediate spaces $\mathcal{X}_l$, including $\mathcal{H}$ must be finite dimensional. The case where the final output space is infinite dimensional is addressed in Section 4.4. Breaking with Section 4.2, we are back in the general case with L layers. For $l \leq L$, we assume that there exists $d_l \in \mathbb{N}^*$ such that $\mathcal{X}_l = \mathbb{R}^{d_l}$.

4.3.1 A Gradient Descent Scheme

The objective function of Problem (4.1), viewed as a function of $(f_L \circ \ldots \circ f_1)(x_1), \ldots, (f_L \circ \ldots \circ f_1)(x_n), \|f_1\|_{\mathcal{H}_1}, \ldots, \|f_L\|_{\mathcal{H}_L}$ satisfies the condition on V needed to establish Theorem 4.1. After applying it (with $L_0 = L$), Problem (4.1) boils down to the problem of finding the $\varphi^*_{l,i}$'s, which are finite dimensional. This crucial observation shows that our problem can be solved in a computable manner, although its convexity still cannot be ensured (see Section 4.2).

The objective only depending on the $\varphi_{l,i}$'s, Problem (3.3) can be approximately solved by Gradient Descent (GD). We now specify the gradient derivation in the decomposable OVKs case, *i.e.* for any $l \leq L$ we assume that there exists a scalar kernel $k_l \colon \mathcal{X}_l \times \mathcal{X}_l \to \mathbb{R}$, and $A_l \in \mathcal{L}(X_{l+1})$ positive semidefinite such that $\mathcal{K}_l(x, x') = k_l(x, x')A_l$. For $l \leq L$, let $\Phi_l = (\varphi_{l,1}, \ldots, \varphi_{l,n})^\top \in \mathbb{R}^{n \times d_l}$ storing the coefficients $\varphi_{l,i}$ in rows, and $K_l \in \mathbb{R}^{n \times n}$ such that $\left[K_l\right]_{i,i'} = k_l(x_i^{(l-1)}, x_{i'}^{(l-1)})$, with Theorem 4.1's notation: $x_i^{(l)} = f_l \circ f_{l-1} \circ \ldots \circ f_1(x_i)$ for $i \leq n$ and $l \leq L$. Let $l_0 \leq L$ and $i_0 \leq n$, the gradient of the distortion term reads:

$$\left(\nabla_{\varphi_{l_0,i_0}} \frac{1}{2n} \sum_{i=1}^{n} \left\| \phi(y_i) - f_L \circ \ldots \circ f_1(x_i) \right\|_{\mathcal{H}}^2 \right)^\top \tag{4.2}$$
$$= -\frac{1}{n} \sum_{i=1}^{n} \left(\phi(y_i) - {x_i}^{(L)} \right)^\top \mathbf{Jac}_{{x_i}^{(L)}}(\varphi_{l_0,i_0}).$$

On the other hand, $\|f_l\|^2_{\mathcal{H}_l}$ may be rewritten as:

$$\begin{aligned}
\|f_l\|^2_{\mathcal{H}_l} &= \left\langle f_l, f_l \right\rangle_{\mathcal{H}_l}, \\
&= \left\langle \sum_{i=1}^{n} \mathcal{K}_l\left(\cdot, {x_i}^{(l-1)}\right) \varphi_{l,i} \ , \ \sum_{i'=1}^{n} \mathcal{K}_l\left(\cdot, {x_{i'}}^{(l-1)}\right) \varphi_{l,i'} \right\rangle_{\mathcal{H}_l}, \\
&= \sum_{i,i'=1}^{n} \left\langle \mathcal{K}_l\left(\cdot, {x_i}^{(l-1)}\right) \varphi_{l,i} \ , \ \mathcal{K}_l\left(\cdot, {x_{i'}}^{(l-1)}\right) \varphi_{l,i'} \right\rangle_{\mathcal{H}_l}, \\
&= \sum_{i,i'=1}^{n} \left\langle \varphi_{l,i} \ , \ \mathcal{K}_l\left({x_i}^{(l-1)}, {x_{i'}}^{(l-1)}\right) \varphi_{l,i'} \right\rangle_{\mathcal{X}_l}, \\
&= \sum_{i,i'=1}^{n} k_l\left({x_i}^{(l-1)}, {x_{i'}}^{(l-1)}\right) \left\langle \varphi_{l,i} \ , \ A_l \ \varphi_{l,i'} \right\rangle_{\mathcal{X}_l}.
\end{aligned}$$

So that $\|f_l\|^2_{\mathcal{H}_l}$ may depend on φ_{l_0,i_0} in two ways: 1) if $l_0 = l$, there is a direct dependence of the second quadratic term, 2) but note also that for $l_0 < l$, the $\varphi_{l_0,i}$ have an influence on the ${x_i}^{(l-1)}$ and so on the first term. This remark leads to the following formulas:

$$\nabla_{\Phi_l} \|f_l\|^2_{\mathcal{H}_l} = 2\, K_l \Phi_l A_l, \tag{4.3}$$

with $\nabla_{\Phi_l} F \coloneqq \left(\nabla_{\varphi_{l,1}} F, \dots, \nabla_{\varphi_{l,n}} F\right)^\top \in \mathbb{R}^{n \times d_l}$ storing the gradients of any real-valued function F with respect to the $\varphi_{l,i}$ in rows.

And when $l_0 < l$, it holds:

$$\begin{aligned}
\left(\nabla_{\varphi_{l_0,i_0}} \|f_l\|^2_{\mathcal{H}_l}\right)^\top &= \sum_{i,i'=1}^{n} [N_l]_{i,i'} \left(\nabla_{\varphi_{l_0,i_0}} k_l\left({x_i}^{(l-1)}, {x_{i'}}^{(l-1)}\right)\right)^\top, \\
&= \sum_{i,i'=1}^{n} [N_l]_{i,i'} \Bigg[\left(\nabla^{(1)} k_l\left({x_i}^{(l-1)}, {x_{i'}}^{(l-1)}\right)\right)^\top \mathbf{Jac}_{{x_i}^{(l-1)}}(\varphi_{l_0,i_0}) \\
&\qquad + \left(\nabla^{(2)} k_l\left({x_i}^{(l-1)}, {x_{i'}}^{(l-1)}\right)\right)^\top \mathbf{Jac}_{{x_{i'}}^{(l-1)}}(\varphi_{l_0,i_0}) \Bigg], \\
&= \sum_{i,i'=1}^{n} [N_l]_{i,i'} \left(\nabla^{(1)} k_l\left({x_i}^{(l-1)}, {x_{i'}}^{(l-1)}\right)\right)^\top \mathbf{Jac}_{{x_i}^{(l-1)}}(\varphi_{l_0,i_0}) \\
&\quad + \sum_{i',i=1}^{n} [N_l]_{i',i} \left(\nabla^{(1)} k_l\left({x_{i'}}^{(l-1)}, {x_i}^{(l-1)}\right)\right)^\top \mathbf{Jac}_{{x_{i'}}^{(l-1)}}(\varphi_{l_0,i_0}), \\
&= 2 \sum_{i,i'=1}^{n} [N_l]_{i,i'} \left(\nabla^{(1)} k_l\left({x_i}^{(l-1)}, {x_{i'}}^{(l-1)}\right)\right)^\top \mathbf{Jac}_{{x_i}^{(l-1)}}(\varphi_{l_0,i_0}),
\end{aligned} \tag{4.4}$$

where $\nabla^{(1)} k_l(x, x')$, respectively $\nabla^{(2)} k_l(x, x')$, denote the gradients of $k_l(\cdot, \cdot)$ with respect to the first (respectively second) coordinate and evaluated in (x, x'), and N_l the $n \times n$ matrix such that $[N_l]_{i,i'} = \langle \varphi_{l,i}, A_l \varphi_{l,i'} \rangle_{\mathcal{X}_l}$.

Assuming that the Jacobian matrices $\mathbf{Jac}_{{x_i}^{(L)}}(\varphi_{l_0,i_0})$ are known (their computation is detailed in Section 4.3.2), the norm part of the gradient is computable, and combining Equations (4.2) to (4.4) using the linearity of the gradient yields the complete formula.

Remark 4.4. *If n, L, and p denote respectively the number of samples, the number of layers, and the size of the largest latent space, the algorithm complexity is no more than $\mathcal{O}(n^2 L p)$ for objective evaluation, and $\mathcal{O}(n^3 L^2 p^3)$ for gradient derivation. Hence, it appears natural to consider stochastic versions of GD. But as shown by Equation (4.4), the norms gradients involve the computation of many Jacobians. Selecting a mini-batch does not affect these terms, which are the most time consuming. Thus, the expected acceleration due to stochasticity must not be so important. But a* doubly stochastic *scheme inspired from Dai et al. (2014), where both the points on which the objective is evaluated, as well as the coefficients to be updated, are chosen randomly at each iteration, might be of high interest since it would dramatically decrease the number of Jacobians computed. This approach, potentially combined with kernel approximations such as* Nyström's method *(Williams and Seeger, 2001) or* Random Fourier Features *(Rahimi and Recht, 2008; Brault et al., 2016) constitute a promising research direction to decrease the computational cost of the procedure.*

We now detail how the Jacobian matrices can be computed in an efficient fashion.

4.3.2 Detail of Jacobians Computation

All previously written gradients involve Jacobian matrices. Their computation is to be detailed now. First notice that $\mathbf{Jac}_{{x_i}^{(l)}}(\varphi_{l_0,i_0})$ only makes sense if $l_0 \leq l$. Indeed, ${x_i}^{(l)}$ is completely independent from φ_{l_0,i_0} otherwise. Let us first detail ${x_i}^{(l)}$ and use the linearity of the Jacobian operator :

$$\mathbf{Jac}_{{x_i}^{(l)}}(\varphi_{l_0,i_0}) = \sum_{i'=1}^{n} \mathbf{Jac}_{k_l\left({x_i}^{(l-1)},{x_{i'}}^{(l-1)}\right)A_l\ \varphi_{l,i'}}(\varphi_{l_0,i_0}).$$

Just as in the norm gradient case, there are two different outputs depending on whether $l = l_0$ (this gives an initialization), or $l > l_0$ (this leads to a recurrence formula).

Own Jacobian ($l = l_0$) :

$$\begin{aligned}
\mathbf{Jac}_{{x_i}^{(l)}}(\varphi_{l,i_0}) &= \sum_{i'=1}^{n} \mathbf{Jac}_{k_l\left({x_i}^{(l-1)},{x_{i'}}^{(l-1)}\right)A_l\ \varphi_{l,i'}}(\varphi_{l,i_0}), \\
&= \sum_{i'=1}^{n} k_l\left({x_i}^{(l-1)},{x_{i'}}^{(l-1)}\right)\mathbf{Jac}_{A_l\ \varphi_{l,i'}}(\varphi_{l,i_0}), \\
&= [K_l]_{i,i_0}\ A_l. \qquad (4.5)
\end{aligned}$$

Higher order Jacobian ($l > l_0$) :

$$\begin{aligned}
\mathbf{Jac}_{{x_i}^{(l)}}(\varphi_{l_0,i_0}) &= \sum_{i'=1}^{n} \mathbf{Jac}_{k_l\left({x_i}^{(l-1)},{x_{i'}}^{(l-1)}\right)A_l\ \varphi_{l,i'}}(\varphi_{l_0,i_0}), \\
&= \sum_{i'=1}^{n} A_l\ \varphi_{l,i'}\left(\nabla_{\varphi_{l_0,i_0}}\ k_l\left({x_i}^{(l-1)},{x_{i'}}^{(l-1)}\right)\right)^{\top}, \\
&= A_l \sum_{i'=1}^{n} \varphi_{l,i'}\Bigg[\ \left(\nabla^{(1)}k_l\left({x_i}^{(l-1)},{x_{i'}}^{(l-1)}\right)\right)^{\top}\mathbf{Jac}_{{x_i}^{(l-1)}}(\varphi_{l_0,i_0}) \\
&\qquad + \left(\nabla^{(1)}k_l\left({x_{i'}}^{(l-1)},{x_i}^{(l-1)}\right)\right)^{\top}\mathbf{Jac}_{{x_{i'}}^{(l-1)}}(\varphi_{l_0,i_0})\ \Bigg], \\
&= A_l\left[\sum_{i'=1}^{n} \varphi_{l,i'}\left(\nabla^{(1)}k_l\left({x_i}^{(l-1)},{x_{i'}}^{(l-1)}\right)\right)^{\top}\right]\mathbf{Jac}_{{x_i}^{(l-1)}}(\varphi_{l_0,i_0}) \\
&\quad + A_l\left[\sum_{i'=1}^{n} \varphi_{l,i'}\left(\nabla^{(1)}k_l\left({x_{i'}}^{(l-1)},{x_i}^{(l-1)}\right)\right)^{\top}\mathbf{Jac}_{{x_{i'}}^{(l-1)}}(\varphi_{l_0,i_0})\right], \\
\mathbf{Jac}_{{x_i}^{(l)}}(\varphi_{l_0,i_0}) &= A_l\Bigg[\Phi_l^{\top}\Delta_l\left({x_i}^{(l-1)}\right)\mathbf{Jac}_{{x_i}^{(l-1)}}(\varphi_{l_0,i_0}) \qquad (4.6) \\
&\qquad + \sum_{i'=1}^{n} \varphi_{l,i'}\left(\nabla^{(1)}k_l\left({x_{i'}}^{(l-1)},{x_i}^{(l-1)}\right)\right)^{\top}\mathbf{Jac}_{{x_{i'}}^{(l-1)}}(\varphi_{l_0,i_0})\Bigg],
\end{aligned}$$

with $\Delta_l(x) = \left(\nabla^{(1)} k_l\left(x, {x_1}^{(l-1)}\right), \dots, \nabla^{(1)} k_l\left(x, {x_n}^{(l-1)}\right)\right)^\top$ the $n \times d_{l-1}$ matrix storing the $\nabla^{(1)} k_l\left(x, {x_i}^{(l-1)}\right)$ in rows. These matrices are further explicited for popular kernels, especially at the interesting point $x = {x_i}^{(l-1)}$.

Assuming the $\Delta_l(x)$ matrices are known, we have an expression of $\mathbf{Jac}_{{x_i}^{(l)}}(\varphi_{l_0,i_0})$ that only depends on the $\mathbf{Jac}_{{x_{i'}}^{(l-1)}}(\varphi_{l_0,i_0})$. Thus we can unroll the recurrence formula obtained in Equation (4.6) until $l = l_0$, where Equation (4.5) is used. This gives a general way to recursively compute the Jacobian matrices. Notice that, similarly to a backpropagation, the recursive computation of Jacobians of higher order (*i.e.* difference between l and l_0) makes it memory efficient as lower order matrices are discarded progressively.

Remark 4.5. *An interesting remark can be made about the two-terms structure of the recursive relationship established in Equation (4.6). Indeed, the first term corresponds to the chain rule on ${x_i}^{(l)} = f_l({x_i}^{(l-1)})$ assuming that f_l is constant: $\partial f_l({x_i}^{(l-1)})/\partial\varphi_{l_0,i_0} = \partial f_l({x_i}^{(l-1)})/\partial {x_i}^{(l-1)} \cdot \partial {x_i}^{(l-1)}/\partial\varphi_{l_0,i_0}$ (with notation abuse on ∂ in order to preserve understandability). In opposition, the second term corresponds to a chain rule assuming that ${x_i}^{(l-1)}$ does not vary with φ_{l_0,i_0}, but that f_l does, through the influence of φ_{l_0,i_0} on the supports of f_l, namely the ${x_{i'}}^{(l-1)}$.*

We now detail the matrices $\Delta(x)$ computations for popular kernels.

Detail of the Δ_l Matrices Computations.

In this section we derive the quantities $\nabla^{(1)} k_l\left({x_i}^{(l-1)}, {x_{i'}}^{(l-1)}\right)$ and more specifically the matrices $\Delta_l({x_i}^{(l-1)})$ for $l \leq L$ and $i \leq n$. Notice that all previously computed quantities are independent from the kernel chosen. Actually, the $\Delta_l({x_i}^{(l-1)})$ matrices encapsulate all the kernel specificity of the algorithm. Thus, tailoring a new algorithm by changing the kernels only requires computing the new Δ_l matrices. This flexibility is a key asset of our approach, and more generally a crucial characteristic of kernel methods. In the following, we describe the Δ_l derivation for two popular kernels : the Gaussian and the polynomial ones.

Gaussian kernel :

$$\nabla^{(1)} k_l(x, x') = \nabla_x \left(\exp\left(-\gamma_l \|x - x'\|^2_{\mathcal{X}_{l-1}}\right)\right) = -2\gamma_l \, e^{-\gamma_l \|x-x'\|^2_{\mathcal{X}_{l-1}}} \, (x - x').$$

$$\begin{aligned}
\Delta_l\left({x_i}^{(l-1)}\right) &= \left[\nabla^{(1)} k_l\left({x_i}^{(l-1)}, {x_1}^{(l-1)}\right), \dots, \nabla^{(1)} k_l\left({x_i}^{(l-1)}, {x_n}^{(l-1)}\right)\right]^\top, \\
&= -2\gamma_l \left[e^{-\gamma_l \left\| {x_i}^{(l-1)} - {x_1}^{(l-1)} \right\|^2_{\mathcal{X}_{l-1}}} \left({x_i}^{(l-1)} - {x_1}^{(l-1)}\right), \dots \right. \\
&\qquad \left. \dots, e^{-\gamma_l \left\| {x_i}^{(l-1)} - {x_n}^{(l-1)} \right\|^2_{\mathcal{X}_{l-1}}} \left({x_i}^{(l-1)} - {x_n}^{(l-1)}\right) \right]^\top, \\
&= -2\gamma_l \, \tilde{K}_{l,i} \circ \left(\tilde{X}_i^{(l-1)} - X^{(l-1)}\right),
\end{aligned}$$

where:

- $X^{(l-1)} = \left({x_1}^{(l-1)}, \dots, {x_n}^{(l-1)}\right)^\top \in \mathbb{R}^{n \times d_{l-1}}$ stores the level $l-1$ representations of the x_i's in rows
- $\tilde{X}_i^{(l-1)} = \left({x_i}^{(l-1)}, \dots, {x_i}^{(l-1)}\right)^\top \in \mathbb{R}^{n \times d_{l-1}}$ stores the level $l-1$ representation of x_i n times in rows
- $\tilde{K}_{l,i} \in \mathbb{R}^{n \times n}$ is the k_l Gram matrix between $X^{(l-1)}$ and $\tilde{X}_i^{(l-1)}$ $\left(i.e.\ [\tilde{K}_{l,i}]_{s,t} = k_l\left({x_i}^{(l-1)}, {x_t}^{(l-1)}\right)\right)$
- $\circ$ denotes the Hadamard (termwise) product for two matrices of the same shape

In practice, it is important to note that computing the Δ_l matrices with the Gaussian kernel needs not new calculations, but only uses already computed quantities : the level $l-1$ representations and their Gram matrix.

Polynomial kernel :

$$\nabla^{(1)} k_l(x, x') = \nabla_x \left(a\langle x, x'\rangle + b\right)^c = ca\left(a\langle x, x'\rangle + b\right)^{c-1} x'.$$

$$\begin{aligned}
\Delta_l\left({x_i}^{(l-1)}\right) &= \left[\nabla^{(1)} k_l\left({x_i}^{(l-1)}, {x_1}^{(l-1)}\right), \dots, \nabla^{(1)} k_l\left({x_i}^{(l-1)}, {x_n}^{(l-1)}\right)\right]^\top, \\
&= ca\Bigg[\left(a\left\langle {x_i}^{(l-1)}, {x_1}^{(l-1)}\right\rangle + b\right)^{c-1} {x_1}^{(l-1)}, \dots \\
&\qquad \dots, \left(a\left\langle {x_i}^{(l-1)}, {x_n}^{(l-1)}\right\rangle + b\right)^{c-1} {x_n}^{(l-1)}\Bigg]^\top, \\
\Delta_l\left({x_i}^{(l-1)}\right) &= ca\ \left(\tilde{K}_{l,i}\right)^{\frac{c-1}{c}} \circ X^{(l-1)},
\end{aligned}$$

where we keep the notations introduced in the Gaussian kernel example for $X^{(l-1)}$, $\tilde{K}_{l,i}$ and $\circ$. Note that the exponent on $\tilde{K}_{l,i}$ must be understood as a termwise power, and not a matrix multiplication power.

In practice, it is important to notice that computing the Δ_l matrices with the polynomial kernel only requires a slight and cheap new calculation : putting the - already computed - Gram matrix at layer $l-1$ to the termwise power $(c-1)/c$.

With this last derivation, we finish to show that a Gradient Descent strategy can be considered to learn Deep Kernel Machines solving Problem (4.1) when all spaces are finite dimensional. We now turn to the more involved case where the outputs $\phi(y_i)$'s are infinite dimensional. If a similar Gradient Descent approach can be performed for the inner layers coefficients (the gradient easily passes through the last infinite dimensional layer), an alternative Kernel Ridge Regression resolution is needed to update the last layer's coefficients.

4.4 General Hilbert Space Resolution

Assume that outputs y_i's are embedded through the canonical feature map associated to the Gaussian kernel. Then, every $\phi(y_i)$, so as any element in $\mathcal{H}$, is infinite dimensional. This prevents from the use of a global Gradient Descent strategy, as the final layer's coefficients are valued in $\mathcal{H}$, and cannot be stored for instance. Instead, a procedure alternating Gradient Descent and Kernel Ridge Regression must be designed. It is the purpose of this section to expose it.

4.4.1 An Alternate Approach

In this section, $\mathcal{H}$, the space where outputs y_i's are embedded through ϕ is supposed to be infinite dimensional. In spite of this relaxation, Kernel Autoencoders remain computable. As Theorem 4.1 makes no assumption on the dimensionality of $\mathcal{X}_L = \mathcal{H}$, it can be applied. The only difference is that coefficients $\varphi_{L,i}$'s $\in \mathcal{X}_L^n$ are infinite dimensional, preventing from the use of a global Gradient Descent.

Nevertheless, if the $\varphi_{L,i}$'s are assumed to be fixed, a Gradient Descent (GD) can still be performed on the $\varphi_{l,i}$'s, for $l \leq L-1$. On the other hand, if one assumes these coefficients frozen, the optimal $\varphi_{L,i}$'s are the solutions to a Kernel Ridge Regression (KRR) problem.

Consequently, a hybrid approach alternating GD and KRR is considered. Two issues remain to be addressed:

1. How to compute the KRR in the infinite dimensional space $\mathcal{X}_L$?
2. How to propagate the gradients through $\mathcal{X}_L$?

From now, A_L is assumed to be the identity operator on $\mathcal{H} = \mathcal{X}_L$.

Answer to Question 1). If the $\varphi_{l,i}$'s, $l \leq L-1$ are fixed, then the best $\varphi_{L,i}$'s shall satisfy for all $i \leq n$ (Micchelli and Pontil, 2005):

$$\sum_{i'=1}^{n} \left(\mathcal{K}_L \left({x_i}^{(L-1)}, {x_{i'}}^{(L-1)} \right) + n\Lambda_L \delta_{ii'} \right) \varphi_{L,i'} = \phi(y_i). \tag{4.7}$$

This equality makes it easy to compute the N_L matrix. Indeed, it holds

$$\begin{aligned}
\left\langle \phi(y_j), \phi(y_{j'}) \right\rangle_{\mathcal{H}} &= \left\langle \sum_{i=1}^{n} \left(\mathcal{K}_L \left(x_j^{(L-1)}, x_i^{(L-1)} \right) + n\Lambda_L \delta_{ij} \right) \varphi_{L,i} \ , \right. \\
&\qquad \left. \sum_{i'=1}^{n} \left(\mathcal{K}_L \left(x_{j'}^{(L-1)}, x_{i'}^{(L-1)} \right) + n\Lambda_L \delta_{i'j'} \right) \varphi_{L,i'} \right\rangle_{\mathcal{X}_0}, \\
&= \sum_{i,i'=1}^{n} \left\langle \left(k_L \left(x_j^{(L-1)}, x_i^{(L-1)} \right) + n\Lambda_L \delta_{ij} \right) \varphi_{L,i} \ , \right. \\
&\qquad \left. \left(k_L \left(x_{j'}^{(L-1)}, x_{i'}^{(L-1)} \right) + n\Lambda_L \delta_{i'j'} \right) \varphi_{L,i'} \right\rangle_{\mathcal{X}_0}, \\
k(y_j, y_{j'}) &= \sum_{i,i'=1}^{n} \left(k_L \left(x_j^{(L-1)}, x_i^{(L-1)} \right) + n\Lambda_L \delta_{ij} \right) \\
&\qquad \left(k_L \left(x_{j'}^{(L-1)}, x_{i'}^{(L-1)} \right) + n\Lambda_L \delta_{i'j'} \right) \left\langle \varphi_{L,i}, \varphi_{L,i'} \right\rangle_{\mathcal{X}_0}.
\end{aligned} \tag{4.8}$$

With the notation $K^Y \in \mathbb{R}^{n\times n}$ such that $K^Y_{jj'} = k(y_j, y_{j'})$, Equation (4.8) is equivalent to

$$K^Y_{jj'} = \sum_{i,i'=1}^{n} [K_L + n\Lambda_L \mathbf{I}_n]_{ji} [N_L]_{ii'} [K_L + n\Lambda_L \mathbf{I}_n]_{i'j},$$

or equivalently:

$$K^Y = (K_L + n\Lambda_L \mathbf{I}_n)\ N_L\ (K_L + n\Lambda_L \mathbf{I}_n),$$

so that N_L is given by

$$N_L = (K_L + n\Lambda_L \mathbf{I}_n)^{-1}\ K^Y\ (K_L + n\Lambda_L \mathbf{I}_n)^{-1}. \tag{4.9}$$

Since K_L is recursively derived from K^X, $\Phi_1, \ldots, \Phi_{L-1}$, the optimal matrix N_L (in the sense of the Kernel Ridge Regression) only depends on K^X, K^Y, the coefficient matrices, and the last layer regularization parameter Λ_L. Let N_{KRR} be the function that computes N_L of Equation (4.9) from K^X, K^Y, $\Phi_1, \ldots, \Phi_{L-1}$, and Λ_L.

Thus, we have seen that the resolution of the KRR easily translates into a closed form solution for N_L. Remarkably, only this computable quantity is needed to propagate the gradient through the infinite dimensional layer. This shall answer Question 2).

Answer to Question 2). Assume that the coefficients $\varphi_{L,i}$'s are frozen. As detailed in Section 4.3, the total gradient of the objective is the (weighted) sum of Equations (4.2) to (4.4). Equation (4.3) features no quantity depending on the last layer: it remains unchanged. Equation (4.4) changes for $l = L$. It then only requires the knowledge of the $[N_L]_{ij} = \langle \varphi_{L,i}, \varphi_{L,j} \rangle$, which has been established by Equation (4.9). However, Equation (4.2) involves $\mathbf{Jac}_{x_i^L}(\varphi_{l_0,i_0})$, which does not make sense anymore since $\mathcal{X}_L$ is now infinite dimensional. Nevertheless, $\varphi_{l,i}$ is finite dimensional, and the distortion is a scalar: a gradient does exist. One is just forced to use the differential operator of $\|\phi(y_i) - f_L \circ \ldots \circ f_1(x_i)\|^2_{\mathcal{H}}$ to make it appear.

As a reminder, the chain rule for differentials reads : $d(g \circ f)(x) = dg(f(x)) \circ df(x)$. Let us apply it with $g(\cdot) = \|\cdot\|^2_{\mathcal{H}}$ and $f : \varphi_{l_0,i_0} \mapsto \phi(y_i) - x_i^{(L)}$. Let $h \in \mathcal{X}_{l_0}$ and $h' \in \mathcal{H}$. It holds:

$$\Big(dg(y)\Big)(h') = 2\Big\langle y, h' \Big\rangle_{\mathcal{H}},$$

and

$$\begin{aligned}
\Big(df(\varphi_{l_0,i_0})\Big)(h) &= \left(d\left(x_i - \sum_{i'=1}^{n} k_L\left(x_i^{(L-1)}, x_{i'}^{(L-1)}\right)\varphi_{L,i'}\right)(\varphi_{l_0,i_0})\right)(h), \\
&= -\sum_{i'=1}^{n}\left(d\left(k_L\left(x_i^{(L-1)}, x_{i'}^{(L-1)}\right)\varphi_{L,i'}\right)(\varphi_{l_0,i_0})\right)(h), \\
&= -\sum_{i'=1}^{n}\left(d\left(k_L\left(x_i^{(L-1)}, x_{i'}^{(L-1)}\right)\right)(\varphi_{l_0,i_0})\right)(h)\ \varphi_{L,i'}, \\
&= -\sum_{i'=1}^{n}\left\langle \nabla_{\varphi_{l_0,i_0}} k_L\left(x_i^{(L-1)}, x_{i'}^{(L-1)}\right), h\right\rangle_{\mathcal{H}} \varphi_{L,i'}.
\end{aligned}$$

Combining both expressions with $y = \phi(y_i) - x_i^{(L)}$ gives:

$$\begin{aligned}
&\left(d(\|\phi(y_i) - f_L \circ \ldots \circ f_1(x_i)\|^2_{\mathcal{X}_0})(\varphi_{l_0,i_0})\right)(h) \\
&\qquad = \left(d(g \circ f)(\varphi_{l_0,i_0})\right)(h), \\
&\qquad = \left(dg\left(\phi(y_i) - x_i^{(L)}\right)\right) \circ \left(df(\varphi_{l_0,i_0})\right)(h), \\
&\qquad = 2\left\langle \phi(y_i) - x_i^{(L)}, -\sum_{i'=1}^{n} \left\langle \nabla_{\varphi_{l_0,i_0}} k_L\left(x_i^{(L-1)}, x_{i'}^{(L-1)}\right), h \right\rangle_{\mathcal{X}_{l_0}} \varphi_{L,i'} \right\rangle_{\mathcal{H}}, \\
&\qquad = -2\sum_{i'=1}^{n} \left\langle \nabla_{\varphi_{l_0,i_0}} k_L\left(x_i^{(L-1)}, x_{i'}^{(L-1)}\right), h \right\rangle_{\mathcal{X}_{l_0}} \left\langle \phi(y_i) - x_i^{(L)}, \varphi_{L,i'} \right\rangle_{\mathcal{H}}, \\
&\qquad = \left\langle -2\sum_{i'=1}^{n} \left\langle \phi(y_i) - x_i^{(L)}, \varphi_{L,i'} \right\rangle_{\mathcal{H}} \nabla_{\varphi_{l_0,i_0}} k_L\left(x_i^{(L-1)}, x_{i'}^{(L-1)}\right), h \right\rangle_{\mathcal{X}_{l_0}}.
\end{aligned}$$

A direct identification gives.

$$\nabla_{\varphi_{l_0,i_0}} \left\|\phi(y_i) - x_i^{(L)}\right\|^2_{\mathcal{H}} = -2\sum_{i'=1}^{n} \left\langle \phi(y_i) - x_i^{(L)}, \varphi_{L,i'} \right\rangle_{\mathcal{H}} \nabla_{\varphi_{l_0,i_0}} k_L\left(x_i^{(L-1)}, x_{i'}^{(L-1)}\right). \tag{4.10}$$

Using Equation (4.7), that may also be written $n\Lambda_L \varphi_{L,i} = \phi(y_i) - x_i^{(L)}$, Equation (4.10) simplifies into

$$\nabla_{\varphi_{l_0,i_0}} \left\|\phi(y_i) - x_i^{(L)}\right\|^2_{\mathcal{H}} = -2n\Lambda_L \sum_{i'=1}^{n} [N_L]_{ii'} \nabla_{\varphi_{l_0,i_0}} k_L\left(x_i^{(L-1)}, x_{i'}^{(L-1)}\right). \tag{4.11}$$

Again, only computable or known quantities are involved. The gradient of the full criterion being just the weighted sum of the gradients of the distortion and the norm penalizations, we have thus shown that if N_L is fixed (and known), it is easy to propagate the gradient through the infinite dimensional layer.

Design of the Algorithm. The approach is natural. First freeze the last layer's coefficients (and consequently N_L). The gradient may be easily propagated. We use the shortcut notation $\nabla_{\Phi_l}\left(\hat{\epsilon}_n + \Omega \mid N_L\right)$ in Algorithm 4.1 to denote the gradient of the entire criterion with respect to Φ_l, assuming that N_L is fixed. After one gradient step, $\Phi_1, \ldots, \Phi_{L-1}$ are in turn frozen. The optimal $\varphi_{L,i}$ are then given by the KRR, which results in an update of N_L. Recall that N_{KRR} denotes the function that return the optimal KRR N_L matrix from K^X, K^Y, $\Phi_1, \ldots, \Phi_{L-1}$, and Λ_L. Let T be a number of epochs, and γ_t a step size rule, the approach is summarized in Algorithm 4.1.

Remark 4.6. *The crux of the algorithm is that infinite dimensional coefficients $\varphi_{L,i}$'s are never computed, but only their scalar products. For a Kernel Autoencoder, not knowing the $\varphi_{L,i}$'s is of no importance, as we are interested in the encoding function, which does not rely on them. Yet, one would not have direct access to the reconstructed inputs, but only to their discrepancy with respect to original inputs.*

Algorithm 4.1 General Hilbert Deep Kernel Machine

input : Gram matrices K^X, K^Y

init : $\Phi_1 = \Phi_1^{\text{init}}, \dots, \Phi_{L-1} = \Phi_{L-1}^{\text{init}}$, $N_L = N_{\text{KRR}}\Big(K^X, K^Y, \Phi_1, \dots, \Phi_{L-1}, \Lambda_L\Big)$

for *epoch* t *from* 1 *to* T **do**

  // inner coefficients updates at fixed N_L

  for *layer* l *from* 1 *to* $L-1$ **do**

    $\Phi_l = \Phi_l - \gamma_t \, \nabla_{\Phi_l} \Big(\hat{\epsilon}_n + \Omega \mid N_L\Big)$

  // N_L update

  $N_L = N_{\text{KRR}}\Big(K^X, K^Y, \Phi_1, \dots, \Phi_{L-1}, \lambda_L\Big)$

return $\Phi_1, \dots, \Phi_{L-1}$, N_L

Remark 4.7. *We highlight the fact that, if another update formula were available for N_L, then Algorithm 4.1 could be readily adapted. One could for instance imagine that N_L is not associated to the KRR minimizer, but to another criterion minimizer, such as the ϵ-insensitive Ridge regression, or the Hubert regression. If N_L then does not admit a closed form solution, the exact computation may be replaced by one iteration of an approximation algorithm. Attention should also be paid to the fact that changing the criterion may also change the gradient formula. This possibilities are investigated in Chapter 5.*

Remark 4.8. *As shall be seen in Chapter 5, and as highlighted by Equation (4.13), in many cases, the optimal coefficients $\varphi_{L,i}$ are actually linear combinations of the outputs $\phi(y_i)$. Denoting by W the $n \times n$ weight matrix such that $\varphi_{L,i} = \sum_j w_{ij}\phi(y_j)$, the last layer is fully characterized by W, which is finite dimensional. One can then consider a global Gradient Descent on $\Phi_1, \dots, \Phi_{L-1}, W$.*

The next section makes explicit the computation of the test distortion.

4.4.2 Test Distortion Computation

As previously highlighted, the final layer's coefficients are never computed explicitly. Yet, test distortion may always be computed, as long as dot products between the (kernelized) outputs $\langle \phi(y_{\text{test}}), \phi(y_{\text{train}}) \rangle = k(y_{\text{test}}, y_{\text{train}})$ are known.

Since we have assumed that A_L is the identity operator on $\mathcal{X}_L$, Equation (4.7) simplify to:

$$\forall\, i \leq n, \qquad \sum_{i'=1}^{n} W_{ii'}\, \varphi_{L,i'} = \phi(y_i), \tag{4.12}$$

where $W = K_L + n\Lambda_L \mathbf{I}_n$. It is then easy to show that the

$$\varphi_{L,i'} = \sum_{i=1}^{n} \Big[W^{-1}\Big]_{i'i}\, \phi(y_i) \qquad \forall\, i' \leq n \tag{4.13}$$

are solutions to Equation (4.12) and therefore to Equation (4.7). Notice that using this expansion directly leads to Equation (4.9). But more interestingly, this new writing allows for computing the distortion on a test set.

Indeed, let $x, \phi(y) \in \mathcal{X} \times \mathcal{H} = \mathcal{X}_L$, one has:

$$
\begin{aligned}
&\left\|\phi(y) - f_L \circ \ldots \circ f_1(x)\right\|_{\mathcal{H}}^2 \\
&= \left\|\phi(y) - f_L\left(x^{(L-1)}\right)\right\|_{\mathcal{H}}^2, \\
&= k(y,y) + \left\|f_L\left(x^{(L-1)}\right)\right\|_{\mathcal{H}}^2 - 2\left\langle \phi(y), f_L\left(x^{(L-1)}\right)\right\rangle_{\mathcal{H}}, \\
&= k(y,y) + \left\|\sum_{i=1}^{n} k_L\left(x^{(L-1)}, x_i^{(L-1)}\right)\varphi_{L,i}\right\|_{\mathcal{H}}^2 - 2\left\langle \phi(y), \sum_{i=1}^{n} k_L\left(x^{(L-1)}, x_i^{(L-1)}\right)\varphi_{L,i}\right\rangle_{\mathcal{H}}, \\
&= k(y,y) + \sum_{i,j=1}^{n} k_L\left(x^{(L-1)}, x_i^{(L-1)}\right) k_L\left(x^{(L-1)}, x_j^{(L-1)}\right)\left\langle \varphi_{L,i}, \varphi_{L,j}\right\rangle_{\mathcal{H}} \\
&\quad - 2\sum_{i=1}^{n} k_L\left(x^{(L-1)}, x_i^{(L-1)}\right)\left\langle \phi(y), \varphi_{L,i}\right\rangle_{\mathcal{H}}, \\
&= k(y,y) + \sum_{i,j=1}^{n} v_i [N_L]_{ij} v_j - 2\sum_{i,j=1}^{n} v_i \left[W^{-1}\right]_{i,j}\left\langle \phi(y), \phi(y_j)\right\rangle_{\mathcal{H}}, \\
&= k(y,y) + v^\top N_L v - 2v^\top W^{-1} u,
\end{aligned}
$$

with v the $\mathbb{R}^n$ vector such that its ith entry is equal to $v_i = k_L(x^{(L-1)}, x_i^{(L-1)})$, and u the counterpart for the $\langle \phi(y), \phi(y_j)\rangle_{\mathcal{H}}$. Just like in Section 4.4, knowing only the scalar products in the infinite dimensional space is enough to compute the test distortion, all other quantities involved being finite dimensional and thus naturally computable.

We now close this chapter by presenting some experimental results.

4.5 Numerical Experiments

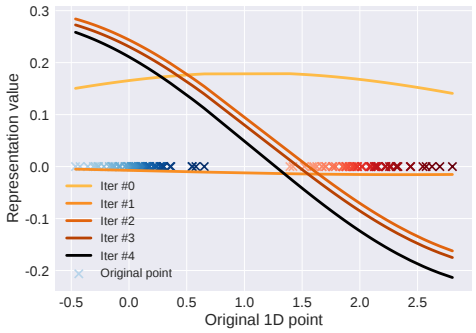
Numerical experiments have been run in order to highlight the capacity of Kernel Autoencoders to extract relevant data representations. We used decomposable OVKs with the identity operator as A, and the Gaussian kernel as k. First, we present insights on the interesting properties of the Kernel Autoencoders representations through 2D examples (Section 4.5.1). Then, we describe more involved experiments on the NCI dataset composed of molecules (Section 4.5.2).

4.5.1 Behavior on Low-Dimensional Problems

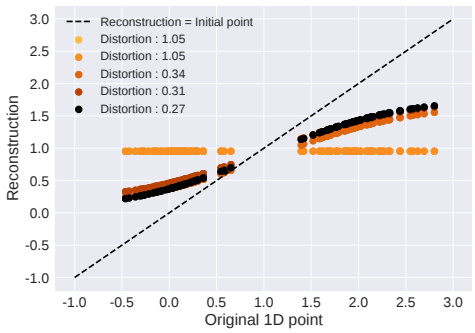
Experiments on low dimensional data have been run since results can be plotted easily. It thus provide useful insights on what Kernel Autoencoders are learning. We have explored several emblematic datasets so as to contrast the *disentangling* and *natural clustering* capacity of Kernel Autoencoders representations.

1D Gaussian Clusters

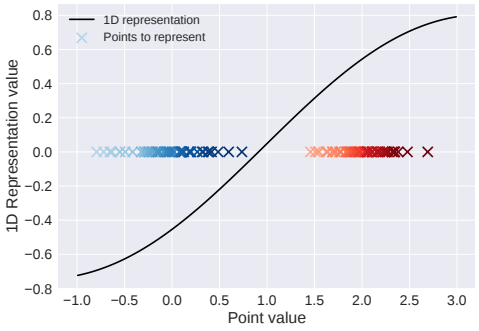
Figure 4.2 gives a look on the algorithm behavior on 1D data. Results on 1D data are displayed and analyzed here as they are easily understandable. Indeed, one dimension of the plot (the x axis) is used to display the original 1D points (the crosses), while their



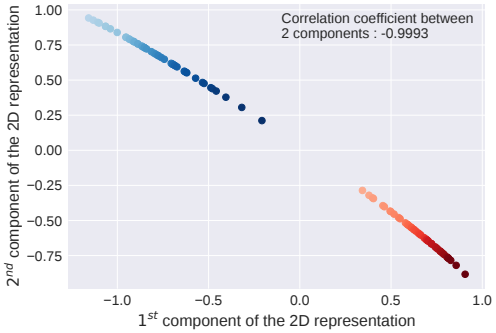
(a) Encoding evolution during fitting



(b) Decoding evolution during fitting



(c) 1D Gaussian clusters and 1D representation



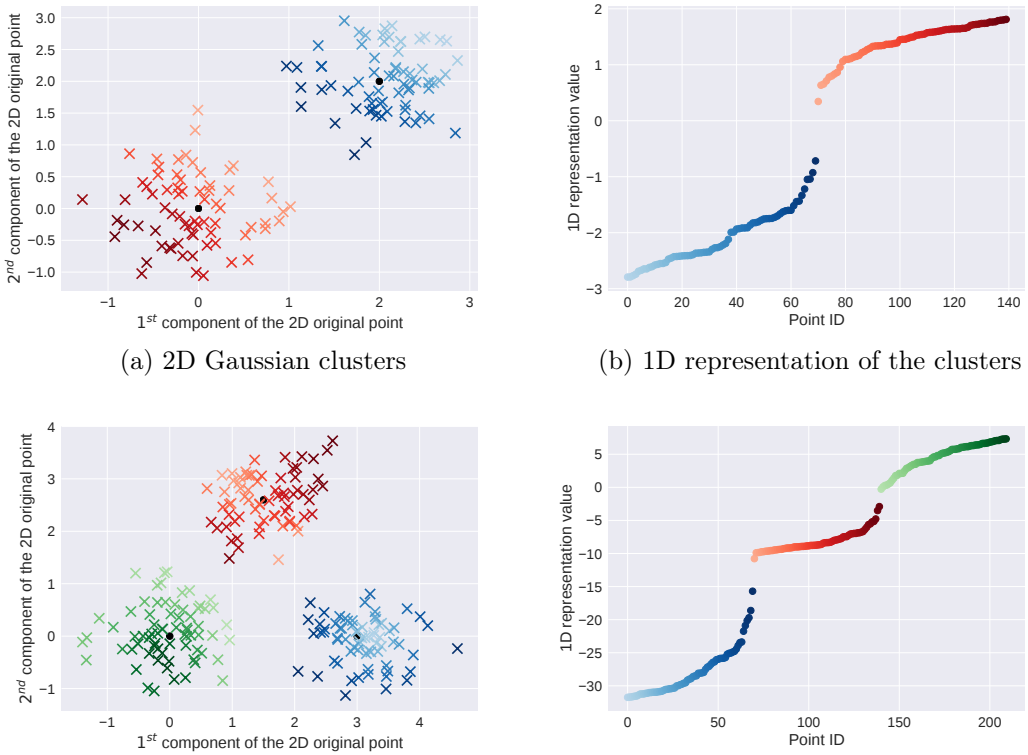
(d) 2D representation of the clusters

Figure 4.2 – Algorithm behavior on 1D data

representations (the $f_1(x_i)$) vary along the y axis. As soon as the original point or the representation needs more than 1 dimension to be plotted, a 2D plot lacks of dimensions to correctly display the behavior of the algorithm. Original data (to be represented) are sampled from 2 Gaussian distributions, of standard deviation 0.1, and with expected value 0 and 2 respectively.

Figures 4.2a and 4.2b show the evolution of the encoding/decoding functions along the iterations of the algorithm. From the initial yellow representation function, obtained by uniform weights, the algorithm learns the black function, which seems satisfying in two ways. First, the representations of the two clusters are easily separable. Points from the first blue cluster (i.e. drawn from the Gaussian centered at 0) have positive representations, while points from the red one (i.e. drawn from the Gaussian centered at 2) have negative ones. If computed in a clustering purpose, the representation thus gives an easy criterion to distinguish the two clusters. Second, in order to be able to reconstruct any point, one must observe variability within each cluster. This way, the reconstruction function can easily reassign every point. On the contrary, the yellow representation function represents all points by almost the same value, which leads necessarily to a uniform (and bad) reconstruction.

Figure 4.2c shows another 1D representation function of the two clusters, obtained with a different initialization. Solutions from Figures 4.2a and 4.2c seem both satisfactory, and highlight the difficulty of optimizing a non-convex criterion. Figure 4.2d shows a 2D encoding of these points. Interestingly, the two components of the 2D representation are highly correlated, underlining that 2D descriptors are over-parametrized for 1D points.



(a) 2D Gaussian clusters

(b) 1D representation of the clusters

(c) 2D Gaussian clusters (3)

(d) 1D representation of the clusters

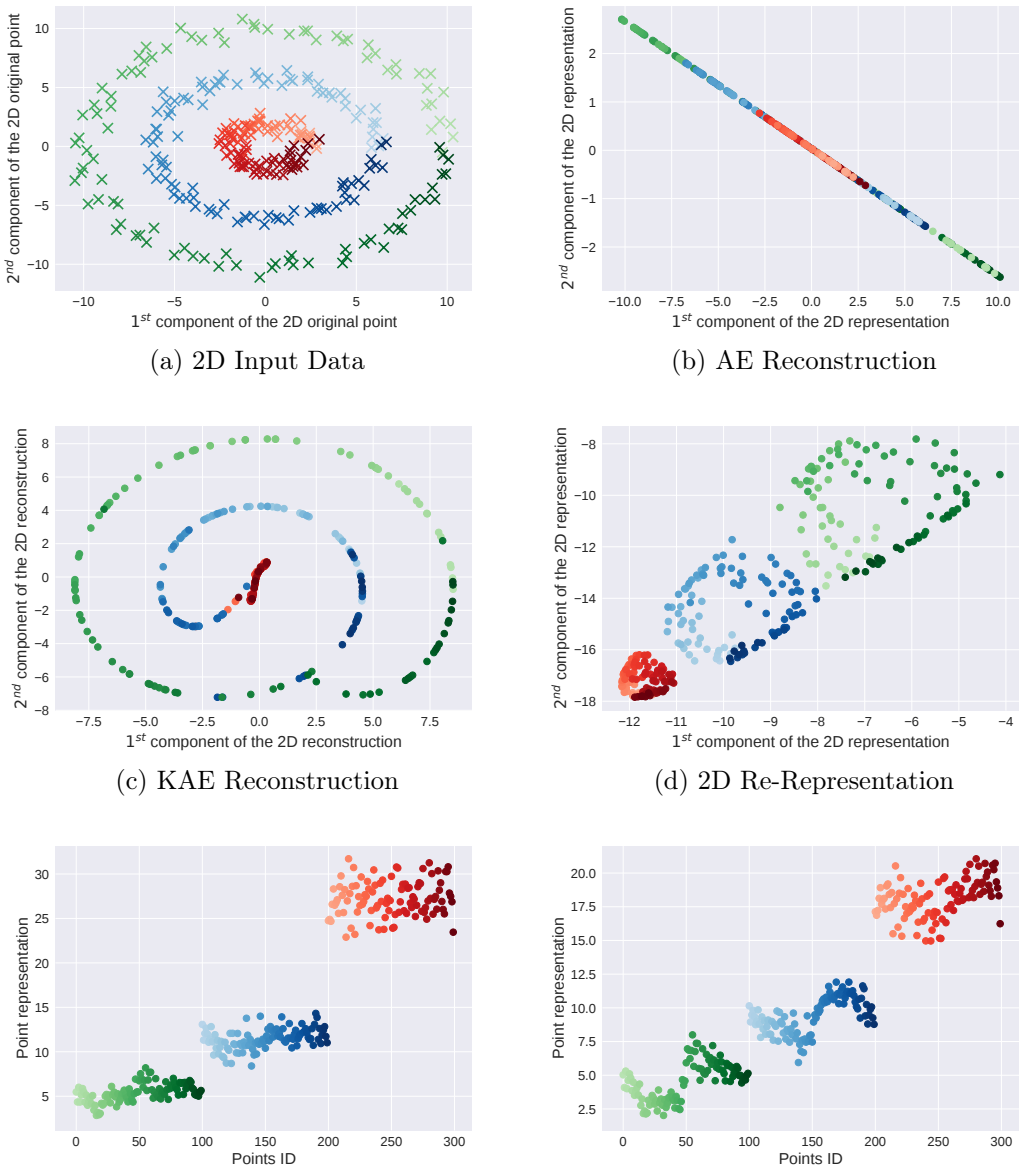
Figure 4.3 – Algorithm Behavior on 2D Gaussian Clusters

2D Gaussian Clusters

Figure 4.3 displays the algorithm's performance on 2D Gaussian clusters. The Kernel Autoencoder architecture is 2-1-2 here. In Figure 4.3a are plotted the original 2D data. Figure 4.3b shows their 1D internal learned representations. The colormap has been designed according to the value of this representation. What can be seen first is that the two clusters remain well separated in the 1 dimensional representation space, validating the *natural clustering* ability of Kernel Autoencoders representations. What is even more interesting is how the clusters are separated. The lighter the blue points are, the most negative representation they have, or in other terms, the *most certain* they are to be in the blue cluster. The converse goes for the darker red points. When looking at Figure 4.3a, one observes that this color gradient matches the distribution: light blue points are the most distant from the red cluster, and conversely for the dark red ones. The algorithm has found the direction that discriminates the two clusters. Similar results are shown for 3 Gaussian clusters on Figures 4.3c and 4.3d.

Concentric Circles

Let us now consider three noisy concentric circles such as in Figure 4.4a. Although the main strength of Kernel Autoencoders is to perform autoencoding on complex data (Section 3.2.3), they can still be applied on vector-valued points. Figures 4.4b and 4.4c show the reconstructions obtained after fitting respectively a 2-1-2 standard and kernel Autoencoders. Since the latent space is 1 dimensional, the 2D reconstructions are on manifolds of the same dimension, hence the curve aspect. What is interesting though



(a) 2D Input Data

(b) AE Reconstruction

(c) KAE Reconstruction

(d) 2D Re-Representation

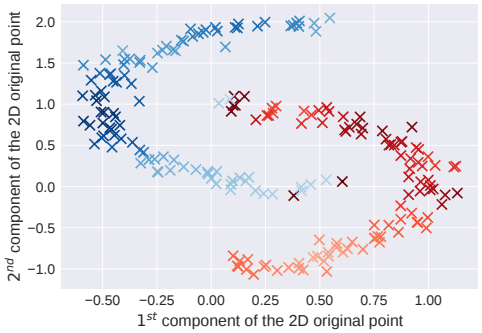
(e) 1D Representation (5 it.)

(f) 1D Representation (20 it.)

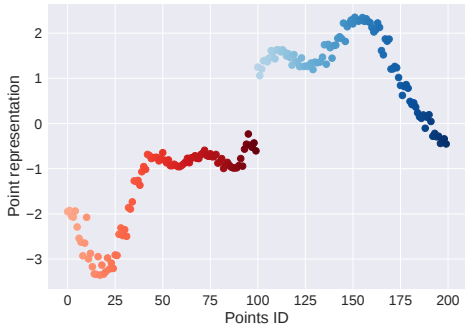
Figure 4.4 – Kernel Autoencoder (KAE) Performance on Noisy Concentric Circles

is that Kernel Autoencoders learn much more complex manifolds than standard ones. Due to its linear limitations (activation functions did not help much here), the standard Autoencoder returns a line, very different from the concentric circle, while the Kernel Autoencoder outputs a more complex manifold, almost reproducing the initial data.

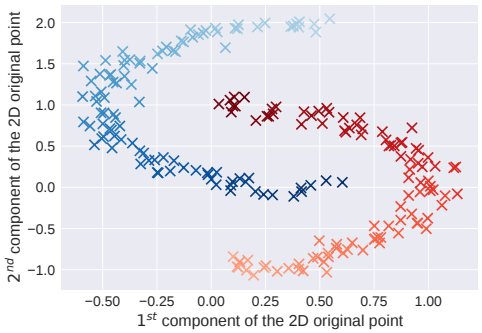
Apart from a good reconstruction, we are also interested in finding representations with attractive properties. The 1D feature found by the previous Kernel Autoencoder is interesting, as it is a discriminative one with respect to the original clusters: points from different circles are mapped around different values (Figure 4.4e). Interestingly, after a few iterations, some variability is introduced around these *cluster values*, so that all codes shall not be mapped back to the same point (Figure 4.4f). This *natural clustering* property was desired in Bengio et al. (2013a) for instance.



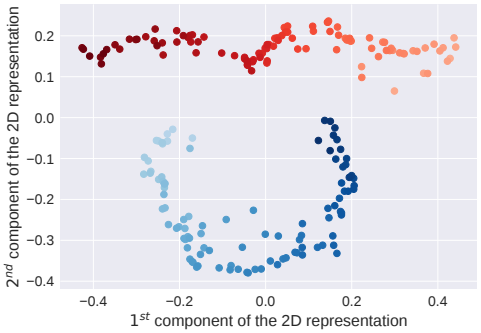
(a) Two moons dataset, colored w.r.t. its 1D representation



(b) 1D representation of the 2 moons



(c) Two moons dataset, colored w.r.t. its 2D representation



(d) 2D representation of the 2 moons

Figure 4.5 – Algorithm behavior on the 2 moons dataset

Finally, a Kernel Autoencoder with 1 hidden layer of size 2 architecture provides the internal representations shown in Figure 4.4d. This new 2D representation has a new interesting disentangling effect: the circle structure is kept in order to preserve the intra-cluster specificity, while the inter-cluster differentiation is ensured by the circles dissociation. Again, the disentangling effect is often a desired feature for representation. One can indeed imagine that classifiers run on the new representations of Figure 4.4d would perform better in average that if they were run on original points of Figure 4.4a.

This visual 2D example thus gives interesting insights on the good properties of the representations computed by Kernel Autoencoders: natural clustering, disentanglement. These two key aspects can notably be found among the list of representations desired properties exposed in Bengio et al. (2013a).

Two Moons Dataset

Finally, Figure 4.5 shows the algorithm's behavior on the so called *two moons dataset*. 2D original points (Figure 4.5a and Figure 4.5c, colored differently according to the representation on their right) are first mapped to a 1D representation (Figure 4.5b). Just as for the 3 concentric circles example, this 1D representation is discriminative, also with intra-cluster variability to reconstruct accurately. The 2D re-representations on Figure 4.5d shows again the disentangling effect of Kernel Autoencoders representations.

Table 4.1 – MSREs on Test Metabolites

Dimension	AE (sigmoid)	AE (relu)	KAE
5	99.81	96.62	**76.38**
10	87.36	84.02	**65.76**
25	72.31	68.77	**51.63**
50	63.00	58.29	**40.72**
100	55.43	48.63	**36.27**

4.5.2 Representation Learning on Molecules

We now present an application of Kernel Autoencoders (KAEs) in chemoinformatics. The motivation is triple. First, such complex data cannot be handled by standard Autoencoders. Second, kernel methods being prominent in the field, data are often stored as Gram matrices, suiting perfectly our framework. Third, finding a compressed representation of a molecule is a problem of highest interest in Drug Discovery. We considered two different problems, one supervised, one unsupervised.

As for the supervised one, we exploited the dataset of Su et al. (2010) from the NCI-cancer. database: it consists in a Gram matrix comparing 2303 molecules by the mean of a Tanimoto kernel (a linear path kernel built using the presence or absence of sequences of atoms in the molecule), as well as the molecules activities in the presence of 59 types of cancer. The dataset containing no vectorial representations of the molecules (but only Gram matrices), only kernel methods were possible to benchmark. As a good representation is supposed to facilitate ulterior learning tasks, we assess the goodness of the representations through the regression scores obtained by Random Forests (RFs) from scikit-learn (Pedregosa et al., 2011) fed with it.

2-layer KAEs with respectively 5, 10, 25, 50 and 100 internal dimension were run, as well as Kernel Principal Component Analyses (KPCAs) with the same number of components. Finally, these representations were given as inputs to RFs. KRR was also added to the comparison. The Normalized Mean Squared Errors (NMSEs), averaged on 10 runs, for all strategies and on the first 8 cancers are displayed in Figure 4.6. Clearly, methods combining a data representation step followed by a prediction one performs better. But the good performance of our approach should not be attributed to the use of RFs only, since the same strategy run with KPCA leads to worse results. Indeed, the KAE 50 + RF strategy outperforms all other procedures on all problems, managing to extract compact and useful feature vectors from the molecules.

The data for the unsupervised problem is taken from Brouard et al. (2016a). It is composed of two sets (a train set of size 5579, and a test set of size 1395), each one containing metabolites under the form of 4136-long binary vectors (called fingerprints), as well as a Gram matrix comparing them. 2-layer standard AEs from Keras (Chollet et al., 2015) with sigmoid and relu activation functions, and 2-layer KAEs with internal layer of size 5, 10, 25, 50 and 100, were trained. In absence of a supervised task, we measured the Mean Squared Reconstruction Errors (MSREs) induced on the test set, and stored them in Table 4.1. Again, the KAE approach shows a systematic improvement.

All Strategies on 8 Cancers Graph

As expected, the greater the dimension of the extracted representations, the better the prediction performance by the RF, for both KAE and KPCA. However, it is worth noticing that for cancer 7, the prediction error increases between the 50 and the 100-long representations. This might be the beginning of an overfitting phenomenon (seen on 8 of the 59 cancer types, always between the 50 and the 100-dimensional representations), as the extracted components may become less relevant, thus misleading the RF in its predictions.

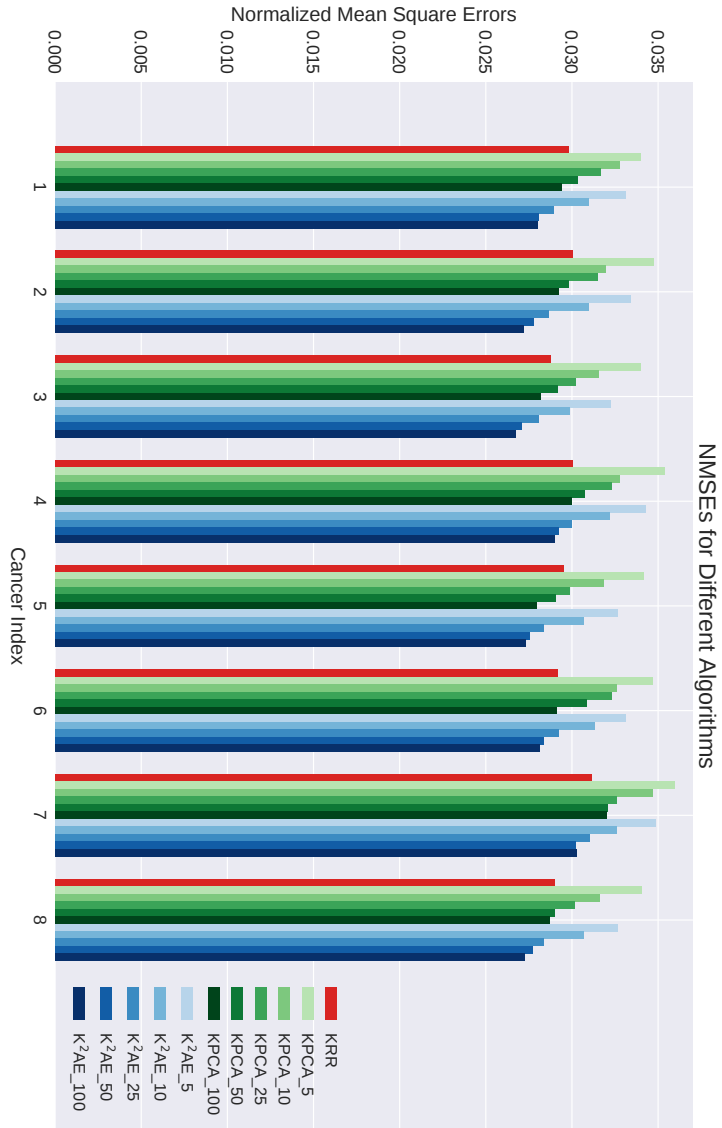


Figure 4.6 – Performance of the Different Strategies on 8 Cancers

Table 4.2 – NMSEs on Molecular Activity for Different Types of Cancers

	KRR	$KPCA_{10}$ + RF	$KPCA_{50}$ + RF	KAE_{10} + RF	KAE_{50} + RF
Canc. 01	0.02978	0.03279	0.03035	0.03097	**0.02808**
Canc. 02	0.03004	0.03194	0.02978	0.03099	**0.02775**
Canc. 03	0.02878	0.03155	0.02914	0.02989	**0.02709**
Canc. 04	0.03003	0.03274	0.03074	0.03218	**0.02924**
Canc. 05	0.02954	0.03185	0.02903	0.03065	**0.02754**
Canc. 06	0.02914	0.03258	0.03083	0.03134	**0.02838**
Canc. 07	0.03113	0.03468	0.03207	0.03257	**0.03018**
Canc. 08	0.02899	0.03162	0.02898	0.03065	**0.02770**
Canc. 09	0.02860	0.02992	0.02804	0.02872	**0.02627**
Canc. 10	0.02987	0.03291	0.03111	0.03170	**0.02910**
Canc. 11	0.03035	0.03258	0.03095	0.03188	**0.02900**
Canc. 12	0.03178	0.03461	0.03153	0.03253	**0.02983**
Canc. 13	0.03069	0.03338	0.03104	0.03162	**0.02857**
Canc. 14	0.03046	0.03340	0.03102	0.03135	**0.02862**
Canc. 15	0.02910	0.03221	0.03066	0.03131	**0.02806**
Canc. 16	0.02956	0.03220	0.02958	0.03060	**0.02779**
Canc. 17	0.03004	0.03413	0.03140	0.03145	**0.02869**
Canc. 18	0.02954	0.03195	0.03005	0.03108	**0.02805**
Canc. 19	0.03003	0.03211	0.03079	0.03178	**0.02832**
Canc. 20	0.02911	0.03179	0.03041	0.03085	**0.02769**
Canc. 21	0.02963	0.03275	0.03023	0.03152	**0.02837**
Canc. 22	0.03075	0.03391	0.03089	0.03263	**0.02958**
Canc. 23	0.03006	0.03286	0.02983	0.03109	**0.02760**
Canc. 24	0.03075	0.03398	0.03112	0.03242	**0.02894**
Canc. 25	0.02977	0.03307	0.03054	0.03159	**0.02824**
Canc. 26	0.03083	0.03358	0.03132	0.03206	**0.02959**
Canc. 27	0.03083	0.03347	0.03116	0.03230	**0.02974**
Canc. 28	0.03061	0.03256	0.03116	0.03185	**0.02918**
Canc. 29	0.03056	0.03360	0.03147	0.03181	**0.02892**
Canc. 30	0.03099	0.03288	0.03100	0.03181	**0.02906**
Canc. 31	0.03082	0.03361	0.03161	0.03242	**0.02986**
Canc. 32	0.03233	0.03562	0.03300	0.03422	**0.03158**
Canc. 33	0.03065	0.03208	0.03045	0.03142	**0.02909**
Canc. 34	0.03326	0.03668	0.03423	0.03486	**0.03183**
Canc. 35	0.03292	0.03587	0.03393	0.03450	**0.03146**
Canc. 36	0.03068	0.03389	0.03122	0.03249	**0.02925**
Canc. 37	0.03023	0.03310	0.03061	0.03130	**0.02878**
Canc. 38	0.03100	0.03487	0.03156	0.03327	**0.02974**
Canc. 39	0.02989	0.03288	0.03149	0.03148	**0.02865**
Canc. 40	0.03166	0.03525	0.03201	0.03352	**0.03010**
Canc. 41	0.03139	0.03501	0.03203	0.03316	**0.03025**
Canc. 42	0.03010	0.03251	0.03013	0.03072	**0.02807**
Canc. 43	0.03042	0.03324	0.03062	0.03144	**0.02806**
Canc. 44	0.02838	0.03045	0.02821	0.02927	**0.02679**
Canc. 45	0.02910	0.03085	0.02895	0.02970	**0.02651**
Canc. 46	0.02969	0.03258	0.02996	0.03111	**0.02834**
Canc. 47	0.03148	0.03438	0.03346	0.03286	**0.03056**
Canc. 48	0.03272	0.03640	0.03397	0.03425	**0.03197**
Canc. 49	0.03305	0.03392	0.03329	0.03334	**0.03148**
Canc. 50	0.03229	0.03637	0.03300	0.03404	**0.03155**
Canc. 51	0.02943	0.03188	0.03028	0.03072	**0.02857**
Canc. 52	0.03309	0.03420	0.03252	0.03335	**0.03130**
Canc. 53	0.03170	0.03340	0.03105	0.03170	**0.02843**
Canc. 54	0.03189	0.03439	0.03164	0.03345	**0.03036**
Canc. 55	0.03082	0.03339	0.03146	0.03207	**0.02892**
Canc. 56	0.03026	0.03327	0.03041	0.03185	**0.02901**
Canc. 57	0.02962	0.03237	0.02990	0.03162	**0.02855**
Canc. 58	0.02883	0.03200	0.02978	0.03058	**0.02783**
Canc. 59	0.02936	0.03208	0.02914	0.03032	**0.02750**

4.6 Conclusion

After a thorough description of the Kernel Autoencoder model in Chapter 3, the goal of Chapter 4 was to address the optimization issues it raises. Despite the non-convexity of the criterion, a Representer Theorem dedicated to the composition architecture makes it possible to learn Kernel Autoencoders via Gradient Descent. When the output space is infinite dimensional, the Gradient Descent must be alternated with Kernel Ridge resolutions. Finally, numerical experiments run on synthetic and molecular datasets exhibit the good properties of representations extracted by Kernel Autoencoders, among which *natural clustering* and *disentanglement*.

The optimization process detailed in this chapter also applies to Deep Kernel Machines, for which outputs (potentially kernelized) may differ from inputs. If experiments focus mainly on autoencoding applications, benchmarking Deep Kernel Machines on tasks such as metabolite identification could be of high interest. In the next chapter, we consider other loss functions at the last layer than the squared norm in the output feature space. So far unused within vv-RKHSs, these new losses define OVK machines that must be solved through duality. They also yield important modifications, such as sparsity, and improve the performances.

5

Dualizing Operator-Valued Kernel Machines

Contents

So far, once the outputs (or the inputs and the outputs in the case of KAE) have been embedded through the feature map ϕ, the sole loss function that have been considered in the output Hilbert space is the squared norm. This is indeed a natural choice, as it can be computed from kernel evaluations only. Furthermore, the operator-valued Kernel Ridge Regression (ov-KRR) problems that consequently arise at the last layer of the architecture admit well-known closed form solutions that allow for the gradient to propagate.

Nevertheless, it is legitimate to wonder if sophisticated approaches such as Deep IOKR and KAE do not suffer from the simplicity of the ov-KRR that tackles the surrogate task. This is all the more true as in both cases, the vv-RKHS norm minimization problem is a proxy for the real task, respectively predicting a molecule or extracting relevant representations. Hence, proceeding with a strong data-fitting term, even though the criterion is not the main goal pursued, and possibly at the expense of a generalization capacity diminution, does not sound as an ideal solution. However considering other losses than the squared norm necessitates more work as one cannot rely on closed form solutions anymore. In contrast, the alternative explored in this chapter leverage a dualization approach.

While brief reminders on duality within kernel methods are exposed in Section 5.1, the main tool of this chapter, referred to as the *Double Representer Theorem*, is stated and proved in Section 5.2. This general theorem is then used to instantiate specific solvable problems for two interesting loss functions, the ϵ-insensitive squared norm and the

Huber loss, in Section 5.3. Section 5.4 focuses on the particular case of *integral losses*, while Section 5.5 finally presents some numerical experiments attesting the benefits of considering other losses than the squared norm. This chapter covers the following work:

▶ **P. Laforgue***, A. Lambert*, L. Brogat-Motte, F. d'Alché-Buc. On the dualization of operator-valued kernel machines. *arXiv preprint arXiv:1910.04621*, 2019.

5.1 Reminders on Duality

In this section, we briefly recall basics about dualization that are needed for the rest of the chapter. In Section 5.1.1 are provided general statements about dualization, while Section 5.1.2 focuses on their application to scalar kernel machines.

5.1.1 General Reminders

In convex optimization (refer to the monographs by *e.g.* Rockafellar (1970); Boyd and Vandenberghe (2004); Nocedal and Wright (2006); Bauschke et al. (2011)), duality is an approach in which the original optimization problem, referred to as the *primal problem*, is reformulated as a *dual problem*, usually easier to solve. Under certain conditions, solutions to both the primal and the dual problems coincide, and solving the dual provides a workaround to the initial problem. These notions are now to be formalized. For the sake of simplicity, the definition-example below has parameter x in $\mathbb{R}^d$, but the extension to general Hilbert spaces can be found in Bauschke et al. (2011).

An optimization problem in standard form can be written as

$$\begin{array}{lll} \min\limits_{x\in\mathbb{R}^d} & f_0(x), & \\ \text{s.t.} & f_i(x)\leq 0, & i\leq m, \\ & h_j(x)=0, & j\leq p, \end{array} \tag{5.1}$$

with f_0, f_i, $i\leq m$, and h_j, $j\leq p$, applications from $\mathbb{R}^d$ to $\mathbb{R}$. The *Lagrangian* associated to (primal) Problem (5.1) is then defined as

$$\mathscr{L}: \begin{pmatrix} \mathbb{R}^d\times\mathbb{R}^m\times\mathbb{R}^p & \to & \mathbb{R} \\ (x,\alpha,\nu) & \mapsto & f_0(x)+\sum_i \alpha_i f_i(x)+\sum_j \nu_j h_j(x) \end{pmatrix},$$

and the *Lagrange dual function* is given by

$$g: \begin{pmatrix} \mathbb{R}^m\times\mathbb{R}^p & \to & \mathbb{R} \\ (\alpha,\nu) & \mapsto & \inf\limits_{x\in\mathbb{R}^d} \mathscr{L}(x,\alpha,\nu) \end{pmatrix}.$$

One can already notice that for any $\alpha\in\mathbb{R}^m_+$ and any $\nu\in\mathbb{R}^p$, it holds $g(\alpha,\nu)\leq p^*$, with p^* the value at the optimum in primal Problem (5.1).

The *Lagrange dual problem* associated to Problem (5.1) then writes

$$\begin{array}{ll} \max & g(\alpha,\nu), \\ \text{s.t.} & \alpha\succeq 0. \end{array} \tag{5.2}$$

The *weak duality* holds for any problem and ensures that $d^*\leq p^*$, with d^* the value at the optimum in dual Problem (5.2). The next proposition states sufficient conditions for the equality to hold.

Proposition 5.1. (Slater's Condition) *Assume that primal Problem (5.1) is convex, i.e. f_0 is convex defined on a convex set, f_i, $i \leq m$, are convex, and h_j, $j \leq p$, are affine. If furthermore the problem is strictly feasible, i.e.*

$$\exists x \in \textbf{relint } \mathrm{dom}_{f_0}, \quad f_i(x) < 0, \ i \leq m, \quad h_j(x) = 0, \ j \leq p,$$

then strong duality *holds: $p^* = d^*$.*

Conditions ensuring that strong duality holds in convex problems are called *constraint qualifications*. The next proposition furnishes other constraint qualifications. They are more used in practice as they help solving the optimization problems.

Proposition 5.2. (Karush-Kuhn-Tucker Conditions) *The following conditions are necessary and sufficient for strong duality to hold in (differentiable) convex problems.*

1. *Primal feasibility: $f_i(x) \leq 0$ for $i \leq m$ and $h_j(x) = 0$ for $j \leq p$.*
2. *Dual feasibility: $\alpha \succeq 0$.*
3. *Complementary slackness: $\alpha_i f_i(x) = 0$ for $i \leq m$.*
4. *First order condition: $\nabla f_0(x) + \sum_i \alpha_i \nabla f_i(x) + \sum_j \nu_j \nabla h_j(x) = 0$.*

We close this subsection with two important definitions, that of the *Fenchel-Legendre* transform, that naturally arises during the derivation of dual problems, and that of the infimal-convolution (Bauschke et al., 2011), which is crucial in this chapter's analysis.

Definition 5.3. *Let $f : \mathbb{R}^d \to \mathbb{R}$. The* Fenchel-Legendre transform *of f is the function $f^\star : \mathbb{R}^d \to \mathbb{R}$ defined as*

$$f^\star(x) = \sup_y \langle x, y \rangle - f(y).$$

Definition 5.4. *Let $f, g : \mathbb{R}^d \to \mathbb{R}$. The* infimal-convolution *of f and g is the function $f \square g : \mathbb{R}^d \to \mathbb{R}$ defined as*

$$f \square g(x) = \inf_y f(y) + g(x - y).$$

Proposition 5.5. *Using Definitions 5.3 and 5.4's notation, it holds*

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

The next subsection then deals with applications of the duality to scalar kernel machines.

5.1.2 Application to Scalar Kernel Machines

The use of duality has a long history in scalar kernel methods. Indeed, if plugging the expansion obtained by the Representer Theorem (Theorem 2.5) into the KRR criterion is enough to derive the optimal coefficient, this is not the case in general. For instance, the Support Vector Machines (SVMs, Cortes and Vapnik (1995)) that addresses binary classification ($y_i \in \{-1, +1\}$) has primal problem

$$\min_{h \in \mathcal{H}_k, \ b \in \mathbb{R}} \quad \frac{1}{n} \sum_{i=1}^n \max\Big(0, 1 - y_i(h(x_i) + b)\Big) + \frac{\Lambda}{2} \|h\|^2_{\mathcal{H}_k},$$

that rewrites with the introduction of slack variables

$$\begin{aligned} \min_{h,b,u} \quad & \sum_{i=1}^{n} u_i + \frac{\Lambda}{2}\|h\|^2_{\mathcal{H}_k}, \\ \text{s.t.} \quad & -u_i \leq 0, && \text{for } i \leq n, \\ & 1 - y_i(h(x_i) + b) - u_i \leq 0, && \text{for } i \leq n. \end{aligned}$$

Using the Karush-Kuhn-Tucker (KKT) conditions, and with the notation $\alpha = (\alpha_i)_{i=1}^n$, the dual problem writes

$$\begin{aligned} \max_{\alpha} \quad & \mathbf{1}^\top \alpha - \frac{1}{2}\alpha^\top \bar{K} \alpha, \\ \text{s.t.} \quad & \mathbf{0} \preceq \alpha \preceq \mathbf{1}/\Lambda, \\ & \alpha^\top \boldsymbol{y} = 0, \end{aligned}$$

with $\bar{K}$ the $n \times n$ matrix such that $\bar{K}_{ij} = y_i k(x_i, x_j) y_j$, and $\hat{h}$ given by $\hat{h} = \sum_i y_i \alpha_i k(\cdot, x_i)$.

This latter problem is a classical Quadratic Program, that can be solved easily, leading to an efficient resolution although the primal problem seems, at first, more complex than that of the KRR for instance. This approach has later been applied to regression (Support Vector Regressors, SVRs, Drucker et al. (1997)), Ridge Regression (Saunders et al., 1998), or Least-Squares SVMs (Suykens et al., 2002) among others.

Yet, very few attempts have been made in the literature to adapt this methodology to general operator-valued kernel machines, although the idea was already present in the Appendix of Brouard et al. (2016b), or in Sangnier et al. (2017) for the finite dimensional case. One reason may be that the dual problem, in its most general form, involves infinite dimensional Lagrange multipliers. This difficulty is overcome in the present chapter through the use of a *Double Representer Theorem*. This is the subject of the next section.

5.2 The Double Representer Theorem

In this section, we state and prove the *Double Representer Theorem*, that makes dual OVK problems computable in most settings (Section 5.2.2). Prior to the statement, we recall in Section 5.2.1 two mentions in the literature of the use of duality within matrix-valued and operator-valued kernels. Section 5.2.3 is finally devoted to the analysis of the hypotheses needed.

5.2.1 Previous Mentions of Duality within vv-RKHSs

For self-containedness, we first recall our learning setting. We want to learn a prediction function h^* from some metric space $\mathcal{X}$ to some (infinite dimensional) output Hilbert space $\mathcal{Y}$, that minimizes among a hypothesis set $\mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$ the risk

$$\mathbb{E}_{Z \sim P}\left[\ell\Big(h(X), Y\Big)\right],$$

where $Z = (X, Y)$ is a random vector valued in $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ with unknown probability distribution P, and $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ is a given loss function. The class $\mathcal{H}$ is assumed to be a vv-RKHS $\mathcal{H}_{\mathcal{K}}$, associated to some OVK $\mathcal{K} : \mathcal{X} \times \mathcal{X} \to \mathcal{L}(\mathcal{Y})$. Following the (regularized) ERM paradigm (recalled in Chapter 1), and given $\mathcal{S}_n = \{(x_i, y_i)_{i=1}^n\} \in \mathcal{Z}^n$

a sample of n i.i.d. realizations of Z, the general form of an OVK learning problem is to find $\hat{h}$ that solves:

$$\min_{h\in\mathcal{H}_\mathcal{K}} \frac{1}{n}\sum_{i=1}^{n}\ell\Big(h(x_i),y_i\Big)+\frac{\Lambda}{2}\|h\|^2_{\mathcal{H}_\mathcal{K}}. \tag{5.3}$$

A crucial tool in kernel methods to address these problems is the *Representer Theorem* (Theorem 2.12), ensuring that $\hat{h}$ actually pertains to a reduced subspace of $\mathcal{H}_\mathcal{K}$. For the sake of self-containedness of the present analysis, it is briefly recalled here.

Theorem 5.6. *There exist $(\hat{\alpha}_i)_{i=1}^n \in \mathcal{Y}^n$ such that Problem (5.3)'s solution $\hat{h}$ is given by*

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

Although Theorem 5.6 drastically downscales the search domain (from $\mathcal{H}_\mathcal{K}$ to $\mathcal{Y}^n$), it gives no additional information about the optimal coefficients $(\hat{\alpha}_i)_{i=1}^n$. One way to gain insight is to perform Problem (5.3)'s dualization, with the notation $\ell_i : y \in \mathcal{Y} \mapsto \ell(y,y_i)$ for any $i \leq n$.

Theorem 5.7. *The solution to Problem (5.3) is given by*

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

with $(\hat{\alpha}_i)_{i=1}^n \in \mathcal{Y}^n$ the solutions to the dual problem

$$\min_{(\alpha_i)_{i=1}^n\in\mathcal{Y}^n} \sum_{i=1}^{n}\ell_i^\star(-\alpha_i)+\frac{1}{2\Lambda n}\sum_{i,j=1}^{n}\Big\langle\alpha_i,\mathcal{K}(x_i,x_j)\alpha_j\Big\rangle_\mathcal{Y}, \tag{5.4}$$

Proof. First, notice that the primal problem

$$\min_{h\in\mathcal{H}_\mathcal{K}} \frac{1}{n}\sum_{i=1}^{n}\ell\Big(h(x_i),y_i\Big)+\frac{\Lambda}{2}\|h\|^2_{\mathcal{H}_\mathcal{K}}$$

can be rewritten

$$\begin{aligned}\min_{h\in\mathcal{H}_\mathcal{K}} \quad & \sum_{i=1}^{n}\ell_i(u_i)+\frac{\Lambda n}{2}\|h\|^2_{\mathcal{H}_\mathcal{K}},\\ \text{s.t.} \quad & u_i = h(x_i) \quad \forall i \leq n.\end{aligned}$$

Therefore, with the notation $\boldsymbol{u}=(u_i)_{i\leq n}$ and $\boldsymbol{\alpha}=(\alpha_i)_{i\leq n}$, the Lagrangian writes

$$\begin{aligned}\mathscr{L}(h,\boldsymbol{u},\boldsymbol{\alpha}) &= \sum_{i=1}^{n}\ell_i(u_i)+\frac{\Lambda n}{2}\|h\|^2_{\mathcal{H}_\mathcal{K}}+\sum_{i=1}^{n}\Big\langle\alpha_i,u_i-h(x_i)\Big\rangle_\mathcal{Y},\\ &= \sum_{i=1}^{n}\ell_i(u_i)+\frac{\Lambda n}{2}\|h\|^2_{\mathcal{H}_\mathcal{K}}+\sum_{i=1}^{n}\langle\alpha_i,u_i\rangle_\mathcal{Y}-\sum_{i=1}^{n}\Big\langle\mathcal{K}(\cdot,x_i)\alpha_i,h\Big\rangle_{\mathcal{H}_\mathcal{K}}.\end{aligned}$$

Differentiating with respect to h and using Definition 5.3, one gets

$$\begin{aligned} g(\boldsymbol{\alpha}) &= \inf_{h\in\mathcal{H}_\mathcal{K},\boldsymbol{u}\in\mathcal{Y}^n} \mathscr{L}(h,\boldsymbol{u},\boldsymbol{\alpha}), \\ &= \sum_{i=1}^n \inf_{u_i\in\mathcal{Y}} \left\{\ell_i(u_i) + \langle \alpha_i, u_i\rangle_\mathcal{Y}\right\} + \inf_{h\in\mathcal{H}_\mathcal{K}} \left\{\frac{\Lambda n}{2}\|h\|^2_{\mathcal{H}_\mathcal{K}} - \sum_{i=1}^n \left\langle \mathcal{K}(\cdot,x_i)\alpha_i, h\right\rangle_{\mathcal{H}_\mathcal{K}}\right\}, \\ &= \sum_{i=1}^n -\ell_i^\star(-\alpha_i) - \frac{1}{2\Lambda n}\sum_{i,j=1}^n \left\langle \alpha_i, \mathcal{K}(x_i,x_j)\alpha_j\right\rangle_\mathcal{Y}, \end{aligned}$$

together with the equality $\hat{h} = \dfrac{1}{\Lambda n}\displaystyle\sum_{i=1}^n \mathcal{K}(\cdot,x_i)\alpha_i$. The conclusion follows immediately. □

Theorem 5.7 can be found in the Appendix of Brouard et al. (2016b) in its most general form presented here, but is only used in the paper for maximum-margin regression and Ridge Regression, for which it is not even necessary. A version for matrix-valued kernels is stated in Sangnier et al. (2017) (Proposition 1 therein), with a focus on ϵ-insensitive losses exclusively.

Compared to Theorem 5.6, dualization thus brings more knowledge about the optimal coefficients, now known to be the solutions to the dual problem also. Notice nonetheless that the Representer Theorem holds true for a much wider class of problems, explaining this deficit of information. As such, Problem (5.4) is however of little interest, as the optimization must be performed on the infinite dimensional space $\mathcal{Y}^n$, which is merely impossible. The next section introduces a *Double Representer Theorem* that permits to get around this difficulty. From now on, all presented results have been established in **Laforgue** et al. (2019c).

5.2.2 Theorem Statement

In order to make Problem (5.4) solvable, we need some (mild) assumptions on the kernel and the loss function. The next one has already been exposed in Chapter 2 (see Definition 2.9 therein), but is recalled here for self-containedness purposes. Here and throughout, $\mathbf{Y}$ denotes $\text{span}\{y_i,\ i\leq n\}$.

Assumption 5.8. *The OVK $\mathcal{K}:\mathcal{X}\times\mathcal{X}\to\mathcal{L}(\mathcal{Y})$ is said to be* separable *if and only if there exist a scalar kernel $k:\mathcal{X}\times\mathcal{X}\to\mathbb{R}$, and a positive semi-definite operator $A\in\mathcal{L}(\mathcal{Y})$ such that:*

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

If furthermore $A=\boldsymbol{I}_\mathcal{Y}$, then $\mathcal{K}$ is said to be identity decomposable.

Under Assumption 5.8, K^X and K^Y_A denote respectively the $n\times n$ input and output Gram matrices such that $[K^X]_{ij} = k(x_i,x_j)$, and $[K^Y_A]_{ij} = \langle y_i, Ay_j\rangle_\mathcal{Y}$. If Assumption 5.8 has already been discussed in Chapter 2, the next one is a bit more general.

Assumption 5.9. *There exist $T\in\mathbb{N}^*$, and for all $t\leq T$ admissible scalar kernels $k_t:\mathcal{X}\times\mathcal{X}\to\mathbb{R}$ and positive semi-definite operators $A_t\in\mathcal{L}(\mathcal{Y})$, such that:*

$$\forall (x,x')\in\mathcal{X}^2,\quad \mathcal{K}(x,x') = \sum_{t=1}^T k_t(x,x')A_t.$$

Similarly, under Assumption 5.9, K_t^X and K_t^Y, for $t \leq T$, denote the $n \times n$ matrices such that $[K_t^X]_{ij} = k_t(x_i, x_j)$, and $[K_t^Y]_{ij} = \langle y_i, A_t y_j \rangle_{\mathcal{Y}}$ for all $t \leq T$. Independently from the linear operators A_t, or A underAssumption 5.8, K^Y denotes the $n \times n$ matrix such that $[K^Y]_{ij} = \langle y_j, y_j \rangle$.

Notice that Assumption 5.9 is by no means restrictive, since every shift-invariant OVK can be approximated arbitrarily closely by kernels satisfying Assumption 5.9 (see *e.g.* Carmeli et al. (2010)). The next assumption is of a different order, and deals with $\mathcal{K}(x_i, x_j)$ invariant subspaces.

Assumption 5.10. $\forall i, j \leq n$, $\mathbf{Y}$ *is an invariant subspace of* $\mathcal{K}(x_i, x_j)$.

Notice that if for all $t \leq T$, A_t keeps Y invariant, then Assumption 5.9 directly implies Assumption 5.10. The next two assumptions define admissible losses through conditions on their Fenchel-Legendre transforms.

Assumption 5.11. $\forall i \leq n,\ \forall (\alpha^{\mathbf{Y}}, \alpha^{\perp}) \in \mathbf{Y} \times \mathbf{Y}^{\perp}$,

$$\ell_i^{\star}(\alpha^{\mathbf{Y}}) \leq \ell_i^{\star}(\alpha^{\mathbf{Y}} + \alpha^{\perp}).$$

Assumption 5.12. $\forall i \leq n, \exists L_i : \mathbb{R}^{n+n^2} \to \mathbb{R}$ *such that* $\forall\ \boldsymbol{\omega} = (\omega_j)_{j=n}^n \in \mathbb{R}^n$,

$$\ell_i^{\star}\left(-\textstyle\sum_{j=1}^n \omega_j\ y_j\right) = L_i\left(\boldsymbol{\omega}, K^Y\right).$$

Equipped with these assumptions and notation, Theorem 5.13 proves that the solutions to Problem (5.4) actually lie in $\mathbf{Y}^n$, ensuring their computability.

Theorem 5.13. *Let* $\mathcal{K}$ *be an OVK satisfying Assumption 5.10, and* $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ *be a loss function with Fenchel-Legendre transforms satisfying Assumptions 5.11 and 5.12. Then, the solution to Problem (5.3) is given by*

$$\hat{h} = \frac{1}{\Lambda n} \sum_{i,j=1}^n \mathcal{K}(\cdot, x_i)\ \hat{\omega}_{ij}\ y_j, \tag{5.5}$$

with $\hat{\Omega} = [\hat{\omega}_{ij}] \in \mathbb{R}^{n \times n}$ *the solution to the computable convex optimization problem*

$$\min_{\Omega \in \mathbb{R}^{n \times n}} \sum_{i=1}^n L_i\left(\Omega_{i:}, K^Y\right) + \frac{1}{2\Lambda n} \mathbf{Tr}\left(\tilde{M}^{\top}(\Omega \otimes \Omega)\right), \tag{5.6}$$

with M *the* $n \times n \times n \times n$ *tensor such that* $M_{ijkl} = \langle y_k, \mathcal{K}(x_i, x_j) y_l \rangle_{\mathcal{Y}}$, *and* $\tilde{M}$ *its rewriting as a* $n^2 \times n^2$ *block matrix such that its* (i, j) *block is the* $n \times n$ *matrix with elements* $\tilde{M}_{st}^{(i,j)} = \langle y_j, \mathcal{K}(x_i, x_s) y_t \rangle_{\mathcal{Y}}$. *If* $\mathcal{K}$ *further satisfies Assumption 5.9, tensor* M *simplifies to* $M_{ijkl} = \sum_{t=1}^T [K_t^X]_{ij} [K_t^Y]_{kl}$ *and the problem rewrites*

$$\min_{\Omega \in \mathbb{R}^{n \times n}} \sum_{i=1}^n L_i\left(\Omega_{i:}, K^Y\right) + \frac{1}{2\Lambda n} \sum_{t=1}^T \mathbf{Tr}\left(K_t^X \Omega K_t^Y \Omega^{\top}\right). \tag{5.7}$$

Proof. Using Theorem 5.7, one gets that $\hat{h} = \frac{1}{\Lambda n} \sum_{i=1}^n \mathcal{K}(\cdot, x_i) \hat{\alpha}_i$, with the $(\hat{\alpha}_i)_{i \leq n}$ satisfying:

$$(\hat{\alpha}_i)_{i=1}^n \in \operatorname*{argmin}_{(\alpha_i)_{i=1}^n \in \mathcal{Y}^n} \sum_{i=1}^n \ell_i^{\star}(-\alpha_i) + \frac{1}{2\Lambda n} \sum_{i,j=1}^n \left\langle \alpha_i, \mathcal{K}(x_i, x_j) \alpha_j \right\rangle_{\mathcal{Y}}.$$

However, this optimization problem cannot be solved in a straightforward manner, as $\mathcal{Y}$ is in general infinite dimensional. Nevertheless, it is possible to bypass this difficulty by noticing that the optimal $(\hat{\alpha}_i)_{i\leq n}$ actually lie in $\mathbf{Y}^n$. Indeed, by Assumptions 5.10 and 5.11, for all for all $(\alpha_i^{\mathbf{Y}})_{i\leq n}, (\alpha_i^{\perp})_{i\leq n} \in \mathbf{Y}^n \times {\mathbf{Y}^{\perp}}^n$, it holds:

$$\sum_{i=1}^{n} \ell_i^\star(-\alpha_i^{\mathbf{Y}}) + \frac{1}{2\Lambda n} \sum_{i,j=1}^{n} \left\langle \alpha_i^{\mathbf{Y}}, \mathcal{K}(x_i,x_j)\alpha_j^{\mathbf{Y}} \right\rangle_{\mathcal{Y}}$$
$$\leq \sum_{i=1}^{n} \ell_i^\star(-\alpha_i^{\mathbf{Y}} - \alpha_i^{\perp}) + \frac{1}{2\Lambda n} \sum_{i,j=1}^{n} \left\langle \alpha_i^{\mathbf{Y}} + \alpha_i^{\perp}, \mathcal{K}(x_i,x_j)(\alpha_j^{\mathbf{Y}} + \alpha_j^{\perp}) \right\rangle_{\mathcal{Y}},$$

ensuring that the optimal coefficients are valued in $\mathbf{Y}^n$. If the inequality about $\ell_i^\star$ follows directly Assumption 5.11, that about $\mathcal{K}(x_i,x_j)$ can be obtained by Assumption 5.10 through the following calculus:

$$\begin{aligned}
&\sum_{i,j=1}^{n} \left\langle \alpha_i^{\mathbf{Y}} + \alpha_i^{\perp}, \mathcal{K}(x_i,x_j)(\alpha_j^{\mathbf{Y}} + \alpha_j^{\perp}) \right\rangle_{\mathcal{Y}} \\
&= \sum_{i,j=1}^{n} \left\langle \alpha_i^{\mathbf{Y}}, \mathcal{K}(x_i,x_j)\alpha_j^{\mathbf{Y}}) \right\rangle_{\mathcal{Y}} + 2\sum_{i,j=1}^{n} \left\langle \alpha_i^{\perp}, \mathcal{K}(x_i,x_j)\alpha_j^{\mathbf{Y}} \right\rangle_{\mathcal{Y}} + \sum_{i,j=1}^{n} \left\langle \alpha_i^{\perp}, \mathcal{K}(x_i,x_j)\alpha_j^{\perp} \right\rangle_{\mathcal{Y}}, \\
&= \sum_{i,j=1}^{n} \left\langle \alpha_i^{\mathbf{Y}}, \mathcal{K}(x_i,x_j)\alpha_j^{\mathbf{Y}}) \right\rangle_{\mathcal{Y}} + \sum_{i,j=1}^{n} \left\langle \alpha_i^{\perp}, \mathcal{K}(x_i,x_j)\alpha_j^{\perp} \right\rangle_{\mathcal{Y}}, \\
&\geq \sum_{i,j=1}^{n} \left\langle \alpha_i^{\mathbf{Y}}, \mathcal{K}(x_i,x_j)\alpha_j^{\mathbf{Y}}) \right\rangle_{\mathcal{Y}},
\end{aligned}$$

where we have used successively Assumption 5.10 and the positiveness of $\mathcal{K}$. So there exists $\Omega = [\omega_{ij}]_{1\leq i,j\leq n} \in \mathbb{R}^{n\times n}$ such that for all $i \leq n$, $\hat{\alpha}_i = \sum_j \omega_{ij}\ y_j$. This proof technique is very similar in spirit to that of the Representer Theorem, and yields an analogous result, the reduction of the search space to a smaller vector space. The dual optimization problem thus rewrites:

$$\begin{aligned}
&\min_{\Omega\in\mathbb{R}^{n\times n}} \sum_{i=1}^{n} \ell_i^\star\left(-\sum_{j=1}^{n} \omega_{ij}\ y_j\right) + \frac{1}{2\Lambda n}\sum_{i,j=1}^{n} \left\langle \sum_{k=1}^{n} \omega_{ik}\ y_k, \mathcal{K}(x_i,x_j) \sum_{l=1}^{n} \omega_{jl}\ y_l \right\rangle_{\mathcal{Y}} \\
&\min_{\Omega\in\mathbb{R}^{n\times n}} \sum_{i=1}^{n} L_i\left((\omega_{ij})_{j\leq n}, K^Y\right) + \frac{1}{2\Lambda n}\sum_{i,j=1}^{n} \omega_{ik}\ \omega_{jl}\ M_{ijkl}, \\
&\min_{\Omega\in\mathbb{R}^{n\times n}} \sum_{i=1}^{n} L_i\left((\omega_{ij})_{j\leq n}, K^Y\right) + \frac{1}{2\Lambda n}\mathbf{Tr}\left(\tilde{M}^\top(\Omega\otimes\Omega)\right),
\end{aligned}$$

with M the $n\times n\times n\times n$ tensor such that $M_{ijkl} = \langle y_k, \mathcal{K}(x_i,x_j)y_l\rangle_{\mathcal{Y}}$, and $\tilde{M}$ its rewriting as a $n^2\times n^2$ block matrix such that its (i,j) block is the $n\times n$ matrix with elements

$$\tilde{M}_{st}^{(i,j)} = \left\langle y_j, \mathcal{K}(x_i,x_s)y_t \right\rangle_{\mathcal{Y}}.$$

The second term is quadratic in Ω, and consequently convex. The L_i's are basically rewritings of the Fenchel-Legendre transforms $\ell_i^\star$'s that ensure the computability of the problem (they only depend on K^Y, which is known). Regarding their convexity, we

know by definition that the $\ell_i^\star$'s are convex. Composing by a linear function preserving the convexity, we know that each L_i is convex with respect to $\Omega_{i:}$, and therefore with respect to Ω.

Thus, we have converted the infinite dimensional primal problem in $\mathcal{H}_\mathcal{K}$ into an infinite dimensional dual problem in $\mathcal{Y}^n$, which in turn is reduced to a convex optimization procedure over $\mathbb{R}^{n\times n}$, that only involves computable quantities.

If $\mathcal{K}$ satisfies Assumption 5.9, the tensor M simplifies to

$$M_{ijkl} = \Big\langle y_k, \mathcal{K}(x_i,x_j)y_l \Big\rangle_\mathcal{Y} = \sum_{t=1}^T k_t(x_i,x_j)\left\langle y_k, A_t y_l\right\rangle_\mathcal{Y} = \sum_{t=1}^T [K_t^X]_{ij}[K_t^Y]_{kl},$$

and the problem rewrites

$$\min_{\Omega\in\mathbb{R}^{n\times n}} \sum_{i=1}^n L_i\left(\Omega_{i:}, K^Y\right) + \frac{1}{2\Lambda n}\sum_{t=1}^T \mathbf{Tr}\left(K_t^X \Omega K_t^Y \Omega^\top\right).$$

□

This theorem can be seen as a Double Representer Theorem, as both theorems share analogous proofs and consequences: a search domain reduction, respectively from $\mathcal{H}_\mathcal{K}$ to $\mathcal{Y}^n$, and $\mathcal{Y}^n$ to $\mathbb{R}^{n\times n}$. The following remarks address computational implications of Theorem 5.13, as well as Assumption 5.10.

Remark 5.14. *Thanks to the* Double Representer Theorem*, the knowledge of the n^4 tensor M is thus the sole requirement to make OVK problems in infinite dimensional output spaces computable. Although it might seem prohibitive at first sight, one has to keep in mind that a n^2 cost is needed to use kernels with infinite dimensional feature maps (like the scalar Gaussian kernel), while the second n^2 cost makes it possible to handle infinite dimensional outputs. In the particular case of a decomposable kernel, one has the simplifying identity $M_{ijkl} = K^X_{ij}K^Y_{kl}$, so that only the knowledge of two n^2 matrices is required.*

Remark 5.15. *The second term of Problem (5.6) can be easily optimized. Indeed, let $\tilde{M}$ be a block matrix such that $\tilde{M}^{(i,j)}_{st} = \tilde{M}^{(s,t)}_{ij}$ for $i,j,s,t \leq n$. Notice $\tilde{M}$ used in Theorem 5.13 satisfies this assumption as a direct consequence of the OVK symmetry property. Then it holds*

$$\frac{\partial \mathbf{Tr}\left(\tilde{M}^\top(\Omega\otimes\Omega)\right)}{\partial \omega_{st}} = 2\mathbf{Tr}\left(\tilde{M}^{(s,t)\top}\Omega\right).$$

Indeed, notice that $\mathbf{Tr}(\tilde{M}^\top(\Omega\otimes\Omega)) = \sum_{i,j=1}^n \omega_{ij}\mathbf{Tr}(\tilde{M}^{(i,j)\top}\Omega)$ and use the symmetry assumption. In the particular case of a decomposable kernel, $\tilde{M}^{(i,j)} = K^X_{i:}{K^Y_{j:}}^\top$ so that

$$\begin{aligned}\frac{\partial \mathbf{Tr}\left(\tilde{M}^\top(\Omega\otimes\Omega)\right)}{\partial \omega_{st}} &= 2\mathbf{Tr}\left(\tilde{M}^{(s,t)\top}\Omega\right)\\ &= 2\sum_{i,j=1}^n \left[K^X_{s:}{K^Y_{t:}}^\top\right]_{ij}\omega_{ij} = 2\sum_{ij=1}^n K^X_{si}K^Y_{tj}\omega_{ij} = 2\left[K^X\Omega K^Y\right]_{st}.\end{aligned}$$

Remark 5.16. *Assumption 5.10 is actually a stronger assumption than that required. One only needs*

$$\sum_{i,j=1}^{n} \langle \alpha_i^{\mathbf{Y}}, \mathcal{K}(x_i, x_j)\alpha_j^{\mathbf{Y}} \rangle_{\mathcal{Y}} \leq \sum_{i,j=1}^{n} \left\langle \alpha_i^{\mathbf{Y}} + \alpha_i^{\perp}, \mathcal{K}(x_i, x_j)(\alpha_j^{\mathbf{Y}} + \alpha_j^{\perp}) \right\rangle_{\mathcal{Y}},$$

for all $(\alpha_i^{\mathbf{Y}})_{i\leq n}, (\alpha_i^{\perp})_{i\leq n} \in \mathbf{Y}^n \times \mathbf{Y}^{\perp n}$. *It is easy to verify that this inequality holds true for any OVK that satisfies Assumption 5.10.*

5.2.3 Admissible Losses

Before studying particular instances of Problem (5.7), we analyze in this section the admissible losses that satisfy Assumptions 5.11 and 5.12.

Proposition 5.17. *The following losses have Fenchel-Legendre transforms satisfying Assumptions 5.11 and 5.12:*

- $\ell_i(y) = f(\langle y, z_i \rangle)$, $z_i \in \mathbf{Y}$ *and* $f : \mathbb{R} \to \mathbb{R}$ *convex. This encompasses maximum-margin regression, obtained with* $z_i = y_i$ *and* $f(t) = \max(0, 1 - t)$.
- $\ell(y) = f(\|y\|)$, $f : \mathbb{R}_+ \to \mathbb{R}$ *convex increasing s.t.* $t \mapsto \frac{f'(t)}{t}$ *is continuous over* $\mathbb{R}_+$. *This includes all power functions* $\frac{\lambda}{\eta}\|y\|_{\mathcal{Y}}^{\eta}$ *for* $\eta > 1$ *and* $\lambda > 0$.
- $\forall \lambda > 0$, *with* $\mathcal{B}_\lambda$ *the centered ball of radius* λ,
  - $\ell(y) = \lambda\|y\|$,
  - $\ell(y) = \lambda\|y\| \log(\|y\|)$,
  - $\ell(y) = \chi_{\mathcal{B}_\lambda}(y)$,
  - $\ell(y) = \lambda(\exp(\|y\|) - 1)$.
- $\ell_i(y) = f(y - y_i)$, *with* $f^\star$ *satisfying Assumptions 5.11 and 5.12.*
- *Any infimal convolution of functions satisfying Assumptions 5.11 and 5.12. This encompasses* ϵ*-insensitive losses (Sangnier et al., 2017), the Huber loss (Huber, 1964), and more generally all Moreau envelopes (Moreau, 1962).*

Proof. The proof technique is the same for all losses: first explicit the Fenchel-Legendre transforms $\ell_i^\star$, then use simple arguments to meet Assumptions 5.11 and 5.12. For instance, any increasing function of $\|\alpha\|$ automatically satisfy the assumptions.

- Assume that ℓ is such that there is $f : \mathbb{R} \to \mathbb{R}$ convex, $\forall i \leq n, \exists z_i \in \mathbf{Y}, \ell_i(y) = f(\langle y, z_i \rangle)$. Then $\ell_i^\star : \mathcal{Y} \to \mathbb{R}$ writes $\ell_i^\star(\alpha) = \sup_{y\in\mathcal{Y}} \langle \alpha, y \rangle - f(\langle y, z_i \rangle)$. If α is not collinear to z_i, this quantity is obviously $+\infty$. Otherwise, assume that $\alpha = \lambda z_i$. The Fenchel-Legendre transform rewrites: $\ell_i^\star(\alpha) = \sup_t \lambda t - f(t) = f^\star(\lambda) = f^\star(\pm\|\alpha\|/\|z_i\|)$. Finally, $\ell_i^\star(\alpha) = \chi_{\mathrm{span}\{z_i\}}(\alpha) + f^\star(\pm\|\alpha\|/\|z_i\|)$. If $\alpha \notin \mathbf{Y}$, then *a fortiori* $\alpha \notin \mathrm{span}\{z_i\}$, so $\ell_i^\star(\alpha^{\mathbf{Y}} + \alpha^{\perp}) = +\infty \geq \ell_i^\star(\alpha^{\mathbf{Y}})$ for all $(\alpha^{\mathbf{Y}}, \alpha^{\perp}) \in \mathbf{Y} \times \mathbf{Y}^{\perp}$. For all $i \leq n$, $\ell_i^\star$ satisfy Assumption 5.11. As for Assumption 5.12, if $\alpha = \sum_{i=1}^n c_i y_i$, then $\chi_{\mathrm{span}\{z_i\}}(\alpha)$ only depends on the $(c_i)_{i\leq n}$ Indeed, assume that $z_i \in \mathbf{Y}$ writes $\sum_j b_j y_j$. Then $\chi_{\mathrm{span}\{z_i\}}(\alpha)$ is equal to 0 if there exists $\lambda \in \mathbb{R}$ such that $c_j = \lambda b_j$ for all $j \leq n$, and to $+\infty$ otherwise. The second term of $\ell_i^\star$ depending only on $\|\alpha\|$, it satisfies Assumption 5.12. This concludes the proof.

- Assume that ℓ is such that there is $f : \mathbb{R}_+ \to \mathbb{R}$ convex increasing, with $f'(t)/t$ continuous over $\mathbb{R}_+$, $\ell(y) = f(\|y\|)$. Although this loss may seem useless at the first sight since ℓ does not depend on y_i, it should not be forgotten that the composition with $y \mapsto y - y_i$ does not affect the validation of Assumptions 5.11 and 5.12 (see below). One has: $\ell^\star(\alpha) = \sup_{y\in\mathcal{Y}} \langle \alpha, y\rangle - f(\|y\|)$. Differentiating with respect to y, one gets: $\alpha = f'(\|y\|)/\|y\| \cdot y$, which is always well define as $t \mapsto f'(t)/t$ is continuous over $\mathbb{R}_+$. Reverting the equality, it holds: $y = f'^{-1}(\|\alpha\|)/\|\alpha\| \cdot \alpha$, and $\ell^\star(\alpha) = \|\alpha\| f'^{-1}(\|\alpha\|) - f \circ f'^{-1}(\|\alpha\|)$. This expression depending only on $\|\alpha\|$, Assumption 5.12 is automatically satisfied. Let us now investigate the monotonicity of $\ell^\star$ w.r.t. $\|\alpha\|$. Let $g : \mathbb{R}_+ \to \mathbb{R}$ such that $g(t) = tf'^{-1}(t) - f \circ f'^{-1}(t)$. Then $g'(t) = f'^{-1}(t) \geq 0$. Indeed, as $f' : \mathbb{R}_+ \to \mathbb{R}_+$ is always positive due to the monotonicity of f, so is f'^{-1}. This final remark guarantees that $\ell^\star$ is increasing with $\|\alpha\|$. It is then direct that $\ell^\star$ fulfills Assumption 5.11.

- Assume that $\ell(y) = \lambda\|y\|$. It is well known that $\ell^\star(\alpha) = \chi_{\mathcal{B}_\lambda}(\alpha)$. The latter being an increasing function of $\|\alpha\|$, it directly fulfills Assumptions 5.11 and 5.12.

- Assume that $\ell(y) = \chi_{\mathcal{B}_\lambda}(y)$. It is well known that $\ell^\star(\alpha) = \lambda\|\alpha\|$. The usual arguments on the monotonicity of $\ell^\star$ w.r.t. $\|\alpha\|$ permit to conclude.

- Assume that $\ell(y) = \lambda\|y\| \log(\|y\|)$. It can be shown that $\ell^\star(\alpha) = \lambda e^{\frac{\|\alpha\|}{\lambda} - 1}$. The same arguments as above apply.

- Assume that $\ell(y) = \lambda(\exp(\|y\|) - 1)$. It can be shown that

$$\ell^\star(\alpha) = \mathbb{1}\{\|\alpha\| \geq \lambda\} \cdot \Big(\|\alpha\| \log(\|\alpha\|/(\lambda e)) + \lambda \Big) .$$

  Again, the Fenchel-Legendre transform is an increasing function of $\|\alpha\|$: it satisfies Assumptions 5.11 and 5.12.

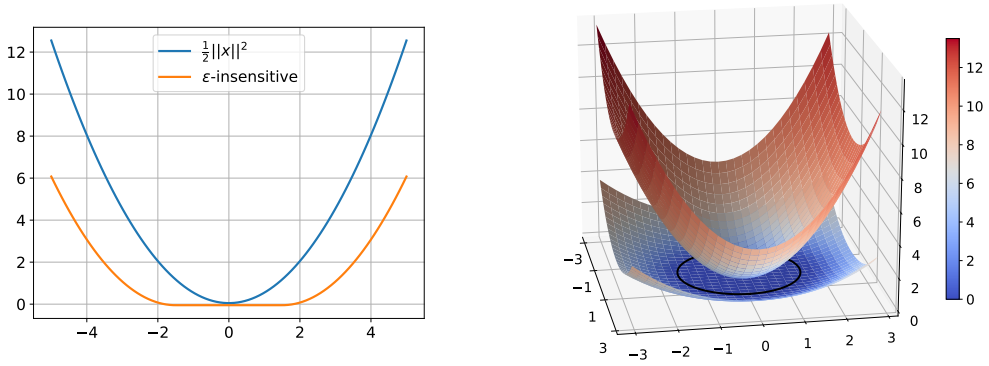
- Assume that $\ell_i(y) = f(y - y_i)$, with f such that $f^\star$ fulfills Assumptions 5.11 and 5.12. Then $\ell_i^\star(\alpha) = \sup_{y\in\mathcal{Y}} \langle \alpha, y\rangle - f(y - y_i) = f^\star(\alpha) + \langle \alpha, y_i\rangle$. If $f^\star$ satisfies Assumptions 5.11 and 5.12, then so does $\ell_i^\star$. This remark is very important, as it gives more sense to loss function based on $\|y\|$ only, since they can be applied to $y - y_i$ now.

- Assume that there exists f, g satisfying Assumptions 5.11 and 5.12 such that $\ell_i(y) = (f \,\square\, g)(y)$. By Proposition 5.5 it holds $(f \,\square\, g)^\star = f^\star + g^\star$, so that if both f and g satisfy Assumptions 5.11 and 5.12, so does $f \,\square\, g$. This last example allows to deal with ϵ-insensitive losses for instance (convolution of a loss and $\chi_{\mathcal{B}_\epsilon}$), the Huber loss (convolution of $\|.\|$ and $\|.\|^2$), or more generally all Moreau envelopes (convolution of a loss and $\frac{1}{2}\|.\|^2$).

□

One can notice that most losses that depend exclusively on norms and dot products satisfy Assumptions 5.11 and 5.12. For losses that leverage the functional nature of elements of $\mathcal{Y}$, specific tools must be used, that are detailed in Section 5.4. For now, we focus on particular instances of Problem (5.7), namely when ℓ is the ϵ-insensitive squared norm, or the Huber loss.

Figure 5.1 – Standard and ϵ-version of the squared norm in 1 and 2 dimensions, $\epsilon = 1.5$.



5.3 Specific Instances of Dual Problems

In this section, we completely derive the dual problems for two interesting losses, the ϵ-insensitive squared norm and the Huber loss.

5.3.1 The ϵ-Insensitive Ridge Regression

As a first go, we recall the important notion of ϵ-insensitive losses. Following in the footsteps of Sangnier et al. (2017), we extend them in a natural way from $\mathbb{R}^p$ to any Hilbert space $\mathcal{Y}$. In order to avoid overwhelming notation, ℓ denotes here the loss taken with respect to one argument only (*i.e.* previously ℓ_i).

Definition 5.18. *Let $\ell : \mathcal{Y} \to \mathbb{R}$ be a convex loss such that $\ell(0) = 0$, and $\epsilon > 0$. The ϵ-insensitive version of ℓ, denoted ℓ_ϵ, is defined by $\ell_\epsilon(y) = (\ell \square \chi_{\mathcal{B}_\epsilon})(y)$, or again:*

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

In other terms, $\ell_\epsilon(y)$ is the smallest value $\ell(z)$ attained by a point z within the ϵ-ball centered at y. Figure 5.1 gives an illustration of ϵ-insensitive losses in one and two dimensions. Interestingly, and as detailed by Theorem 5.19, Problem (5.7) for the ϵ-insensitive squared norm and an identity decomposable kernel admits a very nice writing, allowing for an efficient resolution.

Theorem 5.19. *For an identity decomposable OVK $\mathcal{K}$, the solution to the ϵ-Ridge regression problem*

$$\min_{h \in \mathcal{H}_\mathcal{K}} \quad \frac{1}{2n} \sum_{i=1}^{n} \max\left(\|h(x_i) - y_i\|_{\mathcal{Y}} - \epsilon, 0 \right)^2 + \frac{\Lambda}{2} \|h\|^2_{\mathcal{H}_\mathcal{K}}, \tag{5.8}$$

is given by Equation (5.5), with $\hat{\Omega} = \hat{W} V^{-1}$, and $\hat{W}$ the solution to the Multi-Task Lasso problem

$$\min_{W \in \mathbb{R}^{n \times n}} \quad \frac{1}{2} \left\| AW - B \right\|^2_{\text{Fro}} + \epsilon \, \|W\|_{2,1}, \tag{5.9}$$

with V, A, B such that $K^Y = VV^\top$, $\frac{K^X}{\Lambda n} + \mathbf{I}_n = A^\top A$, and $V = A^\top B$.

Proof. Applying Theorem 5.7 together with the Fenchel-Legendre transforms detailed in the proof of Proposition 5.17, a dual to Problem (5.8) is:

$$\min_{(\alpha_i)_{i=1}^n \in \mathcal{Y}^n} \quad \frac{1}{2}\sum_{i=1}^n \|\alpha_i\|_{\mathcal{Y}}^2 - \sum_{i=1}^n \langle \alpha_i, y_i \rangle_{\mathcal{Y}} + \epsilon \sum_{i=1}^n \|\alpha_i\|_{\mathcal{Y}} + \frac{1}{2\Lambda n} \sum_{i,j=1}^n \left\langle \alpha_i, \mathcal{K}(x_i, x_j)\alpha_j \right\rangle_{\mathcal{Y}},$$

$$\min_{(\alpha_i)_{i=1}^n \in \mathcal{Y}^n} \quad \frac{1}{2}\sum_{i,j=1}^n \left\langle \alpha_i, \left(\delta_{ij}\mathbf{I}_{\mathcal{Y}} + \frac{1}{\Lambda n}\mathcal{K}(x_i, x_j)\right)\alpha_j \right\rangle_{\mathcal{Y}} - \sum_{i=1}^n \langle \alpha_i, y_i \rangle_{\mathcal{Y}} + \epsilon \sum_{i=1}^n \|\alpha_i\|_{\mathcal{Y}}.$$

By virtue of Theorem 5.13, we known that the optimal $(\alpha_i)_{i=1}^n \in \mathcal{Y}^n$ are in $\mathbf{Y}^n$. After the reparametrization $\alpha_i = \sum_j \omega_{ij}\, y_j$, the problem rewrites:

$$\min_{\Omega \in \mathbb{R}^{n\times n}} \quad \frac{1}{2}\mathbf{Tr}\left(\tilde{K}\Omega K^Y \Omega^\top\right) - \mathbf{Tr}\left(K^Y\Omega\right) + \epsilon \sum_{i=1}^n \sqrt{\left[\Omega K^Y \Omega^\top\right]_{ii}}, \tag{5.10}$$

with Ω, $\tilde{K}$, the $n \times n$ matrices such that $[\Omega]_{ij} = \omega_{i,j}$, and $\tilde{K} = \frac{1}{\Lambda n}K^X + \mathbf{I}_n$.

Now, let $K^Y = U\Sigma U^\top = (U\Sigma^{1/2})(U\Sigma^{1/2})^\top = VV^\top$ be the SVD of K^Y, and let $W = \Omega V$. Notice that K^Y is positive semi-definite, and can be made positive definite if necessary, so that V is full rank, and optimizing with respect to W is strictly equivalent to minimizing with respect to Ω. With this change of variable, Problem (5.10) rewrites:

$$\min_{W \in \mathbb{R}^{n\times n}} \quad \frac{1}{2}\mathbf{Tr}\left(\tilde{K}WW^\top\right) - \mathbf{Tr}\left(V^\top W\right) + \epsilon\|W\|_{2,1}, \tag{5.11}$$

with $\|W\|_{2,1} = \sum_i \|W_{i:}\|_2$ the row-wise $\ell_{2,1}$ mixed norm of matrix W, $\tilde{K} = A^\top A$ the SVD of $\tilde{K}$, and B such that $A^\top B = V$. One can then add the constant term $(1/2)\mathbf{Tr}(A^{\top -1}VV^\top A^{-1}) = (1/2)\mathbf{Tr}(BB^\top)$ to the objective function without changing Problem (5.11). One finally gets the Multi-Task Lasso problem:

$$\min_{W \in \mathbb{R}^{n\times n}} \quad \frac{1}{2}\|AW - B\|_{\text{Fro}}^2 + \epsilon\|W\|_{2,1}.$$

□

The Multi-Task Lasso is a very well known problem (Obozinski et al., 2010), that can be solved by *e.g.* Block Coordinate Descent (BCD, Tseng (2001); Tseng and Yun (2009)). This procedure is recalled in Algorithm 5.1, with BST the Block Soft Thresholding operator such that $\text{BST}(x, \tau) = (1 - \tau/\|x\|)_+ \cdot x$, and the objective decrease as stopping criterion for instance. Notice that the Singular Value Decomposition (SVD) of $\tilde{K}$ is not necessary, as only $A^\top A = \tilde{K}$ and $A^\top B = V$ are involved in the computations. Finally, the change of variable $W = \Omega V$ is always licit, since V may be assumed invertible.

If $\mathcal{K}$ is not identity decomposable, but only satisfies Assumption 5.9, Problem (5.7) cannot be simplified as Problem (5.9). Nonetheless, it admits a simple resolution, as detailed in the following lines. After the Ω reparametrization, the problem writes

$$\min_{\Omega \in \mathbb{R}^{n\times n}} \quad \frac{1}{2}\mathbf{Tr}(\Omega K^Y \Omega^\top) - \mathbf{Tr}\left(K^Y\Omega\right) + \epsilon \sum_{i=1}^n \sqrt{\left[\Omega K^Y \Omega^\top\right]_{ii}} + \frac{1}{2\Lambda n}\sum_{t=1}^T \mathbf{Tr}(K_t^X \Omega K_t^Y \Omega^\top),$$

Algorithm 5.1 Block Coordinate Descent (BCD)

input : Gram matrices K^X, K^Y, parameters Λ, ϵ

init : $\widetilde{K} = \frac{1}{\Lambda n} K^X + \mathbf{I}_n$, $K^Y = VV^\top$, $W = \mathbf{0}_{\mathbb{R}^{n \times n}}$

while *stopping criterion* False **do**

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

    $W_{i:} = \mathrm{BST}\left(W_{i:} + \frac{1}{\widetilde{K}_{ii}}\left[V_{i:} - \widetilde{K}_{i:}W\right], \frac{\epsilon}{\widetilde{K}_{ii}}\right)$

return W

$$\min_{W \in \mathbb{R}^{n \times n}} \quad \frac{1}{2}\mathbf{Tr}(WW^\top) + \frac{1}{2\Lambda n}\sum_{t=1}^{T} \mathbf{Tr}(K_t^X W \tilde{K}_t^Y W^\top) - \mathbf{Tr}\left(V^\top W\right) + \epsilon \|W\|_{2,1},$$

with $K^Y = VV^\top$, $W = \Omega V$, $\tilde{K}_t^Y = V^{-1}K_t^Y(V^\top)^{-1}$. Due to the different quadratic terms, this problem cannot be summed up as a Multi-Task Lasso like Problem (5.9). However, it may still be solved, *e.g.* by proximal gradient descent. Indeed, the gradient of the smooth term (*i.e.* all but the $\ell_{2,1}$ mixed norm) reads

$$W + \frac{1}{\Lambda n}\sum_{t=1}^{T} K_t^X W \tilde{K}_t^Y - V, \tag{5.12}$$

while the proximal operator of the $\ell_{2,1}$ mixed norm is

$$\mathrm{prox}_{\epsilon \| \cdot \|_{2,1}}(W) = \begin{pmatrix} | \\ \mathrm{prox}_{\epsilon \| \cdot \|_2}(W_{i:}) \\ | \end{pmatrix} = \begin{pmatrix} | \\ \left(1 - \frac{\epsilon}{\|W_{i:}\|_2}\right)_+ W_{i:} \\ | \end{pmatrix} = \begin{pmatrix} | \\ \mathrm{BST}(W_{i:}, \epsilon) \\ | \end{pmatrix}.$$

Hence, even in the more involved case of an OVK satisfying Assumption 5.9, efficient algorithms to compute the solutions to the dual problem exist. The next remark focuses on the fact that the standard Kernel Ridge Regression is recovered when ϵ is set to 0.

Remark 5.20. *Notice that the solution to Problem (5.9) for $\epsilon = 0$ is exactly that of the standard Ridge Regression. Indeed, coming back to Problem (5.10) and differentiating with respect to Ω, one gets:*

$$\tilde{K}\hat{\Omega}K^Y - K^Y = 0 \iff \hat{\Omega} = \tilde{K}^{-1},$$

which is exactly the standard Kernel Ridge Regression solution (Brouard et al., 2016b).

5.3.2 The Huber Loss Regression

Another framework that nicely falls into our generic resolution methodology is the Huber loss regression scheme (Huber, 1964). Tailored to induce robustness, it is based on the following loss function.

Definition 5.21. *The κ-Huber loss is defined as $\ell_{H,\kappa}(y) = \left(\kappa\|\cdot\|_{\mathcal{Y}} \square \frac{1}{2}\|\cdot\|_{\mathcal{Y}}^2\right)(y)$, or again:*

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

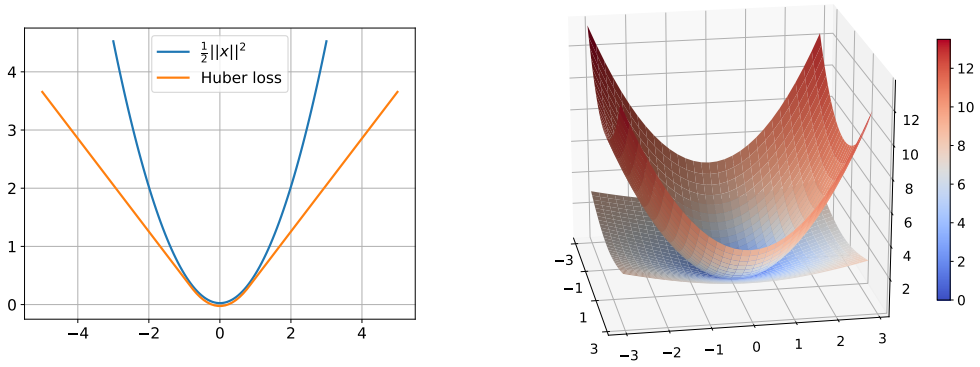


Figure 5.2 – Standard squared norm and Huber loss in 1 and 2 dimensions, $\kappa = 0.8$.

Due to its asymptotic behavior as $\|\cdot\|_{\mathcal{Y}}$, the Huber loss is particularly useful when the training data is heavy tailed or contains outliers. Examples of the Huber loss in one and two dimensions are depicted in Figure 5.2. The following theorem explicits the dual problem for the Huber loss and identity decomposable kernels.

Theorem 5.22. *For an identity decomposable OVK $\mathcal{K}$, the solution to the Huber loss regression problem*

$$\min_{h\in\mathcal{H}_{\mathcal{K}}} \frac{1}{n}\sum_{i=1}^{n} \ell_{H,\kappa}(h(x_i)-y_i) + \frac{\Lambda}{2}\|h\|^2_{\mathcal{H}_{\mathcal{K}}},$$

is given by Equation (5.5), with $\hat{\Omega} = \hat{W}V^{-1}$, and $\hat{W}$ the solution to the constrained least squares problem

$$\begin{aligned} \min_{W\in\mathbb{R}^{n\times n}} \quad & \frac{1}{2}\left\|AW-B\right\|^2_{\mathrm{Fro}}, \\ s.t. \quad & \|W\|_{2,\infty} \leq \kappa, \end{aligned} \tag{5.13}$$

with V, A, and B as in Theorem 5.19.

Proof. Basic manipulations give the Fenchel-Legendre transform of the Huber loss:

$$\begin{aligned} \left(y\mapsto \ell_{H,\kappa}(y-y_i)\right)^{\star}(\alpha) &= \left(\kappa\|\cdot\|_{\mathcal{Y}} \,\square\, \frac{1}{2}\|\cdot\|^2_{\mathcal{Y}}\right)^{\star}(\alpha) + \langle \alpha, y_i\rangle_{\mathcal{Y}}, \\ &= \left(\kappa\|\cdot\|_{\mathcal{Y}}\right)^{\star}(\alpha) + \left(\frac{1}{2}\|\cdot\|^2_{\mathcal{Y}}\right)^{\star}(\alpha) + \langle \alpha, y_i\rangle_{\mathcal{Y}}, \\ &= \chi_{\mathcal{B}_\kappa}(\alpha) + \frac{1}{2}\|\alpha\|^2_{\mathcal{Y}} + \langle \alpha, y_i\rangle_{\mathcal{Y}}. \end{aligned}$$

The rest of the proof is similar to that of Theorem 5.19, except that the $\ell_{2,1}$ mixed norm $\|W\|_{2,1}$ is replaced by constraints on the norms of W's lines. □

Again, the complex dual problem in $\mathcal{Y}^n$ boils down to a well known tractable one in $\mathbb{R}^{n\times n}$. Problem (5.13) can be solved by Projected Gradient Descent (PGD) for instance. See Algorithm 5.2, with γ a predefined stepsize, and Proj the Projection operator such that $\mathrm{Proj}(x,\tau) = \min(\tau/\|x\|, 1)\cdot x$. Analogously to Theorem 5.19, Problem (5.13) for a

Algorithm 5.2 Projected Gradient Descent (PGD)

input : Gram matrices K^X, K^Y, parameters Λ, κ

init : $\widetilde{K} = \frac{1}{\Lambda n} K^X + \mathbf{I}_n$, $K^Y = VV^\top$, $W = \mathbf{0}_{\mathbb{R}^{n\times n}}$

while *stopping criterion* False **do**

  $W = W - \gamma(\widetilde{K}W - V)$ // gradient step

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

    $W_{i:} = \mathrm{Proj}\left(W_{i:}, \kappa\right)$ // projection step

return W

kernel fulfilling Assumption 5.9 is more complex to write, but not to solve. It rewrites

$$\begin{aligned} \min_{W\in\mathbb{R}^{n\times n}} \quad & \frac{1}{2}\mathbf{Tr}(WW^T) + \frac{1}{2\Lambda n}\sum_{t=1}^{T}\mathbf{Tr}(K_t^X W \tilde{K}_t^Y W^\top) - \mathbf{Tr}\left(V^\top W\right), \\ \text{s.t.} \quad & \|W_{i:}\|_2 \leq \kappa \quad \forall i \leq n, \end{aligned}$$

The gradient term is again given by Equation (5.12), while the projection remains unchanged. The only change thus occurs in the gradient step of Algorithm 5.2, with a replacement by the above formula.

The two regression frameworks detailed in Sections 5.3.1 and 5.3.2 have interesting applications in structured prediction and structured representation learning, that are to be detailed now.

5.3.3 Applications

The ability to predict in infinite dimensional Hilbert spaces unlocks many applications, such as structured prediction and structured representation learning. In this section, we give a formal description of these tasks, and highlight the benefit of using the losses we have defined earlier.

Structured Prediction. Assume one is interested in learning a predictive decision rule f from a set $\mathcal{X}$ to a complex structured space $\mathcal{Z}$. To bypass the absence of norm on $\mathcal{Z}$, one may design a (scalar) kernel k on $\mathcal{Z}$, whose canonical feature map $\phi : z \mapsto k(\cdot, z)$ transforms any element of $\mathcal{Z}$ into an element of the (scalar) RKHS associated to k, denoted $\mathcal{Y}$ $(= \mathcal{H}_k)$. One may then use the vv-RKHS theory to learn a predictive function h from $\mathcal{X}$ to $\mathcal{Y}$, as in the previous sections:

$$\hat{h} = \operatorname*{argmin}_{h\in\mathcal{H}_\mathcal{K}} \frac{1}{n}\sum_{i=1}^{n}\ell\left(h(x_i), \phi(z_i)\right) + \frac{\Lambda}{2}\|h\|^2_{\mathcal{H}_\mathcal{K}}. \tag{5.14}$$

Once the function $\hat{h}$ is learned, the final predictions in $\mathcal{Z}$ are obtained by solving the inverse problem:

$$f(x) = \operatorname*{argmin}_{z\in\mathcal{Z}} \|\phi(z) - \hat{h}(x)\|_\mathcal{Y}.$$

The whole procedure is depicted in Figure 5.3. While previous works are restricted to identity decomposable kernels with the standard Ridge regression (Brouard et al., 2016b), the duality framework we have developed allows for many more losses and kernels. The use of an ϵ-insensitive loss in Problem (5.14), in particular, seems all the more adequate as the criterion is not the final task targeted, but rather a surrogate

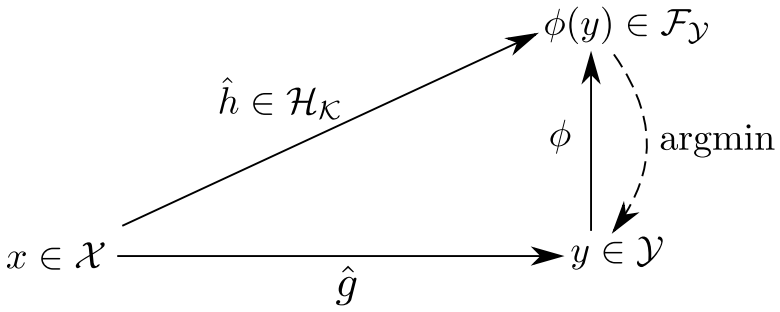


Figure 5.3 – Output Kernel Regression

one. Indeed, inducing small mistakes, that does not harm the inverse problem, while improving generalization, sounds as a suitable compromise. The Huber loss, that does not penalize heavily big errors, benefits from the same type of arguments. Experimental results endorsing the soundness of these new losses are presented in Section 5.5.

Structured Representation Learning. Extracting vectorial representations from structured inputs is another task that can be tackled through vv-RKHSs. If standard neural net functions are not able to produce reconstructions in the input space, because the latter is to complex for instance, it is still possible to embed the datapoints into a Hilbert space. Then, composing functions in vv-RKHSs results in a Kernel Autoencoder (KAE, Figure 5.4) that outputs finite codes by minimizing the (regularized) discrepancy:

$$\frac{1}{2n}\sum_{i=1}^{n} \|\phi(x_i) - f_2 \circ f_1(\phi(x_i))\|_\mathcal{Y}^2 + \Lambda \operatorname{Reg}(f_1, f_2). \tag{5.15}$$

Again, this criterion is not the real goal, but rather a proxy to make the internal representation meaningful. Therefore, all incentives to use ϵ-insensitive losses or the Huber loss still apply. The induced ϵ and Huber KAEs, obtained by changing the loss in Problem (5.15), are optimized following Algorithm 5.3. The layers are fully characterized by coefficients Φ_1 and Φ_2 (Theorem 4.1, and Sections 4.3 and 4.4). Coefficients Φ_1 are finite dimensional, and can be updated by Gradient Descent. Coefficients Φ_2 are infinite dimensional, but reparametrized into W_2, which is updated through the BCD or PGD of Algorithms 5.1 and 5.2. Empirical benefits of these losses for KAE are highlighted by Section 5.5's experiments. Notice that the alternate approach can be replaced by a full Gradient Descent strategy, as all parameters (Φ_1, W_2) are now finite dimensional.

Algorithm 5.3 ϵ-Insensitive and Huber KAEs

input : Gram matrix K^X, Λ, ϵ or κ
init : $K^X = VV^\top$, $\Phi_1 = \Phi_1^{\text{init}}$, $W_2 = \mathbf{0}_{\mathbb{R}^{n\times n}}$
while *stopping criterion* False **do**
  // Φ_1 update at fixed W_2
  $\Phi_1 = \Phi_1 - \gamma \nabla_{\Phi_1} \Big(\text{Reconstruction} \mid W_2\Big)$
  Compute $K_2(\Phi_1)$
  // W_2 update at fixed Φ_1
  $W_2 = \text{BCD}(K_2, K^X, \Lambda, \epsilon)$ // if ϵ-insensitive
  $W_2 = \text{PGD}(K_2, K^X, \Lambda, \kappa)$ // if Huber
return Φ_1, W_2

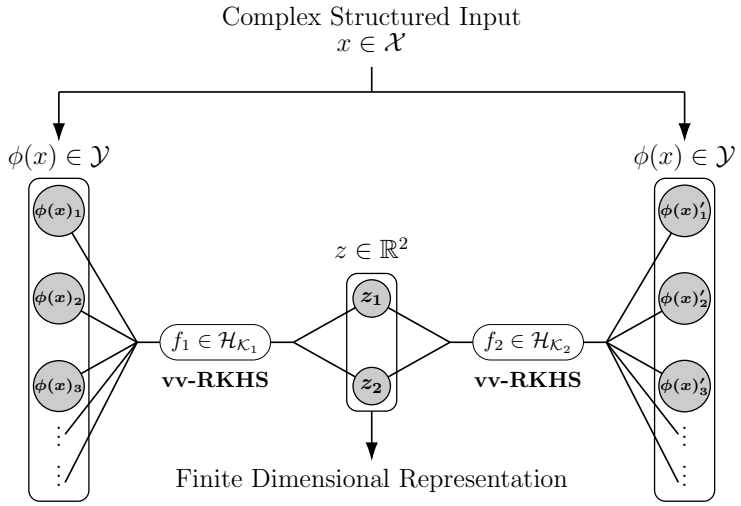


Figure 5.4 – 2-Layer Kernel Autoencoder

5.4 Handling Integral Losses

When the loss ℓ does not depend directly upon the scalar product or the norm in $\mathcal{Y}$, the assumptions developed in Section 5.2 to prove Theorem 5.13 are hard to verify. The dual problem is then seemingly intractable, as no decomposition of the $(\alpha_i)_{i=1}^n$ on a finite basis can be exhibited. Moreover, the Fenchel-Legendre transforms $\ell_i^\star$ may not be computable due to a lack of compatibility between ℓ_i and the scalar product in $\mathcal{Y}$.

Integral losses over function spaces stand as good examples of such a case. These losses, depicted in Equation (5.16), are key to solve function-to-function regression tasks (see Ramsay and Silverman (2007) for an extensive description of challenges involving functional data analysis), as well as continuums of tasks (Brault et al., 2019). Such losses take the form

$$\begin{aligned} l\colon L^2[\Theta,\mu] \times L^2[\Theta,\mu] &\to \mathbb{R} \\ (f,g) &\mapsto \int_\Theta l_\theta(f(\theta),g(\theta))\mathrm{d}\mu(\theta). \end{aligned} \tag{5.16}$$

where μ is a probability measure over some compact set $\Theta \subset \mathbb{R}$, and $l_\theta\colon \mathbb{R}\times\mathbb{R}\to\mathbb{R}$ is a family of loss functions indexed by $\theta \in \Theta$, such that l is well defined.

In our setting, $\mathcal{Y}$ is a space of functions defined over Θ which can be continuously embedded into $L^2[\Theta,\mu]$ by means of an inclusion operator I. For all $g \in L^2[\Theta,\mu]$, l_g relates to $l(\cdot,g)$ and $\ell_g = l_g \circ I$ is the loss function at point g defined on $\mathcal{Y}$. Given that $(x_i,y_i)_{i=1}^n \in \mathcal{X}\times L^2[\Theta,\mu]$ are i.i.d. samples, the problem within the empirical risk minimization framework reads

$$\min_{h\in\mathcal{H}_\mathcal{K}} \quad \frac{1}{n}\sum_{i=1}^n \ell_{y_i}(h(x_i)) + \frac{\Lambda}{2}\|h\|^2_{\mathcal{H}_\mathcal{K}}. \tag{5.17}$$

Note that the $(y_i)_{i=1}^n$ are functions which do not necessarily belong to $\mathcal{Y}$, since $\mathcal{Y}$ is the output space of the candidate functions within the vv-RKHS $\mathcal{H}_\mathcal{K}$. Below, we give a few examples of family of losses $(l_\theta)_{\theta\in\Theta}$, and emphasize on the usefulness of the associated Problem (5.17). Again, similar descriptions can be found in Section 2.2.2, but they are recalled here for self-containedness and clarity. For each example, $\Theta = [0,1]$.

- $l_\theta(s,t) = \frac{1}{2}(s-t)^2$. This setting corresponds to the Ridge regression in the function-valued regression framework. Especially, it coincides with Kadri et al. (2016) when the $(x_i)_{i=1}^n$ are functions.
- $l_\theta(s,t) = \max(\theta(t-s),(\theta-1)(t-s))$. This loss function, called *the pinball loss* (Koenker, 2005), is used at fixed θ to perform conditional quantile regression of random variables $X, Y \in \mathbb{R}^d \times \mathbb{R}$ from i.i.d. samples $(x_i, y_i)_{i=1}^n$. The minimization of its integral counterpart yields an estimate of the conditional quantile function when applied to $(x_i, y_i)_{i=1}^n$, the $(y_i)_{i=1}^n$ being considered as constant functions in $L^2[\Theta, \mu]$.
- $l_\theta(s,t) = |\theta - \mathbb{1}_{\{-1\}}(t)| \max(0, 1-ts)$. Given $\theta \in [0,1]$, this loss function is used in cost-sensitive classification (Zadrozny and Elkan, 2001). The coefficient $|\theta - \mathbb{1}_{\{-1\}}(t)|$ is asymmetric with respect to the two classes $t \in \{-1, 1\}$. It models a different impact for mistakes committed on one class or another. Minimizing the integral loss lifts the need to choose the asymmetric coefficient, rarely known in practice, and allows a practitioner to evaluate the effect of this asymmetry posterior to the learning phase, since the algorithm outputs a maximum-margin classifier as a function of θ.

Dualization of Problem (5.17) is performed exactly the same way than in Theorem 5.7 and leads to Problem (5.4). The choice of $\mathcal{H}_\mathcal{K}$ has to be driven by the feasibility of solving the dual problem in $\mathcal{Y}^n$, as well as being large enough to model the target function. OVKs of the form $\mathcal{K}(x,z) = k_\mathcal{X}(x,z)\mathbf{I}_{\mathcal{H}_k}$, where $k_\mathcal{X}$ and k are scalar kernels respectively defined on $\mathcal{X} \times \mathcal{X}$ and $\Theta \times \Theta$ are legitimate candidates for this (see Remark 5.23).

Remark 5.23. *The framework of vv-RKHS enjoys a nice interpretation when the kernel is separable. Let $k : \Theta \times \Theta \to \mathbb{R}$ be a (scalar) kernel, and $\mathcal{H}_k$ its associated RKHS. Then, choosing $\mathcal{Y} = \mathcal{H}_k$, the vv-RKHS associated to the identity decomposable OVK $\mathcal{K} = k_\mathcal{X}\mathbf{I}_{\mathcal{H}_k}$ is isomorphic to the tensor product $\mathcal{H}_{k_\mathcal{X}} \otimes \mathcal{H}_k$, so that functions in $\mathcal{H}_\mathcal{K}$ may be seen as functions of two variables (x, θ) in the (scalar) RKHS associated to the kernel $k_\mathcal{X} \cdot k$ (Carmeli et al., 2010).*

Before providing an expression of the Fenchel-Legendre transform of integral losses, we recall few properties of RKHSs (see Steinwart and Christmann (2008)) that are useful to solve Problem (5.4). Let Θ be a compact subset of $\mathbb{R}$ and $k\colon \Theta \times \Theta \to \mathbb{R}$ be a positive definite kernel, associated to the RKHS $\mathcal{H}_k$.

Assumption 5.24. *The kernel k is continuous.*

Proposition 5.25. *Grant Assumption 5.24. Then $\mathcal{H}_k$ is a subspace of $L^2[\Theta, \mu]$ and the canonical inclusion $I_k\colon \mathcal{H}_k \to L^2[\Theta, \mu]$ is a bounded operator whose adjoint denoted $T_k\colon L^2[\Theta, \mu] \to \mathcal{H}_k$ is given for all $g \in L^2[\Theta, \mu]$ by $T_k g = \int_\Theta k(\cdot, \theta) g(\theta) \mathrm{d}\mu(\theta)$.*

In particular, Proposition 5.25 ensures that for all $(\alpha, g) \in \mathcal{H}_k \times L^2[\Theta, \mu]$, it holds $\langle \alpha, T_k g \rangle_{\mathcal{H}_k} = \langle \alpha, g \rangle_{L^2[\Theta,\mu]}$. Continuity of k also grants a spectral decomposition for its integral operator, as stated in Proposition 5.26.

Proposition 5.26. *Assume that Assumption 5.24 holds. Denote by $L_k = I_k T_k$. There exist an orthonormal basis $(\psi_m)_{m=1}^\infty$ of $L^2[\Theta, \mu]$, and some $(\lambda_m)_{m=1}^\infty \in \mathbb{R}_+$ ordered in a non-increasing fashion and converging to zero such that $L_k = \sum_{m=1}^\infty \lambda_m \psi_m \otimes \psi_m$.*

Remark 5.27. *Even though each ψ_m is defined up to a null μ-set, it is convenient to work with some $\tilde{\psi}_m$ belonging to both $\mathcal{H}_k$ and the equivalence class in $L^2[\Theta, \mu]$ of ψ_m, which is assumed afterwards.*

Assumption 5.28. *Measure μ is non-degenerate:* $\mathrm{Supp}(\mu) = \Theta$.

Assumption 5.29. *The kernel k is universal, i.e. $\mathcal{H}_k$ is dense in the set of continuous functions from Θ to $\mathbb{R}$.*

Proposition 5.30. *Grant Assumptions 5.24 and 5.28. Then, operator T_k is surjective. If furthermore Assumption 5.29 is satisfied, then T_k is bijective.*

Lemma 5.31 below uses the aforementioned assumptions to link the Fenchel-Legendre transforms of ℓ_y and l_y.

Lemma 5.31. *Let $l\colon L^2(\Theta, \mu) \times L^2(\Theta, \mu) \to \mathbb{R}$ be a continuous loss function. Under Assumptions 5.24, 5.28 and 5.29, it holds*

$$\forall y \in L^2(\Theta, \mu), \quad \ell_y^\star = l_y^\star \circ T_k^{-1}. \tag{5.18}$$

Proof. Define $l_y\colon g \in L^2[\Theta, \mu] \mapsto \int_\theta \ell_\theta(g(\theta), y(\theta))\mathrm{d}\mu(\theta)$ so that $\ell_y = l_y \circ I_k$. Let $\alpha \in \mathcal{H}_k$. Using the bijectivity of T_k, one gets:

$$\begin{aligned}
(l_y \circ I_k)^\star(\alpha) &= \sup_{\xi \in \mathcal{H}_k} \ \langle \alpha, \xi \rangle_{\mathcal{H}_k} - \ l_y \circ I_k(\xi) \\
&= \sup_{\xi \in \mathcal{H}_k} \ \langle T_k (T_k)^{-1}(\alpha), \xi \rangle_{\mathcal{H}_\|} - \ l_y \circ I_k(\xi) \\
&= \sup_{\xi \in \mathcal{H}_k} \ \langle (T_k)^{-1}(\alpha), I_k(\xi) \rangle_{L^2[(0,1)]} - \ l_y \circ I_k(\xi).
\end{aligned}$$

Since $\xi \mapsto \langle (T_k)^{-1}(\alpha), I_k(\xi) \rangle_{L^2[\Theta,\mu]} - l_y \circ I_k(\xi)$ is continuous, and $\mathcal{H}_k$ is dense in $L^2[\Theta, \mu]$ (Carmeli et al., 2010), it holds that

$$\sup_{\xi \in \mathcal{H}_k} \ \langle (T_k)^{-1}(\alpha), I_k(\xi) \rangle_{L^2[\Theta,\mu]} - \ l_y \circ I_k(\xi) = \sup_{f \in L^2[\Theta,\mu]} \ \langle (T_k)^{-1}\alpha, f \rangle_{L^2[\Theta,\mu]} - \ l_y(f)$$

which gives $(l_y \circ I_k)^\star = l_y^\star \circ (T_k)^{-1}$. □

Lemma 5.31 makes explicit the relationship between $\ell_y^\star$ and $l_y^\star$. It turns out that the scalar product in $L^2[\Theta, \mu]$ is well suited to l_y, so that $l_y^\star$ admits a simple expression, as stated by the theorem below.

Theorem 5.32. *Let $l_\theta\colon \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ be a family of loss functions indexed by $\theta \in \Theta$. Let $(y, g) \in L^2[\Theta, \mu] \times L^2[\Theta, \mu]$. If $\int_\Theta \min\left(0, l_{\theta, y(\theta)}^\star(g(\theta))\right)\mathrm{d}\mu(\theta) > -\infty$, then*

$$l_y^\star(g) = \int_\Theta l_{\theta, y(\theta)}^\star(g(\theta))\mathrm{d}\mu(\theta). \tag{5.19}$$

where $\forall t \in \mathbb{R}$, $l_{\theta,t}^\star$ stands for $l_\theta(\cdot, t)^\star$.

Proof. Recall that $l_y \colon g \in L^2[\Theta,\mu] \mapsto \int_\theta l_\theta(g(\theta), y(\theta))\mathrm{d}\mu(\theta)$. Let $f \in L^2[\Theta,\mu]$.

$$\begin{aligned} l_y^\star(f) &= \sup_{g\in L^2[\Theta,\mu]} \langle f, g\rangle_{L^2[\Theta,\mu]} - l_y(g) \\ &= \sup_{g\in L^2[\Theta,\mu]} \int_\Theta f(\theta)g(\theta)\mathrm{d}\mu(\theta) - \int_\Theta l_\theta(g(\theta), y(\theta))\mathrm{d}\mu(\theta) \\ &= \sup_{g\in L^2[\Theta,\mu]} \int_\Theta f(\theta)g(\theta) - l_\theta(g(\theta), y(\theta))\mathrm{d}\mu(\theta) \\ &\leq \int_\Theta \sup_{t\in\mathbb{R}} \; f(\theta)t - l_\theta(t, y(\theta))\mathrm{d}\mu(\theta) \\ &\leq \int_\Theta l^\star_{\theta,y(\theta)}(f(\theta))\mathrm{d}\mu(\theta), \end{aligned}$$

where $\forall s \in \mathbb{R}$, $l^\star_{\theta,s}$ stands for $l_\theta(\cdot, s)^\star$. Since $\int_\Theta \min\left(0, \ell^\star_{\theta,y(\theta)}(f(\theta))\right)\mathrm{d}\mu(\theta) > -\infty$, it holds that $\int_\Theta \ell^\star_{\theta,y(\theta)}(f(\theta))\mathrm{d}\mu(\theta) \in]-\infty, +\infty]$ is well defined, and equality is attained. □

Remark 5.33. *The instantiation of Equation (5.19) for specific loss functions gives:*

- *When* $l_\theta(s,t) = \frac{1}{2}(t-s)^2$, $\forall (y,g) \in (L^2[\Theta,\mu])^2$,

$$l_y^\star(g) = \frac{1}{2}\|g\|^2_{L^2[\Theta,\mu]} + \langle g, y\rangle_{L^2[\Theta,\mu]}$$

- *When* $l_\theta(s,t) = \max\left(\theta(t-s), (\theta-1)(t-s)\right)$, *for all* $g \in L^2[\Theta,\mu]$, *and* y *constant in* $L^2[\Theta,\mu]$,

$$l_y^\star(g) = \chi_{\theta-1\leq\cdot\leq\theta}(g) + y\int_\Theta g(\theta)\mathrm{d}\mu(\theta)$$

  where $\chi_{\theta-1\leq\cdot\leq\theta}$ *is to be understood in* $L^2[\Theta,\mu]$, *that is up to a null* μ*-set.*

- *When* $l_\theta(s,t) = |\theta - \mathbb{1}_{\{-1\}}(t)|\max\left(0, 1-ts\right)$, *for all* $g \in L^2[\Theta,\mu]$, *and* $y = \pm 1$ *constant in* $L^2[\Theta,\mu]$,

$$l_y^\star(g) = y\int_\Theta g(\theta)\mathrm{d}\mu(\theta) + \chi_{0\leq\cdot\leq|\theta-\mathbb{1}_{\{-1\}}(y)|}(-yg)$$

The key idea of the approach is then to find good candidates $(g_i)_{i=1}^n \in L^2[\Theta,\mu]$ such that $(\alpha_i)_{i=1}^n = (T_k g_i)_{i=1}^n \in \mathcal{H}_k$ are close to the solution of the dual problem.

Theorem 5.34. *Let* $\mathcal{K} = k_\mathcal{X}\mathbf{I}_{\mathcal{H}_k}$ *be an OVK such that* k *satisfies Assumptions 5.24 and 5.29. Assume also that Assumption 5.28 holds. The solution to Problem (5.17) is then given by*

$$\hat{h} = \frac{1}{\Lambda n}\sum_{i=1}^n k_\mathcal{X}(\cdot, x_i)T_k\hat{g}_i$$

with $(\hat{g}_i)_{i=1}^n \in (L^2[\Theta,\mu])^n$ *minimizing*

$$\sum_{i=1}^n l^\star_{y_i}(-g_i) + \frac{1}{2\Lambda n}\sum_{i,j=1}^n k_\mathcal{X}(x_i,x_j)\langle g_i, L_k g_j\rangle \tag{5.20}$$

Proof. Use $\alpha_i = T_k g_i$ for $i \leq n$, and Equation (5.18). □

Although Problem (5.20) phrases the optimization problem in a new space, it remains hard to solve since $L^2[\Theta, \mu]$ is infinite dimensional. To circumvent this difficulty, the research of the $(g_i)_{i=1}^n$ will be performed in a finite dimensional subspace adapted to the problem, namely $\text{span}\{\psi_m, m \leq M\}$, where $\{\psi_m\}$, $m \leq M$ are the eigenvectors associated to the M largest eigenvalues of L_k. With the notation $S = \text{diag}(\lambda_m, m \leq M)$, an approximate dual problem reads:

$$\min_{\beta \in \mathbb{R}^{n \times M}} \sum_{i=1}^{n} l_{y_i}^\star \left(- \sum_{m=1}^{M} \beta_{im} \psi_m \right) + \frac{1}{2\Lambda n} \mathbf{Tr}\left(K^X \beta S \beta^\top \right). \tag{5.21}$$

Remark 5.35. *The eigendecomposition of L_k is dependent both in k and μ, and can be approximately solved using the Galerkin method (Chatelin, 2011), or by solving a differential equations derived from the eigenvalue problem. However, given that the optimal kernel k is unknown, one can choose a Hilbertian basis $\{\psi_m\}_{m=1}^\infty$ of $L^2[\Theta, \mu]$ and a non-increasing sequence $(\lambda_m)_{m=1}^\infty \in \mathbb{R}_+^*$ to construct the kernel k, which gives direct access to the eigendecomposition of T_k.*

Below are presented ways to solve Problem (5.21) for various loss functions. In the following, $R \in \mathbb{R}^{n \times M}$ refers to the matrix such that for all $i \leq n$, and all $m \leq M$, $R_{im} = \langle \psi_m, y_i \rangle_{L^2[\Theta, \mu]}$.

Ridge Regression. When $l_\theta(s, t) = \frac{1}{2}(t - s)^2$, Problem (5.21) reads

$$\min_{\beta \in \mathbb{R}^{n \times M}} \mathbf{Tr}\left(\frac{1}{2} \beta \beta^\top + \frac{1}{2\Lambda n} K^X \beta S \beta^\top - \beta R^\top \right), \tag{5.22}$$

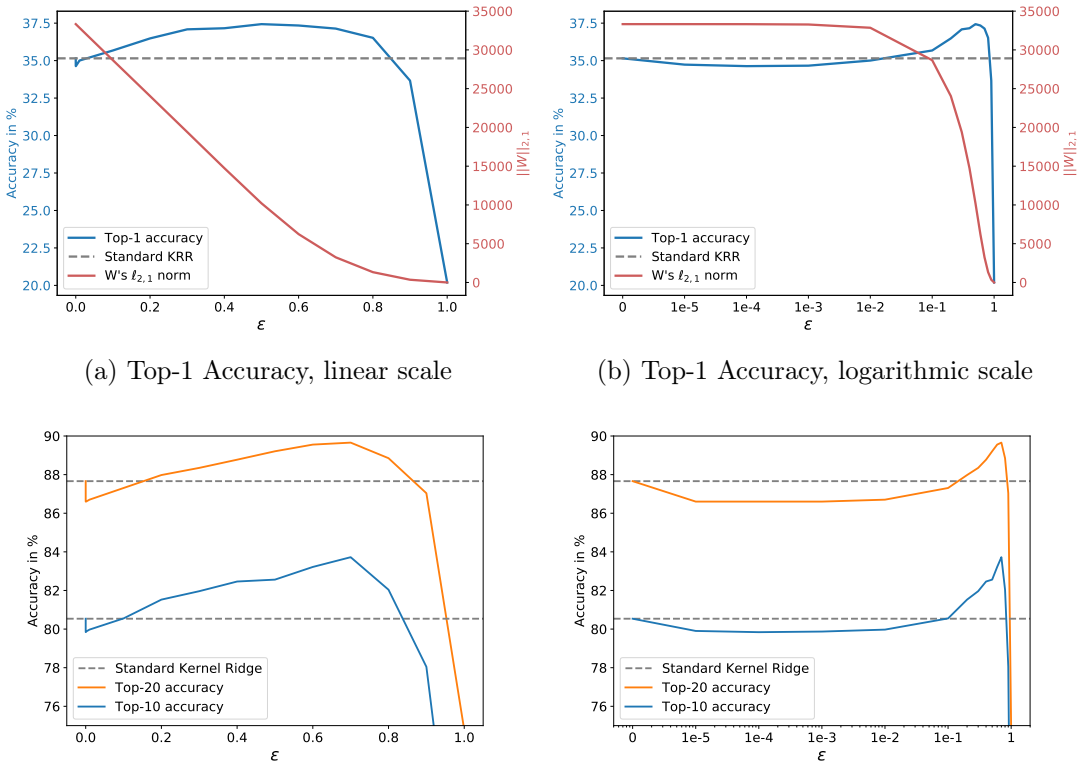
so that it boils down to the minimization of a quadratic form. Setting the gradient to zero yields a solution $\hat{\beta} = (I + K^X/(\Lambda n) \otimes S)^{-1} R$, where $K^X \otimes S$ is the block operator matrix such that $(K^X \otimes S)_{ij} = k_{\mathcal{X}}(x_i, x_j) S$. The inversion of this operator can be performed using its spectral decomposition, and $\hat{\beta}$ coincides with the closed-form solution given in (Kadri et al., 2016).

Dealing with Lipschitz Losses. When $(l_\theta)_{\theta \in \Theta}$ is a family of Lipschitz loss functions, $l_y^\star(g)$ may take $+\infty$ as value if g is not in the feasible set of the dual problem. This induces an additional difficulty to the resolution of Problem (5.21), since the finite dimensional space $\text{span}\{\psi_m, m \leq M\}$ may not be stable with respect to the projection on the feasible set, which annihilates any hope for a vanilla proximal gradient descent.

Application to Huber Loss. Function-to-function regression has mainly been dealt with through the minimization of an empirical L^2 risk. However, in the spirit of Section 5.3.2, this task can be tackled using a Huber loss, which induces robustness. The approximate dual problem is then Problem (5.22) under the additional constraint that $\|\beta\|_{2,\infty} \leq \kappa$, and it can be solved through PGD. Experimental results endorsing this approach are presented in section 5.5.

5.5 Numerical Experiments

Numerical experiments have been run in order to show the benefit of using more sophisticated losses than the standard squared norm in output Hilbert spaces. We focus on three applications: structured prediction, structured representation learning, and functional regression.



(a) Top-1 Accuracy, linear scale

(b) Top-1 Accuracy, logarithmic scale

(c) Top-10/20 Accuracies, linear scale

(d) Top-10/20 Accuracy, logarithmic scale

Figure 5.5 – ϵ-Insensitive Output Kernel Regression Results

κ	Top 1	Top 10	Top 20	$\|W\|_{2,1}$
0.5	38.0	83.5	89.6	2789.6
1.0	**38.9**	**83.8**	**89.9**	5572.4
1.5	38.6	83.7	89.8	8231.9

Table 5.1 – Huber test accuracies (%) with respect to κ

Structured prediction. We consider the problem of identifying metabolites based on their mass spectra (Brouard et al., 2016a). We investigate the advantages of substituting the standard Ridge Regression for its ϵ-insensitive version or a Huber regression. Outputs (metabolites) are embedded into an infinite dimensional Hilbert space by means of a Tanimoto-Gaussian kernel with 0.72 bandwidth. The data is composed of 6974 mass spectra, and algorithms are compared through the top-k accuracies, $k = 1, 10, 20$.

As expected, a wide range of ϵ's induce substantial improvements compared to Ridge Regression (see Figure 5.5). This improvement comes with a norm reduction until the collapsing point at $\epsilon = 1$. The Huber results are gathered in Table 5.1, showing valuable gains for all κ's.

Structured representation learning. Again, the introduction of an ϵ-insensitive algorithm allows to improve generalization while inducing sparsity (Figure 5.6). This makes the ϵ-insensitive framework particularly promising in the context of surrogate approaches.

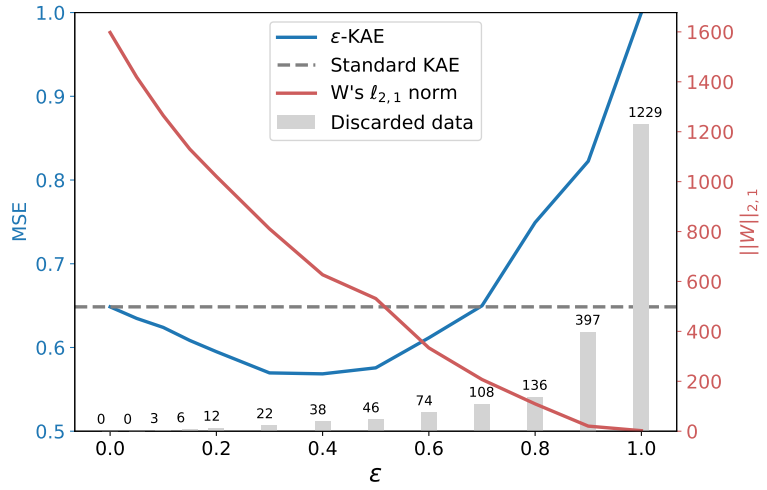


Figure 5.6 – Test Mean Squared Error (MSE) w.r.t. ϵ.

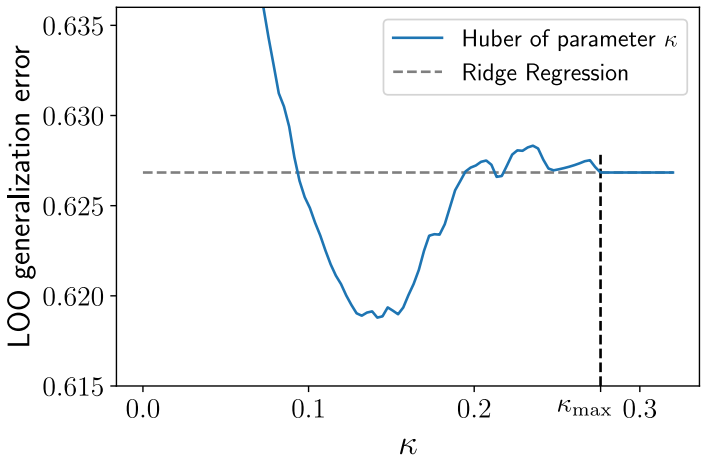


Figure 5.7 – LOO error with respect to κ.

Function-to-Function Regression. Our goal here is to predict lip acceleration from electromyography (EMG) signals (Ramsay and Silverman, 2007). The dataset consists of 32 records of the trajectory of the lower lip associated to EMG records of the muscle that controls it, augmented by 4 outliers to assess the robustness of our approach. We solve Problem (5.21) with a Huber loss for various levels κ. Usefulness of minimizing the Huber loss is illustrated in Figure 5.7 by computing the Leave-One-Out (LOO) generalization error associated to each model. Models trained with Huber loss may generalize better depending on κ. As κ grows larger than $\kappa_{\max}$, the constraint on $\|\beta\|_{2,\infty}$ becomes void and we recover the Ridge Regression solution. We used $k_{\mathcal{X}}(x_1, x_2) = \int_0^1 \exp(|x_1(\theta) - x_2(\theta)|)\mathrm{d}\theta$, $(\psi_m)_{m=1}^M$ the harmonic basis in sine and cosine of $L^2[0,1]$, $(\lambda_m)_{m=1}^M = (1/(m+1)^2)_{m=1}^M$ and $M = 16$.

5.6 Conclusion

This chapter presents an extended analysis of the duality principle within vv-RKHSs, allowing for the use of new loss functions. The particular case of convolved losses is tackled, offering novel ways to enforce sparsity and robustness. This opens an avenue for new applications on structured data (*e.g.* anomaly detection, robust prediction), whose generalization guarantees remain to be investigated. The use of kernel approximations, such as Random Fourier Features (Rahimi and Recht, 2008; Brault et al., 2016) or Nyström's method (Williams and Seeger, 2001) represent a promising research direction, as the analysis presented in this chapter would benefit twice from it: in input and in output. These new loss functions can be applied at the last layers of deep kernel architectures developed in Chapters 3 and 4, enriching their framework. The *Double Representer Theorem* ensures a finite dimensional parametrization even when outputs are infinite dimensional (through the Ω matrix rather than the infinite dimensional coefficients $\varphi_{L,i}$). This suggests new algorithms to optimize deep kernel machines, based on full Gradient Descent strategies.

Part II

Reliable Machine Learning

The second part of this manuscript investigates alternatives to the sample mean as substitutes to the expectation in the Empirical Risk Minimization (ERM) framework.

As a reminder, given $\mathcal{S}_n = \{(x_i, y_i)_{i \leq n}\} \in (\mathcal{X} \times \mathcal{Y})^n$ an i.i.d. sample distributed as P, and a hypothesis set $\mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$, the ERM paradigm consists in replacing

$$\min_{h \text{ measurable}} \mathcal{R}(h) = \mathbb{E}_P \left[\ell \left(h(X), Y \right) \right] \qquad \text{by} \qquad \min_{h \in \mathcal{H}} \widehat{\mathcal{R}}_n(h) = \frac{1}{n} \sum_{i=1}^{n} \ell \left(h(x_i), y_i \right).$$

In Part I, we have focused on a specific choice of hypothesis set $\mathcal{H}$. The goal of Part II is to address the other approximation made during the transformation of the ideal problem into a computable one: the replacement of the unknown expectation by the empirical mean. Indeed, this implicitly assume that the empirical mean is a good estimate of the expectation. However, in many practical use cases (*e.g.* heavy-tailed distribution, presence of outliers, biased training data), this is not the case.

As a first go, Chapter 6 collects some basic notions about U-statistics, which are crucial tools when studying pairwise learning criteria. Their strong concentration properties are incidentally needed to analyze the mean estimators introduced in the next chapter.

The Median-of-Means is indeed a robust mean estimator at the core of Chapter 7. Its basic principle, taking the median of smaller independent estimators, is then extended both to randomization and U-statistics. This results in several novel (pairwise) mean estimators with provable guarantees, except for the Median-of-Incomplete-U-Statistics.

By their construction, Median-of-Means-like estimators exhibit interesting robustness properties. This is further exploited in Chapter 8, where Median-of-Means minimizers are shown to behave well in presence of outliers. As for the Median-of-Means tournament procedure, it yields strong guarantees under mild assumptions on distribution P. Both approaches are extended to U-statistics, allowing for robust pairwise learning strategies.

Finally, Chapter 9 addresses a totally different issue: that of sample bias. If training data comes from several biased samples, computing blindly the empirical mean yields a biased estimate of the risk. Alternatively, from the knowledge of the biasing functions, it is possible to reweight observations so as to build an unbiased estimate of the test distribution. The known asymptotic guarantees are first made non-asymptotic, and then translated into guarantees for the debiased risk minimizers. This new framework provides a general adaptation of the ERM paradigm to the case of biased data.

6

Reminders on U-statistics

Contents

As seen in Chapter 1, a vast majority of machine learning problems may be cast as the following problem:

$$\min_{h \text{ measurable}} \quad \mathbb{E}_{Z\sim P}\left[\ell(h, Z)\right],$$

with ℓ a loss function, and the notation abuse $\ell(h, z)$ for $\ell(h(x), y)$. As P is unknown in practice, the Empirical Risk Minimization (ERM) approach suggests solving instead

$$\min_{h\in\mathcal{H}} \quad \frac{1}{n}\sum_{i=1}^{n} \ell(h, z_i),$$

where $\{z_i\}_{i\leq n}$ are n i.i.d. realizations of Z, and $\mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$ the hypothesis set. However, not all machine learning problems can be reduced to a simple empirical mean. Indeed, in cases such as *ranking* or *metric learning* for instance, one compares pairs of observations. The objective one would ideally optimize then reads

$$\min_{h\in\mathcal{H}} \quad \mathbb{E}_{Z,Z'\sim P\otimes P}\left[\ell(h, Z, Z')\right].$$

To adapt the ERM methodology, one then needs an unbiased estimator of an expectation over two independent identically distributed random variables. This is precisely the purpose of U-statistics, introduced by Wassily Hoeffding in the 1940s (Hoeffding, 1948), with U standing for *unbiased*, to furnish such estimators.

After having formally defined U-statistics in their most general form in Section 6.1, we exhibit in Section 6.2 several cases, both in statistics and machine learning, where U-statistics are typically used. Section 6.3 is devoted to properties of U-statistics that are used in the proofs of Chapters 7 and 8, while in Section 6.4 are presented extensions around U-statistics that incidentally appear in further chapters.

6.1 Definition

As a first go, let us recall the definition of a generalized U-statistic of arbitrary degree.

Definition 6.1. *Let $d \in \mathbb{N}^*$, and $\{Z_i\}_{i\leq n}$ be a collection of $n \geq d$ i.i.d. random variables, valued in some metric space $\mathcal{Z}$, with distribution $F(dz)$. Let $h : \mathcal{Z}^d \to \mathbb{R}$ be a measurable function, square integrable with respect to the probability distribution $F^{\otimes d}$. Assume in addition (without loss of generality) that h is symmetric in its d arguments. The U-statistic of degree d with kernel h is then defined as*

$$U_n(h) = \frac{1}{\binom{n}{d}} \sum_I h\left(Z_{I_1}, \ldots, Z_{I_d}\right),$$

where the symbol $\sum_I$ refers to the summation over all unordered subsets $I = \{I_1, \ldots, I_d\}$ of d integers chosen in $\{1, \ldots, n\}$.

One may see that Definition 6.1 already generalizes the standard sample mean, that is obtained with $d = 1$, and $h(z) = z$. However, U-statistics can be made even more general by involving several samples. For the sake of completeness, we also define such multisample U-statistics, although most statistical learning applications only require the single sample version. Unless otherwise specified, all subsequent analyses and properties are established for single sample U-statistics.

Definition 6.2. *Let $S \geq 1$ and $(d_1, \ldots, d_S) \in \mathbb{N}^{*S}$. Let $\boldsymbol{Z}_{\{1,\ldots,n_s\}} = (Z_1^{(s)}, \ldots, Z_{n_s}^{(s)})$, for $s \leq S$, be S independent samples of sizes $n_s \geq d_s$ and composed of i.i.d. random variables valued in some measurable spaces $\mathcal{Z}_s$ with distribution $F_s(dz)$ respectively. Let $h : \mathcal{Z}_1^{d_1} \times \ldots \times \mathcal{Z}_S^{d_S} \to \mathbb{R}$ be a measurable function, square integrable with respect to the probability distribution $F_1^{\otimes d_1} \otimes \ldots \otimes F_S^{\otimes d_S}$. Assume in addition (without loss of generality) that h is symmetric within each block of argument $\boldsymbol{Z}^{(s)}$ valued in $\mathcal{Z}_s^{d_s}$, for $s \leq S$. The generalized (or S-sample) U-statistic of degrees $(d_1, \ldots, d_S)$ with kernel h is then defined as*

$$U_{\boldsymbol{n}}(h) = \frac{1}{\prod_{s=1}^S \binom{n_s}{d_s}} \sum_{I_1} \cdots \sum_{I_S} h\left(\boldsymbol{Z}_{I_1}^{(1)}, \ldots, \boldsymbol{Z}_{I_S}^{(S)}\right),$$

where the symbol $\sum_{I_s}$ refers to the summation over all $\binom{n_s}{d_s}$ subsets $\boldsymbol{Z}_{I_s}^{(s)} = (Z_{i_1}^{(s)}, \ldots, Z_{i_{d_s}}^{(s)})$ related to a set I_s of d_s indexes $1 \leq i_1 < \ldots < i_{d_s} \leq n_s$, and $\boldsymbol{n} = (n_1, \ldots, n_S)$.

In order to provide more intuition on these notions, we continue by giving examples of U-statistics, both in the statistics and statistical learning literatures.

6.2 Examples

As already mentioned, the standard sample mean is a particular instance of a U-statistic. However, more convincing and illustrative examples are to be listed now.

6.2.1 Occurrences in Statistics

We start by giving three examples taken from the statistics literature. Each one has its specificity, as there is one U-statistic of degree 2, one U-statistic based on non-scalar observations, and one multisample U-statistic.

Sample Variance. A very classical example is the empirical variance. Given a sample $\{Z_i\}_{i\leq n}$, it reads

$$\hat{\sigma}_n^2 = \frac{1}{n(n-1)} \sum_{1\leq i<j\leq n} (Z_i - Z_j)^2,$$

which is a one sample U-statistic of degree 2, with kernel $h(z, z') = (z - z')^2/2$.

Kendall's τ. Definitions 6.1 and 6.2 do not require observations to be real-valued. Let $\{Z_i = (X_i, Y_i)\}_{i\leq n}$ be a sample of n i.i.d. bivariate random vectors. In order to measure pairs concordance, one may compute Kendall's τ, which reads

$$\tau = \frac{4}{n(n-1)} \sum_{1\leq i<j\leq n} \mathbb{1}\left\{(Y_j - Y_i)(X_j - X_i) > 0\right\} - 1.$$

This is a one sample U-statistic of degree 2 with kernel $h(z, z') = h((x, y), (x', y')) = 2 \cdot \mathbb{1}\{(y' - y)(x' - x) > 0\} - 1$. A Kendall's τ close to 1 means that pairs are globally concordant, while a value close to -1 suggests a negative dependence, and a value around 0 is a sign of independence.

Mann-Whitney Statistic. Assume now that our goal is to estimate $\theta = \mathbb{P}\{Z \leq Z'\}$, for two random variables Z and Z'. Given two i.i.d. samples $\{Z_i\}_{i\leq m}$ and $\{Z'_j\}_{j\leq n}$, distributed as Z and Z' respectively, the natural empirical estimator of θ, called the Mann-Whitney statistic, is

$$\hat{\theta}_{m,n} = \frac{1}{m\ n} \sum_{i=1}^{m} \sum_{j=1}^{n} \mathbb{1}\left\{Z_i \leq Z'_j\right\}.$$

This statistic is an example of a two samples U-statistic of degrees $(1, 1)$.

U-statistics are also present in the statistical learning literature. Indeed, following the Empirical Risk Minimization paradigm, one is often encouraged to minimize the empirical version of the risk. Even if most of the times this empirical risk takes the form of a simple empirical mean, there are still many situations in which the summation is done over pairs of observations. It is the purpose of the following subsection to present three examples of learning problems that write as a U-statistic.

6.2.2 Occurrences in Statistical Learning

Although less ubiquitous in the statistical learning literature than standard empirical means, U-statistics are nonetheless perfectly suited to describe and tackle many learning tasks. Three of them are now to be described, that write as one sample U-statistics of degree 2.

Clustering. In clustering, the goal is to find a partition $\mathcal{P}$ of the feature space $\mathcal{Z}$ so that pairs of observations independently drawn from a certain distribution F on $\mathcal{Z}$ within a same cell of $\mathcal{P}$ are more similar with respect to a given metric $d : \mathcal{Z}^2 \to \mathbb{R}_+$ than pairs lying in different cells. Let $\Phi_{\mathcal{P}}(z, z') = \sum_{\mathcal{C}\in\mathcal{P}} \mathbb{1}\{(z, z') \in \mathcal{C}^2\}$ for a partition candidate $\mathcal{P}$. Based on an i.i.d. training sample $Z_1, \ldots, Z_n$, the Empirical Risk Minimization paradigm leads to minimizing the U-statistic, referred to as *empirical clustering risk* (see Clémençon (2014) and references therein):

$$\widehat{\mathcal{W}}_n(\mathcal{P}) = \frac{2}{n(n-1)} \sum_{1\leq i<j\leq n} d(Z_i, Z_j) \cdot \Phi_{\mathcal{P}}(Z_i, Z_j).$$

Metric Learning. In metric learning (Bellet et al., 2015), one goes the other way around. From given similarities (*e.g.* pertaining to the same cell, being sampled from the same class), the practitioner aims at learning a metric $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$, such that points supposedly close are also similar with respect to d. Given a sample $\{(X_i, Y_i)\}_{i \leq n}$, drawn i.i.d. from a random pair (X, Y) valued in $\mathcal{X} \times \mathcal{Y}$, one may construct *a priori similarities* Y_{ij} (*e.g.* $Y_{ij} = 2 \cdot \mathbb{1}\{Y_i = Y_j\} - 1$ if the labels are classes) and then minimize, for a given tolerance ϵ, the empirical risk:

$$\widehat{\mathcal{R}}_n(d) = \frac{2}{n(n-1)} \sum_{1 \leq i < j \leq n} \mathbb{1}\left\{ Y_{ij} \cdot (d(X_i, X_j) - \epsilon) > 0 \right\}.$$

Apart from the monograph by Bellet et al. (2015), the interested reader may refer to Bellet et al. (2013) and Bellet and Habrard (2015), that focus respectively on robustness aspects and structured data in metric learning. Statistical perspectives may also be found in Bellet et al. (2012), or more recently in Vogel et al. (2018), with a pointwise optimization of the ROC curve.

Pairwise Ranking. In pairwise ranking, the objective is to learn from independent labeled data $(X_1, Y_1), \ldots, (X_n, Y_n)$, drawn as a generic random pair $(X, Y) \in \mathcal{X} \times \mathbb{R}$, a ranking rule $r : \mathcal{X}^2 \to \{-1, 0, +1\}$ that permits to predict, among two objects (X, Y) and (X', Y') chosen at random, which one is preferred: (X, Y) is preferred to (X', Y') when $Y > Y'$ and, in this case, one would ideally have $r(X, X') = +1$, the rule r being supposed anti-symmetric (*i.e.* $r(x, x') = -r(x', x)$ for all $(x, x') \in \mathcal{X}^2$). This can be formulated as the problem of minimizing the U-statistic known as the *empirical ranking risk* (see Clémençon et al. (2005)):

$$\widehat{\mathcal{L}}_n(r) = \frac{2}{n(n-1)} \sum_{1 \leq i < j \leq n} \mathbb{1}\left\{ r(X_i, X_j) \cdot (Y_i - Y_j) \leq 0 \right\}.$$

Now that U-statistics have been clearly defined and illustrated, we next focus on some fundamental properties that are useful in further chapters.

6.3 Basic Properties

We address here two main topics: expectation and variance of a U-statistic, and its concentration properties Extensions such as incomplete U-statistics and V-statistics are deferred to Section 6.4.

6.3.1 Expectation and Variance

Introduced by Hoeffding in the 1940s (Hoeffding, 1948), U-statistics aims at estimating

$$\theta(h) = \mathbb{E}\left[h(Z_1, \ldots, Z_d) \right],$$

for a given symmetric kernel h, and i.i.d. random variables $\{Z_i\}_{i \leq d}$. Given a sample $\{Z_i\}_{i \leq n}$ with $n \geq d$, a first unbiased estimator of θ is $h(Z_1, \ldots, Z_d)$. But this estimate is unnatural as it does not use the full sample. To remedy this problem, one can average this statistic over all unordered d-tuples pertaining to $\{1, \ldots, n\}$, leading to Definition 6.1. The obtained $U_n(h)$ is therefore also an unbiased estimator of θ (U stands for *unbiased*), and one can prove that its variance is lower than that of $h(Z_1, \ldots, Z_d)$.

Indeed, since $U_n(h)$ is an average (over all permutations), it can be expressed as the following conditional expectation

$$U_n(h) = \mathbb{E}\left[h(Z_1, \dots, Z_d) \mid Z_{(1)}, \dots, Z_{(n)}\right],$$

where $Z_{(1)}, \dots, Z_{(n)}$ denote the data $Z_1, \dots, Z_n$, but sorted in increasing order.

Then, Jensen's inequality yields

$$\begin{aligned}
\mathrm{Var}\left(U_n(h)\right) &= \mathbb{E}\left[(U_n(h) - \theta)^2\right], \\
&= \mathbb{E}\left[\left(\mathbb{E}\left[h(Z_1, \dots, Z_d) \mid Z_{(1)}, \dots, Z_{(n)}\right] - \theta\right)^2\right], \\
&\leq \mathbb{E}\left[\mathbb{E}\left[\left(h(Z_1, \dots, Z_d) - \theta\right)^2 \mid Z_{(1)}, \dots, Z_{(n)}\right]\right], \\
&\leq \mathbb{E}\left[\left(h(Z_1, \dots, Z_d) - \theta\right)^2\right], \\
&\leq \mathrm{Var}\left(h(Z_1, \dots, Z_d)\right).
\end{aligned}$$

It can further be shown by a Lehmann-Scheffé argument that $U_n(h)$ is the unbiased estimator of $\theta(h) = \mathbb{E}[h(Z_1, \dots, Z_d)]$ with minimal variance. Let us now investigate more deeply the form taken by $\mathrm{Var}(U_n(h))$.

$$\begin{aligned}
\mathrm{Var}\left(U_n(h)\right) &= \mathrm{Cov}\left(\frac{1}{\binom{n}{d}} \sum_I h\left(Z_{I_1}, \dots, Z_{I_d}\right), \frac{1}{\binom{n}{d}} \sum_{I'} h\left(Z_{I'_1}, \dots, Z_{I'_d}\right)\right), \\
&= \frac{1}{\binom{n}{d}^2} \sum_{I, I'} \mathrm{Cov}\left(h\left(Z_{I_1}, \dots, Z_{I_d}\right), h\left(Z_{I'_1}, \dots, Z_{I'_d}\right)\right).
\end{aligned}$$

Given the symmetry of h, $\mathrm{Cov}(h(Z_{I_1}, \dots, Z_{I_d}), h(Z_{I'_1}, \dots, Z_{I'_d}))$ only depends on the number of common variables in I and I'. For $c \leq d$, let $\zeta_c(h) = \mathrm{Cov}(h(Z_{I_1}, \dots, Z_{I_d}), h(Z_{I'_1}, \dots, Z_{I'_d}))$ when c variables are common. It is now enough to count how many times each case occurs. One has $\binom{n}{d}$ choices for variables in I, $\binom{d}{c}$ choices for components in I, and $\binom{n-d}{d-c}$ choices for variables in I'. Hence, noticing that $\zeta_0(h) = 0$, it holds

$$\mathrm{Var}\left(U_n(h)\right) = \frac{1}{\binom{n}{d}} \sum_{c=1}^{d} \binom{d}{c} \binom{n-d}{d-c} \zeta_c(h). \tag{6.1}$$

An upper bound can thus immediately be derived:

$$\begin{aligned}
\mathrm{Var}\left(U_n(h)\right) &= \sum_{c=1}^{d} \frac{d!^2}{c!(d-c)!^2} \, \frac{(n-d)(n-d-1)\dots(n-2d+c+1)}{n(n-1)\dots(n-d+1)} \, \zeta_c(h), &(6.2)\\
&\leq \sum_{c=1}^{d} \frac{d!^2}{c!(d-c)!^2} \, \frac{\zeta_c(h)}{n(n-1)\dots(n-c+1)}. &(6.3)
\end{aligned}$$

For the particular case $d = 2$, one can also use the *Second Hoeffding's Decomposition*

$$U_n(h) - \theta(h) = \frac{2}{n}\sum_{i=1}^{n} h_1(X_i) + \frac{2}{n(n-1)} \sum_{1\le i<j\le n} h_2(X_i, X_j),$$

with

$$h_1(Z_1) = \mathbb{E}\left[h(Z_1, Z_2) \mid Z_1\right] - \theta(h) \qquad \sigma_1^2(h) = \mathrm{Var}\left(h_1(Z_1)\right), \tag{6.4}$$

$$h_2(Z_1, Z_2) = h(Z_1, Z_2) - h_1(Z_1) - h_1(Z_2) - \theta(h) \qquad \sigma_2^2(h) = \mathrm{Var}\left(h_2(Z_1, Z_2)\right). \tag{6.5}$$

It is then direct to see that

$$\sigma^2(h) = \mathrm{Var}\left(h(Z_1, Z_2)\right) = 2\sigma_1^2(h) + \sigma_2^2(h). \tag{6.6}$$

and

$$\mathrm{Var}\left(U_n(h)\right) = \frac{4}{n}\sigma_1^2(h) + \frac{2}{n(n-1)}\sigma_2^2(h). \tag{6.7}$$

Notice that in general $\zeta_c(h)$ and $\sigma_c^2(h)$ are different. Indeed, one has

$$\zeta_2(h) = \mathrm{Cov}\left(h(Z, Z'), h(Z, Z')\right) = \mathrm{Var}\left(h(Z, Z')\right) = \sigma^2(h) = 2\sigma_1^2(h) + \sigma_2^2(h),$$

and

$$\begin{aligned}
\zeta_1(h) &= \mathrm{Cov}\left(h(Z, Z'), h(Z, Z'')\right), \\
&= \mathbb{E}\left[h(Z, Z')h(Z, Z'')\right] - \mathbb{E}\left[h(Z, Z')\right]\mathbb{E}\left[h(Z, Z'')\right], \\
&= \mathbb{E}\left[\mathbb{E}\left[h(Z, Z')h(Z, Z'') \mid Z\right]\right] - \theta^2(h), \\
&= \mathbb{E}\left[\mathbb{E}\left[h(Z, Z') \mid Z\right]\mathbb{E}\left[h(Z, Z'') \mid Z\right]\right] - \theta^2(h), \\
&= \mathbb{E}\left[h_1(Z)^2\right] - \mathbb{E}\left[h_1(Z)\right]^2, \\
&= \sigma_1^2(h).
\end{aligned}$$

Replacing these values in Equation (6.2), one recovers

$$\begin{aligned}
\mathrm{Var}\left(U_n(h)\right) &= \frac{4(n-2)}{n(n-1)}\zeta_1(h) + \frac{2}{n(n-1)}\zeta_2(h), \\
&= \frac{4(n-2)}{n(n-1)}\sigma_1^2(h) + \frac{2}{n(n-1)}(2\sigma_1^2(h) + \sigma_2^2(h)), \\
&= \frac{4}{n}\sigma_1^2(h) + \frac{2}{n(n-1)}\sigma_2^2(h).
\end{aligned}$$

Finally, the trick used to compute the variance of $U_n(h)$ can be readily applied to the multisample setting. Denoting $\zeta_{c_1,\dots,c_S}(h) = \mathrm{Cov}\left(h(\boldsymbol{Z}_{I_1}^{(1)}, \dots, \boldsymbol{Z}_{I_S}^{(S)}), h(\boldsymbol{Z}_{I_1'}^{(1)}, \dots, \boldsymbol{Z}_{I_S'}^{(S)})\right)$ when c_s variables are common in I_s and I_s' for $s \le S$, one gets

$$\mathrm{Var}\left(U_n(h)\right) = \frac{1}{\prod_{s=1}^{S}\binom{n_s}{d_s}^2} \sum_{c_1=0}^{d_1} \cdots \sum_{c_S=0}^{d_S} \prod_{s=1}^{S} \binom{n_s}{d_s}\binom{d_s}{c_s}\binom{n_s - d_s}{d_s - c_s}\zeta_{c_1,\dots,c_S}(h). \tag{6.8}$$

So much care has been given to explicit U-statistics variances because they play a crucial role for the estimators defined in the next sections, influencing for instance their concentration rates. When dealing with U-statistics of degree 2, Equation (6.7) will be preferred, while Equations (6.1) and (6.3) will be used to generalize to U-statistics of arbitrary degree. Finally, although results for multisample U-statistics are not stated in this manuscript, Equation (6.8) suggests that a careful analysis (in particular to the different samples sizes) should lead to similar guarantees.

6.3.2 Concentration Properties

A first idea that comes to mind to derive concentration properties for U-statistics is the bounded difference inequality (see Appendix A). Indeed, although the functional takes the form of an average, the non independence between pairs that share an observation prevents from the use of standard Hoeffding's inequality. Thus, applying the bounded differences inequality to a U-statistic of degree d and bounded kernel h in a direct manner yields

$$\mathbb{P}\left\{\left|U_n(h) - \theta(h)\right| > t\right\} \leq 2\exp\left(-\frac{t^2}{2\|h\|_\infty^2}\frac{n}{d^2}\right).$$

But the dependence in d may be improved, as revealed by the following proposition due to Hoeffding.

Proposition 6.3. *(Hoeffding's Inequality for U-Statistics, Hoeffding (1963)). Let $d \in \mathbb{N}^*$, $\{Z_i\}_{i\leq n}$ be $n \geq d$ independent realizations of a $\mathcal{Z}$-valued random variable Z, and $h : \mathcal{Z}^d \to \mathbb{R}$ bounded such that $\mathbb{E}[h(Z_1, \ldots, Z_d)] = \theta(h)$. Then, with $U_n(h)$ defined as in Definition 6.1, it holds for any $t > 0$*

$$\mathbb{P}\left\{\left|U_n(h) - \theta(h)\right| > t\right\} \leq 2\exp\left(-\frac{t^2}{2\|h\|_\infty^2}\frac{n}{d}\right).$$

Notice that, similarly to the standard mean (Theorem 7.3), this bound requires h to be bounded, which is not the case in general (*e.g.* variance of unbounded random variable). The estimators presented in Chapter 7 remedy this limitation, and exhibit similar exponential guarantees on the sole assumption that the second order moment of h is finite. The interested reader may finally refer to Maurer et al. (2019) for a recent extension of Bernstein's inequality with applications to U-statistics concentration.

6.4 Extensions

This last section deals with important extensions around U-statistics. In Section 6.4.1 we focus on *incomplete U-statistics*, that aim at downscaling the computational cost of U-statistics by sampling the pairs summed, instead of building all possible combinations. Finally, Section 6.4.2 analyses V-statistics, that allow one observation to appear several times in the kernel, inducing a more complex dependence structure.

6.4.1 Incomplete U-Statistics

One major practical drawback of U-statistics is their computational cost, as it involves the summation of $\mathcal{O}(n^d)$ terms, when d is the degree of the U-statistic. The concept of *incomplete U-statistic* (Blom, 1976) precisely permits to address this computational

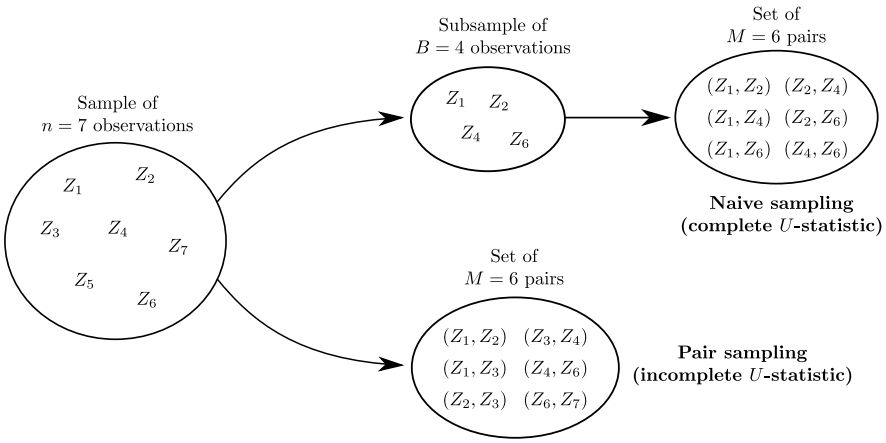


Figure 6.1 – Incomplete U-statistic Procedure

issue and achieve a trade-off between scalability and variance reduction. In one of its simplest forms (we focus here on the case $d = 2$), it consists in selecting a subsample of size $M \geq 1$ by sampling with replacement in the set of pairs of observations that can be formed from the original sample (for sampling without replacement, refer to *e.g.* Serfling (1974)). Setting $\Lambda = \{(i, j) : 1 \leq i < j \leq n\}$, and denoting by $\{(i_1, j_1), \ldots, (i_M, j_M)\} \subset \Lambda$ the subsample drawn by Monte-Carlo, the incomplete version of the U-statistic defined in Definition 6.1 is

$$\tilde{U}_M(h) = \frac{1}{M} \sum_{m \leq M} h(X_{i_m}, X_{j_m}).$$

It is direct to see that $\tilde{U}_M(h)$ is also an unbiased estimator of $\theta(h)$ and that its variance is necessarily larger than that of the complete U-statistic based on the full sample (as it uses less pairs). Formally, one has

$$\mathrm{Var}\left(\tilde{U}_M(h)\right) = \left(1 - \frac{1}{M}\right) \mathrm{Var}(U_n(h)) + \frac{\sigma^2(h)}{M}. \tag{6.9}$$

Observe that the difference between the variances vanishes as M increases. In contrast, $\tilde{U}_M(h)$ exhibits an interesting variance among estimates based on M pairs only. Indeed, for a complete U-statistic to involve only M pairs, one has to select no more than $\sqrt{M}$ observations (roughly). Equation (6.7) indicates that the variance would be in $\mathcal{O}(1/\sqrt{M})$, where the incomplete U-statistic has a $\mathcal{O}(1/M)$ one. This difference is due to the redundancy of the complete U-statistic, that always pairs the same observations, while the incomplete version is allowed to browse the whole dataset. Refer to Figure 6.1, reproduced from Clémençon et al. (2016), for a visual representation of the phenomenon. This reduced variance property has major consequences, inducing for instance scalable statistical learning strategies, such as that investigated in Clémençon et al. (2016). As for practical applications, the interested reader may refer to Bertail and Tressou (2006) for a nice utilization of incomplete U-statistics in food risk assessment.

6.4.2 V-statistics

The next and final topic we address is the analysis of the closely related V-statistics. Named after Richard von Mises, these statistics are built exactly the same way as U-statistics, except that they allow multiple replications of an observation within the

kernel. Thus, for a V-statistic of degree d, the average is made over n^d terms, rather than $\binom{n}{d}$ for the associated U-statistic. Let us give a formal definition.

Definition 6.4. *Let $d \in \mathbb{N}^*$, and $\{Z_i\}_{i\leq n}$ be a collection of $n \geq d$ i.i.d. random variables, valued in some metric space $\mathcal{Z}$, with distribution $F(dz)$. Let $h : \mathcal{Z}^d \to \mathbb{R}$ be a measurable function, square integrable with respect to the probability distribution $F^{\otimes d}$. Assume in addition (without loss of generality) that h is symmetric in its d arguments. The V-statistic of degree d with kernel h is then defined as*

$$V_n(h) = \frac{1}{n^d} \sum_{i_1=1}^{n} \dots \sum_{i_d=1}^{n} h\left(Z_{i_1}, \dots, Z_{i_d}\right).$$

Observe that the allowed replications completely break the independence assumption, and consequently many proofs about U-statistics. Indeed, proofs often rely on a specific representation of the U-statistic, written as the sum of independent random variables, which is not possible here. Finally, notice that, although one cannot guarantee that $\mathbb{E}[V_n(h)] = \theta(h)$, deviation probabilities may still be easily obtained via the comparison to the corresponding U-statistic.

Proposition 6.5. *Let $d \in \mathbb{N}^*$, $\{Z_i\}_{i\leq n}$ be $n \geq d$ independent realizations of a $\mathcal{Z}$-valued random variable Z, and $h : \mathcal{Z}^d \to \mathbb{R}$ bounded such that $\mathbb{E}[h(Z_1, \dots, Z_d)] = \theta(h)$. Then, with $V_n(h)$ as in Definition 6.4, it holds for any $t > 0$:*

$$\mathbb{P}\left\{\left|V_n(h) - \theta(h)\right| > t + \frac{d(d-1)\|h\|_\infty}{n}\right\} \leq 2\exp\left(-\frac{nt^2}{2d\|h\|_\infty^2}\right).$$

Proof. First notice that $n|U_n(h) - V_n(h)| \leq d(d-1)\|h\|_\infty$. Indeed, let $W_n(h)$ denote the average of all terms $h(Z_{i_1}, \dots, Z_{i_d})$ with equality $i_j = i_k$ for at least one pair $j \neq k$. One has

$$\begin{aligned}
\sum_{i_1=1}^{n} \dots \sum_{i_d=1}^{n} h(Z_{i_1}, \dots Z_{i_d}) &= \sum_{\substack{i_1,\dots i_d \\ j\neq k \,\Rightarrow\, i_j \neq i_k}} h(Z_{i_1}, \dots Z_{i_d}) + \sum_{\substack{i_1,\dots i_d \\ \exists j\neq k,\ i_j = i_k}} h(Z_{i_1}, \dots Z_{i_d}), \\
n^d V_n(h) &= \frac{n!}{(n-d)!} U_n(h) + \left(n^d - \frac{n!}{(n-d)!}\right) W_n(h),
\end{aligned}$$

so that it holds

$$n^d(U_n(h) - V_n(h)) = \left(n^d - \frac{n!}{(n-d)!}\right)(U_n(h) - W_n(h)).$$

One may easily show by recurrence that $n^d - n!/(n-d)! = n^d - n(n-1)\dots(n-d+1)$ is positive and smaller than $d(d-1)/2\ n^{d-1}$ for all $n \geq 1$, which gives:

$$n|U_n(h) - V_n(h)| \leq d(d-1)\|h\|_\infty.$$

We point out that this bound is essentially tight as $(1/n^{d-1})(n^d - n(n-1)\dots(n-d+1))$ tends to $d(d-1)/2$ as n goes to infinity.

Using Proposition 6.3, and one can now bound the deviation probability.

$$\mathbb{P}\left\{\left|V_n(h)-\theta(h)\right| > t + \frac{d(d-1)\|h\|\infty}{n}\right\} \leq 2\exp\left(-\frac{nt^2}{2d\|h\|_\infty^2}\right),$$

which concludes the proof. □

Thus, despite the possible replications and the bias of $V_n(h)$, it is still possible to control its deviations around $\theta(h)$.

Further references about U-statistics include Hoeffding (1948) and Hoeffding (1963) for his seminal works, Hajek (1968); Grams et al. (1973) for a study and use of projection techniques, van der Vaart (1998) (Chapter 12 therein) for an excellent introduction, de la Peña and Giné (1999) for the introduction of decoupling arguments, or Lee (1990) as a general account of properties and asymptotic theory. Giné et al. (2000) provides a more recent development on moment inequalities for U-statistics, while complements about V-statistics may be found in Serfling (1980).

6.5 Conclusion

The next chapter deals with robust mean estimators relying on the *Median-of-Means* (MoM) principle. Originally developed for standard means, this procedure is further extended to U-statistics. But the use of U-statistics in Chapter 7 cannot be limited to this adaptation. Indeed, as shall be seen in particular in Sections 7.2 and 7.4, the analyses of randomized versions of MoM-like estimators crucially rely on U-statistics and their remarkable concentration properties.

7

Robust Mean Estimators

Contents

As already mentioned, in the ERM paradigm, one substitutes the (intractable) problem

$$\min_{h \text{ measurable}} \quad \mathbb{E}_{Z\sim P}\Big[\ell(h,Z)\Big] \qquad \text{to} \qquad \min_{h\in\mathcal{H}} \quad \frac{1}{n}\sum_{i=1}^{n}\ell(h,Z_i),$$

implicitly assuming that the empirical mean is a *good* estimate of the expectation. While this assumption seems sounded in presence of sub-Gaussian data, it is known to fail in heavy tailed scenarios. A natural question that arises next is: do mean estimates that overcome this difficulty exist? The answer is affirmative, and it is precisely the purpose of this chapter to introduce and analyze several of such robust mean estimators.

Estimators presented here are based on the *Median-of-Means* principle (Nemirovsky and Yudin, 1983), which is recalled in Section 7.1. Extensions based on randomizations (Section 7.1), tailored to U-statistics (Section 7.3), or both at the same time (refer to Section 7.4) are then detailed. They all come from the following publication:

► **P. Laforgue**, S. Clémençon, P. Bertail. On medians of (Randomized) pairwise means. In *Proceedings of International Conference on Machine Learning*, 2019.

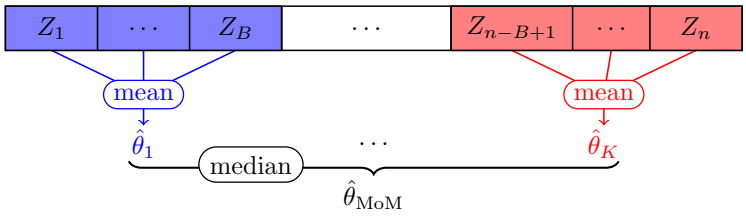


Figure 7.1 – MoM's procedure

7.1 The Median-of-Means Estimator

The first estimator we present is the Median-of-Means (MoM). It is the building block for all estimators presented afterwards: in Section 7.2, we analyze a randomized version of the MoM, while Sections 7.3 and 7.4 are dedicated to versions tailored to U-statistics, respectively standard and randomized.

7.1.1 Definition

The Median-of-Means (MoM) estimator is a mean estimator introduced independently in convex optimization by Nemirovsky and Yudin (1983), in computational complexity theory by Jerrum et al. (1986), or for memory-efficient frequency moments estimation by Alon et al. (1999). The construction of the MoM estimator starts by designing independent weakly concentrated estimators of the mean, say by computing empirical means on disjoint groups of observations. The second step consists in taking the median of the previous estimators, leading to a robust estimate. Formally, recall that given a collection of real numbers $\{Z_i\}_{i\leq n} \in \mathbb{R}^n$, its empirical mean is defined as

$$\frac{1}{n}\sum_{i=1}^{n} Z_i,$$

while its median is

$$Z_{\sigma(\frac{n+1}{2})} \text{ if } n \text{ is odd, and } Z_{\sigma(\frac{n}{2})} \text{ otherwise,}$$

with σ a permutation of $\{1,\dots,n\}$ such that $Z_{\sigma(1)} \leq \dots \leq Z_{\sigma(n)}$. The MoM estimator is then defined as follows.

Definition 7.1. *Let $\mathcal{S}_n = \{Z_i\}_{i\leq n}$ be a sample of n independent realizations of a real-valued random variable Z. Let $K \leq n$, and partition $\mathcal{S}_n$ into K blocks $(B_k)_{k\leq K}$ of size $B = \lfloor n/K \rfloor$ (the possible $K-1$ remaining observations may be ignored). For $k \leq K$, let $\hat{\theta}_k$ denote the empirical mean over B_k. Namely, $\hat{\theta}_k = \frac{1}{B}\sum_{i\in B_k} Z_i$. The Medians-of-Means (MoM) estimator is then given by*

$$\hat{\theta}_{\text{MoM}} = \text{median}(\hat{\theta}_1,\dots,\hat{\theta}_K).$$

One may find in Figure 7.1 a visual representation of the MoM's building procedure. It will notably be useful to draw comparisons with the randomized versions we introduce in future sections (see Figures 7.2 to 7.6).

Despite this conceptual simplicity, the MoM estimator exhibits strong concentration properties, even for heavy-tailed random variables. In particular, it compares very favorably to the empirical mean, that requires much stronger assumptions to reach a similar sub-Gaussian behavior.

7.1.2 Concentration Properties

The MoM estimator has witnessed a particular resurgence of interest since the seminal works by Audibert and Catoni (2011) and Catoni (2012). Indeed, the general idea of these works is to analyze mean estimators through their deviation probabilities – rather than via the mean squared error – giving the MoM a central role: leveraging the strong concentration properties of the median, it achieves a sub-Gaussian behavior, even for of heavy-tailed random variables, while the empirical mean typically necessitates bounded or sub-Gaussian data. The counterpart to this efficiency is the careful choice of the number of blocks, which plays a crucial role in the performance, and may depend on the targeted confidence. The good concentration properties of the MoM are now to be highlighted through their comparison to that of the empirical mean. We start by recalling Hoeffding's inequality, which ensures the empirical mean to be sub-Gaussian if the observations are bounded.

Lemma 7.2. *(Hoeffding's Lemma, Hoeffding (1963)) Let Z be a centered real random variable such that there exist $a, b \in \mathbb{R}^2$ such that $a < Z < b$. Then, for any $\lambda > 0$, the following holds*

$$\mathbb{E}\left[e^{\lambda Z}\right] \leq e^{\frac{\lambda^2 (b-a)^2}{8}}.$$

Theorem 7.3. *(Hoeffding's Inequality, Hoeffding (1963)) Let $\{Z_i\}_{i \leq n}$ be n independent realizations of a real random variable Z with expectation θ and such that there exist $a, b \in \mathbb{R}^2$ such that $a < Z < b$. Then, for any $\delta > 0$ it holds*

$$\mathbb{P}\left\{\left|\frac{1}{n}\sum_{i=1}^{n} Z_i - \theta\right| > (b-a)\sqrt{\frac{\ln(2/\delta)}{2n}}\right\} \leq \delta.$$

Proof.

$$\begin{aligned}
\mathbb{P}\left\{\sum_{i=1}^{n}(Z_i - \theta) > t\right\} &= \mathbb{P}\left\{e^{\lambda \sum_{i=1}^{n}(Z_i-\theta)} > e^{\lambda t}\right\} \quad \text{for any } \lambda > 0, \\
&\leq e^{-\lambda t}\, \mathbb{E}\left[\prod_{i=1}^{n} e^{\lambda(Z_i-\theta)}\right], \\
&\leq e^{-\lambda t + \frac{n\lambda^2 (b-a)^2}{8}}, \\
\mathbb{P}\left\{\sum_{i=1}^{n}(Z_i - \theta) > t\right\} &\leq e^{-\frac{2t^2}{n(b-a)^2}}, \\
\mathbb{P}\left\{\left|\frac{1}{n}\sum_{i=1}^{n} Z_i - \theta\right| > t\right\} &\leq 2e^{-\frac{2nt^2}{(b-a)^2}},
\end{aligned}$$

where we have used successively Markov's inequality, the independence of the Z_i's, Lemma 7.2, and optimizing the bound for $\lambda > 0$, attained in $\lambda = 4t/(n(b-a)^2) > 0$. Reverting the bound leads to the desired result. □

Remark 7.4. *In its full generality, Hoeffding's inequality does not need the observations to be i.i.d., but only independent. It leads to a more general result involving the bounds of each random variable Z_i. However, this general version still requires the boundedness of the random variables, which is precisely the point we want to emphasize.*

We continue by relaxing the boundedness assumption, and consider now sub-Gaussian random variables. Guarantees almost identical to that of Theorem 7.3 may be derived very easily, as one only needs to apply the definition of sub-Gaussianity (Definition 7.5) instead of Lemma 7.2.

Definition 7.5. *A real random variable Z is said to be σ sub-Gaussian if it satisfies*

$$\forall \lambda > 0, \quad \mathbb{E}\left[e^{\lambda(Z-\mathbb{E}[Z])}\right] \leq e^{\sigma^2\lambda^2/2}.$$

Remark 7.6. *Notice that thanks to Lemma 7.2, every bounded real random variable Z such that $|Z| \leq M$ is also M sub-Gaussian.*

Proposition 7.7. *Let $\{Z_i\}_{i\leq n}$ be n independent realizations of a random variable Z σ sub-Gaussian and with expectation θ. Then, for any $\delta > 0$ it holds*

$$\mathbb{P}\left\{\left|\frac{1}{n}\sum_{i=1}^{n} Z_i - \theta\right| > \sigma\sqrt{\frac{2\ln(2/\delta)}{n}}\right\} \leq \delta.$$

Proof. The proof is identical to that of Theorem 7.3, except that Definition 7.5 is used instead of Lemma 7.2. □

We have recalled these basic tools in order to highlight the high concentration capacities of the MoM estimator, that, unlike the empirical mean, does not require bounded data (such as in Theorem 7.3) or sub-Gaussian data (like for Proposition 7.7), but only a finite second order moment to concentrate nicely around the expectation. This behavior is explicited by the following proposition.

Proposition 7.8. *Let $\{Z_i\}_{i\leq n}$ be n independent realizations of a real random variable Z with expectation θ and finite variance σ^2. Then, for any $\delta \in [e^{1-2n/9}, 1[$, choosing $K = \left\lceil \frac{9}{2}\ln(1/\delta)\right\rceil$, it holds*

$$\mathbb{P}\left\{\left|\hat{\theta}_{\text{MoM}} - \theta\right| > 3\sqrt{6}\sigma\sqrt{\frac{1+\ln(1/\delta)}{n}}\right\} \leq \delta.$$

Proof. Let $t > 0$, and $\hat{I}_{k,t} = \mathbb{1}\{|\hat{\theta}_k - \theta| > t\}$ for $k \leq K$. Observe that the $\hat{I}_{k,t}$ are i.i.d. Bernoulli variables with same parameter $\hat{p}_t = \mathbb{P}\{|\hat{\theta}_k - \theta| > t\} \leq \text{Var}(\hat{\theta}_k)/t^2 \leq \sigma^2/(Bt^2)$. In addition, observe that

$$\left\{\left|\hat{\theta}_{\text{MoM}} - \theta\right| > t\right\} \subset \left\{\sum_{k=1}^{K} \hat{I}_{k,t} \geq \frac{K}{2}\right\},$$

such that, applying Hoeffding's inequality to the bounded $\hat{I}_{k,t}$, one gets

$$\begin{aligned}
\mathbb{P}\left\{\left|\hat{\theta}_{\text{MoM}}-\theta\right|>t\right\} &\leq \mathbb{P}\left\{\frac{1}{K}\sum_{k=1}^{K}\hat{I}_{k,t}\geq\frac{1}{2}\right\},\\
&\leq \mathbb{P}\left\{\frac{1}{K}\sum_{k=1}^{K}\hat{I}_{k,t}-\hat{p}_t\geq\frac{1}{2}-\frac{\sigma^2}{Bt^2}\right\},\\
\mathbb{P}\left\{\left|\hat{\theta}_{\text{MoM}}-\theta\right|>t\right\} &\leq \exp\left(-2K\left(\frac{1}{2}-\frac{\sigma^2}{Bt^2}\right)^2\right).
\end{aligned}$$

At this point, ignoring that K and B should be integers, choosing $K=\frac{\ln(1/\delta)}{2(\frac{1}{2}-\lambda)^2}$, $\lambda<1/2$, makes the upper bound equal to δ, as long as $\frac{\sigma^2}{Bt^2}=\lambda$. Reverting in t gives

$$t=\sigma\sqrt{\frac{1}{\lambda B}}=\sigma\sqrt{\frac{K}{\lambda n}}=\sigma\sqrt{\frac{1}{2\lambda(\frac{1}{2}-\lambda)^2}\,\frac{\ln(1/\delta)}{n}}.$$

Optimizing in λ leads to an optimal rate of $3\sqrt{3}\sigma\sqrt{\frac{\ln(1/\delta)}{n}}$, attained in $\lambda=1/6$, for $K=\frac{9}{2}\ln(1/\delta)$. Now, using $K=\left\lceil\frac{9}{2}\ln(1/\delta)\right\rceil$ preserves the inequality in δ, while taking $B=\left\lfloor\frac{n}{K}\right\rfloor\geq\frac{n}{2K}\geq\frac{n}{2\left\lceil\frac{9}{2}\ln(1/\delta)\right\rceil}\geq\frac{n}{9(1+\ln(1/\delta))}$ leads to the slightly modified final result:

$$\mathbb{P}\left\{\left|\hat{\theta}_{\text{MoM}}-\theta\right|>3\sqrt{6}\sigma\sqrt{\frac{1+\ln(1/\delta)}{n}}\right\}\leq\delta.$$

□

Remark 7.9. *As previously evoked, the counterpart to such sub-Gaussian behavior is the careful choice of K. It should be of the order $\ln(1/\delta)$, inducing that the estimator changes with the confidence targeted. The MoM is a so called δ-*dependent estimator*. This remark has therefore a consequence on the range of confidences achievable, since we need $K=\left\lceil\frac{9}{2}\ln(1/\delta)\right\rceil\leq n$. This phenomenon is shared by all MoM-based estimators presented in this manuscript.*

Remark 7.10. *Notice that the optimization in λ is restricted to the interval $]0,1/2[$, so that the use of Hoeffding's inequality in the first part of the proof is always permitted.*

Remark 7.11. *The proof uses Hoeffding's inequality to bound the deviation of the sum of the indicator random variables $\hat{I}_{k,t}$, as in Hsu and Sabato (2016), but using the Binomial law as in Devroye et al. (2016) would also have been possible. Changes only occur on constants: the $3\sqrt{6}$ derived here is smaller than the $2\sqrt{2}e$ of Devroye et al. (2016), but larger than the 6 of Hsu and Sabato (2016). It comes at the price of a number $\left\lceil\frac{9}{2}\ln(1/\delta)\right\rceil$ of blocks needed, to be compared to the $\left\lceil\ln(1/\delta)\right\rceil$ of Devroye et al. (2016). Hsu and Sabato (2016) are able to exhibit a lower constant only thanks to the extra assumption that $K\leq n/4$, which naturally has also an impact on the range of achievable confidences δ.*

Remark 7.12. *The first inequality used* $\{|\hat{\theta}_{\mathrm{MoM}} - \theta| > t\} \subset \{\sum_k \hat{I}_{k,t} \geq \frac{K}{2}\}$ *may seem a bit rough. However, improving on this inclusion does not lead to significant gains in the final bound. One may use instead*

$$\left\{\left|\hat{\theta}_{\mathrm{MoM}} - \theta\right| > t\right\} = \left\{\sum_{k=1}^{K} \mathbb{1}\{\hat{\theta}_k - \theta > t\} \geq \frac{K}{2}\right\} \cup \left\{\sum_{k=1}^{K} \mathbb{1}\{\hat{\theta}_k - \theta < -t\} \geq \frac{K}{2}\right\}.$$

After applying the union bound, one can only hope to improve on the constant factor of Proposition 7.8, at the price of an extra assumption ensuring the deviations to be symmetrical for instance. Thus, it is not of great interest to try refining this step.

One may possibly argue that Proposition 7.8 only addresses real random variables, whereas concentration properties for the mean easily extend to the multivariate case. However, the MoM has also been extended to the multidimensional setting, while preserving the nature of the guarantees that can be derived.

7.1.3 Extension to Multidimensional Random Variables

Generalizing the concept of median to the multidimensional setting does not admit a single answer, nor does the extension of the MoM. For instance, Hsu and Sabato (2016) and Joly et al. (2017); Lugosi and Mendelson (2017) introduce alternatives to the scalar median, either based on growing balls intersection or pairwise distance comparisons, to extend the MoM to any metric space. Another natural approach is that of Minsker et al. (2015), that uses the geometric median in Banach spaces. We now detail some of their results, starting by recalling the definition of the geometric median.

Definition 7.13. *Let* $\mathcal{Z}$ *be a Banach space with norm* $\|\cdot\|$*, and let* μ *be a probability measure on* $(\mathcal{Z}, \|\cdot\|)$*. The geometric median of* μ *is given by*

$$z_{\mathrm{med}} = \operatorname*{argmin}_{y \in \mathcal{Z}} \int_{\mathcal{Z}} \Big(\|y - z\| - \|z\|\Big)\, d\mu(z).$$

Henceforth, the *geometric MoM* will refer to the multivariate mean estimator that is built exactly as the standard MoM, except that a geometric median is used instead of the standard scalar one. The rest of the notation remains unchanged.

Using geometrical arguments (see Lemma 2.1 in Minsker et al. (2015)), it can be shown that the deviations of the geometric MoM can be controlled by the study of the indicator variables that correspond to the individual deviations, just as in the scalar case. This dependence is explicited in the following proposition.

Proposition 7.14. *(Theorem 3.1 in Minsker et al. (2015)) For* $0 < p < \alpha < 1/2$*, define* $C_\alpha = (1-\alpha)\sqrt{\frac{1}{1-2\alpha}}$*, and* $\psi(\alpha, p) = (1-\alpha)\ln\frac{1-\alpha}{1-p} + \alpha \ln\frac{\alpha}{p}$*. Let* $t > 0$ *such that for all* $k \leq K$ *it holds*

$$\mathbb{P}\Big\{\|\hat{\theta}_k - \theta\| > t\Big\} \leq p.$$

Then one has

$$\mathbb{P}\Big\{\|\hat{\theta}_{\mathrm{MoM}} - \theta\| > C_\alpha t\Big\} \leq e^{-K\psi(\alpha,p)}.$$

Proof. Using successively Lemma 2.1 in Minsker et al. (2015), the introduction of a binomial random variable $W \sim B(K, p)$ and Chernoff bound, one gets

$$\mathbb{P}\left\{\|\hat{\theta}_{\text{MoM}} - \theta\| > C_\alpha t\right\} \leq \mathbb{P}\left\{\sum_{k=1}^{K} \mathbb{1}\{\|\hat{\theta}_k - \theta\| > t\} > \alpha K\right\} \leq \mathbb{P}\left\{W > \alpha K\right\} \leq e^{-K\psi(\alpha,p)}.$$

□

This result directly leads to a concentration inequality for the geometric MoM.

Corollary 7.15. *(Corollary 4.1 in Minsker et al. (2015)) Let $\mathcal{Z}$ be a separable Hilbert space with norm $\|\cdot\|$. Let $\{Z_i\}_{i\leq n}$ be n independent realizations of a $\mathcal{Z}$-valued random variable Z with expectation θ and covariance operator $\Sigma = \mathbb{E}[(Z-\theta)\otimes(Z-\theta)]$ of finite trace. Set $\alpha^* = 7/18$, and $p^* = 0.1$. Then, for any $\delta \in]0,1[$ such that $K = \left\lfloor \frac{\ln(1/\delta)}{\psi(\alpha^*,p^*)} \right\rfloor + 1$ is lower than $n/2$, it holds*

$$\mathbb{P}\left\{\left\|\hat{\theta}_{\text{MoM}} - \theta\right\| > 11\sqrt{\frac{\mathbf{Tr}(\Sigma)\ln(1.4/\delta)}{n}}\right\} \leq \delta.$$

As for computational aspects, we underline that many recent contributions focus on downscaling the computational cost of MoMs in high dimension. Among them, Hopkins (2018), for instance, proposes an algorithmic approach to compute multivariate MoMs in polynomial time.

So far, all results about the MoM estimator, whether scalar or multivariate, are based on the assumption that the weakly concentrated estimates on which is applied the median are independent. It is the purpose of the following section to investigate a relaxed randomized version of the MoM that does not rely anymore on this independence assumption, while preserving strong guarantees. From now on, all estimates described are taken from **Laforgue** et al. (2019b).

7.2 The Median-of-Randomized-Means Estimator

Randomization is a classical alternative to data segmentation in many situations. For instance, it can be used for model selection as a substitute to cross-validation, or to compute estimator stability via bootstrap aggregation. In this subsection, we explore randomization, instead of the initial segmentation, for the MoM estimator, leading to the novel *Median-of-Randomized-Means* (MoRM) estimator.

7.2.1 Definition and Motivations

Intuitively, the estimator is built exactly the same way as the MoM, except that the blocks on which the intermediate empirical means are computed are no longer a partition of the original dataset, but rather randomly sampled. For each block, a subset of constant size B (to be explicited afterwards) is sampled, without replacement so that one observation cannot be present twice in a block. This *Sampling Without Replacement* will be referred to as SWoR thereafter. As for the different blocks, they are sampled independently and with replacement, so that one observation may pertain to different blocks, jeopardizing the independence assumption. Formally, each random block $\mathcal{B}_k$

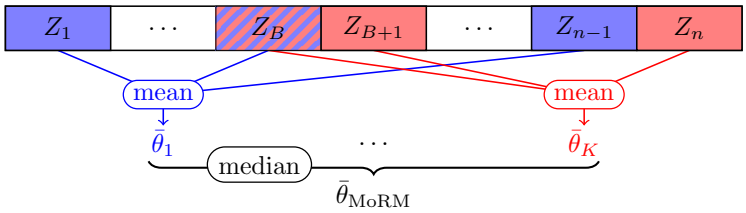


Figure 7.2 – MoRM's procedure

is fully characterized by a random vector $\boldsymbol{\epsilon}_k = (\epsilon_{k,1}, \ldots, \epsilon_{k,n}) \in \{0,1\}^n$ such that $\epsilon_{k,i}$ is equal to 1 if observation i is selected in $\mathcal{B}_k$, and to 0 otherwise. The $\boldsymbol{\epsilon}_k$'s are i.i.d. random vectors, uniformly distributed on the set $\Lambda_{n,B} = \left\{\boldsymbol{\epsilon} \in \{0,1\}^n : \mathbf{1}^\top \boldsymbol{\epsilon} = B\right\}$, of cardinality $\binom{n}{B}$.

We now give a formal definition of the randomized version of the new mean estimate.

Definition 7.16. *Let $\mathcal{S}_n = \{Z_i\}_{i \leq n}$ be a sample of n independent realizations of a real-valued random variable Z. Let $K \in \mathbb{N}^*$, $B \leq n$, and $(\mathcal{B}_k)_{k \leq K}$ be K blocks of size B, sampled independently from $\mathcal{S}_n$ by SWoR. For $k \leq K$, let $\bar{\theta}_k$ denote the empirical mean over $\mathcal{B}_k$. Namely, $\bar{\theta}_k = \frac{1}{B}\sum_{i \in \mathcal{B}_k} Z_i = \frac{1}{B}\sum_{i=1}^n \epsilon_{k,i} Z_i$, with the $\boldsymbol{\epsilon}_k$'s drawn i.i.d. and uniformly over $\Lambda_{n,B}$. The Median-of-Randomized-Means (MoRM) estimator is then given by*

$$\bar{\theta}_{\text{MoRM}} = \text{median}(\bar{\theta}_1, \ldots, \bar{\theta}_K).$$

The estimator's construction is also depicted in Figure 7.2, which should be considered together with Figure 7.1 in order to perceive the similarities and differences with the standard MoM estimator.

Although clearer after examination of further sections, some motivations for such a randomization may nevertheless be exposed as of now.

- First of all, the MoRM procedure induces a very flexible framework, in which the number of blocks K may not be limited by n, nor observations set aside because n is not divided by K.
- Empirically, when performing a MoM Gradient Descent (MoM GD, Section 8.4), it is often needed to shuffle the blocks at each step in order to avoid local minima (see e.g. Remark 5 in Lecué et al. (2018)). While this shuffling may seem artificial and "ad hoc" within a standard MoM GD, it is already included and controlled in a MoRM GD.
- Finally, with in mind the extension of the *Median-of-U-Statistics* (Section 7.3) to the *Median-of-Incomplete-U-Statistics* (see Section 7.4.3), which should be of particular interest thanks to the reduced variance property, it is first needed to exhibit guarantees on the randomization of the simple MoM.

As previously underlined, the counterpart to the proposed randomization is that the small intermediate estimators $\bar{\theta}_k$'s are no longer independent. Despite this relaxation, concentration inequalities for the MoRM estimator can be derived, as revealed by the following subsection.

7.2.2 Concentration Properties

Without independence of the $\bar{\theta}_k$'s, on which Proposition 7.8's proof crucially relies, it is naturally more involved to derive a concentration inequality for the MoRM. However, one can leverage the independence of the randomized blocks conditionally to the data, as well as the specificity of the SWoR to derive guarantees similar to that of the MoM.

Proposition 7.17. *Let $\mathcal{S}_n = \{Z_i\}_{i\leq n}$ be a sample of n independent realizations of a real random variable Z with expectation θ and finite variance σ^2. Then, for any $\tau \in]0, 1/2[$, for any $\delta \in [2e^{-4\tau^2 n/9}, 1[$, choosing $K = \left\lceil \frac{\ln(2/\delta)}{2(\frac{1}{2}-\tau)^2} \right\rceil$ and $B = \left\lfloor \frac{8\tau^2 n}{9\ln(2/\delta)} \right\rfloor$, it holds*

$$\mathbb{P}\left\{ \left| \bar{\theta}_{\mathrm{MoRM}} - \theta \right| > \frac{3\sqrt{3}\ \sigma}{2\ \tau^{3/2}} \sqrt{\frac{\ln(2/\delta)}{n}} \right\} \leq \delta.$$

Proof. Let $t > 0$, and $\bar{I}_{\boldsymbol{\epsilon}_k,t} = \mathbb{1}\{|\bar{\theta}_k - \theta| > t\}$ for $k \leq K$. Notice that the $\bar{I}_{\boldsymbol{\epsilon}_k,t}$'s have two sources of randomness: one from the data $\mathcal{S}_n$, and one from the randomness of the block, materialized by $\boldsymbol{\epsilon}_k$. Just like in the classic argument used to prove Proposition 7.8, it holds

$$\left\{ \left| \bar{\theta}_{\mathrm{MoRM}} - \theta \right| > t \right\} \subset \left\{ \sum_{k=1}^{K} \bar{I}_{\boldsymbol{\epsilon}_k,t} \geq \frac{K}{2} \right\}.$$

In order to benefit from the conditional independence of the blocks given the original sample $\mathcal{S}_n$, we may condition upon $\mathcal{S}_n$ and consider the variability induced by the $\boldsymbol{\epsilon}_k$'s only. Then, the global deviation of the average $\frac{1}{K}\sum_{k=1}^{K} \bar{I}_{\boldsymbol{\epsilon}_k,t}$ may be decomposed into

1) its deviation, solely due to the $\boldsymbol{\epsilon}_k$'s, from its conditional expectation

$$\bar{U}_{n,t} = \mathbb{E}_{\boldsymbol{\epsilon}} \left[\frac{1}{K} \sum_{k=1}^{K} \bar{I}_{\boldsymbol{\epsilon}_k,t} \;\middle|\; \mathcal{S}_n \right],$$

2) the deviation of this conditional expectation $\bar{U}_{n,t}$ from the overall expectation

$$\bar{p}_t = \mathbb{E}[\bar{U}_{n,t}] = \mathbb{E}_{\mathcal{S}_n}\Big[\mathbb{E}[\bar{I}_{1,t} \mid \mathcal{S}_n]\Big] = \mathbb{E}[\bar{I}_{1,t}] = \mathbb{P}\Big\{|\bar{\theta}_1 - \theta| > t\Big\}.$$

We have $\forall \tau \in]0, 1/2[$:

$$\begin{aligned}
\mathbb{P}\left\{ \left| \bar{\theta}_{\mathrm{MoRM}} - \theta \right| > t \right\} &\leq \mathbb{P}\left\{ \frac{1}{K}\sum_{k=1}^{K} \bar{I}_{\boldsymbol{\epsilon}_k,t} \geq \frac{1}{2} \right\}, \\
&\leq \mathbb{P}\left\{ \frac{1}{K}\sum_{k=1}^{K} \bar{I}_{\boldsymbol{\epsilon}_k,t} - \bar{U}_{n,t} + \bar{U}_{n,t} - \bar{p}_t \geq \frac{1}{2} - \bar{p}_t + \tau - \tau \right\}, \\
&\leq \mathbb{P}\left\{ \frac{1}{K}\sum_{k=1}^{K} \bar{I}_{\boldsymbol{\epsilon}_k,t} - \bar{U}_{n,t} \geq \frac{1}{2} - \tau \right\} + \mathbb{P}\left\{ \bar{U}_{n,t} - \bar{p}_t \geq \tau - \bar{p}_t \right\}.
\end{aligned}$$

As announced, we deal with the first term of last inequation's right side by conditioning upon $\mathcal{S}_n$, and using Hoeffding's inequality for i.i.d. averages:

$$\begin{aligned}\mathbb{P}\left\{\frac{1}{K}\sum_{k=1}^{K}\bar{I}_{\boldsymbol{\epsilon}_k,t}-\bar{U}_{n,t}\geq\frac{1}{2}-\tau\right\} &\leq \mathbb{E}_{\mathcal{S}_n}\left[\mathbb{P}_{\boldsymbol{\epsilon}}\left\{\frac{1}{K}\sum_{k=1}^{K}\bar{I}_{k,t}-\bar{U}_{n,t}\geq\frac{1}{2}-\tau\;\middle|\;\mathcal{S}_n\right\}\right],\\ &\leq \mathbb{E}_{\mathcal{S}_n}\left[\exp\left(-2K\left(\frac{1}{2}-\tau\right)^2\right)\right],\\ &\leq \exp\left(-2K\exp\left(\frac{1}{2}-\tau\right)^2\right).\end{aligned}$$

As for the second term, one may observe that

$$\bar{U}_{n,t} \;=\; \frac{1}{\binom{n}{B}}\sum_{\boldsymbol{\epsilon}\in\Lambda(n,B)}\bar{I}_{\boldsymbol{\epsilon},t} \;=\; \frac{1}{\binom{n}{B}}\sum_{I}\mathbb{1}\left\{\left|\frac{1}{B}\sum_{j=1}^{B}X_{I_j}-\theta\right|>t\right\},$$

where the symbol $\sum_I$ refers to the summation over all unordered subsets I of B integers chosen in $\{1,\ldots,n\}$. $\bar{U}_{n,t}$ is therefore a U-statistic of degree B (see Definition 6.1), with symmetric kernel $h(Z_1,\ldots,Z_B) = \mathbb{1}\{|(1/B)\sum_{j=1}^{B}X_j-\theta|>t\}$. This second term being independent from the $\boldsymbol{\epsilon}_k$'s, we can use Hoeffding's inequality for U-statistics (Proposition 6.3), together with the fact that $\bar{p}_t \leq \frac{\sigma^2}{Bt^2}$ (cf Appendix B.1). It yields

$$\begin{aligned}\mathbb{P}_{\mathcal{S}_n}\left\{\bar{U}_{n,t}-\bar{p}_t\geq\tau-\bar{p}_t\right\} &\leq \mathbb{P}_{\mathcal{S}_n}\left\{\bar{U}_{n,t}-\bar{p}_t\geq\tau-\frac{\sigma^2}{Bt^2}\right\},\\ &\leq \exp\left(-2\frac{n}{B}\left(\tau-\frac{\sigma^2}{Bt^2}\right)^2\right).\end{aligned}$$

Finally, we get

$$\mathbb{P}\left\{\left|\bar{\theta}_{\mathrm{MoRM}}-\theta\right|>t\right\}\leq\exp\left(-2K\left(\frac{1}{2}-\tau\right)^2\right)+\exp\left(-2\frac{n}{B}\left(\tau-\frac{\sigma^2}{Bt^2}\right)^2\right).$$

Choosing $K=\left\lceil\frac{\ln(2/\delta)}{2(\frac{1}{2}-\tau)^2}\right\rceil$ ensures that the first term is lower than $\delta/2$. Ignoring that B should be an integer, choosing $B=\frac{2n(\tau-\lambda)^2}{\ln(2/\delta)}$, $\lambda<\tau$, makes the second term equal to $\delta/2$ as long as $\frac{\sigma^2}{Bt^2}=\lambda$. Reverting in t gives

$$t=\sigma\sqrt{\frac{1}{\lambda B}}=\sigma\sqrt{\frac{1}{2\lambda(\tau-\lambda)^2}\,\frac{\ln(2/\delta)}{n}}.$$

Optimizing in λ leads to an optimal rate of $\frac{3\sqrt{3}}{2\sqrt{2}\tau^{3/2}}\sigma\sqrt{\frac{\ln(2/\delta)}{n}}$, attained in $\lambda=\tau/3$, for $B=\frac{8\tau^2 n}{9\ln(2/\delta)}$. Now, using $B=\left\lfloor\frac{8\tau^2 n}{9\ln(2/\delta)}\right\rfloor\geq\frac{4\tau^2 n}{9\ln(2/\delta)}$ preserves the inequality in $\delta/2$, while scaling t by $\sqrt{2}$, leading to the final result. □

At this point, a few remarks can be made in order to compare Propositions 7.8 and 7.17.

Remark 7.18. *First, notice that the number K of randomized blocks is completely arbitrary in the MoRM procedure and may even exceed n. Consequently, it is always possible to build the $\left\lceil \frac{\ln(2/\delta)}{2(\frac{1}{2}-\tau)^2} \right\rceil$ blocks prescribed, and there is no restriction on the acceptable range of confidences δ achievable due to K.*

Remark 7.19. *Then, the size B of the blocks can be chosen completely independently from K, inducing a slight improvement with $\ln(2/\delta)$ instead of $1+\ln(1/\delta)$ at the numerator. Notice that, as in Proposition 7.8, the optimization in λ is restricted to the interval $]0,\tau[$, so that the use of Hoeffding's inequality in the first part of the proof is always permitted. Observe also that B never exceeds n. Indeed for all $\tau \in]0,1/2[$, $\frac{8\tau^2}{9\ln(2/\delta)}$ does not exceeds 1 as long as δ is lower than $2\exp(-2/9) \approx 1.6$, which is always true. Still, B needs to be greater than 1, which results in a restriction on the acceptable range of confidences δ achievable, such as specified.*

Remark 7.20. *Third, the proposed bound involves an additional parameter τ, that can be arbitrarily chosen in $]0,1/2[$. As may be revealed by examination of the proof, the choice of this extra parameter reflects a trade-off between the deviations induced by $\boldsymbol{\epsilon}$ or by $\mathcal{S}_n$, that depends on K and B respectively. The larger τ, the larger K, the larger the confidence range, the larger B and the lower the constant factor. Since one can pick K arbitrarily large, and that B never exceeds n, τ should be chosen as large as possible in $]0,1/2[$. This way, one asymptotically achieves the $3\sqrt{6}$ constant factor of Proposition 7.8. However, the price of such an improvement is the construction of a higher number of blocks in practice. This seems sounded, as one needs more randomized ornithorynque blocks to* see *all observations, what a partition does by design. For a comparable number of blocks ($\tau = 1/6$), the constant in Proposition 7.17 becomes $27\sqrt{2}$.*

Remark 7.21. *Finally, about the term $\ln(2/\delta)$ that appears, instead of $\ln(1/\delta)$. It only comes out from a crude analysis during the proof. Indeed, K and B have been chosen so that both exponential terms are equal to $\delta/2$, but one could of course consider splitting the two terms into $(1-\kappa)\delta$ and $\kappa\delta$ for any $\kappa \in]0,1[$. This, way, choosing*

$$K = \left\lceil \frac{\ln\left(\frac{1}{(1-\kappa)\delta}\right)}{2(\frac{1}{2}-\tau)^2} \right\rceil \quad \text{and} \quad B = \left\lfloor \frac{8\tau^2 n}{9\ln\left(\frac{1}{\kappa\delta}\right)} \right\rfloor$$

leads to a $\ln(1/\kappa\delta)$ term instead. With the possibility to choose κ as close as possible to 1, one asymptotically recovers the $\ln(1/\delta)$ rate of Proposition 7.8.

Just as for MoM, let us now consider extensions of the framework we just developed. If the extension to multivariate random variables is quite easy, the use of the SWoR sampling seems to be crucial, and it cannot be replaced by any other sampling scheme.

7.2.3 Alternatives and Extensions

Guarantees being proven for the SWoR sampling, it is natural to wonder if other sampling schemes could lead to similar bounds. We discuss this alternative in the following subsection, as well as the possibility to apply MoRM to multivariate data.

Alternative Sampling Schemes

One possible alternative that naturally comes to mind is to use a Monte-Carlo sampling, which allows replacement within a block. However, the theoretical analysis of such a variant is much more challenging, as the conditional expectation takes the form of a V-statistic, instead of a U-statistic.

Let $\tilde{\theta}_k$ denote the empirical means obtained on the K Monte-Carlo samples of size B, $\tilde{I}_{k,t}$, $\tilde{U}_{n,t}$ and $\tilde{p}_t$ the counterparts of respectively $\bar{I}_{k,t}$, $\bar{U}_{n,t}$ and $\bar{p}_t$ introduced in the proof of Proposition 7.17. If $\mathbb{P}_{\boldsymbol{\epsilon}}\{\frac{1}{K}\sum_{k=1}^{K}\tilde{I}_{k,t} - \tilde{U}_{n,t} > \frac{1}{2} - \tau \mid \mathcal{S}_n\}$ can still be bounded using the conditional version of Hoeffding's inequality, $\mathbb{P}_{\mathcal{S}_n}\{\tilde{U}_{n,t} - \tilde{p}_t > \tau - \tilde{p}_t\}$ cannot be treated the same way as in Proposition 7.17, since $\tilde{U}_{n,t}$ is not anymore a U-statistic of degree B. More precisely,

$$\tilde{U}_{n,t} = \frac{1}{n^B}\sum_{i_1=1}^{n}\cdots\sum_{i_B=1}^{n}\mathbb{1}\left\{\left|\frac{1}{B}\sum_{j=1}^{B}Z_{i_j} - \theta\right| > t\right\}$$

is a V-statistic of degree B (see Section 6.4.2), with same kernel as $\bar{U}_{n,t}$. Proposition 6.5 actually encourages to study the deviation of $\tilde{U}_{n,t}$ with respect to $\bar{p}_t$ (the expected value of the associated U-statistic $\bar{U}_{n,t}$). But one can only achieve

$$\mathbb{P}_{\mathcal{S}_n}\left\{\tilde{U}_{n,t} - \bar{p}_t > \tau - \bar{p}_t\right\} \leq \exp\left(-\frac{n}{2B}\left(\tau - \frac{\sigma^2}{Bt^2}\right)^2\right) + \frac{B(B-1)}{n},$$

which is clearly not sufficient due to the second term.

The use of the bounded differences inequality (McDiarmid (1989), Appendix A) is not satisfactory neither. Indeed, considering $\tilde{U}_{n,t}$ as a function of the n i.i.d. variables $\{Z_i\}_{i\leq n}$, one has to bound $|\tilde{U}_{n,t} - \tilde{U}'_{n,t}|$, where $\tilde{U}'_{n,t}$ is the same quantity as $\tilde{U}_{n,t}$, but obtained on a sample $\mathcal{S}'_n$ in which only one observation differs from $\mathcal{S}_n$. Formally

$$\left|\tilde{U}_{n,t} - \tilde{U}'_{n,t}\right| = \left|\mathbb{P}\left\{|\tilde{\theta}_1 - \theta| > t \mid \mathcal{S}_n\right\} - \mathbb{P}\left\{|\tilde{\theta}'_1 - \theta| > t \mid \mathcal{S}'_n\right\}\right|.$$

If the observation that differs between $\mathcal{S}_n$ and $\mathcal{S}'_n$ is not drawn in the Monte-Carlo block, which happens with probability $\left(1 - \frac{1}{n}\right)^B$, the difference is null. Otherwise, it is strictly lower than 1. Finally, we get

$$\left|\tilde{U}_{n,t} - \tilde{U}'_{n,t}\right| \leq 1 - \left(1 - \frac{1}{n}\right)^B \leq \frac{B}{n},$$

and the bounded difference inequality yields

$$\mathbb{P}_{\mathcal{S}_n}\left\{\tilde{U}_{n,t} - \tilde{p}_t > \tau - \tilde{p}_t\right\} \leq \exp\left(-2\frac{n}{B^2}\left(\tau - \tilde{p}_t\right)^2\right).$$

However, the B^2 at the denominator, instead of B in the proof of Proposition 7.17 makes the bound unusable, not even considering that the Chebyshev upper-bound for $\tilde{p}_t$ is larger than that of $\bar{p}_t$ (the variance is naturally larger due to the allowance of possible replications within a block in the Monte-Carlo scheme).

The same phenomenon occurs for a Bernoulli sampling, in which each observation has a probability B/n to be drawn in a block, independently from the others, potentially leading to blocks of different sizes. $|\tilde{U}_{n,t} - \tilde{U}'_{n,t}|$ is still bounded by the probability of the differing observation to be in the block, which is directly B/n here.

Hence, although these variants have been shown experimentally to provide reasonable results (see Section 7.5), their theoretical analyses cannot be tackled using the same arguments as the ones employed for Proposition 7.17, and the SWoR seems to be the only sampling scheme that comes along with easily provable guarantees.

Extension to Multidimensional Random Variables

Another way to extend the MoRM estimator is to consider multidimensional random variables. Among approaches extending MoMs to random vectors (see Section 7.1.3), that of Minsker et al. (2015) could be readily adapted to MoRM. Indeed, once Lemma 2.1 therein has been applied, the estimator's deviation probability is bounded by the deviation probability of a sum of indicator random variables (proof of Proposition 7.14). This lemma being a consequence of the nature of the geometric median only, a geometric MoRM would also benefit from it. The rest of the proof can be handled exactly as for Proposition 7.17, leading to guarantees for the geometric MoRM.

Now that we have proved that guarantees may still be derived without independence between blocks, we can address problems where this question is all the more present. Namely, the estimation of U-statistics. Indeed, a U-statistic may depend on several variables, making the segmentation particularly harmful: once the partition is set, an observation cannot be used on another block, at any of the kernel entry, multiplying the partitioning damage by roughly the degree of the U-statistic.

7.3 The Median-of-U-Statistics Estimator

As shall be seen in this section, the MoM procedure extends nicely and very naturally to the problem of estimating U-statistics. Like for standard means, it yields estimators that do not require the kernel to be bounded (see Proposition 6.3) to exhibit strong concentration properties. Notice that for the sake of simplicity, we first restrict ourselves to U-statistics of degree 2, general statements for U-statistics of arbitrary degree being deferred to Section 7.3.3.

7.3.1 Definition

Rather than the mean of an integrable random variable, we assume now that the quantity of interest is of the form $\theta(h) = \mathbb{E}[h(Z_1, Z_2)]$, where Z_1 and Z_2 are i.i.d. random vectors with distribution $F(dz)$, and $h : \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}$ is a symmetric measurable mapping, square integrable with respect to $F \otimes F$. A direct adjustment of the MoM estimator consists in replacing the standard empirical means by pairwise means, *i.e.* U-statistics of degree 2, as detailed in Definition 7.22 and Figure 7.3.

Definition 7.22. *Let $\mathcal{S}_n = \{Z_i\}_{i \leq n}$ be a sample of n independent realizations of a $\mathcal{Z}$-valued random variable Z, and h as described in the previous paragraph. Let $K \leq n$, and partition $\mathcal{S}_n$ into K blocks $(B_k)_{k \leq K}$ of size $B = \lfloor n/K \rfloor$ (the possible $K - 1$ remaining observations may be ignored). For $k \leq K$, let $\hat{U}_k(h)$ denote the (complete) U-statistic built on B_k. Namely, $\hat{U}_k(h) = \frac{2}{B(B-1)} \sum_{\substack{i,j \in B_k^2 \\ i<j}} h(Z_i, Z_j)$. The Median-of-$U$-Statistics*

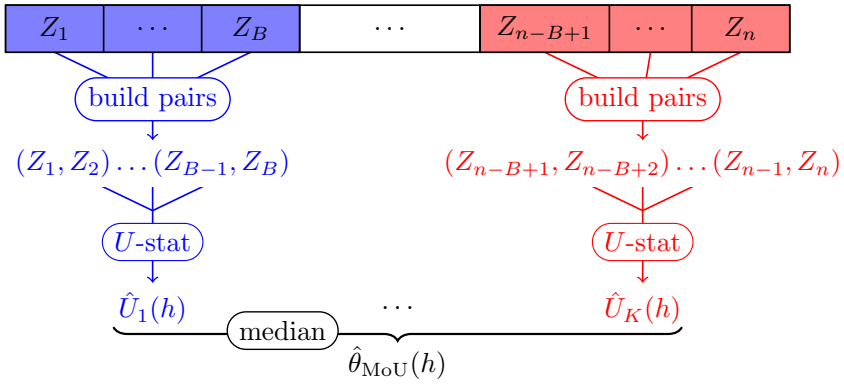


Figure 7.3 – MoU's procedure

(MoU) estimator is then given by

$$\hat{\theta}_{\mathrm{MoU}}(h) = \mathrm{median}(\hat{U}_1(h), \dots, \hat{U}_K(h)).$$

This estimator resembles a lot the standard MoM, and comparable guarantees may be established using the same proof techniques, as shall be detailed in the next subsection.

7.3.2 Concentration Properties

Just as for the standard MoM, strong concentration guarantees can be derived for the MoU estimator, under minimal assumptions (finite second order moment). When analyzing U-statistics, almost all quantities depend on the kernel h chosen. In order to avoid heavy notation, the dependence in h may be ignored in the subsequent analyses when it is clear from context.

Proposition 7.23. *Let $\{Z_i\}_{i\leq n}$ be n independent realizations of a $\mathcal{Z}$-valued random variable Z, and $h : \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}$ symmetric such that $\mathbb{E}[h(Z_1, Z_2)] = \theta(h) < +\infty$, and $\mathrm{Var}(h(Z_1, Z_2)) = \sigma^2(h) < +\infty$. Then, for any $\delta \in [e^{1-2n/27}, 1[$, with $K = \left\lceil \frac{9}{2} \ln(1/\delta) \right\rceil$, it holds*

$$\mathbb{P}\left\{ \left| \hat{\theta}_{\mathrm{MoU}}(h) - \theta(h) \right| > C_1(h) \sqrt{\frac{1 + \ln(1/\delta)}{n}} + C_2(h) \, \frac{1 + \ln(1/\delta)}{n} \right\} \leq \delta,$$

with $C_1(h) = 6\sqrt{6}\, \sigma_1(h)$ and $C_2(h) = 18\sqrt{3}\, \sigma_2(h)$, $\sigma_1(h)$ and $\sigma_2(h)$ being defined as in Equations (6.4) and (6.5) respectively.

Proof. The proof technique is very similar to that of previous propositions. For the K blocks, let $\hat{J}_{k,t} = \mathbb{1}\{|\hat{U}_k(h) - \theta(h)| > t\}$. Again, observe that $\mathbb{P}\left\{ \left| \hat{\theta}_{\mathrm{MoU}}(h) - \theta(h) \right| > t \right\}$ is lower than

$$\mathbb{P}\left\{ \frac{1}{K} \sum_{k=1}^{K} \hat{J}_{k,t} - \hat{q}_t \geq \frac{1}{2} - \hat{q}_t \right\},$$

where $\hat{q}_t = \mathbb{E}[\hat{J}_{1,t}] = \mathbb{P}\{|\hat{U}_1(h) - \theta(h)| > t\}$. By virtue of Chebyshev's inequality and Equation (6.7), one has

$$\hat{q}_t \leq \frac{\operatorname{Var}\left(\hat{U}_1(h)\right)}{t^2} = \frac{1}{t^2}\left(\frac{4\sigma_1^2(h)}{B} + \frac{2\sigma_2^2(h)}{B(B-1)}\right). \tag{7.1}$$

Using Hoeffding's inequality, the deviation probability can thus be bounded by

$$\exp\left(-2K\left(\frac{1}{2} - \frac{1}{t^2}\left(\frac{4\sigma_1^2(h)}{B} + \frac{2\sigma_2^2(h)}{B(B-1)}\right)\right)^2\right).$$

At this point, ignoring that K and B should be integers, choosing $K = \frac{\ln(1/\delta)}{2(\frac{1}{2}-\lambda)^2}$, $\lambda < 1/2$, makes the upper bound equal to δ, as long as $\frac{1}{t^2}\left(\frac{4\sigma_1^2(h)}{B} + \frac{2\sigma_2^2(h)}{B(B-1)}\right) = \lambda$. Reverting in t gives

$$t = \sqrt{\frac{1}{\lambda}\left(\frac{4\sigma_1^2(h)}{B} + \frac{2\sigma_2^2(h)}{B(B-1)}\right)} \leq 2\sigma_1(h)\sqrt{\frac{1}{\lambda B}} + \sqrt{2}\sigma_2(h)\sqrt{\frac{1}{\lambda B(B-1)}}.$$

The first term is obviously the dominant one, and is similar to that of Proposition 7.8. The optimization in λ is the same, and one should pick $\lambda = 1/6$, and $K = \frac{9}{2}\ln(1/\delta)$. Using $K = \left\lceil \frac{9}{2}\ln(1/\delta) \right\rceil$ preserves the inequality in δ. Taking $B = \left\lfloor \frac{n}{K} \right\rfloor$, together with the fact that $B \geq B - 1 \geq \left\lfloor \frac{n}{K} \right\rfloor - 1 \geq \frac{n}{2K} \geq \frac{n}{2\left\lceil \frac{9}{2}\ln(1/\delta) \right\rceil} \geq \frac{n}{9(1+\ln(1/\delta))}$ when $K \leq n/3$ (see Appendix C), one gets

$$t \leq 6\sqrt{6}\sigma_1(h)\sqrt{\frac{1+\ln(1/\delta)}{n}} + 18\sqrt{3}\sigma_2(h)\frac{1+\ln(1/\delta)}{n},$$

that allows to complete the proof. □

Remark 7.24. *This bound for U-statistics involves two terms. The first one, which is dominant, is almost the same as that of MoM in Proposition 7.8. The* 2 *factor difference derives from the difference in variances:* $4\sigma_1^2(h)$ *instead of* σ^2*. The second term also comes from the variance expression of the U-statistic, which features an additional part in* $1/B^2$ *approximately. It is thus roughly the square of the first term.*

Remark 7.25. *In Proposition 7.23, we have sacrificed a bit generality for the sake of simplicity. Indeed, approximations on the upper bound for* t *can only be made if* $K \leq n/3$*. It has a direct consequence on the range of attainable confidences* δ*, which differs from that of Propositions 7.8 and 7.17. Finally, notice that, as in previous propositions, a particular care has been given to the fact that Hoeffding's inequality is always used on positive deviations.*

The rest of the section is devoted to the analysis of several alternatives and extensions to the MoU estimator. It especially includes a generalization of Proposition 7.23 to U-statistics of arbitrary degree, and a discussion about related works.

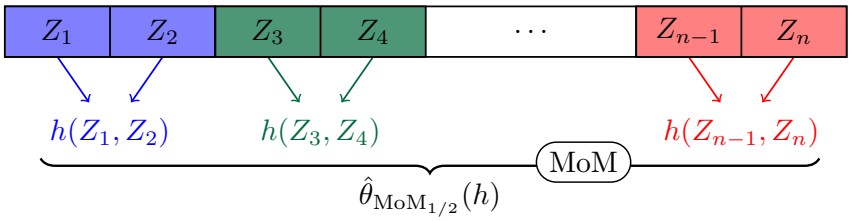


Figure 7.4 – MoM on Half procedure.

7.3.3 Alternatives and Extensions

In this subsection, we explore several alternatives and extensions to the MoU estimator: one based on a direct application of MoM to the pairs, one extension for U-statistics of arbitrary degree, and one estimator proposed in Joly and Lugosi (2016) that uses distinct disjoint blocks for each coordinate in h.

The *MoM on Half* Alternative

Building independent blocks by partitioning naturally induces independent U-statistics. Nevertheless, one can perfectly imagine directly selecting independent pairs and building U-statistics based on them. Observe however that creating the biggest possible blocks of independent pairs (between blocks) boils down to partitioning the data and creating all pairs within independent blocks, exactly as in Definition 7.22. The opposite strategy is to create pairs all independent from each other and then apply directly MoM's procedure to the $\lfloor n/2 \rfloor$ new independent observations created. Such a procedure is exposed in Figure 7.4.

Applying Proposition 7.8 to the $\lfloor n/2 \rfloor$ independent observations $\left\{h(Z_i, Z_{i+1})\right\}_{i \leq \lfloor n/2 \rfloor}$, one gets that for any $\delta \in [e^{1-2\lfloor n/2 \rfloor/9}, 1[$, it holds

$$\mathbb{P}\left\{\left|\hat{\theta}_{\mathrm{MoM}_{1/2}}(h) - \theta(h)\right| > 3\sqrt{6}\sigma(h)\sqrt{\frac{1+\ln(1/\delta)}{\lfloor n/2 \rfloor}}\right\} \leq \delta.$$

If compared to Proposition 7.23, this bound involves approximately a $6\sqrt{6}\sigma(h)$ constant, instead of $6\sqrt{6}\sigma_1(h)$ for the corresponding dominant term. Recalling that $\sigma^2(h) = 2\sigma_1^2(h) + \sigma_2^2(h)$ (Equation (6.6)), this strategy misses a $\sqrt{2}$ factor, in addition to $\sigma_2(h)$. For this latter term, the *MoM on Half* procedure exhibits a rate which is the square root of that of $\hat{\theta}_{\mathrm{MoU}}(h)$. This difference is due to the variances of the independent estimates used in both methods. MoU uses all pairs within a partition block, leading to a variance of the order $\sigma_1^2(h)/B + \sigma_2^2(h)/B^2$ for the base estimates, while the *MoM on Half* only uses the predefined pairs, set once and for all, inducing a higher $\sigma^2(h)/B$ variance. The *MoM on Half* procedure is however less computationally demanding, as much fewer pairs are involved, and its theoretical weakness is less predominant as $\sigma_2(h)$ decreases.

MoU for U-Statistics of Arbitrary Degree

Adapting the proof of Proposition 7.23 to U-statistics of arbitrary degree only requires to reconsider the variance expression. Instead of using the formula of Equation (6.7), plugging Equation (6.3)'s upper bound is enough to get a counterpart to Equation (7.1):

$$\hat{q}_t \leq \frac{\mathrm{Var}\left(\hat{U}_1(h)\right)}{t^2} \leq \frac{1}{t^2}\left(\sum_{c=1}^{d} \frac{d!^2\zeta_c(h)}{c!(d-c)!^2}\ \frac{1}{B(B-1)\dots(B-c+1)}\right).$$

It yields

$$t = \sqrt{\frac{1}{\lambda}\sum_{c=1}^{d} \frac{d!^2\zeta_c(h)}{c!(d-c)!^2}\ \frac{1}{B(B-1)\dots(B-c+1)}},$$

whose dominant term is obviously that attained by $c = 1$, and is similar to that in the proof of Proposition 7.23. The optimization in λ thus does not change (one should choose $\lambda = 1/6$ and $K = 9/2\ln(1/\delta)$). Using Appendix C, that ensures $B - d + 1 = \left\lfloor \frac{n}{K} \right\rfloor - d + 1 \geq \frac{n}{2K}$ as long as $K \leq \frac{n}{2d-1}$, and breaking the square root into pieces, one finally gets

$$\begin{aligned} t &\leq \sum_{c=1}^{d} \frac{d!}{(d-c)!}\sqrt{\frac{6\zeta_c(h)}{c!}}\left(\frac{2K}{n}\right)^{c/2}, \\ &\leq \sum_{c=1}^{d} \frac{3^c d!}{(d-c)!}\sqrt{\frac{6\zeta_c(h)}{c!}}\left(\frac{1+\ln(1/\delta)}{n}\right)^{c/2}. \end{aligned}$$

Proposition 7.26. *Let $d \in \mathbb{N}^*$, $\{Z_i\}_{i\leq n}$ be $n \geq d$ independent realizations of a $\mathcal{Z}$-valued random variable Z, and $h : \mathcal{Z}^d \to \mathbb{R}$ symmetric such that $\mathbb{E}[h(Z_1,\dots,Z_d)] = \theta(h) < +\infty$, and $\mathrm{Var}(h(Z_1,\dots,Z_d)) = \sigma^2(h) < +\infty$. Then, for any $\delta \in [e^{1-\frac{2n}{9(2d-1)}}, 1[$, choosing $K = \left\lceil \frac{9}{2}\ln(1/\delta) \right\rceil$, it holds*

$$\mathbb{P}\left\{\left|\hat{\theta}_{\mathrm{MoU}}(h) - \theta(h)\right| > \sum_{c=1}^{d} \mathscr{C}(c,d,h)\left(\frac{1+\ln(1/\delta)}{n}\right)^{c/2}\right\} \leq \delta,$$

with $\mathscr{C}(c,d,h) = \frac{3^c d!}{(d-c)!}\sqrt{\frac{6\zeta_c(h)}{c!}}$ for $c \leq d$, and $\zeta_c(h)$ defined as in Equation (6.1).

Remark 7.27. *Notice that constants of Proposition 7.23 may not be recovered, since different variance expressions have been used: one with the $\sigma_c(h)$, one with the $\zeta_c(h)$. The rates are however of the same order, and the range of admissible confidences δ remains unchanged.*

Remark 7.28. *As already noticed in Section 6.3, using Equation (6.8), should allow to derive guarantees for multisample U-statistics, up to a careful management of the different sample sizes.*

Related Works

Among studies on robust estimation of U-statistics, mention has to be made of the work by Minsker and Wei (2018). The angle taken by authors is however completely different from that proposed in this section. The U-statistic is viewed as a M-estimator,

minimizing a criterion involving the quadratic loss. The proposed estimator is then the M-estimator solving the same criterion, except that a different loss function is used. This loss function is designed to induce robustness, while being close enough to the square loss to derive guarantees.

The work by Joly and Lugosi (2016) is much closer to the estimator of Definition 7.22. Authors also build upon the MoM methodology, and starts by partitioning the dataset into K disjoint blocks of roughly equal size. But rather than computing (complete) U-statistics on each block, authors advocate to proceed as follows: 1) select a collection of d blocks, where d is the degree of the U-statistic, 2) compute the average of all the $h(Z_{i_1}, \dots, Z_{i_d})$, where i_j is allowed to vary in block j, 3) get such an estimate for every (unordered) collection of d blocks among the K original ones, and 4) finally take the median. The possibility of considering only the "diagonal blocks" (*i.e.* that every i_j varies in the same block) is also evoked – but not investigated – and corresponds exactly to Definition 7.22. Joly and Lugosi (2016) also establish sharper bounds for degenerate U-statistics, that could be adapted to our setting based on a sharper variance control.

One advantage of MoU is its lower computational cost: only $K\binom{n/K}{d}$ terms are summed, compared to the $\binom{K}{d}(n/K)^d$ of Joly and Lugosi (2016). Furthermore, it is easier to analyze theoretically, thanks to the independence of the complete block U-statistics. On the contrary, two base estimators of Joly and Lugosi (2016) are dependent as soon as they share one block among their respective collections of d blocks. However, authors did not consider MoU due to the "waste" it induces. Indeed, when computing complete U-statistics on blocks, one observation is only paired to a smaller fraction of other observations (namely, those in the same block). It yields redundancy, and consequently a higher variance, exactly as for complete U-statistics compared to incomplete ones (see Section 6.4.1 and Figure 6.1 therein).

Another remedy to this redundancy is to merge the MoRM and the MoU procedures to build the Median-of-Randomized-U-Statistics. Considering complete U-statistics on randomized blocks, or even incomplete U-statistics, then allows every observation to be paired to any other one. It is precisely the purpose of the subsequent section to study at length this approach.

7.4 The Median-of-Randomized-U-Statistics Estimator

A first and natural way to extend the randomized framework of MoM to U-statistics is surely to draw random SWoR blocks as in MoRM, compute the complete U-statistics on these blocks and take the median. As for MoRM, an observation may appear in several blocks, hence the need for a more complex proof, relying on conditioning upon the data. Nevertheless, and similarly to MoM/MoRM, guarantees derived in the randomized setting are of same nature as that in the partition case, and one can even recover constant factors asymptotically.

7.4.1 Definition

We start by defining formally the new estimator introduced, through Definition 7.29 and Figure 7.5. We keep the notation introduced in Section 7.2, namely the randomized blocks $(\mathcal{B}_k)_{k\leq K}$ are characterized by the random vectors $\boldsymbol{\epsilon}_k$, uniformly distributed over $\Lambda_{n,B} = \{\boldsymbol{\epsilon} \in \{0,1\}^n : \mathbf{1}^\top \boldsymbol{\epsilon} = B\}$.

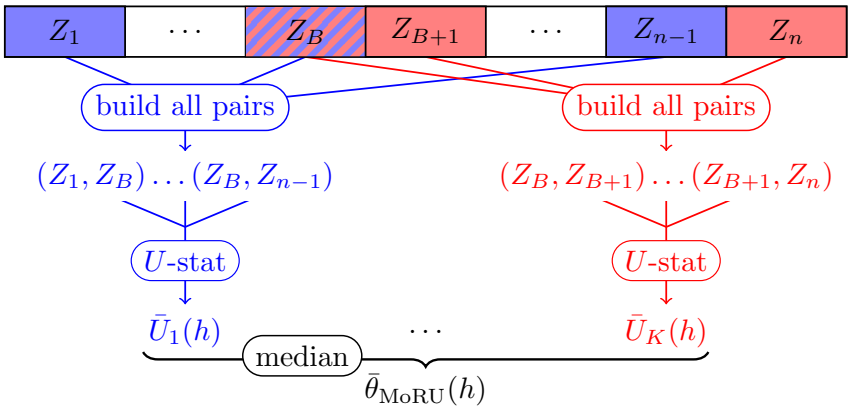


Figure 7.5 – MoRU's procedure

Definition 7.29. *Let $\mathcal{S}_n = \{Z_i\}_{i\leq n}$ be a sample of n independent realizations of a $\mathcal{Z}$-valued random variable Z, and h as in Definition 7.22. Let $K \in \mathbb{N}^*$, $B \leq n$ and $(\mathcal{B}_k)_{k\leq K}$ be K blocks of size B, sampled independently from $\mathcal{S}_n$ by SWoR. For $k \leq K$, let $\bar{U}_k(h)$ denote the (complete) U-statistic built on the block $\mathcal{B}_k$. Namely, $\bar{U}_k(h) = \frac{2}{B(B-1)} \sum_{\substack{i,j\in\mathcal{B}_k^2 \\ i<j}} h(Z_i, Z_j) = \frac{2}{B(B-1)} \sum_{i<j} \boldsymbol{\epsilon}_{k,i}\boldsymbol{\epsilon}_{k,j} h(Z_i, Z_j)$. The Median-of-Randomized-U-Statistics (MoRU) estimator is then given by*

$$\bar{\theta}_{\mathrm{MoRU}}(h) = \mathrm{median}(\bar{U}_1(h), \dots, \bar{U}_K(h)).$$

Let us now investigate the concentration properties of this new estimate.

7.4.2 Concentration Properties

Despite this randomization, the concentration properties of the MoRU are not affected. Using the same conditioning trick as for the proof of MoRM is sufficient. One even recovers the constant factors of MoU asymptotically.

Proposition 7.30. *Let $\{Z_i\}_{i\leq n}$ be n independent realizations of a $\mathcal{Z}$-valued random variable Z, and $h : \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}$ symmetric such that $\mathbb{E}[h(Z_1, Z_2)] = \theta(h) < +\infty$, and $\mathrm{Var}(h(Z_1, Z_2)) < +\infty$. Then, for any $\tau \in]0, 1/2[$, for any $\delta \in [2e^{-2\tau^2 n/9}, 1[$, choosing $K = \left\lceil \frac{\ln(2/\delta)}{2(\frac{1}{2}-\tau)^2} \right\rceil$ and $B = \left\lfloor \frac{8\tau^2 n}{9\ln(2/\delta)} \right\rfloor$, it holds*

$$\mathbb{P}\left\{ \left| \bar{\theta}_{\mathrm{MoRU}}(h) - \theta(h) \right| > C_1(h,\tau)\sqrt{\frac{\ln(2/\delta)}{n}} + C_2(h,\tau)\,\frac{\ln(2/\delta)}{n} \right\} \leq \delta,$$

with $C_1(h,\tau) = (\tau/3)^{-\frac{3}{2}}\,\sigma_1(h)$, and $C_2(h,\tau) = 2(2\tau/3)^{-\frac{5}{2}}\,\sigma_2(h)$, $\sigma_1(h)$ and $\sigma_2(h)$ being defined as in Equations (6.4) and (6.5) respectively.

Proof. Analogously to Proposition 7.17's proof, define

$$\bar{J}_{\boldsymbol{\epsilon}_k,t} = \mathbb{1}\{|\bar{U}_k(h) - \theta(h)| > t\} \quad \forall k \leq K,$$
$$\bar{W}_{n,t} = \mathbb{E}_{\boldsymbol{\epsilon}}\left[\frac{1}{K}\sum_{k=1}^{K} \bar{J}_{\boldsymbol{\epsilon}_k,t} \;\middle|\; \mathcal{S}_n\right],$$
$$\bar{q}_t = \mathbb{E}[\bar{W}_{n,t}] = \mathbb{E}[\bar{J}_{\boldsymbol{\epsilon}_1,t}] = \mathbb{P}\{|\bar{U}_1(h) - \theta(h)| > t\}.$$

Using the same conditioning, we get for any $\tau \in]0, 1/2[$

$$\mathbb{P}\left\{\left|\bar{\theta}_{\text{MoRU}}(h) - \theta(h)\right| > t\right\} \leq \mathbb{E}\left[\mathbb{P}\left\{\frac{1}{K}\sum_{k=1}^{K} \bar{J}_{\boldsymbol{\epsilon}_k,t} - \bar{W}_{n,t} \geq \frac{1}{2} - \tau\right\} \;\middle|\; \mathcal{S}_n\right] + \mathbb{P}\left\{\bar{W}_{n,t} - \bar{q}_t \geq \tau - \bar{q}_t\right\}.$$

Once again, observe that

$$\bar{W}_{n,t} \;=\; \frac{1}{\binom{n}{B}} \sum_{I} \mathbb{1}\left\{\left|\frac{2}{B(B-1)} \sum_{1\leq j<j'\leq B} h(Z_{I_j}, Z_{I_{j'}}) - \theta(h)\right| > t\right\},$$

with $\sum_I$ defined like in Definition 6.1, is a U-statistic of degree B, with bounded symmetric kernel $h(Z_1, \ldots, Z_B) = \mathbb{1}\left\{\left|\frac{2}{B(B-1)} \sum_{1\leq j<j'\leq B} h(X_{I_j}, X_{I_{j'}}) - \theta(h)\right| > t\right\}$.

One may also show (Appendix B.1) that

$$\bar{q}_t \leq \frac{1}{t^2}\left(\frac{4\sigma_1^2(h)}{B} + \frac{2\sigma_2^2(h)}{B(B-1)}\right),$$

so that the use of conditional Hoeffding's inequality and the Hoeffding's inequality for U-statistics (Proposition 6.3) applied to $\bar{W}_{n,t}$ leads to the following upper bound

$$\exp\left(-2K\left(\frac{1}{2} - \tau\right)^2\right) + \exp\left(-2\frac{n}{B}\left(\tau - \frac{1}{t^2}\left(\frac{4\sigma_1^2(h)}{B} + \frac{2\sigma_2^2(h)}{B(B-1)}\right)\right)^2\right).$$

Choosing $K = \left\lceil \frac{\ln(2/\delta)}{2(\frac{1}{2}-\tau)^2} \right\rceil$ ensures that the first term is lower than $\delta/2$. Choosing $B = \frac{2(\tau-\lambda)^2 n}{\ln(2/\delta)}$, $\lambda \leq \tau$, ensures that the second term is also bounded by $\delta/2$, as long as

$$t = \sqrt{\frac{1}{\lambda}\left(\frac{4\sigma_1^2(h)}{B} + \frac{2\sigma_2^2(h)}{B(B-1)}\right)} \leq 2\sigma_1(h)\sqrt{\frac{1}{\lambda B}} + \sqrt{2}\sigma_2(h)\sqrt{\frac{1}{\lambda B(B-1)}}.$$

The dominant term is obviously the first one, and it is similar to that in Proposition 7.17. The optimization is λ is the same, and one should pick $\lambda = \tau/3$, and $B = \frac{8\tau^2 n}{9\ln(2/\delta)}$. Changing to $B = \left\lfloor \frac{8\tau^2 n}{9\ln(2/\delta)} \right\rfloor$, and using $B \geq B-1 \geq \frac{4\tau^2 n}{9\ln(2/\delta)}$ for $B \geq 3$ (see Appendix C), does not change the inequality in $\delta/2$ while leading to the slightly modified final result

$$t \leq \frac{3\sqrt{3}}{\tau^{3/2}}\sigma_1(h)\sqrt{\frac{\ln(2/\delta)}{n}} + \frac{9\sqrt{6}}{4\tau^{5/2}}\sigma_2(h)\,\frac{\ln(2/\delta)}{n}.$$

□

Remark 7.31. *Again, one may observe that constants of Proposition 7.23 are recovered by letting τ tend to $1/2$, that every Hoeffding's inequality use is valid, and that we do have $3 \leq B \leq n$ with the prescribed expressions.*

Although randomized, the MoRU does not benefit from the interesting lower variance property of the incomplete U-statistics. Trying to leverage this attractive characteristic is at the core of the next subsection.

7.4.3 Alternatives and Extensions

As already briefly mentioned in the last paragraph of Section 7.3, computing complete U-statistics on randomized blocks is not the only way to authorize every observation to be paired to any other one. Another strategy would involve directly sampling from the pairs and computing incomplete U-statistics (see Section 6.4.1). Such an approach is developed in the following subsection, as well as a generalization of the MoRU's concentration bound (Proposition 7.30) to U-statistics of arbitrary degree.

The Medians-of-Incomplete-U-Statistics Estimator

In Section 6.4.1, we have highlighted that building incomplete U-statistics rather than complete ones may diminish the variance of the final estimate obtained. In order to incorporate this remark into the MoRU framework, it is completely possible to imagine a Median-of-Incomplete-U-Statistics, as described in Definition 7.32 and Figure 7.6. The underlying idea is that taking the median of estimates with lower variances should necessarily induce an improvement in the performance of the overall estimator.

Definition 7.32. *Let $\mathcal{S}_n = \{Z_i\}_{i \leq n}$ be a sample of n independent realizations of a $\mathcal{Z}$-valued random variable Z, and h as in Definition 7.22. Let $K \in \mathbb{N}$, $M \leq n(n-1)/2$, and $(\mathcal{P}_k)_{k \leq K}$ be K blocks of pairs of size M, sampled independently and uniformly over the $n(n-1)/2$ possible pairs, with or without replacement. For $k \leq K$, let $\tilde{U}_k(h)$ denote the incomplete U-statistic that is built on the M pairs of the block of pairs $\mathcal{P}_k$. Namely, $\tilde{U}_k(h) = \frac{1}{M} \sum_{(i,j) \in \mathcal{P}_k} h(Z_i, Z_j)$. The Median-of-Incomplete-U-Statistics (MoIU) estimator is then given by*

$$\tilde{\theta}_{\mathrm{MoIU}}(h) = \mathrm{median}(\tilde{U}_1(h), \ldots, \tilde{U}_K(h)).$$

Although this new procedure is expected to improve the performances thanks to the reduced variances of the incomplete U-statistics, it is harder to analyze theoretically. The first thing that can be noticed is that there is no independence between the base estimates $\tilde{U}_k(h)$. So one would be tempted to use the same proof path as for MoRM and MoRU. Unfortunately, in this setting, the conditional expectation of the sum of the indicator variables, which reads

$$\tilde{W}_{n,t} = \frac{1}{\binom{\frac{n(n-1)}{2}}{M}} \sum_{I} \mathbb{1}\left\{ \left| \frac{1}{M} \sum_{j=1}^{M} h\left(Z_{I_j^{(1)}}, Z_{I_j^{(2)}}\right) - \theta(h) \right| > t \right\},$$

where the symbol $\sum_I$ refers to the summation over all unordered subsets I of M integers chosen in $\{1, \ldots, n(n-1)/2\}$, and $(I_j^{(1)}, I_j^{(2)})$ represents the j^{th} pair of subset I, cannot be identified as a U-statistic, exactly as in Section 7.2.3.

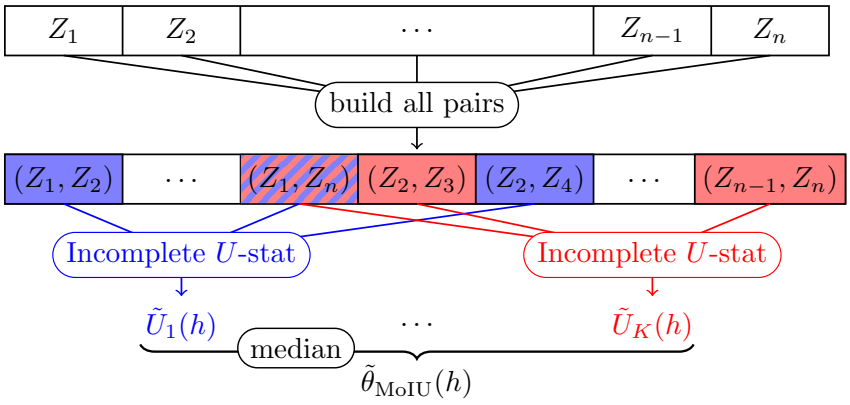


Figure 7.6 – MoIU's procedure

Similarly, the use of the bounded differences inequality is not sufficient here. Assume that one observation changes (so $n-1$ pairs of observations change). The probability of sampling at least one of the changed pairs among the M ones is

$$1 - \left(1 - \frac{2}{n}\right)^M \leq \frac{2M}{n} \qquad \text{with replacement,}$$

$$1 - \frac{\binom{(n-1)(n-2)/2}{M}}{\binom{n(n-1)/2}{M}} \leq \frac{2M}{n-2} \qquad \text{without replacement.}$$

Refer to Appendix D for details about this last inequality. The maximal deviation of the functional if one observation is changed being equal to the previously explicited probabilities, the bounded differences inequality (for the replacement case) yields a bound in $\exp(-n(\tau - \tilde{q}_t)^2/(4M^2))$. Recall that $\tilde{q}_t$ may be bounded using Chebyshev's inequality and is of the order $\mathcal{O}(1/t^2M)$ thanks to the variance formula of the incomplete U-statistic (Equation (6.9)). Hence, to derive reasonable guarantees, we need M to be of the order $\mathcal{O}(n)$, which jeopardizes the bound previously obtained. The analysis is similar without replacement.

Exactly like when other sampling schemes were tested within the MoRM procedure (section 7.2.3), trying to incorporate incomplete U-statistics into MoRU compromises every proof techniques we have developed so far. Despite the absence of theoretical guarantees, empirical evidences of Section 7.5 tend however to validate this approach. A future key research direction would consist in a deeper investigation of this estimator, which was one of the motivations to start considering randomizing MoMs.

MoRU for U-Statistics of Arbitrary Degree

Just as for MoU, a version for MoRU for U-statistics of arbitrary degree is available. The proof technique is very similar to that of Proposition 7.26 so we only state the proposition.

Proposition 7.33. *Let $d \in \mathbb{N}^*$, $\{Z_i\}_{i\leq n}$ be $n \geq d$ independent realizations of a $\mathcal{Z}$-valued random variable Z, and $h : \mathcal{Z}^d \to \mathbb{R}$ symmetric such that $\mathbb{E}[h(Z_1, \ldots, Z_d)] = \theta(h) < +\infty$, and $\mathrm{Var}(h(Z_1, \ldots, Z_d)) = \sigma^2(h) < +\infty$. Then, for any $\tau \in]0, 1/2[$, for any $\delta \in [2e^{-4\tau^2 n/9d}, 1[$, choosing $K = \left\lceil \frac{\ln(2/\delta)}{2(\frac{1}{2}-\tau)^2} \right\rceil$ and $B = \left\lfloor \frac{8\tau^2 n}{9\ln(2/\delta)} \right\rfloor$, it holds*

$$\mathbb{P}\left\{ \left| \bar{\theta}_{\mathrm{MoRU}}(h) - \theta(h) \right| > \sum_{c=1}^{d} \mathscr{C}(c, d, h, \tau) \left(\frac{\ln(2/\delta)}{n} \right)^{c/2} \right\} \leq \delta,$$

with $\mathscr{C}(c, d, h, \tau) = \frac{d!}{(d-c)!} \left(\frac{3}{2\tau} \right)^c \sqrt{\frac{3\zeta_c(h)}{\tau c!}}$ for $c \leq d$, and $\zeta_c(h)$ as in Equation (6.1).

The next and final section of this chapter is devoted to numerical experiments that corroborate the theoretical findings of previous sections.

7.5 Estimation Experiments

We now present numerical experiments supporting the relevance of the MoM variants analyzed in the previous sections. In this section, simulations only focus on estimation problems. Refer to Section 8.6 for experiments on learning problems. Both MoRM and MoRU are studied, through the estimation of the mean and the variance of several laws, three of which being heavy tailed. Performances are assessed in two ways: the quadratic risks, reported in the Tables, and the empirical deviation probabilities (*i.e.* the empirical quantiles) as suggested by Catoni (2012), summarized by the Figures.

7.5.1 MoRM Experiments

Considering inference of the expectation of four pre-specified distributions (Gaussian, Student, Log-normal and Pareto), based on a sample of size $n = 1000$, seven estimators are compared below: standard MoM, and six MoRM estimators, related to different sampling schemes (SWoR, Monte-Carlo) or different values of the hyperparameter τ. Results are obtained through 5000 replications of the estimation procedures. Beyond the quadratic risk (Table 7.1), the estimators accuracies are assessed via the deviation probabilities, *i.e.* empirical quantiles for different confidence levels δ (Figure 7.7). As highlighted in Remark 7.20, $\tau = 1/6$ leads to (approximately) the same number of blocks as in the MoM procedure. However, MoRM usually select blocks of cardinality lower than n/K, so that the MoRM estimator with $\tau = 1/6$ uses less examples than the MoM. Proposition 7.17 exhibits a higher constant for MoRM in that case, and it is confirmed empirically here. The choice $\tau = 3/10$ guarantees that the number of MoRM blocks multiplied by their cardinality is equal to n. This way, MoRM uses as much samples as MoM. Nevertheless, the increased variability leads to a slightly lower performance in this case. Finally, $\tau = 9/20$ is chosen to be close to $1/2$, as suggested by Remark 7.20. In this setting, the two constant factors are (almost) equal, and MoRM even empirically shows a systematic improvement compared to MoM. Note that the quantile curves should be decreasing. However, the estimators being δ-dependent, different experiments are run for each value of δ, and the rare little increases are due to this random effect.

Table 7.1 – Quadratic Risks for the Mean Estimation, $\delta = 0.001$

	Normal $(0, 1)$	Student (3)
MoM	1.49e-3 ± 2.18e-3	4.10e-3 ± 5.84e-3
$\text{MoRM}_{1/6,\ \text{SWoR}}$	1.37e-2 ± 1.89e-2	2.95e-2 ± 4.45e-2
$\text{MoRM}_{1/6,\ \text{MC}}$	1.37e-2 ± 1.90e-2	2.92e-2 ± 4.36e-2
$\text{MoRM}_{3/10,\ \text{SWoR}}$	2.55e-3 ± 3.61e-3	6.02e-3 ± 8.68e-3
$\text{MoRM}_{3/10,\ \text{MC}}$	2.64e-3 ± 3.72e-3	6.22e-3 ± 8.95e-3
$\text{MoRM}_{9/20,\ \text{SWoR}}$	1.05e-3 ± 1.48e-3	**2.64e-3 ± 3.72e-3**
$\text{MoRM}_{9/20,\ \text{MC}}$	**1.05e-3 ± 1.46e-3**	2.65e-3 ± 3.74e-3
	Log-normal $(0, 1)$	Pareto (3)
MoM	6.97e-3 ± 9.48e-3	**1.02 ± 6.12e-2**
$\text{MoRM}_{1/6,\ \text{SWoR}}$	6.21e-2 ± 7.88e-2	1.12 ± 1.50e-1
$\text{MoRM}_{1/6,\ \text{MC}}$	6.17e-2 ± 7.14e-2	1.13 ± 1.49e-1
$\text{MoRM}_{3/10,\ \text{SWoR}}$	1.24e-2 ± 1.61e-2	1.05 ± 7.04e-2
$\text{MoRM}_{3/10,\ \text{MC}}$	1.28e-2 ± 1.65e-2	1.06 ± 7.30e-2
$\text{MoRM}_{9/20,\ \text{SWoR}}$	**4.97e-3 ± 6.68e-3**	1.03 ± 4.90e-2
$\text{MoRM}_{9/20,\ \text{MC}}$	4.99e-3 ± 6.73e-3	1.03 ± 4.88e-2

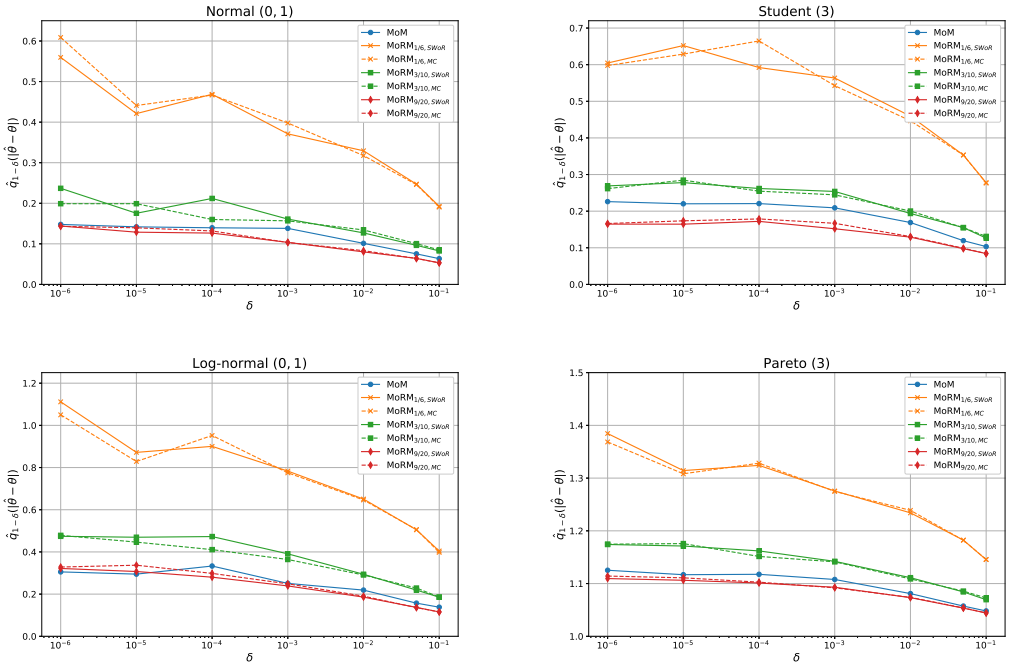


Figure 7.7 – Empirical Quantiles for the Different Mean Estimators on 4 Laws

Table 7.2 – Quadratic Risks for the Variance Estimation, $\delta = 0.001$

	Normal $(0, 1)$	Student (3)
$\text{MoU}_{1/2;1/2}$	4.09e-3 ± 5.79e-3	1.73 ± 28.36
$\text{MoU}_{\text{Partition}}$	3.24e-3 ± 4.48e-3	**3.82e-1 ± 3.19e-1**
$\text{MoRU}_{\text{SWoR}}$	5.04e-3 ± 7.05e-3	5.12e-1 ± 3.88
$\text{MoIU}_{1/6,\ \text{SWoR}}$	2.06e-3 ± 2.85e-3	1.78 ± 34.72
$\text{MoIU}_{1/6,\ \text{MC}}$	**2.05e-3 ± 2.81e-3**	1.65 ± 26.22
$\text{MoIU}_{3/10,\ \text{SWoR}}$	2.16e-3 ± 3.01e-3	1.14 ± 16.95
$\text{MoIU}_{3/10,\ \text{MC}}$	2.11e-3 ± 2.88e-3	1.22 ± 17.47
	Log-normal $(0, 1)$	Pareto (3)
$\text{MoU}_{1/2;1/2}$	2.61 ± 23.50	1.36 ± 36.80
$\text{MoU}_{\text{Partition}}$	**1.62 ± 1.42**	**9.30e-2 ± 5.65e-2**
$\text{MoRU}_{\text{SWoR}}$	2.01 ± 4.85	9.70e-2 ± 7.12e-2
$\text{MoIU}_{1/6,\ \text{SWoR}}$	2.51 ± 21.90	1.38 ± 40.13
$\text{MoIU}_{1/6,\ \text{MC}}$	2.62 ± 24.80	1.51 ± 42.91
$\text{MoIU}_{3/10,\ \text{SWoR}}$	2.07 ± 14.83	8.50e-1 ± 21.99
$\text{MoIU}_{3/10,\ \text{MC}}$	2.17 ± 15.24	8.90e-1 ± 22.29

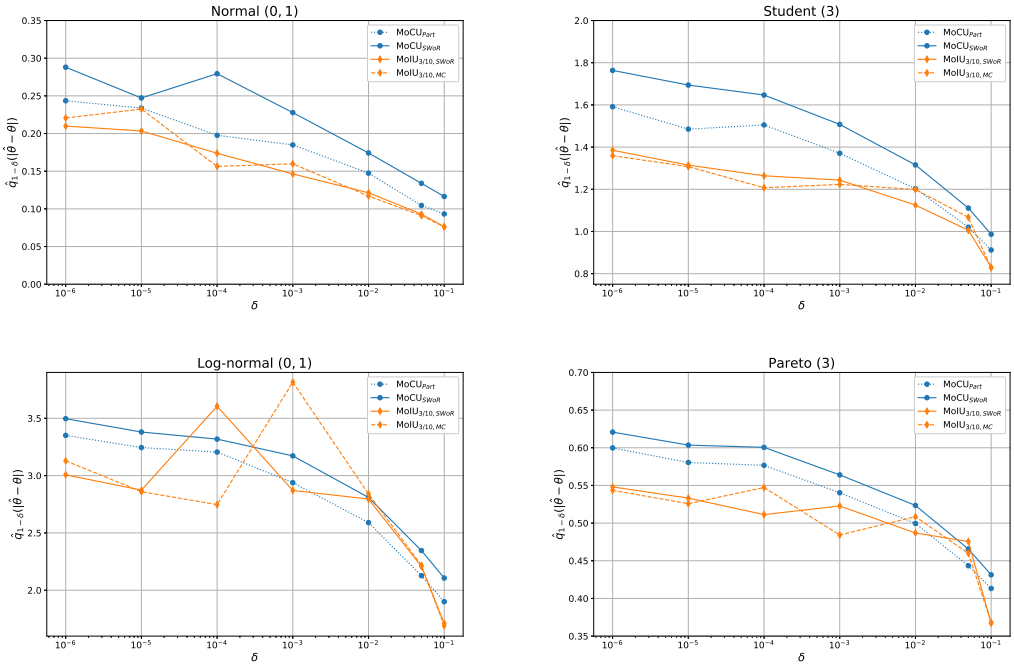


Figure 7.8 – Empirical Quantiles for the Different Variance Estimators on 4 Laws

7.5.2 MoRU Experiments

As for the estimation of U-statistics, we aim at estimating the variance of the four laws used above. Recall that the variance is a single sample U-statistic of degree 2 with kernel $h(z, z') = (z - z')^2/2$. Again, estimators are assessed through their quadratic risk (Table 7.2) and empirical quantiles (Figure 7.8). The empirical quantiles confirm the quadratic risks results: the τ parameter is crucial, making MoRM the worst or the best estimate depending on its value. The sampling scheme seems not to affect much the performance, even if the MC scenario is much more complex to analyze theoretically. The partitioning MoU seems to outperform every other estimate. One explanation can be that an extreme value may *corrupt* only one block within this method, whereas randomized versions can suffer from it in several blocks.

7.6 Conclusion

In this chapter, we have recalled the principle of the Median-of-Means estimator, as well as the deviation inequalities it satisfies. The estimator has further been extended in several ways, either by considering random blocks and/or U-statistics, that involve summing over pairs of observations. Moreover, guarantees of the same order as that of the standard estimator have been derived, crucially relying on U-statistics concentration properties (Chapter 6). A tighter analysis of V-statistics concentration should allow to consider sampling with replacement schemes, so far neglected for technical reasons.

As shall be shown in Chapter 8, learning with Median-of-Means can be performed through minimizing a MoM estimate of the risk, or by tournaments procedures. The U-statistics extensions here introduced then allows to tackle pairwise learning problems. As for the randomized version of MoM, it induces an adaptation of Gradient Descent that naturally escapes the local minima.

8

Robust Learning via Medians-of-(Randomized-Pairwise)-Means

Contents

One of the most direct use in statistical learning of the MoM estimator can surely be found in Bubeck et al. (2013). Authors use several robust mean estimators, as Catoni's M-estimator and the MoM, to design bandits strategies when the data is heavy-tailed.

However, the robustness benefits of the MoM go far beyond the mean estimation of a reward function in reinforcement learning. Indeed, another natural approach to combine statistical learning and MoM-inspired estimators is undoubtedly to adapt the ERM paradigm with robust estimates of the risk, instead of the standard empirical mean generally used. Thus, the MoM's principle was already applied in Lerasle and Oliveira (2011), where authors advocate to minimize a MoM estimate of the risk, with application to Lasso estimators and least-squares density estimation. In Brownlees et al. (2015), on the other hand, it is a Catoni's version of the risk that is proposed to be optimized. Finally, Hsu and Sabato (2014, 2016) tackle least-squares and sparse linear regression by using an extension of the MoM to arbitrary metric spaces (see Section 7.1.3).

While more recent works by Lecué and Lerasle (2017); Lecué et al. (2018) have further enriched the MoM-ERM framework with an algorithmic optimization strategy inspired from Gradient Descent (GD), other approaches introduced by Lugosi and Mendelson (2016); Lugosi et al. (2019) incorporate the MoM's principle into statistical learning theory through the *tournament procedures*.

In this chapter, we focus on the last two approaches, with a goal of extending them to the pairwise framework, and to the randomized estimators when it is possible. The standard MoM-ERM paradigm is first recalled in Section 8.1, followed by its randomized (Section 8.2) and pairwise (Section 8.3) extensions. Section 8.4 is devoted to a general statement of all MoM-like Gradient Descent strategies, while tournament procedures, standard and pairwise, are addressed in Section 8.5. Lastly, some numerical experiments are gathered in Section 8.6, while concluding remarks are collected in Section 8.7.

Aside from Sections 8.1 and 8.5.1, all results presented in this chapter are taken from:

▶ **P. Laforgue**, S. Clémençon, P. Bertail. On medians of (Randomized) pairwise means. In *Proceedings of International Conference on Machine Learning*, 2019.

In particular, notions introduced in Chapter 7 are crucial to the proofs derivation. The tools used to control MoRM's deviations are identical to those needed in Section 7.2, while Sections 8.3 and 8.5.2 build upon the concept of MoU (see Definition 7.22). All proofs presented in this chapter make use of the *small ball* method, or at least an adaptation of it, developed in Mendelson (2014) and Koltchinskii and Mendelson (2015) to handle empirical processes. The interested reader may refer to Lecué and Mendelson (2013) and Mendelson (2016, 2017) for more references on this precise subject.

8.1 Minimizing a MoM Estimate of the Risk

As extensively explained in Chapter 7, MoM-like estimates provide interesting mean estimators when data are heavy-tailed. Then, it is natural to study the results obtained if the celebrated ERM paradigm, that advocates to minimize an empirical mean of the risk, is slightly modified to the minimization of a MoM-like version of the risk. Interestingly, the guarantees that can be derived in this MoM framework exhibit the robustness of the MoM minimizers to outliers in the training dataset.

Our first focus is the MoM minimizer $\hat{f}_{\text{MoM}}$. It is the minimizer of a MoM estimate of the risk, that is formally defined as

$$\hat{f}_{\text{MoM}} = \underset{f\in\mathcal{F}}{\text{argmin}} \left\{ \widehat{\mathcal{R}}_{\text{MoM}}(f) = \hat{\mathbb{E}}_{\text{MoM}}[\ell_f] = \text{median}\left(\sum_{i\in B_1} \ell(f, Z_i), \ldots, \sum_{i\in B_K} \ell(f, Z_i) \right) \right\}, \tag{8.1}$$

where $(B_k)_{k\leq K}$ is a partition of $\mathcal{S}_n$. One can already notice that it is *partition dependent*. This feature makes the optimization procedure particularly challenging (Section 8.4), in addition to the non-linearity of the median operator.

We start by reproducing the proof of Theorem 2 in Lecué et al. (2018), that bounds the excess risk of the MoM minimizer in presence of outliers in the training sample. The variant obtained for MoRM is detailed and analyzed in Section 8.2, while Mo(R)U extensions are tackled in Section 8.3, with proofs focusing mainly on parts that differ from the standard mean scenarios. The theorem first needs the following assumptions.

Assumption 8.1. *There exists* $\sigma_{\mathcal{F}} > 0$ *such that:* $\sup_{f\in\mathcal{F}} \|f\|_{L^2} = \sqrt{\mathbb{E}\left[f(X)^2\right]} \leq \sigma_{\mathcal{F}}$.

Assumption 8.2. *The training sample* $\mathcal{S}_n$ *is composed of* informative *observations, sampled from the law of interest, indexed by* I *(*$\#\mathsf{I} = n_{\mathsf{I}}$*), and* outliers*, potentially adversarial, indexed by* O *(*$\#\mathsf{O} = n_{\mathsf{O}}$*). Let* K *denote the set of block indexes such that the associated block contains no outlier:* $\mathsf{K} = \{k \leq K : B_k \cap \mathsf{O} = \emptyset\}$ *(*$\#\mathsf{K} = n_{\mathsf{K}}$*), and* J *the set of all indexes contained in blocks containing no outliers:* $\mathsf{J} = \cup_{k\in\mathsf{K}} B_k$*. We now need a slightly modified version of the Rademacher complexity, and assume that it is finite:*

$$\mathscr{R}(\mathcal{F}) = \max_{\mathsf{A}\in\{\mathsf{I},\mathsf{J}\}} \mathbb{E}\left[\sup_{f\in\mathcal{F}} \sum_{i\in\mathsf{A}} \sigma_i f(Z_i) \right] < +\infty,$$

with $(\sigma_i)_{i\in\mathsf{A}}$ *being* $\#\mathsf{A}$ *i.i.d. Rademacher random variables.*

Remark 8.3. *This Rademacher complexity definition differs in two ways from that introduced in Chapter 1: it is taken with respect to either* I *or* J*, and thus is not rescaled by the number of points considered. In order to avoid accumulating notation, we now consider that $\mathscr{R}(\mathcal{F})$ refers to the quantity defined in Assumption 8.2.*

While the previous assumption characterizes the complexity of $\mathcal{F}$, the following one describes a form of Lipschitz continuity of the loss function.

Assumption 8.4. *There exists $L > 0$ such that for all $(x, y) \in \mathcal{X} \times \mathcal{Y}$, for all $f, f' \in \mathcal{F}^2$:*

$$|\ell_f(x, y) - \ell_{f'}(x, y)| \leq L|f(x) - f'(x)|.$$

Remark 8.5. *Assumption 8.4 holds for classical relaxations of the $0-1$ loss such as the* hinge loss *or the* logistic loss*. In these examples, L may be chosen equal to 1.*

We now state the theorem that upper bounds the excess risk of the MoM minimizer.

Theorem 8.6. *Grant Assumptions 8.1, 8.2 and 8.4. Assume that $n \geq K \geq 4n_{\mathsf{O}}$, and let $\Delta = 1/4 - n_{\mathsf{O}}/K$. Then with probability at least $1 - 2\exp(-2\Delta^2 K)$ it holds*

$$\mathcal{R}(\hat{f}_{\mathrm{MoM}}) \leq \mathcal{R}(f^*) + 16L \max\left(\sigma_{\mathcal{F}}\sqrt{\frac{K}{n}}, \frac{8\mathscr{R}(\mathcal{F})}{n}\right).$$

Proof. Using the fact that $\hat{f}_{\mathrm{MoM}}$ minimizes $\widehat{\mathcal{R}}_{\mathrm{MoM}}(\ell_f)$ over $\mathcal{F}$, one gets

$$\begin{aligned}
\mathcal{R}(\hat{f}_{\mathrm{MoM}}) - \mathcal{R}(f^*) &\leq \mathcal{R}(\hat{f}_{\mathrm{MoM}}) - \widehat{\mathcal{R}}_{\mathrm{MoM}}(\hat{f}_{\mathrm{MoM}}) + \widehat{\mathcal{R}}_{\mathrm{MoM}}(f^*) - \mathcal{R}(f^*), \\
&\leq 2\sup_{f\in\mathcal{F}} \left|\widehat{\mathcal{R}}_{\mathrm{MoM}}(f) - \mathcal{R}(f)\right|, \\
&\leq 2\sup_{f\in\mathcal{F}} \left|\hat{\mathbb{E}}_{\mathrm{MoM}}(\ell_f - \mathbb{E}[\ell_f])\right|. \qquad (8.2)
\end{aligned}$$

Focus now on the deviation of its right-hand side. One has

$$\mathbb{P}\left\{\sup_{f\in\mathcal{F}} \hat{\mathbb{E}}_{\mathrm{MoM}}(\ell_f - \mathbb{E}[\ell_f]) > t\right\} \leq \mathbb{P}\left\{\sup_{f\in\mathcal{F}} \sum_{k=1}^{K} \mathbb{1}\left\{\hat{\mathbb{E}}_k[\ell_f] - \mathbb{E}[\ell_f] > t\right\} \geq \frac{K}{2}\right\}, \qquad (8.3)$$

with the notation $\hat{\mathbb{E}}_k[\ell_f] = \frac{1}{B}\sum_{i\in B_k} \ell_f(Z_i)$.

Introducing $\psi : t \mapsto (t-1)\mathbb{1}\{1 \leq t \leq 2\} + \mathbb{1}\{t \geq 2\}$ and noticing that $\psi(t) \geq \mathbb{1}\{t \geq 2\}$:

$$\begin{aligned}
&\sup_{f\in\mathcal{F}} \sum_{k=1}^{K} \mathbb{1}\left\{\hat{\mathbb{E}}_k[\ell_f] - \mathbb{E}[\ell_f] > t\right\} \\
&\quad\leq \sup_{f\in\mathcal{F}} \sum_{k\in\mathsf{K}} \mathbb{E}\left[\psi\left(\frac{2(\hat{\mathbb{E}}_k[\ell_f] - \mathbb{E}[\ell_f])}{t}\right)\right] + (K - n_{\mathsf{K}}) \\
&\quad+ \sup_{f\in\mathcal{F}} \sum_{k\in\mathsf{K}} \psi\left(\frac{2(\hat{\mathbb{E}}_k[\ell_f] - \mathbb{E}[\ell_f])}{t}\right) - \mathbb{E}\left[\psi\left(\frac{2(\hat{\mathbb{E}}_k[\ell_f] - \mathbb{E}[\ell_f])}{t}\right)\right].
\end{aligned}$$

As $\psi(t) \leq \mathbb{1}\{t \geq 1\}$, it holds:

$$\begin{aligned}\mathbb{E}\left[\psi\left(\frac{2(\hat{\mathbb{E}}_k[\ell_f] - \mathbb{E}[\ell_f])}{t}\right)\right] &\leq \mathbb{P}\left\{\hat{\mathbb{E}}_k[\ell_f] - \mathbb{E}[\ell_f] \geq \frac{t}{2}\right\}, \\ &\leq \frac{4\ \mathrm{Var}\left(\hat{\mathbb{E}}_k[\ell_f]\right)}{t^2}, \\ &\leq \frac{8L^2}{Bt^2}\ \mathbb{E}\left[f(X)^2\right]. \end{aligned} \tag{8.4}$$

Noticing that $(K - n_{\mathsf{K}}) \leq n_{\mathsf{O}}$, one has:

$$\begin{aligned}&\sup_{f\in\mathcal{F}} \sum_{k=1}^{K} \mathbb{1}\left\{\hat{\mathbb{E}}_k[\ell_f] - \mathbb{E}[\ell_f] > t\right\} \\ &\quad\leq K\left(\frac{8L^2\sigma_{\mathcal{F}}^2}{Bt^2} + \frac{n_{\mathsf{O}}}{K}\right. \\ &\qquad\left. + \sup_{f\in\mathcal{F}} \frac{1}{K}\sum_{k\in\mathsf{K}} \psi\left(\frac{2(\hat{\mathbb{E}}_k[\ell_f] - \mathbb{E}[\ell_f])}{t}\right) - \mathbb{E}\left[\psi\left(\frac{2(\hat{\mathbb{E}}_k[\ell_f] - \mathbb{E}[\ell_f])}{t}\right)\right]\right)\end{aligned}$$

As $0 \leq \psi \leq 1$, the bounded-difference inequality yields that for any $y > 0$, it holds with probability at least $1 - e^{-2Ky^2}$

$$\begin{aligned}&\sup_{f\in\mathcal{F}} \frac{1}{K}\sum_{k\in\mathsf{K}} \psi\left(\frac{2(\hat{\mathbb{E}}_k[\ell_f] - \mathbb{E}[\ell_f])}{t}\right) - \mathbb{E}\left[\psi\left(\frac{2(\hat{\mathbb{E}}_k[\ell_f] - \mathbb{E}[\ell_f])}{t}\right)\right] \\ &\quad\leq \mathbb{E}\left[\sup_{f\in\mathcal{F}} \frac{1}{K}\sum_{k\in\mathsf{K}} \psi\left(\frac{2(\hat{\mathbb{E}}_k[\ell_f] - \mathbb{E}[\ell_f])}{t}\right) - \mathbb{E}\left[\psi\left(\frac{2(\hat{\mathbb{E}}_k[\ell_f] - \mathbb{E}[\ell_f])}{t}\right)\right]\right] + y.\end{aligned}$$

Now, by symmetrization arguments (see Giné and Zinn (1984) for instance), one has

$$\begin{aligned}&\mathbb{E}_{\mathcal{S}_n}\left[\sup_{f\in\mathcal{F}} \frac{1}{K}\sum_{k\in\mathsf{K}} \psi\left(\frac{2(\hat{\mathbb{E}}_k[\ell_f] - \mathbb{E}[\ell_f])}{t}\right) - \mathbb{E}\left[\psi\left(\frac{2(\hat{\mathbb{E}}_k[\ell_f] - \mathbb{E}[\ell_f])}{t}\right)\right]\right] \\ &\quad\leq 2\mathbb{E}_{\mathcal{S}_n,\boldsymbol{\sigma}}\left[\sup_{f\in\mathcal{F}} \frac{1}{K}\sum_{k\in\mathsf{K}} \sigma_k\psi\left(\frac{2(\hat{\mathbb{E}}_k[\ell_f] - \mathbb{E}[\ell_f])}{t}\right)\right],\end{aligned}$$

with $(\sigma_k)_{k\leq n_{\mathsf{K}}}$ being n_{K} i.i.d. Rademacher random variables. Then, by the contraction principle, since ψ is 1-Lipschitz with $\psi(0) = 0$:

$$\mathbb{E}_{\mathcal{S}_n,\boldsymbol{\sigma}}\left[\sup_{f\in\mathcal{F}} \frac{1}{K}\sum_{k\in\mathsf{K}} \sigma_k\psi\left(\frac{2(\hat{\mathbb{E}}_k[\ell_f] - \mathbb{E}[\ell_f])}{t}\right)\right] \leq \mathbb{E}_{\mathcal{S}_n,\boldsymbol{\sigma}}\left[\sup_{f\in\mathcal{F}} \frac{2}{Kt}\sum_{k\in\mathsf{K}} \sigma_k(\hat{\mathbb{E}}_k[\ell_f] - \mathbb{E}[\ell_f])\right]$$

Finally, by the symmetrization principle:

$$\mathbb{E}_{\mathcal{S}_n,\boldsymbol{\sigma}}\left[\sup_{f\in\mathcal{F}}\sum_{k\in\mathsf{K}}\sigma_k(\hat{\mathbb{E}}_k[\ell_f]-\mathbb{E}[\ell_f])\right]\leq\frac{2}{B}\,\mathbb{E}_{\mathcal{S}_n,\boldsymbol{\sigma}}\left[\sup_{f\in\mathcal{F}}\sum_{i\in\mathsf{J}}\sigma_i\ell_f(Z_i)\right]\leq\frac{2L}{B}\,\mathscr{R}(\mathcal{F}).$$

Combining all inequalities, it holds with probability at least $1-e^{-2Ky^2}$

$$\sup_{f\in\mathcal{F}}\sum_{k=1}^{K}\mathbb{1}\left\{\hat{\mathbb{E}}_k[\ell_f]-\mathbb{E}[\ell_f]>t\right\}\leq K\left(\frac{8L^2\sigma_{\mathcal{F}}^2K}{nt^2}+\frac{n_{\mathsf{O}}}{K}+y+\frac{8L\mathscr{R}(\mathcal{F})}{tn}\right)\tag{8.5}$$

Setting $\Delta=\frac{1}{4}-\frac{n_{\mathsf{O}}}{K}$, $y=\Delta$, and $t=8L\max\left(\sigma_{\mathcal{F}}\sqrt{\frac{K}{n}},\frac{8\mathscr{R}(\mathcal{F})}{n}\right)$, one gets:

$$\mathbb{P}\left\{\sup_{f\in\mathcal{F}}\sum_{k=1}^{K}\mathbb{1}\left\{\hat{\mathbb{E}}_k[\ell_f]-\mathbb{E}[\ell_f]>t\right\}\leq\frac{K}{2}\right\}\geq1-e^{-2K\Delta^2},$$

that implies:

$$\mathbb{P}\left\{\sup_{f\in\mathcal{F}}\hat{\mathbb{E}}_{\mathrm{MoM}}(\ell_f-\mathbb{E}[\ell_f])>8L\max\left(\sigma_{\mathcal{F}}\sqrt{\frac{K}{n}},\frac{8\mathscr{R}(\mathcal{F})}{n}\right)\right\}\leq e^{-2K\Delta^2},$$

and by the symmetry of the previous analysis:

$$\mathbb{P}\left\{\sup_{f\in\mathcal{F}}\left|\hat{\mathbb{E}}_{\mathrm{MoM}}(\mathbb{E}[\ell_f]-\ell_f)\right|>8L\max\left(\sigma_{\mathcal{F}}\sqrt{\frac{K}{n}},\frac{8\mathscr{R}(\mathcal{F})}{n}\right)\right\}\leq2e^{-2K\Delta^2}.$$

Combining with Equation (8.2), one finally gets that it holds with probability at least $1-2e^{-2\Delta^2K}$

$$\mathcal{R}(\hat{f}_{\mathrm{MoM}})\leq\mathcal{R}(f^*)+16L\max\left(\sigma_{\mathcal{F}}\sqrt{\frac{K}{n}},\frac{8\mathscr{R}(\mathcal{F})}{n}\right).$$

□

What is remarkable with Theorem 8.6 is that guarantees may be derived despite the presence of outliers in the dataset. If one takes a look back to Equation (8.1), an intuitive explanation might be the following: as $(B_k)_{k\leq K}$ is a partition of $\mathcal{S}_n$, at most n_{O} small empirical risk estimates are contaminated; and if K is large enough (compared to n_{O}), one may hope that a non-contaminated small estimate is chosen as the median.

We now extend the previous analysis to the case of a MoRM minimizer. An additional difficulty lies in the fact that blocks are now sampled at random (and not a partition), so that outliers may be selected more than once. If Theorem 8.6 is reproduced from Lecué et al. (2018), results presented in the next section are new and part of this manuscript's contribution.

8.2 Minimizing a MoRM estimate of the Risk

In this subsection, we thus analyze the impact of considering a MoRM minimizer rather than the MoM minimizer advocated in Theorem 8.6. As a reminder, it is defined as

$$\hat{f}_{\mathrm{MoRM}} = \operatorname*{argmin}_{f\in\mathcal{F}} \left\{ \bar{R}_{\mathrm{MoRM}}(f) = \bar{\mathbb{E}}_{\mathrm{MoRM}}[\ell_f] = \mathrm{median}\left(\sum_{i\in\mathcal{B}_k} \ell(f, Z_i),\ k \le K \right) \right\},$$

where $(\mathcal{B}_k)_{k\le K}$ are K i.i.d. subsamples of size B sampled uniformly over $\mathcal{S}_n$, without replacement within a block, but with replacement from one block to another. Refer to Section 7.2.1 for more details. Before proceeding, we need an additional assumption.

Assumption 8.7. *Let $B \in \mathbb{N}^*$. For all $f \in \mathcal{F}$, for all $t > 0$, define $q_{f,t} : \mathcal{Z}^B \to \{0,1\}$ such that*

$$q_{f,t}(Z_1, \dots, Z_B) = \mathbb{1}\left\{ \frac{1}{B}\sum_{i=1}^{B} \ell_f(Z_i) - \mathbb{E}[\ell_f] > t \right\}.$$

For all $t > 0$, set $\mathcal{Q}_{\mathcal{F},t} = \{q_{f,t} : f \in \mathcal{F}\}$. There exists $d < \infty$ such that

$$\sup_{t>0} \dim_{VC}\left(\mathcal{Q}_{\mathcal{F},t}\right) < d,$$

with $\dim_{VC}(\mathcal{C})$ the Vapnik-Chervonenkis dimension *of any class $\mathcal{C}$.*

An important corollary of Assumption 8.7 is an upper-bound on the growth functions of the $\mathcal{Q}_{\mathcal{F},t}$. It is obtained by Sauer's lemma, as detailed below.

Corollary 8.8. *Assume that $n_{\mathsf{I}} = KB$, and that the $\mathcal{Q}_{\mathcal{F},t}$ satisfy Assumption 8.7. Then, for all $t > 0$, for any $K \ge \sqrt{\frac{d}{\lambda} n_{\mathsf{I}} \ln n_{\mathsf{I}}}$ it holds*

$$\Pi_{\mathcal{Q}_{\mathcal{F},t}}(K) = \max_{\boldsymbol{Z}^{(k)}\in\mathcal{Z}^B,\ k\le K} \#\left\{ \left(q_{f,t}(\boldsymbol{Z}^{(1)}), \dots, q_{f,t}(\boldsymbol{Z}^{(K)}) \right) : f \in \mathcal{F} \right\} \le e^{\lambda K}.$$

Proof. Sauer's lemma gives

$$\Pi_{\mathcal{Q}_{\mathcal{F},t}}(K) \le \left(1 + \binom{n_{\mathsf{I}}}{B} \right)^d \le n_{\mathsf{I}}^{Bd} = n_{\mathsf{I}}^{\frac{n_{\mathsf{I}} d}{K}}.$$

On the other hand, one has

$$\sqrt{\frac{d}{\lambda}\, n_{\mathsf{I}} \ln n_{\mathsf{I}}} \le K, \quad \text{or again} \quad \frac{n_{\mathsf{I}} d}{K} \ln n_{\mathsf{I}} \le \lambda K, \quad \text{so that} \quad n_{\mathsf{I}}^{\frac{n_{\mathsf{I}} d}{K}} \le e^{\lambda K}.$$

□

We can now state the theorem that bounds the excess risk of the MoRM minimizer.

Theorem 8.9. *Grant Assumptions 8.1, 8.2, 8.4 and 8.7. Furthermore, assume that K satisfies $n_{\mathsf{I}} \ge K \ge \max\left\{16 n_{\mathsf{O}},\ 8\sqrt{d n_{\mathsf{I}} \ln n_{\mathsf{I}}},\ \frac{n_{\mathsf{I}}}{8}\right\}$, and let $\Delta = \frac{1}{16} - \frac{n_{\mathsf{O}}}{K}$, and $B = n_{\mathsf{I}}/K$. Then with probability at least $1 - 6\exp\left(-K \min\left\{2\Delta^2, \frac{1}{64}\right\}\right)$ it holds*

$$\mathcal{R}(\hat{f}_{\mathrm{MoRM}}) \le \mathcal{R}(f^*) + 32 L \max\left(\sigma_{\mathcal{F}} \sqrt{\frac{K}{n_{\mathsf{I}}}}, \frac{16 \mathscr{R}(\mathcal{F})}{n_{\mathsf{I}}} \right).$$

Proof. The beginning of the proof is similar to that of Theorem 8.6. We have

$$\mathcal{R}(\hat{f}_{\text{MoRM}}) - \mathcal{R}(f^*) \leq 2 \sup_{f \in \mathcal{F}} \left| \bar{\mathbb{E}}_{\text{MoRM}}(\ell_f - \mathbb{E}[\ell_f]) \right|,$$

and

$$\mathbb{P}\left\{ \sup_{f \in \mathcal{F}} \bar{\mathbb{E}}_{\text{MoRM}}(\ell_f - \mathbb{E}[\ell_f]) > t \right\} \leq \mathbb{P}\left\{ \sup_{f \in \mathcal{F}} \frac{1}{K} \sum_{k=1}^{K} \mathbb{1}\left\{ \bar{\mathbb{E}}_k[\ell_f] - \mathbb{E}[\ell_f] > t \right\} \geq \frac{1}{2} \right\},$$

where we have used the notation $\bar{\mathbb{E}}_k[\ell_f] = \frac{1}{B}\sum_{i \in \mathcal{B}_k} \ell_f(Z_i) = \frac{1}{B}\sum_{i=1}^{n} \epsilon_{k,i}\ \ell_f(Z_i)$, and $\bar{\mathbb{E}}_{\text{MoRM}}(\ell_f) = \text{median}(\bar{\mathbb{E}}_1[\ell_f], \ldots, \bar{\mathbb{E}}_K[\ell_f])$. For the rest of the proof, we introduce

$$\bar{I}_{k,t}(f) = \mathbb{1}\left\{ \bar{\mathbb{E}}_k[\ell_f] - \mathbb{E}[\ell_f] > t \right\}.$$

The proof is then similar to that of MoRM estimator's concentration (see Section 7.4.2): we condition upon the data and decompose the total deviation into 1) the deviation of $\bar{I}_{k,t}(f)$ from its conditional expectation $\mathbb{E}[\bar{I}_{k,t}(f) \mid \mathcal{S}_n]$, and 2) that of the U-statistic $\mathbb{E}[\bar{I}_{k,t}(f) \mid \mathcal{S}_n]$ from the overall expectation.

Another important thing to notice is that K is now a random variable, that depends on the outliers present in each random block $\mathcal{B}_k$. In order to avoid problems when considering sums over $k \in \mathsf{K}$, we proceed as follows:

$$\begin{aligned}
\sup_{f \in \mathcal{F}} \frac{1}{K} \sum_{k=1}^{K} \mathbb{1}\left\{ \bar{\mathbb{E}}_k[\ell_f] - \mathbb{E}[\ell_f] > t \right\} &\leq \frac{K - n_{\mathsf{K}}}{K} + \sup_{f \in \mathcal{F}} \frac{1}{K} \sum_{k \in \mathsf{K}} \bar{I}_{k,t}(f), \\
&\leq \frac{K - n_{\mathsf{K}}}{K} + \sup_{f \in \mathcal{F}} \frac{1}{K} \sum_{k=1}^{K} \bar{I}^{\mathsf{I}}_{k,t}(f),
\end{aligned}$$

with $\bar{I}^{\mathsf{I}}_{k,t}(f)$ equal to $\bar{I}_{k,t}(f)$, but taken exclusively on *informative* points. The last inequality just means that we have artificially added $K - n_{\mathsf{K}}$ positive terms in order to have a sum independent of K. In the following, expectation are thus taken with respect to $\mathcal{S}_n^{\mathsf{I}}$, the set of informative points, but in order to avoid overwhelming notation, we drop the I superscript. The key part to keep in mind is that *outliers* have been taken care of through the $(K - n_{\mathsf{K}})/K$ term. Finally, we use ψ as introduced in Theorem 8.6. With the notation

$$\bar{\Psi}_{k,t}(f) = \psi\left(\frac{2(\bar{\mathbb{E}}_k[\ell_f] - \mathbb{E}[\ell_f])}{t} \right) \geq \bar{I}_{k,t}(f),$$

we have:

$$\begin{aligned}
\sup_{f \in \mathcal{F}} \frac{1}{K} \sum_{k=1}^{K} \mathbb{1}\left\{ \bar{\mathbb{E}}_k[\ell_f] - \mathbb{E}[\ell_f] > t \right\} \leq\ & \underbrace{\frac{K - n_{\mathsf{K}}}{K}}_{(A)} + \underbrace{\sup_{f \in \mathcal{F}} \frac{1}{K} \sum_{k=1}^{K} \bar{I}_{k,t}(f) - \mathbb{E}\left[\bar{I}_{k,t}(f) \mid \mathcal{S}_n \right]}_{(B)} \\
&+ \underbrace{\sup_{f \in \mathcal{F}} \mathbb{E}\left[\bar{\Psi}_{1,t}(f) \mid \mathcal{S}_n \right] - \mathbb{E}\left[\bar{\Psi}_{1,t}(f) \right]}_{(C)} + \underbrace{\sup_{f \in F} \mathbb{E}\left[\bar{\Psi}_{1,t}(f) \right]}_{(D)}.
\end{aligned}$$

We now bound each term successively.

Bounding (A)**.** As already discussed above, one major difference with MoM minimizer is that n_{K} is now a random variable, that cannot be upper bounded by n_{O} almost surely. By the independence of the blocks, $K - n_{\mathsf{K}}$ is a Binomial random variable, with K trials, and parameter p equal to the probability that one block selects at least one outlier. It can be shown using the same technique as in Appendix D that $p \leq n_{\mathsf{O}}B/n$. Therefore $K - n_{\mathsf{K}}$ as an expected value equal to KBn_{O}/n. In the case where $KB = n$, we recover the upper bound used for MoM. As $K - n_{\mathsf{K}}$ is now random, we have to bound its deviation from its expected value. It can be done using Hoeffding's inequality for instance. We get that for all $z_A > 0$, it holds with probability at least $1 - \exp(-2z_A^2/K)$

$$K - n_{\mathsf{K}} \leq \frac{KBn_{\mathsf{O}}}{n} + z_A,$$

or again, with probability at least $1 - \exp(-2z_A^2 K)$ it holds

$$(A) \leq \frac{Bn_{\mathsf{O}}}{n} + z_A.$$

Bounding (D)**.** It can be bounded exactly as for Equation (8.4), but with the variance of $\bar{\mathbb{E}}_k[\ell_f]$ instead of that of $\hat{\mathbb{E}}_k[\ell_f]$. However, as proved in Chapter 7, the variance is equal in the non-replacement scenario, and using Assumptions 8.11 and 8.13, one finally gets that it holds almost surely

$$(D) \leq \frac{8L^2\sigma_{\mathcal{F}}^2}{Bt^2}.$$

Bounding (B)**.** One may recognize the deviation of an incomplete U-statistic (see Section 6.4.1) from its complete version. This question has been addressed for instance in Clémençon et al. (2013). The key part is to discard the supremum by means of the growth function (or the VC-dimension), hence the necessity to keep $\bar{I}_{k,t}(f)$ that takes a finite number of values, and not using $\bar{\Psi}_{k,t}(f)$ directly. The use of standard Hoeffding's inequality conditioned upon $\mathcal{S}_n$ with the union bound then allows to finish. Corollary 8.8 together with the hypothesis of the theorem implies that

$$\Pi_{\mathcal{Q}_{\mathcal{F},t}}(K) = \max_{\mathcal{S}_n} \left| \left\{ \left(\bar{I}_{1,t}(f), \ldots, \bar{I}_{K,t}(f) \right) : f \in \mathcal{F} \right\} \right| \leq e^{\frac{K}{64}}.$$

Then, it holds with probability at least $1 - \exp\left((1/64 - 2z_B^2)K\right)$

$$(B) \leq z_B.$$

Bounding (C)**.** First, an application of the bounded-differences inequality yields that it holds with probability at least $1 - \exp\left(-2z_C^2 n_{\mathsf{I}}/B^2\right)$

$$(C) \leq \mathbb{E}\left[\sup_{f \in \mathcal{F}} \mathbb{E}\left[\bar{\Psi}_{1,t}(f) \mid \mathcal{S}_n \right] - \mathbb{E}\left[\bar{\Psi}_{1,t}(f) \right] \right] + z_C.$$

The key part is then to consider the conditional expectation as a U-statistic of degree B and to rewrite it as an average of sums of i.i.d. blocks, as it is done in Clémençon et al. (2008). For the sake of simplicity, we assume from here that $n_{\mathsf{I}} = KB$. As

discussed at length in Section 7.1, the MoRM framework is usually way more flexible, but it simplifies the computation here. Indeed, it holds

$$\mathbb{E}\left[\bar{\Psi}_{k,t}(f) \mid \mathcal{S}_n\right] = \frac{1}{\binom{n_{\mathsf{I}}}{B}} \sum_{I} \psi\left(\frac{2\left(\frac{1}{B}\sum_{j=1}^{B} \ell_f(Z_{I_j}) - \mathbb{E}[\ell_f]\right)}{t}\right),$$

$$= \frac{1}{n_{\mathsf{I}}!} \sum_{\pi} \frac{1}{K} \sum_{k=1}^{K} \psi\left(\frac{2\left(\frac{1}{B}\left(\ell_f(Z_{\pi(k)}) + \ell_f(Z_{\pi(K+k)}) + \ldots + \ell_f(Z_{\pi((B-1)K+k)})\right) - \mathbb{E}[\ell_f]\right)}{t}\right).$$

Combining this writing with the previous inequality, we get that it holds with probability at least $1 - \exp(-2z_C^2 K^2/n_{\mathsf{I}})$ by

$$(C) \leq \mathbb{E}\left[\sup_{f\in\mathcal{F}} \frac{1}{K}\sum_{k=1}^{K} \psi\left(\frac{2\left(\frac{1}{B}\sum_{b=1}^{B} \ell_f(Z_{(b-1)K+i}) - \mathbb{E}[\ell_f]\right)}{t}\right) - \mathbb{E}\left[\bar{\Psi}_{1,t}(f)\right]\right] + z_C.$$

From here, we recover an expression similar to what is encountered in the proof of Theorem 8.6, in the sense that the K blocks of variables are independent, so that the symmetrization inequality, the concentration principle, the symmetrization inequality again, and the contraction principle finally yield

$$\begin{aligned}
&\mathbb{E}\left[\sup_{f\in\mathcal{F}} \frac{1}{K}\sum_{k=1}^{K} \psi\left(\frac{2\left(\frac{1}{B}\sum_{b=1}^{B} \ell_f(Z_{(b-1)K+i}) - \mathbb{E}[\ell_f]\right)}{t}\right) - \mathbb{E}\left[\bar{\Psi}_{1,t}(f)\right]\right] \\
&\leq 2\mathbb{E}_{\mathcal{S}_n,\boldsymbol{\sigma}}\left[\sup_{f\in\mathcal{F}} \frac{1}{K}\sum_{k=1}^{K} \sigma_k \psi\left(\frac{2\left(\frac{1}{B}\sum_{b=1}^{B} \ell_f(Z_{(b-1)K+i}) - \mathbb{E}[\ell_f]\right)}{t}\right)\right], \\
&\leq \frac{4}{n_{\mathsf{I}} t}\mathbb{E}_{\mathcal{S}_n,\boldsymbol{\sigma}}\left[\sup_{f\in\mathcal{F}} \sum_{k=1}^{K} \sigma_k \sum_{b=1}^{B} \ell_f(Z_{(b-1)K+i}) - \mathbb{E}[\ell_f]\right], \\
&\leq \frac{8}{n_{\mathsf{I}} t}\mathbb{E}_{\mathcal{S}_n,\boldsymbol{\sigma}}\left[\sup_{f\in\mathcal{F}} \sum_{i\in\mathsf{I}} \sigma_i \ell_f(Z_i)\right], \\
&\leq \frac{8L}{n_{\mathsf{I}} t}\mathscr{R}(\mathcal{F}).
\end{aligned}$$

Combining all bounds. Combining the bounds for $(A), (B), (C)$ and (D), we get that it holds with probability at least $1-\exp(-2z_A^2 K)-\exp((1/64-2z_B^2)K)-\exp(-2z_C^2 K^2/n_{\mathsf{I}})$

$$\sup_{f\in\mathcal{F}} \frac{1}{K}\sum_{k=1}^{K} \mathbb{1}\left\{\bar{\mathbb{E}}_k[\ell_f] - \mathbb{E}[\ell_f] > t\right\} \leq \frac{n_{\mathsf{O}}}{K} + \frac{8L^2 K\sigma_{\mathcal{F}}^2}{n_{\mathsf{I}} t^2} + \frac{8L}{n_{\mathsf{I}} t}\mathscr{R}(\mathcal{F}) + z_A + z_B + z_C.$$

Choosing $z_A = \Delta = \frac{1}{16} - \frac{n_O}{K}$, $z_B = \frac{1}{8}$, $z_C = \sqrt{\frac{n_I}{128K}}$, and $t \geq 16L \max\left(\sigma_{\mathcal{F}}\sqrt{\frac{K}{n_I}}, \frac{16\mathscr{R}(\mathcal{F})}{n_I}\right)$, it holds with probability at least $1 - \exp(-2\Delta^2 K) - 2\exp(-K/64)$

$$\sup_{f\in\mathcal{F}} \frac{1}{K}\sum_{k=1}^{K} \mathbb{1}\left\{\bar{\mathbb{E}}_k[\ell_f] - \mathbb{E}[\ell_f] > t\right\} \leq \frac{1}{16} + \frac{1}{32} + \frac{1}{32} + \frac{1}{8} + \sqrt{\frac{n_I}{128K}}.$$

If $n_I \leq 8K$, the latter is lower than $1/2$, and we get

$$\mathbb{P}\left\{\sup_{f\in\mathcal{F}} \left|\bar{\mathbb{E}}_{\mathrm{MoRM}}(\ell_f - \mathbb{E}[\ell_f])\right| > 16L\max\left(\sigma_{\mathcal{F}}\sqrt{\frac{K}{n_I}}, \frac{16\mathscr{R}(\mathcal{F})}{n_I}\right)\right\} \leq 6e^{-K\min\left\{2\Delta^2, \frac{1}{64}\right\}}.$$

The conclusion follows directly. □

Remark 8.10. *One important remark that can be made is that n_I appears in the bound, rather than n. This comes from the introduction of the expectation of the randomized mean given the (informative) data. Unfortunately, this seems inevitable, and it may constitute a limitation of the randomized framework compared to the partition one. This difference however completely vanishes if the dataset contains no outliers.*

In the spirit of what has been done in Chapter 7, we now consider the extension to minimizers of MoM-like estimates of a U-statistic criterion.

8.3 The MoM-U Minimizers

In this subsection, we extend the MoM minimizer scheme to learning criteria that write as U-statistics. Recall that several examples of such criteria can be found in Section 6.2.2. We are thus interested in finding a function $h : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ that minimizes

$$\mathbb{E}_{Z,Z'}\left[\ell(h, Z, Z')\right],$$

with Z, Z' independent identically distributed as P. As P is unknown, one often choose to minimize instead

$$\frac{2}{n(n-1)}\sum_{i<j}\ell(h, Z_i, Z_j),$$

which is the complete U-statistics based on the full sample $\mathcal{S}_n = \{Z_1, \ldots, Z_n\}$. Here, we rather analyze the properties of

$$\hat{h}_{\mathrm{MoM}-U} \in \operatorname*{argmin}_{h\in\mathcal{H}} \hat{\mathbb{E}}_{\mathrm{MoM}-U}\left[\ell(h, Z, Z')\right],$$

with $\hat{\mathbb{E}}_{\mathrm{MoM}-U}$ being a generic notation for any MoM inspired estimate of a U-statistic risk, that might be based either on the *MoM on Half* estimator (see Section 7.3.3), the *Median-of-U-Statistics* estimator (Section 7.3), or the *Median-of-Randomized-U-Statistics* estimator (Section 7.4). For the *Median-of-U-Statistics* estimator, and with $(B_1, \ldots, B_K)$ denoting a partition of $\mathcal{S}_n$ into K blocks of cardinal B. we have

$$\hat{\mathbb{E}}_{\mathrm{MoU}}[\ell_h] = \frac{2}{B(B-1)}\ \mathrm{median}\left(\sum_{\substack{i,j\in B_1^2\\ i<j}}\ell_h(Z_i, Z_j), \ldots, \sum_{\substack{i,j\in B_K^2\\ i<j}}\ell_h(Z_i, Z_j)\right),$$

Analogously to Theorem 8.6, we need the following assumptions to proceed.

Assumption 8.11. *There exists* $\sigma_{\mathcal{H}} > 0$ *such that:* $\sup_{h\in\mathcal{H}} \|h\|_{L^2} = \sqrt{\mathbb{E}\left[h(X,X')^2\right]} \leq \sigma_{\mathcal{H}}$.

Assumption 8.12. *The training sample* $\mathcal{S}_n$ *is still composed of* informative *points, sampled from the law of interest, indexed by* I*, and* outliers*, potentially adversarial, that are indexed by* O*. The notation* K *and* J *remains unchanged. We now introduce a Rademacher complexity tailored to* U*-statistics:*

$$\mathscr{R}(\mathcal{H}) = \max_{\mathsf{A}\in\{\mathsf{I},\mathsf{J}\}} \mathbb{E}\left[\sup_{h\in\mathcal{H}} \sum_{i\leq\left\lfloor \#\mathsf{A}/2\right\rfloor} \sigma_i\, h\left(Z_i, Z_{\left\lfloor \#\mathsf{A}/2\right\rfloor+i}\right)\right] < +\infty,$$

with $(\sigma_i)_{i\leq\left\lfloor \#\mathsf{A}/2\right\rfloor}$ *being* $\left\lfloor \#\mathsf{A}/2\right\rfloor$ *i.i.d. Rademacher random variables.*

We also need an analogous assumption on the Lipschitz continuity of ℓ.

Assumption 8.13. *There exists* $L > 0$ *such that for all* $z = (x,y), z' = (x',y') \in (\mathcal{X}\times\mathcal{Y})^2$*, for all* $h, h' \in \mathcal{H}^2$*:*

$$|\ell_h(z,z') - \ell_{h'}(z,z')| \leq L|h(x,x') - h'(x,x')|.$$

As suggested in Section 7.3.3, one first way to extend the MoM vision to U-statistics is to build independent pairs $h(Z_i, Z_{\lfloor n/2\rfloor+i})$ and then apply a traditional MoM on the new $\lfloor n/2\rfloor$ observations created. This method precisely exhibits the quantity defined in Assumption 8.12, and one gets a direct adaptation of Theorem 8.6.

Theorem 8.14. *Grant Assumptions 8.11 to 8.13. Assume that* $\lfloor n/2\rfloor > K > 4n_{\mathsf{O}}$*, and let* $\Delta = 1/4 - n_{\mathsf{O}}/K$*. Then with probability at least* $1 - 2\exp(-2\Delta^2 K)$ *it holds*

$$\mathcal{R}(\hat{h}_{\mathrm{MoM}_{1/2}}) \leq \mathcal{R}(h^*) + 16L\max\left(\sigma_{\mathcal{H}}\sqrt{\frac{K}{\lfloor n/2\rfloor}}, \frac{8\mathscr{R}(\mathcal{H})}{\lfloor n/2\rfloor}\right).$$

A bit more complex is the case of the $\hat{\mathbb{E}}_{\mathrm{MoU}}$ minimizer. Indeed, unlike for the $\hat{\mathbb{E}}_{\mathrm{MoM}_{1/2}}$ minimizer, the computation of this estimator involves pairs that are not independent. Therefore, some parts of Theorem 8.6's proof needs to be adapted.

Theorem 8.15. *Grant Assumptions 8.11 to 8.13. Assume that* $n > K > 4n_{\mathsf{O}}$*, and let* $\Delta = 1/4 - n_{\mathsf{O}}/K$*. Then with probability at least* $1 - 2\exp(-2\Delta^2 K)$ *it holds*

$$\mathcal{R}(\hat{h}_{\mathrm{MoU}}) \leq \mathcal{R}(h^*) + 16L\max\left(\sigma_{\mathcal{H}}\sqrt{\frac{K}{n}}, \frac{8\mathscr{R}(\mathcal{H})}{\lfloor n/2\rfloor}\right).$$

Proof. Using the independence between the blocks, everything can be reused until the second use of the symmetrization inequality. We have to bound

$$\mathbb{E}_{\mathcal{S}_n,\boldsymbol{\sigma}}\left[\sup_{h\in\mathcal{H}}\sum_{k\in\mathsf{K}}\sigma_k(\hat{\mathbb{E}}_k[\ell_h] - \mathbb{E}[\ell_h])\right] = \mathbb{E}_{\mathcal{S}_n,\boldsymbol{\sigma}}\left[\sup_{h\in\mathcal{H}}\sum_{k\in\mathsf{K}}\sigma_k\left(\frac{2}{B(B-1)}\sum_{\substack{i,j\in B_k^2\\ i<j}}\ell_h(Z_i,Z_j) - \mathbb{E}[\ell_h]\right)\right].$$

The key part is then to use a decoupling argument, that transforms the U-statistic Rademacher average into a sum of independent observations. This can be done for instance using the same lines as in Lemma A.1 in Clémençon et al. (2008):

$$
\begin{aligned}
&\mathbb{E}_{\mathcal{S}_n,\boldsymbol{\sigma}}\left[\sup_{h\in\mathcal{H}}\sum_{k\in\mathsf{K}}\sigma_k\left(\frac{2}{B(B-1)}\sum_{\substack{i,j\in B_k^2\\ i<j}}\ell_h(Z_i,Z_j)-\mathbb{E}[\ell_h]\right)\right]\\
&=\mathbb{E}_{\mathcal{S}_n,\boldsymbol{\sigma}}\left[\sup_{h\in\mathcal{H}}\sum_{k\in\mathsf{K}}\sigma_k\left(\frac{1}{B!}\sum_{\pi}\frac{1}{\lfloor B/2\rfloor}\sum_{i_k=1}^{\lfloor B/2\rfloor}\ell_h(Z_{i_k},Z_{\lfloor B/2\rfloor+i_k})-\mathbb{E}[\ell_h]\right)\right],\\
&\leq\frac{1}{B!}\sum_{\pi}\mathbb{E}_{\mathcal{S}_n,\boldsymbol{\sigma}}\left[\sup_{h\in\mathcal{H}}\sum_{k\in\mathsf{K}}\sigma_k\left(\frac{1}{\lfloor B/2\rfloor}\sum_{i_k=1}^{\lfloor B/2\rfloor}\ell_h(Z_{i_k},Z_{\lfloor B/2\rfloor+i_k})-\mathbb{E}[\ell_h]\right)\right],\\
&=\mathbb{E}_{\mathcal{S}_n,\boldsymbol{\sigma}}\left[\sup_{h\in\mathcal{H}}\sum_{k\in\mathsf{K}}\sigma_k\left(\frac{1}{\lfloor B/2\rfloor}\sum_{i_k=1}^{\lfloor B/2\rfloor}\ell_h(Z_{i_k},Z_{\lfloor B/2\rfloor+i_k})-\mathbb{E}[\ell_h]\right)\right],\\
&\leq\frac{2}{\lfloor B/2\rfloor}\mathbb{E}_{\mathcal{S}_n,\boldsymbol{\sigma}}\left[\sup_{h\in\mathcal{H}}\sum_{i\in\mathsf{J}}\sigma_i\;\ell_h\left(Z_i,Z_{\lfloor n/2\rfloor+i}\right)\right],\\
&\leq\frac{2L}{\lfloor B/2\rfloor}\mathscr{R}(\mathcal{H}),
\end{aligned}
$$

where we have used successively a specific writing of U-statistics, Jensen's inequality, the symmetrization inequality on independent observations already used in the proof of Theorem 8.6 (the notation i_k refers to the i^{th} index of block B_k), and the Lipschitz continuity of ℓ. The rest of the proof is analogous to that of Theorem 8.6. □

Remark 8.16. *Theorems 8.14 and 8.15 are very similar. The advantage of using complete U-statistics instead of the average based on independent pairs can be seen on the range of admissible K's, as well as on the first term in the* max. *Recovering the same second term is expected, as the Rademacher averages over all pairs that appear when using* MoU *are converted into Rademacher averages over independent pairs, as for* $\text{MoM}_{1/2}$.

Minimizing MoRU or MoIU estimates of the risk suffers from the same drawbacks as MoRM. In particular, the absence of independence between the blocks prevents from the use of standard symmetrization arguments, and makes results harder to derive.

The last part of this section is devoted to the design of algorithms capable of computing the solutions to the above mentioned problems.

8.4 The Mo(R)M and Mo(R)U Gradient Descents

Now that the benefits of minimizing a MoM estimate of the risk have been established, we design algorithms to compute the desired solutions. They are based on Gradient Descent (GD), and adapted to the MoM framework.

Assume that $\mathcal{F}$ (respectively $\mathcal{H}$ for U-statistics) is parametrized, so that $f=f_u, u\in\mathbb{R}^p$ ($h=h_u$ respectively). Minimizing the MoM risk can be done as described in Figure 8.1.

Median-of-Means Gradient Descent

1. Partition the dataset.
2. Compute the empirical risk on each block.
3. Select the block with the median empirical risk.
4. Compute a GD step based on the batch that corresponds to the median block.
5. Iterate.

Figure 8.1 – Median-of-Means Gradient Descent (MoM GD)

This algorithm can be seen as a batch Gradient Descent, with a specific criterion to select the batch: it is the block with median empirical risk. This way, we can expect that blocks containing possible outliers are not selected, ensuring a safe descent. On the contrary, this algorithm should yield slower convergence on non-corrupted datasets, as the *least informative* data points are selected at each iteration. These phenomena are illustrated in Section 8.6 for instance.

However, it has been shown empirically (Remark 5 in Lecué et al. (2018)) that the previously described procedure often finds local minima. This happens when the same block is always selected as the median block, and one then minimizes the empirical risk based on this specific block. In order to avoid this problem, one can artificially decide to randomize the partition at each iteration. This way, it is highly improbable that the same block is selected twice in a row.

It is direct to see that the procedure described in steps 1-5 above can be readily adapted to MoRM. Furthermore, using a MoRM estimate, random by nature, directly prevents from finding a local minimum. There is no need to introduce an artificial randomization through the change of partition. The same remark can be made for the MoU/MoRU gradient descents. All the procedures are summarized in Algorithm 8.1. Notice that in order to simplify the notation, the random blocks usually denoted $\mathcal{B}_k$ are denoted as B_k in Algorithm 8.1, so that the notation can be shared among different versions.

To analyze the convergence of Algorithm 8.1's iterates, we need the next assumptions. In order to avoid the statements of four different theorems (MoM, MoRM, MoU, MoRU), we state general assumptions valid in all settings. It is of course possible to restrict them to the particular case of interest.

Assumption 8.17. *There exists $M_\ell > 0$ such that $\forall u \in \mathbb{R}^p$, and P-almost $z, z' \in \mathcal{Z}^2$,*

$$\left\|\nabla_u \ell(f_u, z)\right\|_2 \leq M_\ell,$$
$$\left\|\nabla_u \ell(h_u, z, z')\right\|_2 \leq M_\ell.$$

Assumption 8.18. *The sequence of steps $(\gamma_t)_{t\in\mathbb{N}^*}$ satisfies*

$$\sum_{t\in\mathbb{N}^*} \gamma_t = \infty, \qquad \sum_{t\in\mathbb{N}^*} \gamma_t^2 < \infty.$$

Algorithm 8.1 MoM/MoRM/MoU/MoRU Gradient Descents

input : $\mathcal{S}_n = \{Z_1, \dots, Z_n\}$, $K \in \mathbb{N}^*$ ($\leq n/2$ for MoM/MoU), $B \leq n$ for MoRM/MoRU
$T \in \mathbb{N}^*$, $(\gamma_t)_{t \leq T} \in \mathbb{R}_+^T$

init : $u_0 \in \mathbb{R}^p$

for *epoch* t *from* 1 *to* T **do**

// Select the median block

if MoM *or* MoU **then**

Choose a random permutation π of $[\![1, n]\!]$

Build a partition $B_1, \dots, B_K$ of $\{\pi(1), \dots, \pi(n)\}$

if MoRM *or* MoRU **then**

Sample K i.i.d. SWoR blocks $B_1, \dots, B_K$ among $\mathcal{S}_n$

for $k \leq K$ **do**

$R_k = \sum_{i \in B_k} \ell(f_{u_t}, Z_i)$ // MoM and MoRM

$R_k = \sum_{i<j \in B_k^2} \ell(h_{u_t}, Z_i, Z_j)$ // MoU and MoRU

B_{med} such that $\text{median}(R_1, \dots, R_k) = R_{B_{\text{med}}}$

// Gradient step on selected block

$u_{t+1} = u_t - \gamma_t \sum_{i \in B_{\text{med}}} \ell(f_{u_t}, Z_i)$ // MoM and MoRM

$u_{t+1} = u_t - \gamma_t \sum_{i<j \in B_{\text{med}}^2} \ell(h_{u_t}, Z_i, Z_j)$ // MoU and MoRU

return u_T

Assumption 8.19. *For almost all datasets,* $\exists u_{\min} \in \mathbb{R}^p$ *unique such that*

$$u_{\min} = \operatorname*{argmin}_{u \in \mathbb{R}^p} \mathbb{E}_\pi \left[\hat{\mathbb{E}}_{\text{MoM},\pi}(\ell_{f_u}) \mid \mathcal{S}_n \right],$$

$$u_{\min} = \operatorname*{argmin}_{u \in \mathbb{R}^p} \mathbb{E}_{\boldsymbol{\epsilon}} \left[\hat{\mathbb{E}}_{\text{MoRM}}(\ell_{f_u}) \mid \mathcal{S}_n \right],$$

$$u_{\min} = \operatorname*{argmin}_{u \in \mathbb{R}^p} \mathbb{E}_\pi \left[\hat{\mathbb{E}}_{\text{MoU},\pi}(\ell_{h_u}) \mid \mathcal{S}_n \right],$$

$$u_{\min} = \operatorname*{argmin}_{u \in \mathbb{R}^p} \mathbb{E}_{\boldsymbol{\epsilon}} \left[\hat{\mathbb{E}}_{\text{MoRU}}(\ell_{h_u}) \mid \mathcal{S}_n \right].$$

Assumption 8.20. *For almost all datasets, for almost all* $u \in \mathbb{R}^p$*, for all* $\varepsilon > 0$*,*

$$\inf_{u, \|u - u_{\min}\| > \varepsilon} (u - u_{\min})^\top \mathbb{E}_\pi \left[\nabla_u \hat{\mathbb{E}}_{\text{MoM},\pi}(\ell_{f_u}) \mid \mathcal{S}_n \right] < 0,$$

$$\inf_{u, \|u - u_{\min}\| > \varepsilon} (u - u_{\min})^\top \mathbb{E}_{\boldsymbol{\epsilon}} \left[\nabla_u \hat{\mathbb{E}}_{\text{MoRM}}(\ell_{f_u}) \mid \mathcal{S}_n \right] < 0,$$

$$\inf_{u, \|u - u_{\min}\| > \varepsilon} (u - u_{\min})^\top \mathbb{E}_\pi \left[\nabla_u \hat{\mathbb{E}}_{\text{MoU},\pi}(\ell_{h_u}) \mid \mathcal{S}_n \right] < 0,$$

$$\inf_{u, \|u - u_{\min}\| > \varepsilon} (u - u_{\min})^\top \mathbb{E}_{\boldsymbol{\epsilon}} \left[\nabla_u \hat{\mathbb{E}}_{\text{MoRU}}(\ell_{h_u}) \mid \mathcal{S}_n \right] < 0.$$

Assumption 8.21. *For almost all datasets, for almost all $u \in \mathbb{R}^p$, there exists an open convex set $\mathscr{B}$ containing u such that for any partition of $[\![1, n]\!]$ – any K i.i.d. SWoR blocks – $B_1, \dots, B_K$, there exists $k_{\text{med}} \leq K$ such that for all $v \in \mathscr{B}$*

$$R_{k_{\text{med}}} = \text{median}(R_1, \dots, R_k),$$

with the notation introduced in Algorithm 8.1.

We are now ready to state the convergence theorem.

Theorem 8.22. *Grant Assumptions 8.17 to 8.21. Then, the MoM/MoRM/MoU/MoRU gradient descent algorithms (see Algorithm 8.1) converge:*

$$\|u_T - u_{\min}\| \xrightarrow[T\to\infty]{a.s.} 0.$$

Proof. Assumption 8.21 ensures that for any $t \in \mathbb{N}^*$, there exists a open convex set $\mathscr{B}$ containing u_{t-1} such that for all $u \in \mathscr{B}$ it holds

$$\frac{1}{B} \sum_{i \in B_{k_{\text{med}}}} \nabla_u \ell(f_u, Z_i) = \nabla_u \hat{\mathbb{E}}_{\text{MoM}}(\ell_{f_u}) \quad \left(\text{or } = \nabla_u \hat{\mathbb{E}}_{\text{MoRM}}(\ell_{f_u})\right),$$

$$\frac{2}{B(B-1)} \sum_{\substack{i,j \in B^2_{k_{\text{med}}} \\ i<j}} \nabla_u \ell(h_u, Z_i, Z_j) = \nabla_u \hat{\mathbb{E}}_{\text{MoU}}(\ell_{h_u}) \quad \left(\text{or } = \nabla_u \hat{\mathbb{E}}_{\text{MoRU}}(\ell_{h_u})\right).$$

From here, the rest of the proof is identical to that of the consistency of Stochastic Gradient Descent (SGD), see *e.g.* Bottou (1998). □

We conclude this subsection with a remark on the minimum attained by Algorithm 8.1.

Remark 8.23. *It is important to notice that the minimum attained by Algorithm 8.1 is not a* MoM*, minimizer, but rather the minimizer of the expectation of the* MoM *with respect to all possible permutation in $\mathfrak{S}_n$. The same goes for* MoRM*, with an expectation over the selection vectors $\boldsymbol{\epsilon}$. However, one can expect a concentration of individual* MoM *(respectively* MoRM*,* MoU*,* MoRU*) minimizers around the minimizer of the expectation. Recall that in Chapter 7, we have studied the concentration of* MoRM *and* MoRU *estimates around their expectation with respect to both $\boldsymbol{\epsilon}$ and $\mathcal{S}_n$. We can reasonably expect smaller deviations from an expectation taken with respect to $\boldsymbol{\epsilon}$ only, at fixed $\mathcal{S}_n$.*

The next section investigates another way to use the MoM methodology in learning. The introduced approach completely breaks with ERM or MoM minimizing. Namely, it is based on *tournament procedures*.

8.5 Tournament Procedures

Statistical learning by tournament procedure has been first introduced in Lugosi and Mendelson (2016). It basically consists in segmenting the training data into blocks of equal size, on which the statistical performance of every pair of candidate decision rules are compared. The prediction rule with highest performance on the majority of the blocks is declared as the winner. In the context of nonparametric regression, functions

having won all their duels have been shown to outperform empirical risk minimizers with respect to the mean squared error under minimal assumptions, while exhibiting robustness properties. In Section 8.5.1, we recall the standard tournament procedure, together with its theoretical guarantees. Section 8.5.2 is then devoted to the extension of the procedure to learning problems for which the performance criterion takes the form of an expectation over pairs of observations, as may be the case in pairwise ranking, clustering or metric learning (see Section 6.2.2).

8.5.1 Standard Tournament Procedure

In this subsection, we recall the standard tournament procedure introduced in Lugosi and Mendelson (2016). For now, we restrict our attention to the least squares scalar regression. Given $Z = (X, Y)$ a random variable valued in $\mathcal{X} \times \mathbb{R}$ according to an unknown probability P, our goal is to find

$$f^* = \operatorname*{argmin}_{f \in \mathcal{F}} \mathbb{E}_P \left[\Big(f(X) - Y \Big)^2 \right],$$

with $\mathcal{F}$ a given class of functions contained in $\mathbb{R}^{\mathcal{X}}$. The two classical ways of assessing the output $\hat{f}$ of an algorithm are the L_2 *distance to* f^*

$$\sqrt{\mathbb{E}\left[\Big(\hat{f}(X) - f^*(X) \Big)^2 \right]},$$

and the *excess risk*

$$\mathcal{R}(\hat{f}) - \mathcal{R}(f^*) = \mathbb{E}\left[\Big(\hat{f}(X) - Y \Big)^2 \right] - \mathbb{E}\left[\Big(f^*(X) - Y \Big)^2 \right].$$

While ERM can be shown to provide good solutions when $\mathcal{F}$ is convex and Y is sub-Gaussian (Lecué and Mendelson, 2013), its sensitivity to a small number of atypical points makes it unusable when the data is heavy-tailed. The objective of Theorem 8.25 is to establish the existence of a procedure that exhibits a good $\hat{f}$ even when the data has not well-behaved tails. Before stating the theorem, we need the following assumption.

Assumption 8.24. *The class $\mathcal{F}$ is locally compact and convex. The output random variable Y is square integrable, and there exists $L, \sigma > 0$ such that*

- $\forall (f, f') \in \mathcal{F}^2, \quad \|f - f'\|_{L_4} \leq L \|f - f'\|_{L_2}$,
- $\forall f \in \mathcal{F}, \quad \|f - Y\|_{L_4} \leq L \|f - Y\|_{L_2}$,
- $\|f^* - Y\|_{L_2} \leq \sigma$.

Theorem 8.25. *Grant Assumption 8.24. Then, there exist $c_0, r > 0$ that depend only on L, σ and f^* such that there exists a procedure that based on $\mathcal{S}_n, L, \sigma, r$ selects a function $\hat{f} \in \mathcal{F}$ such that it holds with probability at least $1 - \exp(-c_0 n \min\{1, \sigma^{-2} r^2\})$*

$$\left\| \hat{f} - f^* \right\|_{L_2} \leq cr,$$

and

$$\mathcal{R}(\hat{f}) - \mathcal{R}(f^*) \leq (cr)^2.$$

Standard Tournament Procedure

1. For every pair of candidate functions $(f, f') \in \mathcal{F}^2$, compute a MoM estimate of $\|f - f'\|_{L_1}$ based on the first sub-dataset:

$$\mathscr{D}_{\mathcal{S}_n^{(1)}}(f, f') = \frac{1}{B} \operatorname{median}\left(\sum_{i \in B_k} |f(X_i) - f'(X_i)|, \quad k \leq K \right).$$

2. If $\mathscr{D}_{\mathcal{S}_n^{(1)}}(f, f')$ is *large enough*, then compute a *match* between f and f', *i.e.* a MoM estimate of $\mathcal{R}(f) - \mathcal{R}(f')$ based on the second sub-dataset:

$$\Phi_{\mathcal{S}_n^{(2)}}(f, f') = \frac{1}{B} \operatorname{median}\left(\sum_{i \in B_k} (f(X_i) - Y_i)^2 - (f'(X_i) - Y_i)^2, \quad k \leq K \right).$$

   If it is negative, *i.e.* f has a lower empirical risk on a majority of blocks, then f wins the match. Otherwise, f' is declared as the winner.

3. Gather all candidates that have won all the matches they have been allowed to play, called *champions*. Finally, based on the third sub-dataset, compute a MoM estimate of $(f' - f)(f - Y)$ for every pair (f', f) of champions:

$$\Psi_{\mathcal{S}_n^{(3)}}(f', f) = \frac{1}{B} \operatorname{median}\left(\sum_{i \in B_k} (f'(X_i) - f(X_i)) \cdot (f(X_i) - Y_i), \quad k \leq K \right).$$

   The output of the procedure is a champion $\hat{f}$ winning all its last round home matches $\Psi_{\mathcal{S}_n^{(3)}}(f', f)$ described above.

Figure 8.2 – Standard Tournament Procedure

The proof of Theorem 8.25 can be found in Lugosi and Mendelson (2016). We shall now focus on the description of the procedure that achieves Theorem 8.25's performance. First assume, without loss of generality, that the original dataset $\mathcal{S}_n$ is actually of size $3n$, and that it is divided into three parts of size n: $\mathcal{S}_n^{(1)}, \mathcal{S}_n^{(2)}$ and $\mathcal{S}_n^{(3)}$. The tournament procedure is then summarized in Figure 8.2.

The rationale behind this approach is the following. If f^* is one of the two candidates, since it is only allowed to play matches against distant candidates, it should hopefully win all of them with high probability. Therefore f^* is in the final champions pool, and it can be shown that it should win all its champion's home matches with high probability. Before extending the tournament to pairwise criteria, three remarks can be made.

Remark 8.26. *Comparing Theorem 8.6 and Theorem 8.25, one can see that the latter exhibits faster rates. Actually, it can be seen in the proof that all champions that have won their first matches satisfy* $\mathcal{R}(f) - \mathcal{R}(f^*) \leq \|f - f^*\|_{L_2} \lesssim \sqrt{\ln(1/\delta)/n}$ *with probability* $1 - \delta$*, as the solutions of the* MoM *minimizing. The faster rate* $\ln(1/\delta)/n$ *is obtained after the computation of the final round among the champions.*

Remark 8.27. *As discussed at length in Lugosi and Mendelson (2016), computing the tournament winner is a nontrivial problem as soon as $\mathcal{F}$ does not contain a finite number of elements. However, one could alternatively consider performing a tournament on an ε-coverage of $\mathcal{F}$, while controlling the approximation error of this coverage.*

Remark 8.28. *The adaptation to other losses than the square error is most of the time easy. Indeed, for a positive loss function ℓ, it is always possible to rewrite $\ell(f, x, y) = (\sqrt{\ell(f, z)}) - 0)^2$. Up to a change of variable, x becoming z, and y becoming 0, the previous procedure allows to find an optimal $\sqrt{\ell(f^*, \cdot)}$. If $\ell(f, x, y)$ can be written as $\tilde{\ell}(f(x) - y)$, with $\tilde{\ell}$ invertible, it is then easy to recover f^*.*

The next subsection extends the tournament procedure to pairwise criteria.

8.5.2 Pairwise Tournament Procedure

As shall be seen in the following, the tournament procedure described in the previous subsection naturally extends to criteria that are based on pairs. As a reminder, we first focus on finding

$$h^* = \operatorname*{argmin}_{h \in \mathcal{H}} \mathbb{E}_{P \otimes P} \left[\Big(h(X, X') - t(Y, Y') \Big)^2 \right],$$

with $\mathcal{H} \subset \mathbb{R}^{\mathcal{X} \times \mathcal{X}}$ the function class over which the criterion is optimized and $t : \mathcal{Y}^2 \to \mathbb{R}$ a given function that links a pair to its target. This might be *e.g.* a $0-1$ label indicating if the observations pertain to the same class, or some precomputed distance. Notice that the extension to the general case

$$h^* = \operatorname*{argmin}_{h \in \mathcal{H}} \mathbb{E}_{P \otimes P} \Big[\ell_h(Z, Z') \Big]$$

can be achieved in a similar way as that described in Remark 8.28.

We first need an analog to Assumption 8.24. We then state the main theorem.

Assumption 8.29. *The class $\mathcal{H}$ is locally compact and convex. The output random variable $t(Y, Y')$ is square integrable, and there exists $L, \sigma > 0$ such that*

- $\forall (h, h') \in \mathcal{H}^2, \quad \|h - h'\|_{L_4} \leq L \|h - h'\|_{L_2}$,
- $\forall h \in \mathcal{H}, \quad \|h - t(Y, Y')\|_{L_4} \leq L \|h - t(Y, Y')\|_{L_2}$,
- $\|h^* - t(Y, Y')\|_{L_2} \leq \sigma$.

Theorem 8.30. *Grant Assumption 8.29. Then, there exist $c_0, r > 0$ that depend only on L, σ and h^* such that there exists a procedure that based on $\mathcal{S}_n, L, \sigma, r$ selects a function $\hat{h} \in \mathcal{H}$ such that it holds with probability at least $1 - \exp(-c_0 n \min\{1, \sigma^{-2} r^2\})$*

$$\left\| \hat{h} - h^* \right\|_{L_2} \leq cr,$$

and

$$\mathcal{R}(\hat{h}) - \mathcal{R}(h^*) \leq (cr)^2.$$

Unsurprisingly, this procedure is a tournament where MoM estimates are replaced by MoU estimates. It is summarized in Figure 8.3 (in order to avoid the introduction of new notation, it remains unchanged although U-statistics are now used). As for the proof, it is a direct adaptation of that of Theorem 8.25, with the use of U-statistics tools when necessary. We list below the main changes.

Pairwise Tournament Procedure

1. For every pair of candidate functions $(h, h') \in \mathcal{H}^2$, compute a MoU estimate of $\|h - h'\|_{L_1}$ based on the first sub-dataset:

$$\mathscr{D}_{\mathcal{S}_n^{(1)}}(h, h') = \frac{2}{B(B-1)} \operatorname{median}\left(\sum_{\substack{i,j \in B_k^2 \\ i<j}} \Big| h(X_i, X_j) - h'(X_i, X_j) \Big|, \quad k \leq K \right).$$

2. If $\mathscr{D}_{\mathcal{S}_n^{(1)}}(h, h')$ is *large enough*, then compute a *match* between h and h', *i.e.* a MoU estimate of $\mathcal{R}(h) - \mathcal{R}(h')$ based on the second sub-dataset:

$$\Phi_{\mathcal{S}_n^{(2)}}(h, h') = \frac{2}{B(B-1)} \operatorname{median}\left(\sum_{\substack{i,j \in B_k^2 \\ i<j}} \Big(h(X_i, X_j) - t(Y_i, Y_j) \Big)^2 - \Big(h'(X_i, X_j) - t(Y_i, Y_j) \Big)^2, \quad k \leq K \right).$$

   If it is negative, *i.e.* h has a lower empirical risk on a majority of blocks, then h wins the match. Otherwise, h' is declared as the winner.

3. Gather all candidates that have won all the matches they have been allowed to play, called *champions*. Finally, based on the third sub-dataset, compute a MoU estimate of $(h' - h)(h - t)$ for every pair (h', h) of champions:

$$\Psi_{\mathcal{S}_n^{(3)}}(h', h) = \frac{2}{B(B-1)} \operatorname{median}\left(\sum_{\substack{i,j \in B_k^2 \\ i<j}} \Big(h'(X_i, X_j) - h(X_i, X_j) \Big) \cdot \Big(h(X_i, X_j) - t(Y_i, Y_j) \Big), \quad k \leq K \right).$$

   The output of the procedure is a champion $\hat{h}$ winning all its last round home matches $\Psi_{\mathcal{S}_n^{(3)}}(h', h)$ described above.

Figure 8.3 – Pairwise Tournament Procedure

Definition 8.31. *Let $\lambda_{\mathbb{Q}}(\kappa,\eta,h)$ and $\lambda_{\mathbb{M}}(\kappa,\eta,h)$ be defined as in Lugosi and Mendelson (2016) (see Definitions 2.2 and 2.3 therein). The definitions of $r_E(\kappa,h)$ and $\bar{r}_{\mathbb{M}}(\kappa,h)$ needs however to be adapter. Indeed, let*

$$r_E(\kappa,h) = \inf\left\{ r : \mathbb{E} \sup_{u\in\mathcal{H}_{h,r}} \left| \frac{1}{\sqrt{\lfloor B/2\rfloor}} \sum_{i=1}^{\lfloor B/2\rfloor} \sigma_i\ u(X_i, X_{\lfloor B/2\rfloor+i}) \right| \le \kappa\sqrt{\lfloor B/2\rfloor} r \right\},$$

and

$$\bar{r}_{\mathbb{M}}(\kappa,h) = \inf\left\{ r : \mathbb{E} \sup_{u\in\mathcal{H}_{h,r}} \left| \frac{1}{\sqrt{\lfloor B/2\rfloor}} \sum_{i=1}^{\lfloor B/2\rfloor} \sigma_i\ u(X_i, X_{\lfloor B/2\rfloor+i}) \cdot h(X_i, X_{\lfloor B/2\rfloor+i}) \right| \le \kappa\sqrt{\lfloor B/2\rfloor} r^2 \right\}.$$

Proof of the Oracle Distance. The goal of the oracle distance is to allow matches between distant candidates only. This way, if h^* is selected as one of the candidates, it should win its match against a very different opponent. The fact that the MoU estimate of the L_1 distance between two candidates is a good approximate of their L_2 distance is guaranteed by the following proposition.

Proposition 8.32. *There exist constant $\kappa,\eta,l,c > 0$ and $0 < \alpha < 1 < \beta$, all of them depending only on L for which the following holds. For a fixed $h^* \in \mathcal{H}$, let $d^* = \max\{\lambda_{\mathbb{Q}}(\kappa,\eta,h^*), r_E(\kappa,h^*)\}$. For any $r \ge d^*$, with probability at least $1 - 2\exp(-cn)$, for every $h \in \mathcal{H}$, one has*

- *If $\mathscr{D}_{\mathcal{S}_n^{(1)}}(h,h^*) \ge \beta r$ then $\beta^{-1}\mathscr{D}_{\mathcal{S}_n^{(1)}}(h,h^*) \le \|h-h^*\|_{L_2} \le \alpha^{-1}\mathscr{D}_{\mathcal{S}_n^{(1)}}(h,h^*)$.*
- *If $\mathscr{D}_{\mathcal{S}_n^{(1)}}(h,h^*) < \beta r$ then $\|h-h^*\|_{L_2} \le (\beta/\alpha)\mathscr{D}_{\mathcal{S}_n^{(1)}}(h,h^*)$.*

This proposition is an adaptation of Proposition 3.2 in Lugosi and Mendelson (2016), which is itself a reproduction of Theorem 3.3 in Mendelson (2017). We sketch first the adaptations of the lemmas used in Mendelson (2017).

Lemma 8.33. *For every $q > 2$ and $L \ge 1$, there are constants B and κ_0 that depend only on q and L for which the following holds. If $\|u\|_{L_q} \le L\|u\|_{L_2}$ and $X_1,\dots,X_B$ are independent copies of X, then*

$$\mathbb{P}\left\{ \frac{2}{B(B-1)} \sum_{1\le i<j\le B} |u(X_i,X_j)| \ge \kappa_0\|u\|_{L_2} \right\} \ge 0.9.$$

Proof. The proof is analogous to that of Lemma 3.4 in Mendelson (2017), except that a version of the Berry-Esseen theorem for U-statistics (Callaert et al., 1978) is used instead of the standard one. □

Lemma 8.34. *For every $q > 2$ and $L \ge 1$, there is a constant κ_1 that depends only on q and L for which the following holds. If $X_1,\dots,X_B$ are independent copies of X, then*

$$\mathbb{P}\left\{ \frac{2}{B(B-1)} \sum_{1\le i<j\le B} |u(X_i,X_j)| \le \kappa_1\|u\|_{L_2} \right\} \ge 0.9.$$

Proof. As $\left\{ \frac{2}{B(B-1)} \sum_{i<j} |u(X_i, X_j)| \geq \kappa_1 \|u\|_{L_2} \right\} \subset \left\{ \exists i < j,\ |u(X_i, X_j)| \geq \kappa_1 \|u\|_{L_2} \right\}$, Chebyshev inequality yields

$$\mathbb{P}\left\{ \frac{2}{B(B-1)} \sum_{1 \leq i < j \leq B} |u(X_i, X_j)| \geq \kappa_1 \|u\|_{L_2} \right\} \leq \frac{B(B-1)}{2} \mathbb{P}\left\{ |u(X_i, X_j)| \geq \kappa_1 \|u\|_{L_2} \right\},$$
$$\leq \frac{B(B-1)}{2\kappa_1^2}.$$

Since B only depends on q and L (see proof of Lemma 8.33), so does κ_1. □

Proof of Proposition 8.32. Using Lemma 8.33 and Lemma 8.34 with $u = h - h^*$, together with the union bound, it holds that for every block $\mathcal{B}_k$ one has with probability at least 0.8

$$\kappa_0 \|h - h^*\|_{L_2} \leq \bar{U}_k(|h - h^*|) \leq \kappa_1 \|h - h^*\|_{L_2}. \tag{8.6}$$

Denoting by I_k the indicator of this event, and by $\bar{I}_k$ its complementary, we have $\mathbb{E}[\bar{I}_k] \leq 0.2$. Moreover,

$$\mathbb{P}\left\{ \sum_{k=1}^{K} I_k \geq 0.7K \right\} = 1 - \mathbb{P}\left\{ \frac{1}{K} \sum_{k=1}^{K} \bar{I}_k \geq 0.3 \right\}.$$

When a MoU estimate is used, the I_k are independent, since built on disjoint blocks, and the concentration of Binomial random variables allows to finish. But interestingly, when a MoRU is used, it is straightforward to see that the last term is exactly the same quantity as the one involved in MoU's estimator concentration (Section 7.3.2). The same method can thus be used since an upper bound of $\mathbb{E}[\bar{I}_k]$ is already available. Precisely, choosing $\tau = 0.25 < 0.3$ and recalling $B = \lfloor n/K \rfloor$, it holds

$$\mathbb{P}\left\{ \frac{1}{K} \sum_{k=1}^{K} \bar{I}_k \geq 0.3 \right\} \leq 2 \exp\left(-2(0.05)^2 K \right).$$

So the number of blocks which satisfy (8.6) is larger than $0.7K$ with probability at least $1 - 2\exp(-c_1 K)$ for some positive constant c_1. The rest of the proof is similar to that of Proposition 3.2 in Lugosi and Mendelson (2016). □

Proof of the First Round. As explained above, the goal of the first round is to identify h^*, as it can be shown that it wins all its matches with high probability. It is the purpose of the following proposition.

Proposition 8.35. *Under the assumptions of Theorem 8.25, and using its notation, with probability at least*

$$1 - 2\exp(-c_0 n \min\{1, \sigma^{-2} r^2\}),$$

$\forall\ h \in \mathcal{H}$ *if* $\Phi_{\mathcal{S}_n^{(2)}}(h, h^*) \geq \beta r$ *then* h^* *defeats* h. *In particular* $h^* \in H_{\text{champ}}$, *and* $\forall\ h \in H_{\text{champ}}$, $\|h - h^*\|_{L_2} \leq (\beta/\alpha) r$.

Proof of Proposition 8.35. This proof carefully follows that of Proposition 3.5 in Lugosi and Mendelson (2016) (see Section 5.1 therein), so that only changes induced by pairwise objectives are detailed here.

Proof of pairwise Lemma 5.1 in Lugosi and Mendelson (2016). First, one may rewrite

$$\mathbb{Q}_{h,h'} = \frac{2}{B(B-1)} \sum_{i<j} \Big(h(X_i, X_j) - h'(X_i, X_j) \Big)^2,$$

$$\mathbb{M}_{h,h'} = \frac{4}{B(B-1)} \sum_{i<j} \Big(h(X_i, X_j) - h'(X_i, X_j) \Big) \cdot \Big(h'(X_i, X_j) - t(Y_i, Y_j) \Big),$$

$$R_k(u,t) = \Big| \{(i,j) \in \mathcal{B}_k^2 : i < j, |u(X_i, X_j)| \geq t\} \Big| = \sum_{i<j \in \mathcal{B}_k^2} \mathbb{1}\{|u(X_i, X_j) \geq t|\}.$$

Since pairs are not independent from each others, even if the X_i's are, one cannot use directly the proposed method. Instead, Hoeffding's inequality for U-statistics gives that the probability of each $R_k(h-h', \kappa_0 r)$ to be greater than $\frac{B(B-1)\rho_0}{4}$ is greater than $1-\exp(-B\rho_0^2/4)$. For τ small enough, we still have that this probability is greater than $1-\tau/12$. Aggregating the Bernoulli may be done in two ways. If we deal with a MoU estimate, the independence between the blocks leads to the same conclusion. If a MoRU is used, the remark made for Proposition 8.32 is still valid, and one can conclude.

The next difficulty in proving pairwise Lemma 5.1 arises with the bounded differences inequality for Ψ. If a MoU estimate is used, changing one sample X_i' only affects one block, and generates a $1/K$ difference at most, exactly like with MoM, so that the bound holds the same way. On the contrary, if a MoRU estimate is used, the replaced sample may contaminate all K blocks. The analysis of the MoRU behavior in that case is a bit trickier, and we restrict ourselves to MoU estimates for the matches.

The end of the proof uses symmetrization arguments. This kind of arguments still apply to U-statistics after decoupling, see *e.g.* Arcones and Gine (1993); de la Peña (1992), or p.140 of the monograph by de la Peña and Giné (1999). The analysis based on Lemma A.1. in Clémençon et al. (2008), or the proof of Theorem 8.9 are examples of how to use such arguments. So is the proof of pairwise Lemma 5.1 completed. □

Proof of pairwise Lemma 5.2.

$$\begin{aligned}
\mathbb{P}\left\{ \left| \frac{2}{B(B-1)} \sum_{i<j} U_{i,j} - \mathbb{E}U \right| \geq t \right\} &\leq \frac{2}{B(B-1)t} \mathbb{E} \left| \sum_{i<j} U_{i,j} - \mathbb{E}U \right|, \\
&\leq \frac{2}{B(B-1)t} \sqrt{\mathbb{E} \left| \sum_{i<j} U_{i,j} - \mathbb{E}U \right|^2}, \\
&\leq \frac{2}{B(B-1)t} \sqrt{\sum_{i<j,\ k<l} \mathbb{E}[U_{i,j}U_{j,k}] - (\mathbb{E}U)^2}, \\
&\leq \frac{2}{B(B-1)t} \sqrt{\frac{B(B-1)}{2} \left(\mathbb{E}[U^2] - \big(\mathbb{E}U\big)^2 \right) + \frac{B(B-1)(B-2)}{2} \sigma_1^2}, \\
&\leq \frac{\sqrt{2(B-2)}}{\sqrt{B(B-1)t}} \|U\|_{L_2} \leq \frac{\sqrt{2}}{\sqrt{Bt}} \|U\|_{L_2}.
\end{aligned}$$

The rest of the proof is identical. □

Other tools to prove Proposition 8.35 have already been adapted earlier in the section: Binomial concentration, bounded differences inequality, symmetrization arguments. □

Proof of the Second Round. As explained in Proposition 8.35, the pool of champions satisfies the L_2 proximity to h^*. The second round next aims at exhibiting a candidate with small excess risk. This is explicited in Proposition 8.36.

Proposition 8.36. *Let $H_{\text{champ}} \subset \mathcal{H}$. Under the conditions of Theorem 8.30 and using its notation, it holds with probability at least $1 - 2\exp(-c_0 n \min\{1, \sigma^{-2} r^2\})$*

- *h^* wins all of its home matches, and*
- *if $\mathbb{E}\left[\Psi_{\mathcal{S}_n^{(3)}}(h^*, h)\right] \leq -2r_1^2$ then h loses its home match against h^*.*

Thus, on this event, the set of possible champions is nonempty (since it contains h^*), and any other champion satisfies that $\mathbb{E}\left[\Psi_{\mathcal{S}_n^{(3)}}(h^*, h)\right] \geq -2r_1^2$, and therefore by Lemma 8.37,

$$\mathbb{E}\left[\left(\hat{h}(X, X') - t(Y, Y')\right)^2\right] \leq \mathbb{E}\left[\left(h^*(X, X') - t(Y, Y')\right)^2\right] + 4r_1^2.$$

This is a direct adaptation of Proposition 3.8 in Lugosi and Mendelson (2016). The main tools used are again symmetrization, contraction of a Bernoulli process. We simply recall an adaptation of Lemma 3.6 therein, that links $\Psi_{\mathcal{S}_n^{(3)}}(h^*, h)$ to the excess risk.

Lemma 8.37. *For $\gamma > 0$, if $h \in \mathcal{H}$ satisfies $\mathbb{E}\left[\Psi_{\mathcal{S}_n^{(3)}}(h^*, h)\right] \geq -\gamma t^2$, then*

$$\mathbb{E}\left[\left(\hat{h}(X, X') - t(Y, Y')\right)^2\right] \leq \mathbb{E}\left[\left(h^*(X, X') - t(Y, Y')\right)^2\right] + 2\gamma t^2.$$

Proof. Observe that

$$\begin{aligned}
&\left(h(X, X') - t(Y, Y')\right)^2 - \left(h^*(X, X') - t(Y, Y')\right)^2 \\
&= \left(h(X, X') - h^*(X, X')\right)^2 + 2\left(h(X, X') - h^*(X, X')\right)\left(h^*(X, X') - t(Y, Y')\right),
\end{aligned}$$

and that

$$\begin{aligned}
\left(h(X, X') - h^*(X, X')\right)\left(h^*(X, X') - t(Y, Y')\right) = &-\left(h(X, X') - h^*(X, X')\right)^2 \\
&- \Psi_{\mathcal{S}_n^{(3)}}(h^*, h),
\end{aligned}$$

so that

$$\mathbb{E}\left[\left(h(X, X') - t(Y, Y')\right)^2\right] - \mathbb{E}\left[\left(h^*(X, X') - t(Y, Y')\right)^2\right] \leq -2\mathbb{E}\left[\Psi_{\mathcal{S}_n^{(3)}}(h^*, h)\right] \leq 2\gamma t^2.$$

□

Proof of Theorem 8.30. It is a direct application of Propositions 8.32, 8.35 and 8.36. □

In the last section of this chapter, we display some numerical experiments showing the benefit of using MoM estimates to perform learning.

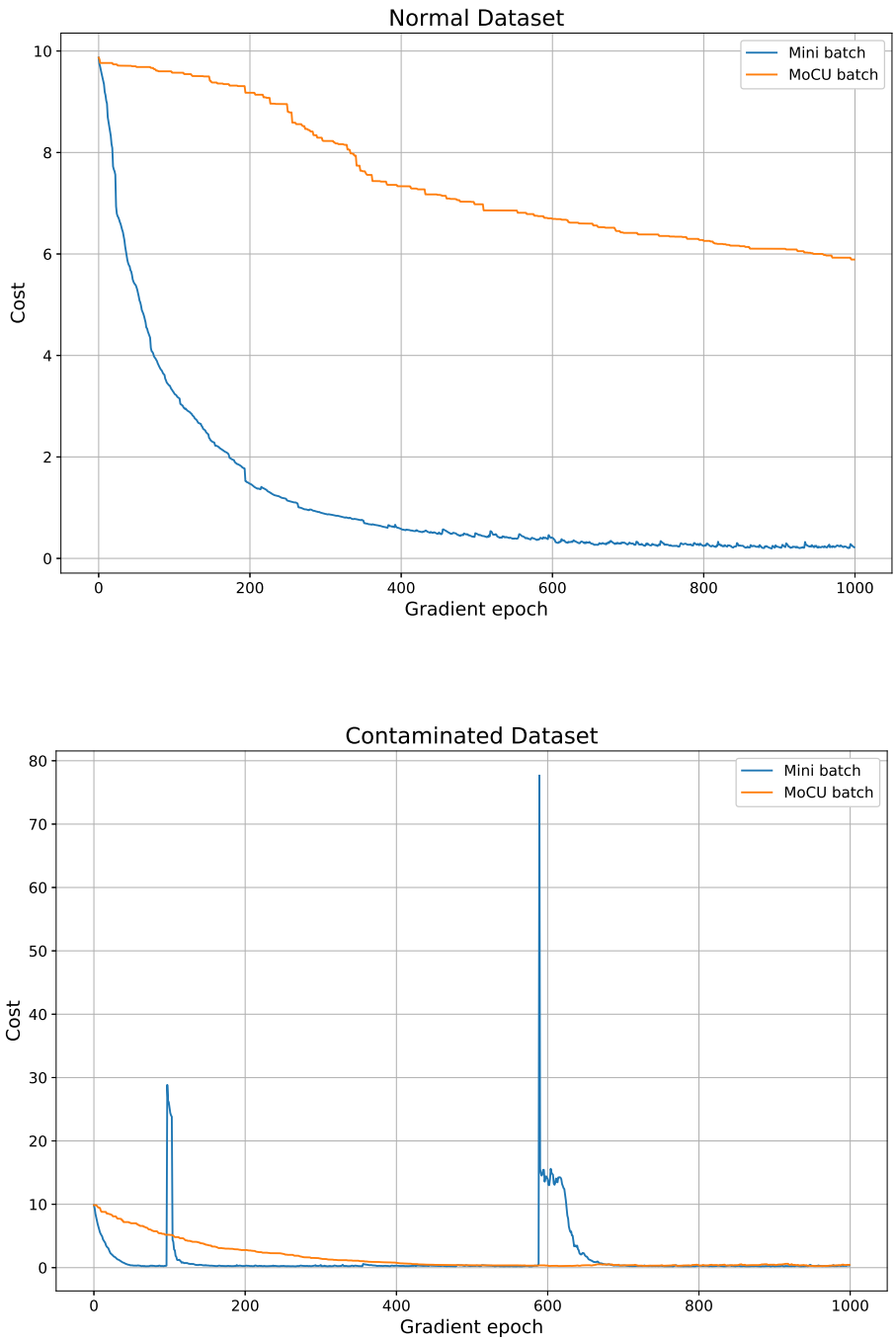


Figure 8.4 – Gradient Descents on normal (top) and contaminated (bottom) datasets.

8.6 Learning Experiments

As explained in Remark 8.27, the tournament winner's computation is often nontrivial. Unfortunately, the pairwise extension of the tournament does not escape this limitation. However, the MoU gradient descent scheme described in Section 8.4 can be set up very easily. Despite the slow convergence rates exhibited by the MoU minimizer, its strength relies on the ability to deal with corrupted datasets. The experiment run is as follows.

It illustrates a *metric learning* problem (see Section 6.2.2), where we try to learn from points that are known to be close or not, a distance that fits them. The search space is the set of all possible Mahalanobis distances, *i.e.* $d(x, y) = \sqrt{(x-y)^\top M(x-y)}$, for some positive definite matrix M. Thus, the optimal distance can be learned through mini-batch gradient descent over the parameter M. The procedure has been run on the iris dataset from scikit-learn (Pedregosa et al., 2011), that we have contaminated with a small number of outliers, in the spirit of Lecué and Lerasle (2017). The gradient descent convergences are depicted in Figure 8.4.

On the normal dataset (top), we can see that standard mini-batches perform well, while MoU mini-batches (or MoCU as the complete U-statistics are computed) induce a much slower convergence. This is expected as the median mini-batch is always selected, that contains the most *normal* observations, and consequently the least *informative* or *discriminative*.

Interestingly, the behavior is drastically different when the experiment is run on the corrupted dataset (bottom). Indeed, although converging with the MoU mini-batches remains slower than with the standard ones on the normal regime, they avoid peaks, that are caused by the presence of one (or more) outlier in the mini-batch. The MoU gradient descent automatically selects mini-batches that does not contain any outlier, and converges slowly but surely.

This experiment is a typical example of the benefits induced by using MoM-based algorithms in presence of outliers. Other examples on standard mean problems can be found in Lecué and Lerasle (2017), while Figure 8.4 gives a perfect illustration of this advantage in a pairwise learning framework.

8.7 Conclusion

In this chapter, we have seen how the MoM's principle can be used to perform learning. Minimizing a MoM estimate of risk, as described in Section 8.1, yields slow rates, but comes with efficient algorithms such as the MoM gradient descents detailed in Section 8.4. The main advantage of this method is its ability to deal with outliers. Other approaches, such as the tournament procedures discussed in Section 8.5, have also been analyzed, providing stronger guarantees even in heavy tailed scenarios. The principal drawback is here the lack of efficient algorithmic resolution. All this methods have been shown to adapt nicely to randomized and pairwise settings, opening the door to *robust ranking* and *robust metric learning*, as illustrated in Section 8.6.

Another promising research direction is the use of such pairwise criteria to perform *robust representation learning*. This can be an objective on its own, or part of a semi-supervised framework, in the spirit of what is proposed in Section 3.2 of Brouard et al. (2016b). One important question that remains to address is to link the Mo(R)M gradient descent solution (*i.e.* the minimizer of the sum over all possible Mo(R)M criteria) to the minimizer of one specific Mo(R)M criterion, for which we have guarantees. Finally, several MoM-based methods such as Le Cam's approach (Lecué and Lerasle, 2019) or MoM minmax estimators (Lecué and Lerasle, 2017) remain to be extended to the U-statistics setting. This would widen the theoretically sounded and algorithmically efficient learning approaches to tackle the robust pairwise learning problem.

9

Learning from Biased Training Samples

Contents

In the previous chapter, we have discussed how to design learning procedures that behave well in presence of heavy-tailed data. However, the few abnormal data points were completely part of the distribution, that was assumed to remain unchanged between the train and the test phases. The goal was simply not to be too much influenced by them. The problematic addressed here is totally different, although as frequent in practice as the previous one. It consists in assuming that the data available in the training stage are not sampled from the test distribution. This phenomenon occurs, for instance, as soon as the data is collected from different strata of one population. In this chapter, we thus develop a theoretically grounded approach to perform *debiased ERM*. It is supported by strong guarantees and a simple practical implementation.

In Section 9.1, we first motivate this analysis by highlighting its importance in today's machine learning. We also emphasize on its generality, as opposed to *covariate shift* in particular. The formal probabilistic framework needed is detailed in Section 9.2, while the debiasing procedure we propose, together with its practical implementation, are discussed in Section 9.3. Finally, guarantees about *debiased ERM*, stated in terms of excess risk, are proved in Section 9.4, and numerical experiments exposed in Section 9.5. This chapter covers the works exposed in the following preprint:

▶ **P. Laforgue**, S. Clémençon. Statistical learning from biased training samples. *arXiv preprint arXiv:1906.12304*, 2019.

9.1 Introduction

Recall first the standard setting of binary classification. The random pair $Z = (X, Y)$ is defined on a probability space with unknown joint probability distribution P, referred to as the *test distribution*. The random vector X, valued in $\mathcal{X} \subset \mathbb{R}^d$, models some information supposedly useful to predict the random binary label Y, taking its values in $\{-1, +1\}$. The objective is to build, from the training dataset $\mathcal{S}_n = \{Z_1, \ldots, Z_n\}$, composed of $n \geq 1$ independent copies of (X, Y), a Borelian predictive function, *i.e.* a classifier $g : \mathcal{X} \to \{-1, +1\}$ that minimizes the error probability of the decision

$$L_P(g) = \mathbb{P}\{Y \neq g(X)\}.$$

This corresponds to the risk of the classifier g, $\mathcal{R}(g)$, with the particular choice of loss function $\ell(g, Z) = \mathbb{1}\{g(X) \neq Y\}$. Empirical Risk Minimization (ERM in short, see *e.g.* Devroye et al. (1996a)) consists in solving instead the minimization problem

$$\min_{g \in \mathcal{G}} \quad \widehat{L}_n(g) = \frac{1}{n} \sum_{i=1}^{n} \ell(g, Z_i) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}\{g(X_i) \neq Y_i\}.$$

One can see $\widehat{L}_n(g)$ as a statistical estimator of the risk $L_P(g)$, obtained by replacing P in L_P with the empirical distribution of the (X_i, Y_i)'s: $\tilde{P}_n = (1/n) \sum_{i=1}^{n} \delta_{Z_i}$. The performance of empirical risk minimizers $\hat{g}_n$ (*i.e.* solutions to the ERM problem) is usually measured in terms of excess risk $L_P(\hat{g}_n) - L_P^*$. It has been studied under different assumptions on $\mathcal{G}$'s complexity (*e.g.* finite VC dimension, Rademacher averages), by means of concentration inequalities for empirical processes, see Boucheron et al. (2013, 2005). However, this approach naturally relies on the assumption that training data are distributed as the test ones, which is often unrealistic in practice.

Motivated by the poor control of the data acquisition process in many applications (see *e.g.* van Miltenburg (2016)), the purpose of the present chapter is therefore to investigate ERM in presence of *sample selection bias*. That is to say in the situation where the samples at disposal for learning a predictive rule g are not distributed as P. This can be viewed as a very specific case of *Transfer Learning*, see Ben-David et al. (2010); Liu et al. (2016). As recently highlighted by Bolukbasi et al. (2016), Zhao et al. (2017) or Burns et al. (2018) among others, representativeness issues do not vanish simply under the effect of the size of the training set. Thus, ignoring selection bias issues may dramatically jeopardize the outputs of machine-learning algorithms, referring to accuracy concerns of course, but also to ethical considerations.

Selection bias can be due to a wide variety of causes, such as the use of a survey scheme to collect observations, censorship, or truncation (see Heckman (1990) or Vella (1998) for instance). The study of its impact on inference methods, as well as techniques to remedy it, have a very long history in Statistics (Heckman, 1979). Depending on the nature of the mechanism causing the sample selection bias – and on that of the statistical information available for learning the decision rule – special cases have been considered in the machine learning literature, for which dedicated approaches have been developed. For instance, the case where some errors occur among the labels of the training data is studied in Lugosi (1992). Extending ERM to the framework of survey training data (when inclusion probabilities are known) is done in Papa et al. (2016), while statistical learning of regression models in the context of right censored training observations is considered in van Belle et al. (2011) and Ausset et al. (2019).

Most of these methods boil down to weighting the training observations with appropriate weights, either based on the *Importance Sampling* approach, or the *Inverse Probability Weighting* technique (IPW in abbreviated form, see *e.g.* Dubin and Rivers (1989) or Winship and Mare (1992) in the context of linear regression models). For instance, these weights are the inverses of the first order inclusion probabilities in the case where data are acquired by means of a survey plan, *cf* Clémençon et al. (2017), and they correspond to estimates of the probability of not being censored in the context of censored data (Ausset et al. (2019) and the references therein). In general, side information about the cause of the selection bias is crucially used to derive explicit forms for the appropriate weights from the observations available (Zadrozny, 2004). Refer also to Rosset et al. (2005) for a study in a semi-supervised framework, to Dudík et al. (2006) for maximum entropy density estimation, or to Lin et al. (2002) for the adaptation of the SVM algorithm to certain bias selection situations.

Recently, a very special case of sample selection bias, referred to as *covariate shift*, has been the subject of a good deal of attention (though it had been already considered by Manski and Lerman (1977) in a simplified version). In this case, the sample selection bias issue is simplified by the hypothesis stipulating that, in supervised problems, only the marginal input distribution may possibly change, the conditional distribution of the output Y given the input X being the same in the learning and predictive stages. One may refer to the rich literature: Shimodaira (2000), Sugiyama and Müller (2005), Huang et al. (2007), or Quionero-Candela et al. (2009) and Sugiyama and Kawanabe (2012). However, in many practical situations, the covariate-shift assumption is not fulfilled. The selection bias mechanism at stake is then way too complex to derive explicit forms for the appropriate weights that would permit to mimic the target distribution P.

In opposition to the aforementioned approaches, the framework we develop here allows to tackle problems where the biasing mechanism at work is very general, provided that certain identifiability hypotheses are satisfied. Precisely, focus is here on the case where statistical learning is based on training data sampled from general *selection bias models*, as originally introduced in Vardi (1985) and Gill et al. (1988) in a context of asymptotic nonparametric estimation of cumulative distribution functions. This very general biased sampling framework accounts for many situations encountered in practice. It covers for instance the (far from uncommon) situation where the samples available are sampled from conditional distributions of (X, Y) given that X lies in specific subsets of the input space $\mathcal{X}$ (assuming that the union of these subsets is equal to X's support). In this general setting, we thus extend ERM to the case of biased training data.

Attention should be paid to the fact that this framework completely encompasses the covariate shift scenario. It can by no means be systematically reduced to a reweighting problem where weights can be straightforwardly deduced (or estimated) from the data. Instead, we propose to build an unbiased empirical estimate of the test distribution by solving a generally nontrivial system of equations. From the solution is then computed a "nearly unbiased" risk estimate. We further establish a tail probability bound for the maximal deviations between the true risk functional and the estimate thus constructed. Based on this result, we prove that minimizers of the "debiased empirical risk" achieve learning rate bounds that are of the same order as those attained by empirical risk minimizers in absence of any bias mechanism. To our knowledge, this is the first time such guarantees are derived for ERM minimizers in such a general framework. We also present results from various numerical experiments, based on synthetic and real data, that provide strong empirical evidence of the relevance of the approach we propose.

We emphasize that, whereas the vast majority of machine-learning techniques dealing with biased data documented in the literature are ad-hoc and seldom supported by a sound theory, the present chapter essentially focuses on generalization issues. If the fact that the biasing functions are known can be seen at first glance as a limitation of the framework developed, one should have in mind that absolutely no learning strategy with statistical guarantees can be designed in absence of any understanding of the biasing mechanism. Furthermore, it is actually far from uncommon in practice that the latter is known (*e.g.* one may know the types of images that are more easily collected or the profiles of individuals who most likely answer a questionnaire). However, the situation where the biasing mechanism is only approximately known is of considerable interest in practice, and investigating to which extent the statistical guarantees established in this chapter are preserved will be the subject of further research.

9.2 Background and Preliminaries

As a first go, we precisely describe the probabilistic framework for ERM based on biased training data we consider. We then briefly recall the rationale behind the approach to nonparametric estimation in biased sampling models developed in Vardi (1985). The subsequent analysis indeed relies partly on this methodology. Here and throughout, we denote by $\lfloor u \rfloor$ the integer part of any real number u, by δ_a the Dirac mass at any point a, and by $||U||_\infty$ the essential supremum of any real-valued random variable U.

9.2.1 The Probabilistic Framework

Let Z be a random vector, taking its values in $\mathcal{Z} \subset \mathbb{R}^q$, $q \geq 1$, with unknown distribution $P(dz)$, and Θ a decision space. Consider a certain loss function $\ell : \mathbb{R}^q \times \Theta \to \mathbb{R}_+$, that is P-integrable for any decision rule $\theta \in \Theta$. Given this theoretical framework, we are interested in solving the *risk minimization* problem

$$\min_{\theta \in \Theta} L_P(\theta), \tag{9.1}$$

where $L_P : \theta \in \Theta \mapsto \mathbb{E}_P[\ell(Z, \theta)] \in \mathbb{R}_+$ is the risk function. In the biased sampling situation we consider here, we cannot rely on the observation of independent copies $Z_1, \ldots, Z_n$ of Z. Statistical learning must be based instead on the observation of $K \geq 1$ independent biased i.i.d. samples $\mathcal{S}_k = \{Z_{k,1}, \ldots, Z_{k,n_k}\}$ of size $n_k \geq 1$. We denote by $n = n_1 + \ldots + n_K$ the size of the pooled sample, and assume that the following classic condition in the multiple samples setting (*e.g.* van der Vaart (1998)) holds true.

Assumption 9.1. *There exist $C < +\infty$, $\lambda_{min}, \underline{\lambda} > 0$, and $(\lambda_1, \ldots, \lambda_K) \in [\lambda_{min}, 1[^K$, with $\sum_k \lambda_k = 1$, such that, for all $k \leq K$, and for all $n \in \mathbb{N}^*$ it holds*

$$\left| \lambda_k - n_k/n \right| \leq C/\sqrt{n} \qquad \text{and} \qquad \underline{\lambda} \leq n_k/n. \tag{9.2}$$

Remark 9.2. *We point out that, in the situation where the vector of sample sizes $(n_1, \ldots, n_K)$ is random, distributed as a multinomial of size n and with parameters $(\lambda_1, \ldots, \lambda_k)$, the bounds (9.2) simultaneously hold true for an appropriate constant C with overwhelming probability. Using Hoeffding's inequality (see Hoeffding (1963)) combined with the union bound for instance, one obtains that, for any $\delta \in]0, 1[$, all these conditions are fulfilled with probability larger than $1 - \delta$ with $C = \sqrt{\log(K/\delta)/2}$, and that $\underline{\lambda} \leq \min_k \lambda_k - C/\sqrt{n}$, provided that $n > C^2 / \min_k \lambda_k$. For simplicity, we restrict*

the subsequent analysis to the situation where the sample sizes are deterministic, the straightforward extension to the random case being left to the reader.

We suppose now that each distribution P_k of the $Z_{k,i}$'s, $k \leq K$, is absolutely continuous with respect to the distribution P, and assume that it is related to it through a known *biasing function* ω_k:

$$\forall z \in \mathcal{Z}, \quad \frac{dP_k}{dP}(z) = \frac{\omega_k(z)}{\Omega_k},$$

where $\Omega_k = \mathbb{E}_P[\omega_k(Z)] = \int_{\mathcal{Z}} \omega_k(z) P(dz)$. Notice that, just like P, the Ω_k's are unknown. This very general framework includes a wide variety of situations encountered in practice, as illustrated by the following examples.

Example 9.3. *We place ourselves in the context of binary classification: $Z = (X, Y)$, $\mathcal{Z} = \mathcal{X} \times \{-1, +1\}$, $q = d + 1$, $\Theta = \mathcal{G}$, and $\ell(Z, g) = \mathbb{1}\{Y \neq g(X)\}$. Consider $K \geq 1$ subsets $\mathcal{X}_1, \ldots, \mathcal{X}_K$ of the input space $\mathcal{X}$, such that $\mu(\mathcal{X}_k) > 0$ for all $k \in \{1, \ldots, K\}$, μ denoting X's marginal distribution. The case where only labeled examples with input observations in $\mathcal{X}_k$ can be collected to form sample $\mathcal{S}_k$, $k \leq K$, corresponds to the situation where $\omega_k(Z) = \mathbb{1}\{X \in \mathcal{X}_k\}$. In this particular case, P_k is (X, Y)'s conditional distribution given $X \in \mathcal{X}_k$.*

Example 9.4. *Consider the distribution-free regression framework, where T is a bounded random duration (i.e. a nonnegative random variable such that $||T||_\infty < +\infty$), and X is a random vector valued in $\mathcal{X} \subset \mathbb{R}^d$, defined on the same probability space, and supposedly useful to predict T. The goal is to learn a regression function $f : \mathcal{X} \to \mathbb{R}$ in a class $\mathcal{F}$ of bounded functions with minimum quadratic risk. In this case, $Z = (X, T)$, $\mathcal{Z} = \mathcal{X} \times \mathbb{R}_+$, $q = d + 1$, $\Theta = \mathcal{F}$, and $\ell(Z, f) = (T - f(X))^2$. Let $K \geq 1$, and $0 < \tau_1 < \ldots < \tau_{K-1} < \tau_K = ||T||_\infty$. Consider the case where, for $k \leq K$, the sample $\mathcal{S}_k$ is formed of censored observations with a deterministic right censorship, i.e. of copies of the pair $(X, \min\{T, \tau_k\})$. This corresponds to the situation where $\omega_k(Z) = \mathbb{1}\{T \leq \tau_k\}$, and P_k is the conditional distribution of (X, T) given $T \leq \tau_k$.*

The following technical assumptions are required in the rate bound analysis carried out in the next section. These hypotheses permit to build a nearly unbiased estimator $\hat{P}_n$ of P from the biased samples $\mathcal{S}_k$ and the biasing functions ω_k.

Assumption 9.5. *The union of the supports of the biased distributions P_k is equal to* $\textsc{Supp}(P)$*, the support of distribution P:*

$$\textsc{Supp}(P) = \bigcup_{k=1}^{K} \Big\{ z \in \mathcal{Z} : \ \omega_k(z) > 0 \Big\}.$$

If Assumption 9.5 is not fulfilled, there is of course absolutely no hope to estimate P on its full support, since no points from $\textsc{Supp}(P) \setminus \bigcup_{k=1}^{K} \{z \in \mathcal{Z} : \ \omega_k(z) > 0\}$ can be sampled. At best, one may thus be able to estimate $P^+ = P(\ \cdot \ | \ \sum_k \omega_k > 0)$.

The next assumption needed states as a graph connectivity condition. Let $\kappa > 0$, and $G_\kappa = (V, A)$ the (undirected) graph with vertices in $V = \{1, \ldots, K\}$, and adjacency matrix $A = (a_{k,l})_{1 \leq k \neq l \leq K}$ defined by $a_{k,l} = \mathbb{1}\{\mathbb{E}_P[\omega_k(Z)\omega_l(Z)] \geq \kappa\}$, *i.e.* vertices k and l are connected if and only if $\mathbb{E}_P[\omega_k(Z)\omega_l(Z)] \geq \kappa$.

Assumption 9.6. *The graph G_κ is connected.*

From an algebraic point of view, one may classically check whether Assumption 9.6 is fulfilled or not by means of a breadth-first search algorithm, or by examining the spectrum of the Laplacian matrix of G_κ for instance (see *e.g.* Godsil and Royle (2001)).

Assumption 9.7. *Let $\xi > 0$. For all $k \in \{1, \ldots, K\}$, $\Omega_k \geq \xi$.*

Notice that, contrary to Gill et al. (1988), Assumptions 9.6 and 9.7 involve explicit lower bounds κ and ξ. Indeed, the subsequent analysis is nonasymptotic, and we need to rely on explicit parameters, as we cannot ensure the positiveness requirements by simply letting n tend to infinity. It is also the purpose of the following assumption to provide constants on which that of Theorem 9.17 are built.

Assumption 9.8. $\exists\, m, M > 0, \quad m \leq \inf_z \max_{k \leq K} \omega_k(z) \quad \text{and} \quad \sup_z \max_{k \leq K} \omega_k(z) \leq M.$

Remark 9.9. *We point out that, in Example 9.3, Assumption 9.5 simply means that $\mathcal{X} = \mathcal{X}_1 \cup \cdots \cup \mathcal{X}_K$, and Assumption 9.7 that $\mu(\mathcal{X}_k) \geq \xi$ for all $k \in \{1, \ldots, K\}$. Assumption 9.8 is always fulfilled with $m = M = 1$, and Assumption 9.6 can be checked in a simple manner, insofar as we have: $\forall 1 \leq k \neq l \leq K$,*

$$e_{k,l} = +1 \Leftrightarrow \mu\left(\mathcal{X}_k \cap \mathcal{X}_l\right) \geq \kappa.$$

In Example 9.4, Assumption 9.5 is always fulfilled by construction, as Assumption 9.8 with $m = M = 1$. Assumption 9.7 means that $\mathbb{P}\{T \leq \tau_1\} \geq \xi$, whereas Assumption 9.6 is always true for any $\kappa \leq \xi$.

We are now equipped to construct an unbiased estimate $\hat{L}_n$ of L_P, based only on the biased training samples $\mathcal{S}_k$, $k \leq K$.

9.2.2 Building an Empirical Error Estimate

The goal pursued here is to build an estimator of the unknown risk L_P based on the K independent biased samples $\mathcal{S}_1, \ldots, \mathcal{S}_K$. The risk estimation procedure we consider follows in the footsteps of the cumulative density function estimation technique in biased models introduced in the seminal contribution of Vardi (1985) (which, incidentally, can be interpreted as nonparametric maximum likelihood estimation). Ignoring the bias selection issue, one may compute the empirical distribution based on the whole pooled sample

$$\tilde{P}_n = \frac{1}{n} \sum_{k=1}^{K} \sum_{i=1}^{n_k} \delta_{Z_{k,i}} = \sum_{k=1}^{K} (n_k/n) \hat{P}_k, \tag{9.3}$$

where $\hat{P}_k = (1/n_k) \sum_{i \leq n_k} \delta_{Z_{k,i}}$ is the empirical distribution based on (biased) sample $\mathcal{S}_k$, $k \leq K$. This discrete random measure is a natural estimator of the linear convex combination of the P_k's given by $\bar{P} = \sum_k \lambda_k P_k$, which is absolutely continuous with respect to P, and whose density can be written as

$$\frac{d\bar{P}}{dP}(z) = \sum_{k=1}^{K} \frac{\lambda_k}{\Omega_k} \omega_k(z).$$

Under Assumption 9.5, it is strictly positive on the whole support of Z, so it holds:

$$P(dz) = \left(\sum_{k=1}^{K} \frac{\lambda_k}{\Omega_k} \omega_k(z) \right)^{-1} \cdot \bar{P}(dz). \tag{9.4}$$

Hence, if estimates $\hat{\Omega}_k$ of the unknown expectations $\mathbb{E}_P[\omega_k(Z)]$ were at our disposal, one could immediately form a *plug-in* estimator of P by replacing $\bar{P}$ and the Ω_k's in Equation (9.4) with their statistical versions, namely $\tilde{P}_n$ and the $\hat{\Omega}_k$'s:

$$\hat{P}_n(dz) = \left(\sum_{k=1}^{K} \frac{n_k}{n\hat{\Omega}_k} \omega_k(z) \right)^{-1} \cdot \tilde{P}_n(dz).$$

In order to estimate the vector $\boldsymbol{\Omega} = (\Omega_1, \ldots, \Omega_K)$, observe that it straightforwardly follows from Equation (9.4) that it is solution to the system of equations

$$\mathbf{1} = (\Gamma_1(\boldsymbol{W}), \ldots, \Gamma_K(\boldsymbol{W})), \tag{9.5}$$

where $\mathbf{1}$ denotes the K-dimensional vector with all components equal to 1 and, for any $k \leq K$ and all $\boldsymbol{W} = (W_1, \ldots, W_K) \in (\mathbb{R}_+^*)^K$,

$$\Gamma_k(\boldsymbol{W}) = \frac{1}{W_k} \int_{z \in \mathcal{Z}} \frac{\omega_k(z)}{\sum_{l=1}^{K} (\lambda_l / W_l) \omega_l(z)} \bar{P}(dz). \tag{9.6}$$

A natural (M-estimation) method to recover $\boldsymbol{\Omega}$ approximately then consists in solving a statistical version of the system above

$$\mathbf{1} = \left(\hat{\Gamma}_1(\boldsymbol{W}), \ldots, \hat{\Gamma}_K(\boldsymbol{W}) \right), \tag{9.7}$$

the $\hat{\Gamma}_l(\boldsymbol{W})$'s being built by replacing λ_l and $\bar{P}$ in Equation (9.6) by n_l/n and $\tilde{P}_n$ respectively. In addition, since the $\hat{\Gamma}_k(\boldsymbol{W})$'s are homogeneous of degree 0 (just like the $\Gamma_k(\boldsymbol{W})$'s), observe that one may build an estimator of P from any solution $\boldsymbol{W}$ of System (9.7) by considering

$$\hat{P}_n = \sum_{k=1}^{K} \sum_{i=1}^{n_k} \left(\frac{\left(\sum_{l=1}^{K} (n_l/(nW_l)) \omega_l(Z_{k,i}) \right)^{-1}}{\sum_{m=1}^{K} \sum_{j=1}^{n_m} \left(\sum_{l=1}^{K} (n_l/(nW_l)) \omega_l(Z_{m,j}) \right)^{-1}} \right) \delta_{Z_{k,i}}. \tag{9.8}$$

Under a slightly weaker version of Assumption 9.6, this inference technique has been investigated from an asymptotic perspective in the context of cumulative distribution function estimation in Gill et al. (1988).

9.3 The Debiased ERM Procedure

We now describe the ERM approach based on biased training samples we promote. It is reproduced from **Laforgue** and Clémençon (2019), as all results exposed hereafter. The complete procedure is summarized in Section 9.3.1, while the resolution of System (9.7) is discussed in Sections 9.3.2 and 9.3.3.

9.3.1 The Complete Procedure

The procedure mainly consists in replacing P in L_P with $\hat{P}_n$ as defined in Equation (9.8), rather than with $\tilde{P}_n = (1/n) \sum_{i=1}^{n} \delta_{Z_i}$. Indeed, the latter is an estimate of the biased distribution $\bar{P} = \sum_k \lambda_k P_k$, while the first one really estimates P. As highlighted by Equation (9.8), this incidentally boils down to reweight the datapoints. The ERM variant we propose in the context of biased training data is summarized in Figure 9.1, and can be implemented in three steps as follows:

ERM Based on Biased Training Samples

- **Input.** Samples $\mathcal{S}_k = \{Z_{k,i},\ i \leq n_k\}$ and biasing functions ω_k, $k \leq K$.
- **Debiasing the raw empirical distribution.** Form the raw distribution based on the pooled sample

$$\tilde{P}_n = \frac{1}{n} \sum_{k=1}^{K} \sum_{i=1}^{n_k} \delta_{Z_{k,i}},$$

(i) compute the functions given by: $\forall \boldsymbol{W} \in (\mathbb{R}_+^*)^K$,

$$\hat{\Gamma}_k(\boldsymbol{W}) = \frac{1}{W_k} \int_{z \in \mathcal{Z}} \frac{\omega_k(z)}{\sum_{l=1}^{K} ((n_l/n)/W_l)\omega_l(z)} \tilde{P}_n(dz) \text{ for } k \leq K$$

(ii) solve System (9.7), producing a solution $\hat{\boldsymbol{W}}$ in $(\mathbb{R}_+^*)^K$ such that $\hat{W}_{K,n} = 1$

(iii) for $k \leq K$ and for $i \leq n_k$, compute the weights

$$\pi_{k,i} = \frac{\left(\sum_{l=1}^{K} (n_l/(n\hat{W}_l))\omega_l(Z_{k,i})\right)^{-1}}{\sum_{m=1}^{K} \sum_{j=1}^{n_m} \left(\sum_{l=1}^{K} (n_l/(n\hat{W}_l))\omega_l(Z_{m,j})\right)^{-1}},$$

and form the "debiased" distribution estimate $\hat{P}_n = \sum_{k=1}^{K} \sum_{i=1}^{n_k} \pi_{k,i} \delta_{Z_{k,i}}$.

- **ERM.** Solve the ERM problem $\min_{\theta \in \Theta} \hat{L}_n(\theta)$, to produce the solution $\hat{\theta}_n$, with $\hat{L}_n(\theta)$ given by

$$\hat{L}_n(\theta) := L_{\hat{P}_n}(\theta) = \sum_{k=1}^{K} \sum_{i=1}^{n_k} \pi_{k,i} \ell(Z_{k,i},\ \theta), \tag{9.9}$$

Figure 9.1 – Debiased ERM Procedure

1. Estimates of the Γ_k's functions are first computed from the pooled empirical distribution $\tilde{P}_n$ (see Equation (9.3)) in order to form System (9.7).
2. The latter is next solved to build the estimate $\hat{P}_n$ (see Equation (9.8)) of P.
3. The decision rule is finally obtained by replacing P by its estimate $\hat{P}_n$ in the risk minimization Problem (9.1) and solving it.

Discussing how to minimize the "nearly debiased" empirical risk in Equation (9.9) (or a smooth/penalized version of it) in practice is beyond the scope of the present paper. However, as discussed in Remark 9.10, one may straightforwardly combine any popular ERM-like learning algorithm with the generic algorithmic approach described above. Before investigating the resolution of System (9.7), a few remarks are in order.

Remark 9.10. *We highlight here that the debiasing procedure exposed in this subsection is by no means computationally expensive. Indeed, the sole difference with standard methods lies in the computation of the weights $\pi_{k,i}$ involved in the risk functional. In addition, it can be readily implemented in a plug-in manner with most machine-learning libraries, using e.g. the* sample_weight *option during the learning stage of scikit-learn's (Pedregosa et al., 2011) predictors.*

Remark 9.11. *From a practical point of view, rather than modifying the objective function using the weights computed at step (iii) in Figure 9.1, one may alternatively sample from distribution in Equation (9.8) given the original data to generate training observations feeding next an untouched version of the learning algorithm.*

We shall now focus on the resolution of System (9.7).

9.3.2 Solving System (9.7)

The resolution of System (9.7) is central in our method. Indeed, this nontrivial step (see Section 9.3.3) is what makes our reweighting so general. While *Inverse Probability Weighting* involves simple weights that can be deduced by *common sense*, the generality of the biasing model studied here necessitates this complex resolution. Hopefully, as shall be seen in this section, the solutions may be approximately computed easily.

Our first focus is on the number of solutions to System (9.7) (possibly none). Indeed, System (9.7) is an empirical estimate of the ideal System (9.5), and the latter has been shown in Gill et al. (1988) to have a unique solution under Assumption 9.6, up to the homogeneity property.

Lemma 9.12. *(Gill et al. (1988), Proposition 1.1 therein) Grant Assumption 9.6. Then, System (9.5) has a unique solution $\boldsymbol{W}^* = (W_1^*, \ldots, W_K^*)$ such that $W_K^* = 1$.*

The following result now reveals that the empirical System (9.7) has also a unique, up to the homogeneity property, solution with overwhelming probability.

Proposition 9.13. *Grant Assumptions 9.1, 9.6 and 9.8. Then, there exists a constant $c_0 > 0$, depending only on κ, $\underline{\lambda}$ and M, such that System (9.7) admits a unique solution $\hat{\boldsymbol{W}}_n = (\hat{W}_{n,1}, \ldots, \hat{W}_{n,K})$ such that $\hat{W}_{n,K} = 1$ with probability at least $1 - K^2 e^{-c_0 n}$.*

Proof. Define the directed graph $\tilde{\mathbf{G}}_n$ with vertices $\{1, \ldots, K\}$ and link $k \to l$ if and only if

$$\int \mathbb{1}\{\omega_k > 0\}(z)\hat{P}_l(dz) > 0, \quad \text{or equivalently iff} \quad \Omega_l \int \omega_k(z)\hat{P}_l(dz) > 0. \tag{9.10}$$

The graph $\tilde{\mathbf{G}}_n$ is said to be *strongly connected* when, for any pair of vertices (k, l), there exist a directed path from k to l, and a directed path from l to k. It is proved in Vardi (1985) (see also Theorem 1.1 in Gill et al. (1988)) that this is a necessary and sufficient condition for System (9.7) to have a unique solution (up to the homogeneity property).

Lemma 9.14. *(Gill et al. (1988), Theorem 1.1) System (9.7) has a unique solution such that $\hat{W}_{n,K} = 1$ if and only if $\tilde{\mathbf{G}}_n$ is strongly connected.*

From an asymptotic perspective, a direct application of the strong law of large numbers ensures that, under Assumption 9.6, the strong connectivity property is fulfilled for n large enough with probability one (*cf* Corollary 1.1 in Vardi (1985)). The rest of the proof is dedicated to a nonasymptotic analysis of the phenomenon.

Let n_E be the number of edges in G_κ. By definition $n_E \leq K(K-1)/2$. Now let $k, l \leq K$ be a pair linked in G_κ. By definition $\mathbb{E}_P[\omega_k(Z)\omega_l(Z)] \geq \kappa$. Since $0 \leq \omega_k(z) \leq M$ for all k and all z, Hoeffding's inequality yields that for any $t > 0$,

$$\mathbb{P}\left\{\Omega_k \int \omega_k(z)\hat{P}_l(dz) - \mathbb{E}_P[\omega_k(z)\omega_l(z)] \leq -t\right\} \leq \exp\left(-\frac{2n_l t^2}{M^2}\right) \leq \exp\left(-\frac{2\underline{\lambda} n t^2}{M^2}\right),$$
$$\mathbb{P}\left\{\Omega_l \int \omega_l(z)\hat{P}_k(dz) - \mathbb{E}_P[\omega_k(z)\omega_l(z)] \leq -t\right\} \leq \exp\left(-\frac{2n_k t^2}{M^2}\right) \leq \exp\left(-\frac{2\underline{\lambda} n t^2}{M^2}\right).$$

Choosing $t = \kappa > 0$, the union bound gives that it holds with probability at least $1 - 2e^{-2\underline{\lambda}\kappa^2 n/M^2}$ both at the same time $k \to l$, and $l \to k$ in $\tilde{\boldsymbol{G}}_n$. Proceeding analogously for every pair connected in G_κ, we get that with probability at least $1 - 2n_E e^{-2\underline{\lambda}\kappa^2 n/M^2}$ all pairs connected in G_κ are connected both ways in $\tilde{\boldsymbol{G}}_n$. Since G_κ is assumed to be connected, this implies that $\tilde{\boldsymbol{G}}_n$ is strongly connected. Noticing that $n_E \leq K^2/2$ and setting $c_0 = 2\underline{\lambda}\kappa^2/M^2$, the proof is finished by applying Lemma 9.14. □

Now that the solution to System (9.7) is proven to be unique with high probability, we shall give some details about the resolution procedure. It directly derives from the writing of the $\hat{\Gamma}_k$. Indeed, it holds for all $k \leq K$:

$$\hat{\lambda}_k \hat{\Gamma}_k(\boldsymbol{W}) = \int \frac{\frac{\hat{\lambda}_k \omega_k}{W_k}}{\sum_{l=1}^K \frac{\hat{\lambda}_l \omega_l}{W_l}} d\tilde{P}_n = \int \frac{e^{u_k}\omega_k}{\sum_{l=1}^K e^{u_l}\omega_l} d\tilde{P}_n = \frac{\partial}{\partial u_k} \int \log\left[\sum_{k=1}^K e^{u_k}\omega_k\right] d\tilde{P}_n,$$

with the change of variable $e^{\boldsymbol{u}} = \hat{\boldsymbol{\lambda}}/\boldsymbol{W}$. Hence, finding $\boldsymbol{W}$ such that for all $k \leq K$ it holds $\hat{\Gamma}_k(\boldsymbol{W}) = 1$, or equivalently $\hat{\lambda}_k \hat{\Gamma}_k(\boldsymbol{W}) = \hat{\lambda}_k$, is equivalent to solving

$$\nabla_{\boldsymbol{u}} \left(\int \log\left[\sum_{k=1}^K e^{u_k}\omega_k\right] d\tilde{P}_n - \hat{\boldsymbol{\lambda}}^\top \boldsymbol{u} \right) = \mathbf{0}.$$

The function inside the gradient operator is actually convex, and can be shown to be strongly convex, hence the uniqueness of the solution (see Gill et al. (1988)). Solving System (9.7) is thus equivalent to rooting the gradient of a (strongly) convex function. This can be easily tackled by means of any Gradient Descent or Robbins-Monro scheme (Robbins and Monro, 1951). Before investigating a *simple* instance of System (9.7) that demonstrates the need for a gradient descent-based resolution, the following remark focuses on the difficulty to check $\tilde{\mathbf{G}}_n$'s connectivity, and the alternatives prescribed.

Remark 9.15. *The strong connectivity of $\tilde{\mathbf{G}}_n$ might be arduous to check, compared to computing the system's solution. Thus, one is rather encouraged to compute the debiasing weights without prior verification. If the graph happens not to be connected, the minimized function is simply convex, and solutions are not unique, but still exist. In this case, one of them will be found by the algorithm and one proceeds just like in the strongly convex scenario. Otherwise, a simple criterion such as the norm of the gradient suffices to alert on the non-convergence, and other bias functions should be chosen, or debiased ERM abandoned and replaced by standard ERM.*

9.3.3 Analysis of *Simple* Systems

The subsequent analysis highlights the difficulty to solve System (9.7) as long as more than one biased sample is involved, even in simplistic settings. This general remark makes the resolution by gradient descent as explained in Section 9.3.2 necessary. It also highlights that the price to pay for the generality of the biasing model considered is that complex systems with nontrivial solutions arise, that must be solved approximately.

For instance, consider the binary classification framework: $Z = (X, Y)$, $\mathcal{Z} = \mathcal{X} \times \{-1,\ +1\}$, $q = d + 1$, $\Theta = \mathcal{G}$, and $\ell(Z, g) = \mathbb{1}\{Y \neq g(X)\}$. The distribution P of the random pair (X, Y) can be either described by X's marginal distribution $\mu(dx)$ and the posterior probability $\eta(X) = \mathbb{P}\{Y = +1 \mid X = x\}$ or else by the triplet (p, F_+, F_-) where $p = \mathbb{P}\left\{Y = +1\right\}$ and $F_\sigma(dx)$ is X's conditional distribution given $Y = \sigma 1$, with $\sigma \in \{-, +\}$.

Consider first the simple case where only one biased training sample $\mathcal{S}_1$ generated by the biased distribution P_1 is at disposal. Suppose furthermore that P_1 is identical to P, but with a different value p_1 (*i.e.* F_+ and F_- remain unchanged). This corresponds to the following biasing function:

$$\omega_1(z) = \frac{dP_1}{dP}(z) = \frac{p_1}{p}\mathbb{1}\{y = +1\} + \frac{1 - p_1}{1 - p}\mathbb{1}\{y = -1\}.$$

Since there is only one training dataset, System (9.7) boils down to one single equation, that admits 1 as a solution, and the debiasing weights $\pi_{k,i} = \pi_{1,i} = \pi_i$ then writes

$$\pi_i = \frac{\omega_1(z_i)^{-1}}{\sum_{j=1}^n \omega_1(z_j)^{-1}} = \frac{1}{C}\left(\frac{p}{p_1}\mathbb{1}\{y_i = +1\} + \frac{1 - p}{1 - p_1}\mathbb{1}\{y_i = -1\}\right),$$

with C the normalizing constant such that the sum of the weights is equal to 1. This is precisely the debiasing weights one would have intuitively expected, and going through the whole debiasing procedure may appear as an excessive tool to compute them.

However, even in this very basic setting, considering more than 1 sample makes the problem considerably more complex. Consider now two biased training samples, biased as in the previous case, with two different p_1 and p_2. Assume in addition that they have the same number of observations. System (9.7) then writes

$$\begin{cases} 1 = 2\lambda_+ \frac{\frac{p_1}{pW_1}}{\frac{p_1}{pW_1} + \frac{p_2}{pW_2}} + 2\lambda_- \frac{\frac{1-p_1}{(1-p)W_1}}{\frac{1-p_1}{(1-p)W_1} + \frac{1-p_2}{(1-p)W_2}}, \\ 1 = 2\lambda_+ \frac{\frac{p_2}{pW_2}}{\frac{p_1}{pW_1} + \frac{p_2}{pW_2}} + 2\lambda_- \frac{\frac{1-p_2}{(1-p)W_2}}{\frac{1-p_1}{(1-p)W_1} + \frac{1-p_2}{(1-p)W_2}}, \end{cases}$$

where λ_+ and λ_- are the proportions of positive (respectively negative) labels in the pooled sample. After setting W_1 to 1, and replacing W_2 by W for notation purposes, it simplifies to:

$$1 = 2\lambda_+ \frac{p_1}{p_1 + \frac{p_2}{W}} + 2\lambda_- \frac{1 - p_1}{1 - p_1 + \frac{1-p_2}{W}},$$

$$\begin{aligned} (p_2 + p_1 W)\left(1 - p_2 + (1 - p_1)W\right) &= 2\lambda_+ p_1 W\left(1 - p_2 + (1 - p_1)W\right) \\ &\quad + 2\lambda_-(1 - p_1)W(p_2 + p_1 W), \end{aligned}$$

$$W^2 + (2\lambda_+ - 1)\left(\frac{1 - p_2}{1 - p_1} - \frac{p_2}{p_1}\right)W + \frac{p_2(1 - p_2)}{p_1(1 - p_1)} = 0,$$

that does not admit any trivial analytical solution. It is still easy to compute (recall that p_1 and p_2 are assumed to be known), but obviously it is not a *common sense* solution. If no simple solution is available in such very simple cases, the computational resolution of System (9.7) seems to be an inevitable step in the general setting, and the GD approach advocated in Section 9.3.2 the only viable option.

Now that the debiased ERM procedure has been thoroughly discussed, the next section establishes the theoretical guarantees carried by the debiased risk minimizers computed.

9.4 Theoretical Guarantees

We now investigate the performance of minimizers of the "debiased" risk estimate of Equation (9.9). The principal results are stated in Section 9.4.1, while intermediate propositions used to derive the main theorem are detailed in Section 9.4.2.

9.4.1 Main Results

We start by a technical assumption needed for the statement of our theorem, see *e.g.* van der Vaart and Wellner (1996).

Assumption 9.16. *The collection of functions $\mathcal{F} = \{\ell(\cdot,\ \theta) :\quad \theta \in \Theta\}$ is a uniform Donsker class (relative to L_1) with polynomial uniform covering numbers, i.e. there exist constants $C_0 > 0$ and $r \geq 1$ such that: $\forall \zeta > 0$,*

$$\sup_Q\ \mathcal{N}\left(\zeta,\ \mathcal{F},\ L_1(Q)\right) \leq C_0(1/\zeta)^r,$$

where the supremum is taken over the set of all probability measures Q on $\mathcal{Z}$, and with $\mathcal{N}\left(\zeta,\ \mathcal{F},\ L_1(Q)\right)$ the number of $L_1(Q)$ balls of radius $\zeta > 0$ needed to cover class $\mathcal{F}$.

This hypothesis is a classical complexity assumption. Notice that the subsequent rate bound analysis can be straightforwardly extended to settings involving other complexity conditions (*e.g.* finite VC dimension, Rademacher averages). For instance, in the binary classification example, recall that if the collection of classifiers $\mathcal{G}$ considered is of finite VC dimension $V < +\infty$, then the collection of functions $\{(x, y) \in \mathcal{X} \times \{-1,\ +1\} \mapsto \mathbb{1}\{Y \neq g(X)\},\ \ g \in \mathcal{G}\}$ satisfies Assumption 9.16 with $r = 2(V-1)$ and $K_0 V (4e)^V \leq C_0$, where K_0 is a universal constant (Theorem 2.6.4 in van der Vaart and Wellner (1996)).

The result stated below now provides a tail bound for the maximal deviations between the risk estimator of Equation (9.9) and the true risk.

Theorem 9.17. *Grant Assumptions 9.1, 9.5 to 9.8 and 9.16. Then there exist constants $c_0, C_1'', C_2'', C_3'' > 0$ such that for any $\delta \in]0, 1 - K^2 e^{-c_0 n}[$ it holds with probability at least $1 - K^2 e^{-c_0 n} - \delta$:*

$$\sup_{\theta \in \Theta} \left| \hat{L}_n(\theta) - L(\theta) \right| \leq \sqrt{K} \left(C_1'' \sqrt{\frac{1}{n} \log\left(\frac{16 C_0 K^3 n^{r/2}}{\delta} \right)} + \frac{C_2'' K}{\sqrt{n}} \right) + \frac{C_3''}{\sqrt{n}}.$$

The proof of Theorem 9.17 is given by several intermediate results further detailed in Section 9.4.2. The bound stated below for the excess of risk of rules obtained by minimization of Equation (9.9) immediately results from the standard bound

$$L(\hat{\theta}_n) - \inf_{\theta \in \Theta} L(\theta) \leq 2 \sup_{\theta \in \Theta} \left| \hat{L}_n(\theta) - L(\theta) \right|,$$

combined with Theorem 9.17. Remarkably, it reveals that minimizers of the "debiased" version of the empirical risk achieve exactly the same learning rate as minimizers of the (unbiased) empirical risk based on $n \geq 1$ independent observations $Z_1, \ldots, Z_n$ drawn from the test distribution P.

Corollary 9.18. *Suppose that the hypotheses of Theorem 9.17 are fulfilled. Let $\hat{\theta}_n$ be any minimizer of Equation (9.9). Then, for any $\delta \in]0,1[$, it holds with probability at least $1-\delta$: $\forall n \geq 1$,*

$$L(\hat{\theta}_n) - \inf_{\theta \in \Theta} L(\theta) \leq 2\sqrt{K}\left(C_1'' \sqrt{\frac{1}{n}\log\left(\frac{32 C_0 K^3 n^{r/2}}{\delta}\right)} + \frac{C_2'' K}{\sqrt{n}}\right) + \frac{C_3''}{\sqrt{n}},$$

as soon as $n \geq \log(2K^2/\delta)/c_0$, where c_0, C_1'', C_2'', C_3'' are the same as in Theorem 9.17.

The next section now focuses on intermediate results needed to derive Theorem 9.17.

9.4.2 Intermediate Results

Here we give the proof of Theorem 9.17. It is done through the succession of intermediate results, stated as lemmas and propositions. The first step, addressed in Section 9.3.2, (Proposition 9.13 therein) consists in proving that, like the "ideal" System (9.5), the empirical System (9.7) has a unique solution with overwhelming probability (up to the homogeneity property). On this event, one can then study the deviation of the (unique) solution to System (9.7), denoted $\hat{\boldsymbol{W}}_n$, with respect to $\boldsymbol{W}^*$, the solution to System (9.5) (see Proposition 9.19). This control next transfers into a control on the deviations of $\hat{\boldsymbol{\Omega}}_n$ with respect to $\boldsymbol{\Omega}$ (Proposition 9.25). Finally, the control on $\hat{\boldsymbol{\Omega}}_n$ results in a control on $\hat{P}_n$, and on the deviation of $\hat{L}_n(\theta)$ with respect to $L(\theta)$ at fixed θ (Proposition 9.26). Chaining arguments finally allow to prove Theorem 9.17.

Step 1: System (9.7) has a unique solution.

Refer to Proposition 9.13 in Section 9.3.2.

Step 2: Deviation of (the unique) solution $\hat{\boldsymbol{W}}_n$.

Now that existence of a (unique) solution $\hat{\boldsymbol{W}}_n$ to System (9.7) is ensured with high probability, the second step of the proof consists in controlling its deviation from the solution $\boldsymbol{W}^*$ of the "true" system in a nonasymptotic fashion. It is the purpose of the following result.

Proposition 9.19. *Grant Assumption 9.1. Then, there exist $C_1, C_2 > 0$ such that for any $\delta > 0$ we have with probability at least $1-\delta$: System (9.7) has a unique solution $\hat{\boldsymbol{W}}_n$ s.t. $\hat{W}_{n,K} = 1$ and*

$$\left\|\hat{\boldsymbol{W}}_n - \boldsymbol{W}^*\right\| \leq \sqrt{K}\left(C_1\sqrt{\frac{\log\left(2K^2/\delta\right)}{n}} + \frac{C_2 K}{\sqrt{n}}\right).$$

More notation is required to prove Proposition 9.19. For any $\boldsymbol{u} = (u_1, \dots, u_K) \in \mathbb{R}^K$, let

$$\bar{D}(\boldsymbol{u}) = \int \log\left[\sum_{k=1}^{K} e^{u_k}\omega_k(z)\right]\bar{P}(dz) - \sum_{k=1}^{K}\lambda_k u_k, \quad \boldsymbol{u}^* = \operatorname*{argmin}_u \ \bar{D}(u),$$

$$\hat{D}_n(\boldsymbol{u}) = \int \log\left[\sum_{k=1}^{K} e^{u_k}\omega_k(z)\right]\tilde{P}_n(dz) - \sum_{k=1}^{K}\hat{\lambda}_k u_k, \quad \hat{\boldsymbol{u}}_n = \operatorname*{argmin}_u \ \hat{D}_n(u),$$

$$\bar{\boldsymbol{D}}' \ \text{ s.t. } \ \bar{\boldsymbol{D}}'(\boldsymbol{u})_l = \int \frac{e^{u_l}\omega_l(z)}{\sum_{k=1}^{K} e^{u_k}\omega_k(z)}\bar{P}(dz) - \lambda_l \quad \forall\, l \le L,$$

$$\hat{\boldsymbol{D}}'_n \ \text{ s.t. } \ \hat{\boldsymbol{D}}'_n(\boldsymbol{u})_l = \int \frac{e^{u_l}\omega_l(z)}{\sum_{k=1}^{K} e^{u_k}\omega_k(z)}\tilde{P}_n(dz) - \hat{\lambda}_l \quad \forall\, l \le L,$$

$$\bar{\boldsymbol{D}}'' \ \text{ s.t. } \ \left[\bar{\boldsymbol{D}}''(\boldsymbol{u})\right]_{l,l'} = \int \left[\frac{e^{u_l}\omega_l(z)\delta_{ll'}}{\sum_{k=1}^{K} e^{u_k}\omega_k(z)} - \frac{e^{u_l}\omega_l(z)e^{u_{l'}}\omega_{l'}(z)}{\left(\sum_{k=1}^{K} e^{u_k}\omega_k(z)\right)^2}\right]\bar{P}(dz) \quad \forall\, l, l' \le L,$$

$$\hat{\boldsymbol{D}}''_n \ \text{ s.t. } \ \left[\hat{\boldsymbol{D}}''_n(\boldsymbol{u})\right]_{l,l'} = \int \left[\frac{e^{u_l}\omega_l(z)\delta_{ll'}}{\sum_{k=1}^{K} e^{u_k}\omega_k(z)} - \frac{e^{u_l}\omega_l(z)e^{u_{l'}}\omega_{l'}(z)}{\left(\sum_{k=1}^{K} e^{u_k}\omega_k(z)\right)^2}\right]\tilde{P}_n(dz) \quad \forall\, l, l' \le L.$$

Observe that Systems (9.5) and (9.7) are equivalent to $\bar{\boldsymbol{D}}'(\boldsymbol{u}^*) = \boldsymbol{0}$ and $\hat{\boldsymbol{D}}'_n(\hat{\boldsymbol{u}}_n) = \boldsymbol{0}$ respectively, with the changes of variables $\boldsymbol{u}^* = \log(\boldsymbol{\lambda}/\boldsymbol{W}^*)$ and $\hat{\boldsymbol{u}}_n = \log(\hat{\boldsymbol{\lambda}}/\hat{\boldsymbol{W}}_n)$. As already mentioned, all these functions are homogeneous of degree 0. Therefore, in the subsequent analysis, we consider them as functions of $K-1$ variables, subject to $W^*_K = \hat{W}_{n,K} = 1$ (or equivalently to $u^*_K = \log(\lambda_k)$ and $\hat{u}_{n,K} = \log(\hat{\lambda}_K)$), in order to ensure uniqueness of the solutions. Lemma 9.20 shows that controlling $\|\hat{\boldsymbol{u}}_n - \boldsymbol{u}^*\|$ is enough to control $\|\hat{\boldsymbol{W}}_n - \boldsymbol{W}^*\|$.

Lemma 9.20. *Suppose that Assumption 9.1 is fulfilled, and let $B > 0$ such that for all $k \le K : \log(1/B) \le \hat{u}_{n,k}$, and $\log(1/B) \le u^*_k$. Then, placing ourselves in the event that System (9.7) has a unique solution $\hat{\boldsymbol{W}}_n$ s.t. $\hat{W}_{n,K} = 1$, we almost-surely have:*

$$\left\|\hat{\boldsymbol{W}}_n - \boldsymbol{W}^*\right\| \le B\left(\left\|\hat{\boldsymbol{u}}_n - \boldsymbol{u}^*\right\| + \frac{C\sqrt{K}}{\sqrt{n}}\right).$$

Proof. Let $B > 0$ such that for all $k \le K : \log(1/B) \le \hat{u}_{n,k}$, and $\log(1/B) \le u^*_k$. Then for all $k \le K$

$$\left|\hat{W}_{n,k} - W^*_k\right| = \left|\hat{\lambda}_k e^{-\hat{u}_{n,k}} - \lambda_k e^{-u^*_k}\right| \le \left|e^{-\hat{u}_{n,k}} - e^{-u^*_k}\right| + \left|\hat{\lambda}_k - \lambda_k\right| e^{-u^*_k},$$

$$\left|\hat{W}_{n,k} - W^*_k\right| \le B\left(|\hat{u}_{n,k} - u^*_k| + \frac{C}{\sqrt{n}}\right),$$

$$\left\|\hat{\boldsymbol{W}}_n - \boldsymbol{W}^*\right\| \le B\left(\left\|\hat{\boldsymbol{u}}_n - \boldsymbol{u}^*\right\| + \frac{C\sqrt{K}}{\sqrt{n}}\right).$$

□

Remark 9.21. *Although B is easy to derive for $\boldsymbol{u}^*$ (indeed, it is direct to see that for all $k \leq K : \xi \leq \Omega_k \leq M$, so that $W_k^* \leq M/\xi$, $u_k^* = \log(\lambda_k/W_k^*) \geq \log(\lambda_{min}\xi/M)$, and finally $B = M/(\lambda_{min}\xi)$), more work is needed to find an explicit lower bound for $\hat{\boldsymbol{u}}_n$. Actually, assumptions on the ω_k functions are necessary for simple derivations.*

First let us assume that the ω_k functions are constant on their domain. This assumption encompasses the Stratified Sampling *framework, in which the ω_k functions are indicator functions of subsets of the input space. For $k, l \leq K$, let $I_{l,k}$ be the set $\{i : \omega_l(Z_{k,i}) \neq 0\}$, and $\#I_{l,k}$ its cardinality. System (9.7) then rewrites*

$$\#\left\{\bigcup_{l' \neq l} I_{l',l}\right\} = \left(\#\left\{\bigcup_{l' \neq l} I_{l',l}\right\} + \sum_{k \neq l} \#I_{l,k}\right) \frac{\omega_l e^{\hat{u}_{n,l}}}{\boldsymbol{\omega}^\top e^{\hat{\boldsymbol{u}}_n}} \qquad \forall l \leq K,$$

$$e^{\hat{u}_{n,l}} = \frac{\#\left\{\bigcup_{l' \neq l} I_{l',l}\right\}}{\#\left\{\bigcup_{l' \neq l} I_{l',l}\right\} + \sum_{k \neq l} \#I_{l,k}} \frac{\boldsymbol{\omega}^\top e^{\hat{\boldsymbol{u}}_n}}{\omega_l} \qquad \forall l \leq K.$$

With $e^{\hat{u}_{n,K}} = \hat{\lambda}_K$, one can compute $\boldsymbol{\omega}^\top e^{\hat{\boldsymbol{u}}_n}$, and then every $\hat{u}_{n,l}$ for $l \leq K-1$. Finding B is then straightforward.

Another case where B can be derived easily is when the ω_k functions have all the same domain. For instance, the may be strictly positive on all the input space (e.g. Gaussian, Laplace, Student). System (9.7) writes

$$n_l = \sum_{k=1}^{K} \sum_{i=1}^{n_k} \frac{e^{\hat{u}_{n,l}} \omega_l(Z_{k,i})}{\sum_{l'=1}^{K} e^{\hat{u}_{n,l'}} \omega_{l'}(Z_{k,i})} \qquad \forall l \leq K.$$

Let $l \leq K-1$. Assume that for every pair (k,i) it holds $e^{\hat{u}_{n,l}} \omega_l(Z_{k,i}) < \frac{n_l}{n_K} e^{\hat{u}_{n,K}} \omega_K(Z_{k,i})$. Summing over k and i then gives $n_l < n_l$. So there exists $k_0 = k_0(l)$ and $i_0 = i_0(l)$ such that $e^{\hat{u}_{n,l}} \omega_l(Z_{k_0,i_0}) \geq \hat{\lambda}_l \omega_K(Z_{k_0,i_0})$, or equivalently $\hat{u}_{n,l} \geq \log\left(\hat{\lambda}_l \omega_K(Z_{k_0,i_0})/\omega_l(Z_{k_0,i_0})\right)$. Taking the minimal lower bound over l gives B.

It is now the purpose of the following lemma to show that a control on $\|\hat{\boldsymbol{u}}_n - \boldsymbol{u}^*\|$ can be achieved by studying the deviation $\|\hat{\boldsymbol{D}}'_n(\boldsymbol{u}^*) - \bar{\boldsymbol{D}}'(\boldsymbol{u}^*)\|$.

Lemma 9.22. *There exists $L > 0$ such that, on the event that System (9.7) has a unique solution $\hat{\boldsymbol{W}}_n$ such that $\hat{W}_{n,K} = 1$, we almost-surely have:*

$$\left\|\hat{\boldsymbol{u}}_n - \boldsymbol{u}^*\right\| \leq L \left\|\hat{\boldsymbol{D}}'_n(\boldsymbol{u}^*) - \bar{\boldsymbol{D}}'(\boldsymbol{u}^*)\right\|.$$

Proof. First notice that for any compact set $\mathcal{C} \subset \mathbb{R}^{K-1}$, there exists $\sigma^* = \sigma^*(\mathcal{C}) > 0$ such that $\forall \boldsymbol{u} \in \mathcal{C}, \quad \mathbf{Sp}(\hat{\boldsymbol{D}}''_n(\boldsymbol{u})) \subset [\sigma^*, +\infty[$. Indeed, it has been shown in Gill et al. (1988) that the matrix $\hat{\boldsymbol{D}}''_n(\boldsymbol{u})$ is positive definite at each point $\boldsymbol{u}$ (see proof of Proposition 1.1 therein). Since $\hat{\boldsymbol{D}}''_n$ is continuous, so is the function associating $\boldsymbol{u}$ to the smallest eigenvalue of $\hat{\boldsymbol{D}}''_n(\boldsymbol{u})$. As a consequence, it attains its minimum on $\mathcal{C}$. Thanks to the previous remark, we know that this minimum, $\sigma^* = \sigma^*(\mathcal{C})$, is strictly positive, and that for all $\boldsymbol{u} \in \mathcal{C}$: $\mathbf{Sp}(\hat{\boldsymbol{D}}''_n(\boldsymbol{u})) \subset [\sigma^*, +\infty[$. Now, let $\mathcal{C}$ be the segment $[\hat{\boldsymbol{u}}_n, \boldsymbol{u}^*]$, and consider

the function $\boldsymbol{F} : [0,1] \to \mathbb{R}^K$ such that $\boldsymbol{F}(t) = \hat{\boldsymbol{D}}'_n \left(\hat{\boldsymbol{u}}_n + t(\boldsymbol{u}^* - \hat{\boldsymbol{u}}_n) \right)$. We have

$$\boldsymbol{F}(1) - \boldsymbol{F}(0) = \left(\int_0^1 \boldsymbol{F}'(t) dt \right),$$

$$\hat{\boldsymbol{D}}'_n(\boldsymbol{u}^*) - \hat{\boldsymbol{D}}'_n(\hat{\boldsymbol{u}}_n) = \left(\int_0^1 \left[\hat{\boldsymbol{D}}''_n(\hat{\boldsymbol{u}}_n + t(\boldsymbol{u}^* - \hat{\boldsymbol{u}}_n)) \right] (\boldsymbol{u}^* - \hat{\boldsymbol{u}}_n) dt \right),$$

$$\hat{\boldsymbol{D}}'_n(\boldsymbol{u}^*) - \bar{\boldsymbol{D}}'(\boldsymbol{u}^*) = \left(\int_0^1 \left[\hat{\boldsymbol{D}}''_n(\hat{\boldsymbol{u}}_n + t(\boldsymbol{u}^* - \hat{\boldsymbol{u}}_n)) \right] dt \right) (\boldsymbol{u}^* - \hat{\boldsymbol{u}}_n),$$

where the integral of a matrix must be understood componentwise. It is then easy to check that the matrix $(\int_0^1 [\hat{\boldsymbol{D}}''_n(\hat{\boldsymbol{u}}_n + t(\boldsymbol{u}^* - \hat{\boldsymbol{u}}_n))] dt)$ is also positive definite with spectrum in $[\sigma^*, +\infty[$, from what we deduce

$$\left\| \boldsymbol{u}^* - \hat{\boldsymbol{u}}_n \right\| \leq \frac{1}{\sigma^*} \left\| \hat{\boldsymbol{D}}'_n(\boldsymbol{u}^*) - \bar{\boldsymbol{D}}'(\boldsymbol{u}^*) \right\|.$$

□

Lemma 9.23. *Let $\hat{h}_n : \mathcal{Z} \to \mathbb{R}$ and $h : \mathcal{Z} \to \mathbb{R}$ be two real-valued functions. Assume that there exist $a, b \in \mathbb{R}^2$ such that: $a \leq h(z) \leq b$ for all $z \in \mathcal{Z}$. If Assumption 9.1 is fulfilled, then it holds with probability at least $1 - \delta$*

$$\left| \int \hat{h}_n(z) \tilde{P}_n(dz) - \int h(z) \bar{P}(dz) \right|$$
$$\leq \sup_z \left| \hat{h}_n(z) - h(z) \right| + \frac{KC \sup_z |h(z)|}{\sqrt{n}} + (b-a) \sqrt{\frac{1}{2\underline{\lambda} n} \log \frac{2K}{\delta}}.$$

Proof.

$$\left| \int \hat{h}_n(z) \tilde{P}_n(dz) - \int h(z) \bar{P}(dz) \right|$$

$$\leq \left| \int \hat{h}_n(z) \tilde{P}_n(dz) - \int h(z) \tilde{P}_n(dz) \right| + \left| \int h(z) \tilde{P}_n(dz) - \int h(z) \bar{P}(dz) \right|,$$

$$\leq \sup_z \left| \hat{h}_n(z) - h(z) \right| + \left| \sum_{k=1}^K \hat{\lambda}_k \int h(z) \hat{P}_k(dz) - \sum_{k=1}^K \lambda_k \int h(z) P_k(dz) \right|,$$

$$\leq \sup_z \left| \hat{h}_n(z) - h(z) \right| + \left| \sum_{k=1}^K \hat{\lambda}_k \int h(z) \hat{P}_k(dz) - \sum_{k=1}^K \hat{\lambda}_k \int h(z) P_k(dz) \right|$$
$$+ \left| \sum_{k=1}^K \hat{\lambda}_k \int h(z) P_k(dz) - \sum_{k=1}^K \lambda_k \int h(z) P_k(dz) \right|,$$

$$\leq \sup_z \left|\hat{h}_n(z) - h(z)\right| + \sum_{k=1}^{K} \hat{\lambda}_k \left| \int h(z)\hat{P}_k(dz) - \int h(z)P_k(dz) \right| + \sup_z |h(z)| \sum_{k=1}^{K} \left|\hat{\lambda}_k - \lambda_k\right|,$$

$$\leq \sup_z \left|\hat{h}_n(z) - h(z)\right| + \frac{KC \sup_z |h(z)|}{\sqrt{n}} + \sum_{k=1}^{K} \hat{\lambda}_k \left| \int h(z)\hat{P}_k(dz) - \int h(z)P_k(dz) \right|.$$

Applying Hoeffding's inequality gives that for all $t > 0$ and all $k \leq K$ it holds

$$\mathbb{P}\left\{ \left| \int h(z)\hat{P}_k(dz) - \int h(z)P_k(dz) \right| \geq t \right\} \leq 2\exp\left(-\frac{2n_k t^2}{(b-a)^2}\right) \leq 2\exp\left(-\frac{2\underline{\lambda} n t^2}{(b-a)^2}\right).$$

A direct application of the union bound finally gives that with probability at least $1-\delta$

$$\left| \int \hat{h}_n(z)\tilde{P}_n(dz) - \int h(z)\bar{P}(dz) \right|$$
$$\leq \sup_z \left|\hat{h}_n(z) - h(z)\right| + \frac{KC \sup_z |h(z)|}{\sqrt{n}} + (b-a)\sqrt{\frac{1}{2\underline{\lambda} n} \log \frac{2K}{\delta}}.$$

□

Lemma 9.24. *If Assumption 9.1 holds, then with probability at least* $1-\delta$

$$\left\| \hat{\boldsymbol{D}}'_n(\boldsymbol{u}^*) - \bar{\boldsymbol{D}}'(\boldsymbol{u}^*) \right\| \leq \sqrt{K} \left(\sqrt{\frac{1}{2\underline{\lambda} n} \log \frac{2K^2}{\delta}} + \frac{(K+1)C}{\sqrt{n}} \right).$$

Proof. Apply Lemma 9.23 on every l component (with $\hat{h}_n = e^{u_l^*}\omega_l / (\sum_k e^{u_k^*}\omega_k) - \hat{\lambda}_l$, and $h = e^{u_l^*}\omega_l / (\sum_k e^{u_k^*}\omega_k) - \lambda_l$), and conclude with the union bound. □

Proposition 9.19 is then proved by combining Lemmas 9.20, 9.22 and 9.24 and setting $C_1 = BL/\sqrt{2\underline{\lambda}}$, and $C_2 = BC(2L+1)$.

Step 3: Deviation of $\hat{\boldsymbol{\Omega}}_n$

Indeed, one must estimate $\boldsymbol{\Omega}$, and not $\boldsymbol{W}^*$. Hopefully, it can be recovered from $\boldsymbol{W}^*$. For $l \leq K$, we have

$$\Omega_l = \int \omega_l(z)P(dz) = \frac{\int \omega_l(z)P(dz)}{\int P(dz)} = \frac{\int \omega_l(z) \left(\sum_{k=1}^{K} \frac{\lambda_k \omega_k(z)}{\Omega_k} \right)^{-1} \bar{P}(dz)}{\int \left(\sum_{k=1}^{K} \frac{\lambda_k \omega_k(z)}{\Omega_k} \right)^{-1} \bar{P}(dz)},$$
$$= \frac{W_l^*}{\int \left(\sum_{k=1}^{K} \frac{\lambda_k \omega_k(z)}{W_k^*} \right)^{-1} \bar{P}(dz)}.$$

The result stated below provides a sharp control of the deviations of the natural estimate $\hat{\Omega}_{n,l} = \hat{W}_{n,l} / \int \left(1 / \sum_{k=1}^{K} \frac{\hat{\lambda}_k \omega_k(z)}{\hat{W}_{n,k}} \right) \tilde{P}_n(dz)$.

Proposition 9.25. *Suppose that Assumptions 9.1 and 9.8 are fulfilled. Then, there exist $C_1', C_2' > 0$ such that for every $\delta > 0$ we have with probability at least $1-\delta$: System (9.7) has a unique solution $\hat{\boldsymbol{W}}_n$ s.t. $\hat{W}_{n,K} = 1$ and*

$$\forall\, l \leq K, \quad \left|\hat{\Omega}_{n,l} - \Omega_l\right| \leq \sqrt{K} \left(C_1' \sqrt{\frac{1}{n} \log \frac{8K^3}{\delta}} + \frac{C_2' K}{\sqrt{n}} \right),$$

$$\textit{which implies} \quad \left\|\hat{\boldsymbol{\Omega}}_n - \boldsymbol{\Omega}\right\| \leq K \left(C_1' \sqrt{\frac{1}{n} \log \frac{8K^3}{\delta}} + \frac{C_2' K}{\sqrt{n}} \right).$$

Proof. Let $B, B' > 0$ such that for all $k \leq K$ it holds $\log(1/B) \leq \hat{u}_{n,k} \leq \log(\underline{\lambda}/B')$ and $\log(1/B) \leq u_k^* \leq \log(\lambda_{\min}/B')$. This assumption ensures that $B' \leq \hat{W}_{n,k} \leq B$ and $B' \leq W_k^* \leq B$ for all $k \leq K$. A similar assumption, has been made in Lemma 9.20. Following the reasoning used in the proof of Lemma 9.20, constant B' can also be made explicit in several specific but quite general cases.

$$\begin{aligned}
\left|\hat{\Omega}_{n,l} - \Omega_l\right| &= \left| \frac{\hat{W}_{n,l}}{\int \left(\sum_{k=1}^K \frac{\hat{\lambda}_k \omega_k(z)}{\hat{W}_{n,k}} \right)^{-1} \tilde{P}_n(dz)} - \frac{W_l^*}{\int \left(\sum_{k=1}^K \frac{\lambda_k \omega_k(z)}{W_k^*} \right)^{-1} \bar{P}(dz)} \right|, \\
&\leq \frac{1}{\int \left(\sum_{k=1}^K \frac{\hat{\lambda}_k \omega_k(z)}{\hat{W}_{n,k}} \right)^{-1} \tilde{P}_n(dz)} \left| \hat{W}_{n,l} - W_l^* \right| \\
&\quad + W_l^* \left| \frac{1}{\int \left(\sum_{k=1}^K \frac{\hat{\lambda}_k \omega_k(z)}{\hat{W}_{n,k}} \right)^{-1} \tilde{P}_n(dz)} - \frac{1}{\int \left(\sum_{k=1}^K \frac{\lambda_k \omega_k(z)}{W_k^*} \right)^{-1} \bar{P}(dz)} \right|, \\
&\leq \frac{M}{B'} \left| \hat{W}_{n,l} - W_l^* \right| \qquad (9.11) \\
&\quad + B \left(\frac{M}{B'} \right)^2 \left| \int \left(\sum_{k=1}^K \frac{\hat{\lambda}_k \omega_k(z)}{\hat{W}_{n,k}} \right)^{-1} \tilde{P}_n(dz) - \int \left(\sum_{k=1}^K \frac{\lambda_k \omega_k(z)}{W_k^*} \right)^{-1} \bar{P}(dz) \right|.
\end{aligned}$$

From Proposition 9.19, we have that with probability at least $1-\delta$

$$\forall l \leq K, \quad \left|\hat{W}_{n,l} - W_l^*\right| \leq \sqrt{K} \left(C_1 \sqrt{\frac{1}{n} \log \frac{2K^2}{\delta}} + \frac{C_2 K}{\sqrt{n}} \right).$$

As for the second term, one has

$$B' \leq \hat{W}_{n,k} \leq B \qquad \text{and} \qquad B' \leq W_k^* \leq B \quad \forall\, k \leq K, \qquad (9.12)$$

so that

$$\frac{B'}{M} \leq \left(\sum_{k=1}^K \frac{\hat{\lambda}_k \omega_k(z)}{\hat{W}_{n,k}} \right)^{-1} \leq \frac{B}{m\underline{\lambda}} \qquad \text{and} \qquad \frac{B'}{M} \leq \left(\sum_{k=1}^K \frac{\lambda_k \omega_k(z)}{W_k^*} \right)^{-1} \leq \frac{B}{m\lambda_{\min}}.$$

Then

$$\begin{aligned}
&\left| \left(\sum_{k=1}^{K} \frac{\hat{\lambda}_k \omega_k(z)}{\hat{W}_{n,k}} \right)^{-1} - \left(\sum_{k=1}^{K} \frac{\lambda_k \omega_k(z)}{W_k^*} \right)^{-1} \right| \\
&\qquad \leq \left(\frac{B}{m\underline{\underline{\lambda}}} \right)^2 \left| \sum_{k=1}^{K} \frac{\hat{\lambda}_k \omega_k(z)}{\hat{W}_{n,k}} - \sum_{k=1}^{K} \frac{\lambda_k \omega_k(z)}{W_k^*} \right|, \\
&\qquad \leq M \left(\frac{B}{m\underline{\underline{\lambda}}} \right)^2 \sum_{k=1}^{K} \left| \frac{\hat{\lambda}_k}{\hat{W}_{n,k}} - \frac{\lambda_k}{\hat{W}_{n,k}} \right| + \left| \frac{\lambda_k}{\hat{W}_{n,k}} - \frac{\lambda_k}{W_k^*} \right|, \\
&\qquad \leq M \left(\frac{B}{m\underline{\underline{\lambda}}} \right)^2 \sum_{k=1}^{K} \frac{\left| \hat{\lambda}_k - \lambda_k \right|}{\hat{W}_{n,k}} + \lambda_k \left| \frac{1}{\hat{W}_{n,k}} - \frac{1}{W_k^*} \right|, \\
&\qquad \leq M \left(\frac{B}{m\underline{\underline{\lambda}}} \right)^2 \left(\frac{KC}{B'\sqrt{n}} + \frac{1}{B'^2} \sum_{k=1}^{K} \lambda_k \left| \hat{W}_{n,k} - W_k^* \right| \right),
\end{aligned}$$

where $\underline{\underline{\lambda}} = \min\{\underline{\lambda}; \lambda_{\min}\}$. Applying again Proposition 9.19, we get that with probability at least $1 - \delta$

$$\left| \left(\sum_{k=1}^{K} \frac{\hat{\lambda}_k \omega_k(z)}{\hat{W}_{n,k}} \right)^{-1} - \left(\sum_{k=1}^{K} \frac{\lambda_k \omega_k(z)}{W_k^*} \right)^{-1} \right| \leq \sqrt{K} \left(\overline{C}_1 \sqrt{\frac{1}{n} \log \frac{2K^2}{\delta}} + \frac{\overline{C}_2 K}{\sqrt{n}} \right),$$

with $\overline{C}_1 = \frac{MB^2 C_1}{\left(m\underline{\underline{\lambda}} B'\right)^2}$, and $\overline{C}_2 = \frac{MB^2}{\left(m\underline{\underline{\lambda}}\right)^2} \left(\frac{C}{B'} + \frac{C_2}{B'^2} \right)$.

Applying Lemma 9.23 with $\hat{h}_n = \left(\sum_{k=1}^{K} \frac{\hat{\lambda}_k \omega_k}{\hat{W}_{n,k}} \right)^{-1}$ and $h = \left(\sum_{k=1}^{K} \frac{\lambda_k \omega_k(z)}{W_k^*} \right)^{-1}$, one gets that with probability at least $1 - \delta$

$$\begin{aligned}
&\left| \int \left(\sum_{k=1}^{K} \frac{\hat{\lambda}_k \omega_k(z)}{\hat{W}_{n,k}} \right)^{-1} \tilde{P}_n(dz) - \int \left(\sum_{k=1}^{K} \frac{\lambda_k \omega_k(z)}{W_k^*} \right)^{-1} \bar{P}(dz) \right| \\
&\qquad \leq \sqrt{K} \left(\overline{\overline{C}}_1 \sqrt{\frac{1}{n} \log \frac{4K^2}{\delta}} + \frac{\overline{\overline{C}}_2 K}{\sqrt{n}} \right),
\end{aligned}$$

with $\overline{\overline{C}}_1 = \overline{C}_1 + \frac{B}{m\lambda_{\min}\sqrt{2\underline{\lambda}}}$, and $\overline{\overline{C}}_2 = \overline{C}_2 + \frac{CB}{m\lambda_{\min}}$. Hence, for $l \leq K$, it holds with probability $1 - \delta$

$$\left| \hat{\Omega}_{n,l} - \Omega_l \right| \leq \sqrt{K} \left(C_1' \sqrt{\frac{1}{n} \log \frac{8K^2}{\delta}} + \frac{C_2' K}{\sqrt{n}} \right),$$

with $C_1' = \frac{C_1 M}{B'} + \frac{\overline{\overline{C}}_1 BM^2}{B'^2}$ and $C_2' = \frac{C_2 M}{B'} + \frac{\overline{\overline{C}}_2 BM^2}{B'^2}$.

Finally, the union bound gives that with probability at least $1-\delta$

$$\left\|\hat{\mathbf{\Omega}}_n - \mathbf{\Omega}\right\| \leq K\left(C_1'\sqrt{\frac{1}{n}\log\frac{8K^3}{\delta}} + \frac{C_2'K}{\sqrt{n}}\right).$$

□

Step 4: Deviation of $\hat{L}_n(\theta)$

The bound for the deviation between $\hat{\mathbf{\Omega}}_n$ and $\mathbf{\Omega}$ next permits, at fixed $\theta \in \Theta$, to describe the concentration properties of the empirical process

$$\left|\hat{L}_n(\theta) - L(\theta)\right| = \left|\int (1/\sum_{k=1}^{K}\frac{\hat{\lambda}_k\omega_k(z)}{\hat{\Omega}_{n,k}})\ell(z,\theta)\tilde{P}_n(dz) - \int (1/\sum_{k=1}^{K}\frac{\lambda_k\omega_k(z)}{\Omega_k})\ell(z,\theta)\bar{P}(dz)\right|.$$

Proposition 9.26. *Suppose that Assumptions 9.1 and 9.8 are fulfilled. Then, there exist $C_1'', C_2'' > 0$ such that for all $\theta \in \Theta$, any $\delta > 0$, we have with probability larger than $1-\delta$: System (9.7) has a unique solution $\hat{\boldsymbol{W}}_n$ s.t. $\hat{W}_{n,K} = 1$ and*

$$\left|\hat{L}_n(\theta) - L(\theta)\right| \leq \sqrt{K}\left(C_1''\sqrt{\frac{1}{n}\log\frac{16K^3}{\delta}} + \frac{C_2''K}{\sqrt{n}}\right).$$

Proof. Assume that $|\ell(z,\theta)| \leq 1$ for all pair (θ, z). Fix $\theta \in \Theta$.

$$\begin{aligned}
&\left|\hat{L}_n(\theta) - L(\theta)\right| \\
&\quad = \left|\mathbb{E}_{\hat{P}_n}[\ell(Z,\theta)] - \mathbb{E}_P[\ell(Z,\theta)]\right| \\
&\quad = \left|\int \left(\sum_{k=1}^{K}\frac{\hat{\lambda}_k\omega_k(z)}{\hat{\Omega}_k}\right)^{-1}\ell(z,\theta)\tilde{P}_n(dz) - \int \left(\sum_{k=1}^{K}\frac{\lambda_k\omega_k(z)}{\Omega_k}\right)^{-1}\ell(z,\theta)\bar{P}(dz)\right|.
\end{aligned}$$

We recover the second term in Equation (9.11), but with $\hat{\Omega}_k$ and Ω_k instead of $\hat{W}_{n,k}$ and W_k^* respectively. The same technique can be used with small changes. Equation (9.12) becomes

$$\frac{mB'\underline{\lambda}}{B} \leq \hat{\Omega}_{n,l} \leq \frac{MB}{B'} \qquad \text{and} \qquad \frac{mB'\lambda_{\min}}{B} \leq \Omega_k \leq M \qquad \forall\, k \leq K,$$

so that

$$\frac{mB'\underline{\lambda}}{MB} \leq \left(\sum_{k=1}^{K}\frac{\hat{\lambda}_k\omega_k(z)}{\hat{\Omega}_{n,k}}\right)^{-1}\ell(z,\theta) \leq \frac{MB}{mB'\underline{\lambda}},$$

and

$$\frac{mB'\lambda_{\min}}{MB} \leq \left(\sum_{k=1}^{K}\frac{\lambda_k\omega_k(z)}{\Omega_k}\right)^{-1}\ell(z,\theta) \leq \frac{M}{m\lambda_{\min}}.$$

Then

$$\left| \left(\sum_{k=1}^{K} \frac{\hat{\lambda}_k \omega_k(z)}{\hat{\Omega}_{n,k}} \right)^{-1} \ell(z,\theta) - \left(\sum_{k=1}^{K} \frac{\lambda_k \omega_k(z)}{\Omega_k} \right)^{-1} \ell(z,\theta) \right|$$

$$\leq \left(\frac{MB}{mB' \underline{\lambda}} \right)^3 \frac{KC}{\sqrt{n}} + \frac{1}{M} \left(\frac{MB}{mB' \underline{\lambda}} \right)^4 \sum_{k=1}^{K} \lambda_k \left| \hat{\Omega}_{n,k} - \Omega_k \right|.$$

Applying Proposition 9.25, we get that with probability at least $1-\delta$

$$\left| \left(\sum_{k=1}^{K} \frac{\hat{\lambda}_k \omega_k(z)}{\hat{\Omega}_{n,k}} \right)^{-1} \ell(z,\theta) - \left(\sum_{k=1}^{K} \frac{\lambda_k \omega_k(z)}{\Omega_k} \right)^{-1} \ell(z,\theta) \right|$$

$$\leq \sqrt{K} \left(\widetilde{C}_1 \sqrt{\frac{1}{n} \log \frac{8K^3}{\delta}} + \frac{\widetilde{C}_2 K}{\sqrt{n}} \right),$$

with $\widetilde{C}_1 = \frac{C_1'}{M} \left(\frac{MB}{mB' \underline{\lambda}} \right)^4$, and $\widetilde{C}_2 = \frac{C_2'}{M} \left(\frac{MB}{mB' \underline{\lambda}} \right)^4 + \left(\frac{MB}{mB' \underline{\lambda}} \right)^3$.

Applying Lemma 9.23 with $\hat{h}_n = \left(\sum_{k=1}^{K} \frac{\hat{\lambda}_k \omega_k}{\hat{\Omega}_{n,k}} \right)^{-1}$ and $h = \left(\sum_{k=1}^{K} \frac{\lambda_k \omega_k(z)}{\Omega_k} \right)^{-1}$, one gets that with probability at least $1-\delta$

$$\left| \int \left(\sum_{k=1}^{K} \frac{\hat{\lambda}_k \omega_k(z)}{\hat{\Omega}_{n,k}} \right)^{-1} \ell(z,\theta) \tilde{P}_n(dz) - \int \left(\sum_{k=1}^{K} \frac{\lambda_k \omega_k(z)}{\Omega_k} \right)^{-1} \ell(z,\theta) \bar{P}(dz) \right|$$

$$\leq \sqrt{K} \left(C_1'' \sqrt{\frac{1}{n} \log \frac{16K^3}{\delta}} + \frac{C_2'' K}{\sqrt{n}} \right),$$

with $C_1'' = \widetilde{C}_1 + \frac{M}{m \lambda_{\min} \sqrt{2\underline{\lambda}}}$, and $C_2'' = \widetilde{C}_2 + \frac{CM}{m \lambda_{\min}}$. □

Step 5: Proof of Theorem 9.17 by Chaining Arguments

Finally, the maximal deviation bound stated in Theorem 9.17 is obtained from the pointwise bound of Proposition 9.26 combined with a classic *chaining* argument (see *e.g.* Dudley (1999)), involving the complexity of class $\mathcal{F}$, *cf* Assumption 9.16. Observe first that: $\forall (\theta,\ \theta') \in \Theta^2$,

$$\left| \hat{L}_n(\theta) - L(\theta) \right| \leq \left| \hat{L}_n(\theta) - \hat{L}_n(\theta') \right| + \left| \hat{L}_n(\theta') - L(\theta_i) \right| + \left| L(\theta') - L(\theta) \right|$$

$$\leq \frac{2MB}{mB' \underline{\lambda}} \left\| \ell(\cdot,\theta) - \ell(\cdot,\theta') \right\|_{L_1(Q)} + \left| \hat{L}_n(\theta') - L(\theta_i) \right|,$$

where $Q = (\tilde{P}_n + \bar{P})/2$, by using the definitions of $\hat{L}_n$ and L, and the upper bounds derived in the previous subsection.

Let $\zeta > 0$, and $\theta_1, \ldots, \theta_{\mathcal{N}(\zeta,\mathcal{F},L_1(Q))}$ a ζ-coverage of $\mathcal{F}$ with respect to $L_1(Q)$. Set $\mathcal{N} = \mathcal{N}(\zeta, \mathcal{F}, L_1(Q))$ for simplicity. Let θ be an arbitrary element of Θ. By definition, there exists $i \leq \mathcal{N}$ such that $\sup_Q \|\ell(\cdot, \theta) - \ell(\cdot, \theta_i)\|_{L_1(Q)} \leq \zeta$. Applying the bound above, we get:

$$\left|\hat{L}_n(\theta) - L(\theta)\right| \leq \frac{2MB}{mB'\underline{\underline{\lambda}}}\zeta + \left|\hat{L}_n(\theta_i) - L(\theta_i)\right|.$$

Proposition 9.26 combined with the union bound also gives that with probability at least $1 - \delta$

$$\begin{aligned}\sup_{i\leq\mathcal{N}} \left|\hat{L}_n(\theta_i) - L(\theta_i)\right| &\leq \sqrt{K}\left(C_1''\sqrt{\frac{1}{n}\log\frac{16\mathcal{N}K^3}{\delta}} + \frac{C_2''K}{\sqrt{n}}\right), \\ &\leq \sqrt{K}\left(C_1''\sqrt{\frac{1}{n}\log\frac{16C_0K^3}{\delta\zeta^r}} + \frac{C_2''K}{\sqrt{n}}\right),\end{aligned}$$

so that it also holds with probability at least $1 - \delta$

$$\sup_{\theta\in\Theta} \left|\hat{L}_n(\theta) - L(\theta)\right| \leq C_3''\zeta + \sqrt{K}\left(C_1''\sqrt{\frac{1}{n}\log\frac{16C_0K^3}{\delta\zeta^r}} + \frac{C_2''K}{\sqrt{n}}\right),$$

with $C_3'' = \frac{2MB}{mB'\underline{\underline{\lambda}}}$. This bound remaining valid for any $\zeta > 0$, one can now optimize on the value of ζ. Choosing $\zeta \sim 1/\sqrt{n}$ finally gives that it holds with probability at least $1 - \delta$:

$$\sup_{\theta\in\Theta} \left|\hat{L}_n(\theta) - L(\theta)\right| \leq \sqrt{K}\left(C_1''\sqrt{\frac{1}{n}\log\frac{16C_0K^3n^{r/2}}{\delta}} + \frac{C_2''K}{\sqrt{n}}\right) + \frac{C_3''}{\sqrt{n}}.$$

□

9.5 Numerical Experiments

In this section, we display some numerical experiments that confirm the benefits of the debiasing approach detailed in Section 9.3. While Section 9.5.1 focuses on synthetic estimation experiments, Section 9.5.2 is devoted to learning experiments on real-world datasets. Experiments have been run in Python, and the code used to perform debiased ERM is publicly available at: https://github.com/plaforgue/db_learn.

9.5.1 Estimation Experiments

The synthetic data here consists of 1000 train and 300 test realizations of a 3-dimensional Gaussian random vector. The goal pursued is to predict the norm of the realizations via four learning algorithms: Linear Regression (LR), Kernel Ridge Regression (KRR), Support Vector Regression (SVR) and Random Forest (RF). They are implemented with default hyperparameters, as focus is not on performances *per se*, but rather on the impact of the debiasing procedure for a given model.

The simplest biasing functions ω_k's one can imagine, and that are used here, are the indicator functions of subspaces of $\mathbb{R}^3$. These functions (or equivalently the subsets) are chosen according to twelve different scenarios, in order to contrast the debiasing effects. When one biasing function is the identity (or one subspace is the whole $\mathbb{R}^3$), the algorithm is also trained on the sole unbiased sample. However, this approach does not benefit from the whole dataset, and performances reported compare unfavorably to debiased ERM. Numerical results are gathered in Tables 9.1 and 9.2. For scenarios in which no subspace is $\mathbb{R}^3$, two lines are displayed: the upper one corresponds to the standard ERM (std), while the second one is achieved through the debiased approach (**dbs**). When one subspace is $\mathbb{R}^3$, a third line is added, which corresponds to the result obtained with training on the sole unbiased sample (ubs).

Let us now thoroughly describe the first six scenarios, that depict situations where selection bias applies directly to the norm of the realizations, and whose visualizations are available in Figure 9.2.

a) To understand scenario a), one must have in mind that 1.5 is approximately the median value of $\|x\|$ when $x \sim \mathcal{N}(\mathbf{0}_3, \mathbf{I}_3)$ (see $\chi^2(3)$ law). Hence, partitioning the whole space using $\mathbb{1}\{\|x\| \leq 1.6\}$ and $\mathbb{1}\{\|x\| \geq 1.4\}$ (the two subspaces must intersect) divides $\mathbb{R}^3$ into parts of roughly equal importance. Considering two samples of equal size, each associated to one of these biasing functions, should therefore be almost equivalent to considering blindly the concatenated sample. Consequently, debiasing ERM in this scenario should not lead to any particular improvement. This is exactly what is verified empirically. As no subset is the full space, no third line is provided. On the contrary, if the samples were of different sizes, one should expect an improvement when using debiasing ERM.

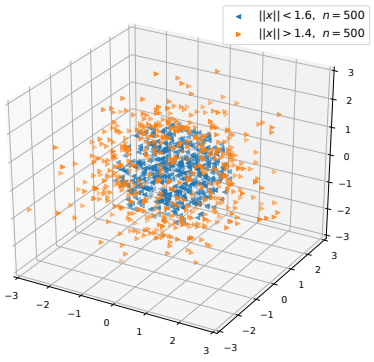
b) In order to emphasize this effect, scenario b) considers strongly concentrated points around 0, with $\mathbb{1}\{\|x\| \leq 0.8\}$. A sample of size 900 is drawn from this part of the space, which usually represents 10% of the distribution, while a 100 long unbiased sample completes the scenario. As expected, the debiasing ERM appears to be less fooled by the outnumbered examples with small norm, and induces a significant improvement compared to the naive ERM. ERM based the sole unbiased sample is also globally outperformed.

c) Scenario c) is similar to scenario b), with less imbalanced samples. Debiasing ERM remains the most successful approach, but by expected lower margins.

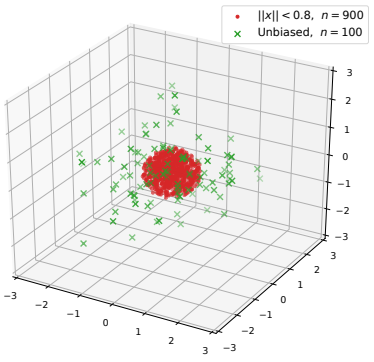
d) What happens if one attempts to fight the selection bias towards $\mathbf{0}_3$ and consider a second sample biased towards great norms, rather than an unbiased one ? It is the purpose of scenarios d) and e) to investigate this option, using $\mathbb{1}\{\|x\| \geq 0.5\}$ as a second biasing function. Almost no change can be acknowledged when the sample sizes are the same as in scenario c) (see scenario d)).

e) However, the advantage of debiasing ERM decreases with the proportion of small norm points, as illustrated by scenario e).

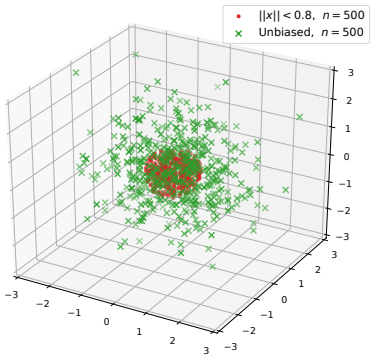
f) Finally, scenario f) illustrates that the number of samples is of low importance. If the sample biased towards small norms is large enough, debiasing ERM outperforms all other methods, even if two additional samples are considered, one biased towards large norms, and one unbiased.



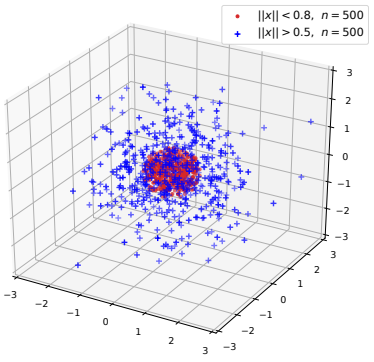
(a) Scenario a)



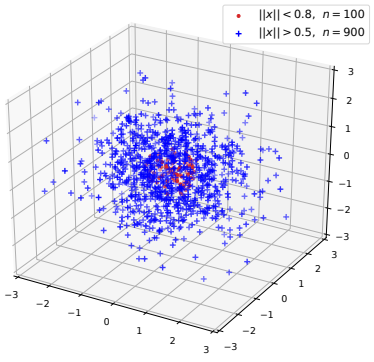
(b) Scenario b)



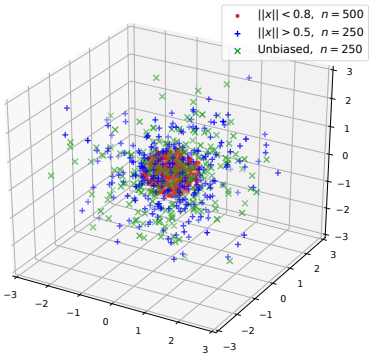
(c) Scenario c)



(d) Scenario d)



(e) Scenario e)



(f) Scenario f)

Figure 9.2 – Different scenarios when selection bias occur on norm

		LR	KRR	SVR	RF
Sc. a)	std	**4.58e-1 ± 4.01e-2**	6.77e-2 ± 2.91e-2	6.62e-3 ± 2.72e-3	**3.36e-2 ± 6.66e-3**
	dbs	4.59e-1 ± 3.97e-2	**6.33e-2 ± 2.81e-2**	**6.54e-3 ± 2.64e-3**	3.39e-2 ± 6.60e-3
Sc. b)	std	1.30e+0 ± 9.81e-2	3.18e-1 ± 7.46e-2	3.77e-2 ± 1.18e-2	1.45e-1 ± 3.20e-2
	dbs	**4.83e-1 ± 4.81e-2**	**1.81e-1 ± 5.64e-2**	4.42e-2 ± 1.33e-2	**1.18e-1 ± 2.78e-2**
	ubs	4.84e-1 ± 4.88e-2	3.40e-1 ± 7.75e-2	**3.04e-2 ± 9.71e-3**	1.31e-1 ± 2.77e-2
Sc. c)	std	7.21e-1 ± 6.63e-2	1.05e-1 ± 3.68e-2	**1.01e-2 ± 4.03e-3**	5.22e-2 ± 1.08e-2
	dbs	**4.61e-1 ± 3.80e-2**	**7.66e-2 ± 3.13e-2**	1.03e-2 ± 3.73e-3	**4.53e-2 ± 9.03e-3**
	ubs	**4.61e-1 ± 3.80e-2**	1.03e-1 ± 3.66e-2	1.06e-2 ± 4.06e-3	4.63e-2 ± 8.93e-3
Sc. d)	std	6.98e-1 ± 6.55e-2	1.01e-1 ± 3.60e-2	**9.82e-3 ± 3.83e-3**	5.09e-2 ± 1.02e-2
	dbs	**4.58e-1 ± 3.84e-2**	**7.51e-2 ± 3.07e-2**	9.92e-3 ± 3.56e-3	**4.43e-2 ± 8.53e-3**
Sc. e)	std	4.60e-1 ± 4.03e-2	6.23e-2 ± 2.74e-2	6.19e-3 ± 2.46e-3	3.35e-2 ± 6.70e-3
	dbs	**4.56e-1 ± 3.82e-2**	**6.01e-2 ± 2.68e-2**	**6.16e-3 ± 2.41e-3**	**3.29e-2 ± 6.32e-3**
Sc. f)	std	7.08e-1 ± 6.80e-2	1.01e-1 ± 3.55e-2	**9.72e-3 ± 3.61e-3**	5.11e-2 ± 1.08e-2
	dbs	**4.59e-1 ± 3.91e-2**	**7.40e-2 ± 2.99e-2**	9.85e-3 ± 3.36e-3	**4.44e-2 ± 8.82e-3**
	ubs	4.65e-1 ± 4.10e-2	1.69e-1 ± 5.10e-2	1.67e-2 ± 5.77e-3	6.86e-2 ± 1.46e-2

Table 9.1 – Mean Squared Errors by 4 Algorithms on the *Norm Biased* Scenarios

All numerical results can be found in Table 9.1 and attest that: 1) ignoring selection bias may have dramatic consequences 2) discarding some data and learning only on the unbiased sample – when it exists – is not a viable solution either, thus endorsing the debiased approach we promote.

One may however argue that results presented in Table 9.1 overestimate the debiasing effect, as bias occurs precisely on the problem's target. In the following, we present similar results obtained when selection bias applies on one component of the Gaussian only, and not on the norm itself. Again, six different scenarios have been investigated, and depicted in Figure 9.3, while complete numerical results are gathered in Table 9.2.

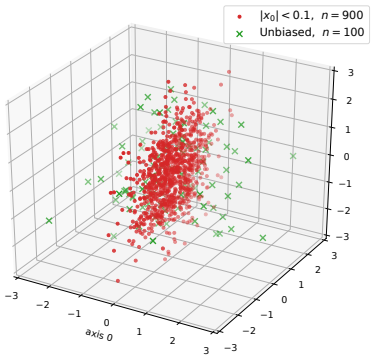
g) , **h)** Scenarios g) and h) are analogous to scenarios b) and c) respectively, except that only one component is biased towards small values with $\mathbb{1}\{|x_0| < 0.1\}$. The improvements induced by debiasing ERM remains substantial, and decrease expectedly as the unbiased sample becomes larger (scenario h)).

i) Scenario i) illustrates that debiasing ERM may improve the results even if a bias applies on large values, using $\mathbb{1}\{x_0 > 1.5\}$ for instance. However, this bias does not distort the predictions towards small norm values, inducing smaller squared norm errors, hence the smaller benefit of debiased ERM.

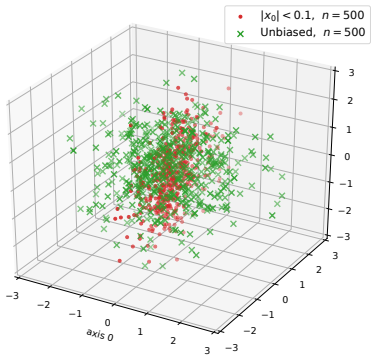
j) Scenario j) is analogous to scenario a), but with 3 samples. It leads to similar conclusions: when the blind concatenated sample is similar to an unbiased sample (the interval $|x_0| < 0.1$ indeed represents 10% of the distribution), debiased ERM is of lower interest.

k) But when the proportions are not respected anymore, as in scenario k), it significantly increases the performances.

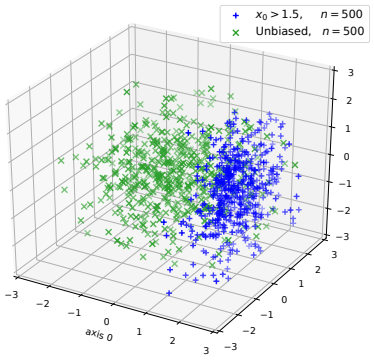
l) Finally, scenario l) involves 4 samples, with similar conclusions as above.



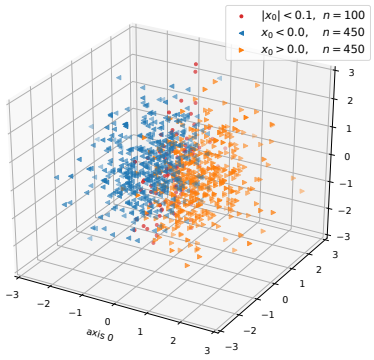
(a) Scenario g)



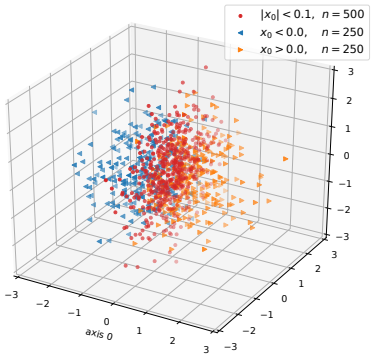
(b) Scenario h)



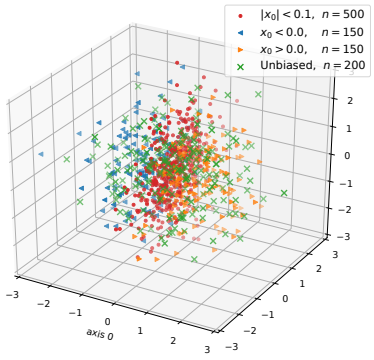
(c) Scenario i)



(d) Scenario j)



(e) Scenario k)



(f) Scenario l)

Figure 9.3 – Different scenarios when selection bias occur on first dimension

		LR	KRR	SVR	RF
Sc. g)	std	5.61e-1 ± 5.66e-2	2.04e-1 ± 5.77e-2	**1.54e-2 ± 5.38e-3**	1.40e-1 ± 3.19e-2
	dbs	**4.80e-1 ± 4.48e-2**	**1.61e-1 ± 5.33e-2**	3.78e-2 ± 1.26e-2	**8.55e-2 ± 2.10e-2**
	ubs	4.82e-1 ± 4.56e-2	3.37e-1 ± 8.14e-2	3.00e-2 ± 1.03e-2	1.29e-1 ± 2.98e-2
Sc. h)	std	4.88e-1 ± 4.66e-2	8.68e-2 ± 3.37e-2	**8.27e-3 ± 3.24e-3**	4.39e-2 ± 9.08e-3
	dbs	**4.59e-1 ± 3.96e-2**	**7.55e-2 ± 3.12e-2**	9.99e-3 ± 3.51e-3	**4.06e-2 ± 8.05e-3**
	ubs	4.59e-1 ± 3.97e-2	1.03e-1 ± 3.73e-2	1.07e-2 ± 3.89e-3	4.64e-2 ± 9.30e-3
Sc. i)	std	5.53e-1 ± 4.84e-2	6.71e-2 ± 2.93e-2	**6.66e-3 ± 2.33e-3**	3.92e-2 ± 7.92e-3
	dbs	**4.58e-1 ± 3.83e-2**	**6.71e-2 ± 2.88e-2**	8.72e-3 ± 3.02e-3	**3.84e-2 ± 7.83e-3**
	ubs	4.58e-1 ± 3.87e-2	1.02e-1 ± 3.74e-2	1.06e-2 ± 3.92e-3	4.60e-2 ± 8.95e-3
Sc. j)	std	**4.57e-1 ± 4.01e-2**	6.44e-2 ± 2.89e-2	**6.36e-3 ± 2.60e-3**	3.33e-2 ± 6.92e-3
	dbs	4.58e-1 ± 3.99e-2	**6.32e-2 ± 2.88e-2**	6.53e-3 ± 2.64e-3	**3.32e-2 ± 6.83e-3**
Sc. k)	std	4.86e-1 ± 4.55e-2	8.72e-2 ± 3.49e-2	**8.34e-3 ± 3.35e-3**	4.40e-2 ± 9.24e-3
	dbs	**4.60e-1 ± 3.98e-2**	**7.64e-2 ± 3.27e-2**	1.00e-2 ± 3.70e-3	**4.09e-2 ± 8.58e-3**
Sc. l)	std	4.88e-1 ± 4.71e-2	8.64e-2 ± 3.32e-2	**8.21e-3 ± 3.18e-3**	4.40e-2 ± 8.82e-3
	dbs	**4.60e-1 ± 3.99e-2**	**7.50e-2 ± 3.10e-2**	9.91e-3 ± 3.46e-3	**4.08e-2 ± 8.29e-3**
	ubs	4.69e-1 ± 4.18e-2	2.04e-1 ± 5.81e-2	1.98e-2 ± 7.00e-3	8.13e-2 ± 1.74e-2

Table 9.2 – Mean Squared Errors by 4 Algorithms on the *1 Component Biased* Scenarios

Again, and although bias does not apply on the target itself, but instead on one simple covariate, the debiasing approach naturally yields improvements, both upon standard and unbiased (on fewer observations) methods.

9.5.2 Learning Experiments

In many practical applications, the data acquisition process cannot be fully mastered, information being collected in several goes over specific strata of the population of interest, and statistical learning then relies on a collection of biased data samples. It is precisely the purpose of the procedure investigated in this chapter to address this crucial issue.

Boston dataset

As an illustration, we consider here the *Boston* housing dataset problem, where the price of a house is to be predicted based on 14 attributes, such as the number of rooms or neighborhood statistics. One may easily conceive that the dataset at disposal is actually composed of two samples: one large open dataset, in which the most expensive houses do not appear for privacy purposes, and a second one, unbiased but smaller, taken *e.g.* from a local estate agency. This setting can be simulated the following way: from the 500 observations available, 400 are kept as a first training sample. Two samples are then derived from it: a biased one with the cheapest houses of size 250, and an unbiased one of size 50. Models are trained on the 300 selected observations, and tested on the other 100 ones first set aside. Numerical results are displayed in Table 9.3 in terms of Mean Squared Errors (MSEs), validating the soundness of the debiasing approach.

Adult dataset

The machine-learning problem associated to the *Adult* dataset, also known as the *Census Income* dataset (freely available at https://archive.ics.uci.edu/ml/datasets/adult), is a binary classification task, where the goal is to predict whether a person's income exceeds 50,000$ a year, based on census data.

	LR	SVR	RF
Standard ERM	26.83 ± 8.13	98.63 ± 17.69	18.26 ± 7.00
Debiased ERM	**25.71 ± 6.59**	**84.73 ± 15.27**	**17.94 ± 7.25**
Unbiased Sample	28.02 ± 7.57	85.05 ± 15.13	19.27 ± 7.30

Table 9.3 – MSEs on *Boston* dataset (100 runs average)

By nature, such data combine population strata, making it the perfect playground for the approach we promote. For instance, and as revealed by Figure 9.4a, the proportion of persons having an income exceeding 50k$ a year substantially depends on the number of years of education. If highly educated people happen to be over-represented in the dataset (which is totally plausible as it is more convenient to poll people concentrated in big cities, usually more educated, than people living in the countryside), it should deteriorate the predictions in absence of a debiasing procedure.

Furthermore, notice that this setting cannot be casted as simple covariate shift, since conditional laws can by no means be considered as identical from one group to another. Figure 9.4b highlights these differences by showing the income's dependence with respect to the age for three different levels of education. The striking difference between the curves makes it impossible to consider the covariate shift as a reasonable assumption. In contrast, the general debiasing framework developed in this article perfectly suits the following situation.

The first experimental protocol is as follows. From the whole set of observations, 1500 are kept for the testing phase. From the rest are sampled two subgroups: one of 12+ years of education people of size 5 900, and one unbiased (*i.e.* sampled from the entire population) of size 100. A Logistic Regression (LogReg) and a Random Forest (RF) are then trained on the concatenation of the 6 000 observations, with and without debiasing procedure, as well as on the small second sample of size 100 only. Numerical results are summarized in Table 9.4 in terms of prediction scores. As expected, the standard LogReg totally collapses, with a deficit of 16% compared to its debiased version. As for the RF, a partition method, it severely suffers from the lack of data when only the small unbiased sample is used. The debiased ERM procedures, however, behaves nicely in all circumstances.

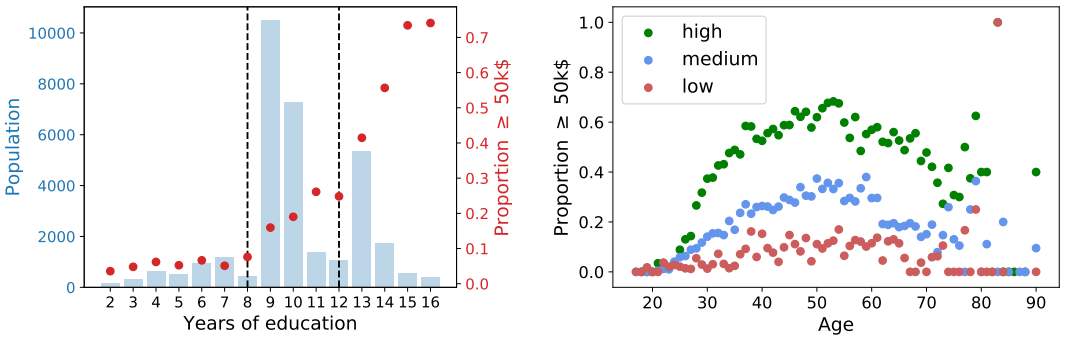
The second protocol is relatively similar. First notice that the age of the subject has a strong impact on his/her probability to earn more than 50k$ a year (see Figure 9.5a). Moreover, and as for the example based on years of education, this scenario cannot be casted as a covariate shift problem. Indeed, the conditional laws cannot be assumed to remain identical. Figure 9.5b illustrates this phenomenon by showing the dependence of the income with respect to the years of education by age group. Clearly, middle age people take more advantage of their education than younger people, which is totally normal as they are working for a longer period. This observation makes simple covariate shift impossible to consider here.

If middle age people happen to be over-represented in the training dataset, it should induce a general over-estimation of the probability, unless the debiasing procedure is used. This setting has been simulated as follows. From the initial observations, 5 000 are kept for the testing phase. From the rest are sampled two subgroups: one of middle age people of size 9 900, and one unbiased (*i.e.* sampled from the entire population) of

size 100. A Logistic Regression (LogReg) and a Random Forest (RF) are then trained on the concatenation of the 10 000 observations, with and without debiasing procedure, as well as on the small second sample of size 100 only. Numerical results are summarized in Table 9.5 in terms of prediction scores. Again, the debiased version of the ERM yields the best performances, and for both algorithms. The gaps are however less spectacular than that presented for the first protocol. It is probably due to a softer biasing effect than the one achieved when it applies to the years of education. The less striking difference between conditional laws (Figure 9.4b and Figure 9.5b) is another marker that the debiasing effect expected in this latter example is less important.

Hence, these three learning examples, either on regression or classification tasks, that cannot be tackled through ordinary covariate shift (whether the bias applies to the target, or the conditional laws obviously change), empirically confirm that ignoring selection bias in the learning procedure – or discarding data to keep only unbiased observations – jeopardizes most algorithms. It thus strongly supports the relevance of the debiasing approach detailed in this chapter.

Figure 9.4 – Impact of *Education* on salary



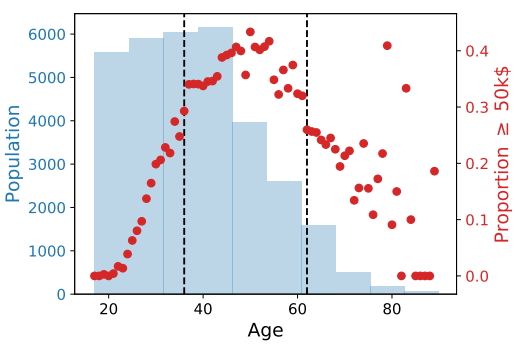
(a) Salary ≥ 50k$ w.r.t. education

(b) Salary ≥ 50k$ w.r.t. age by education level

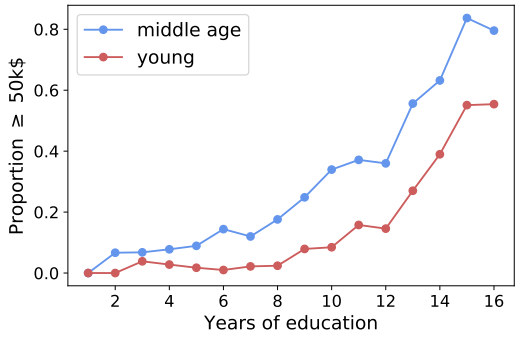
	LogReg	RF
Standard ERM	63.95 ± 1.37	42.73 ± 3.36
Debiased ERM	**79.77 ± 1.72**	**43.58 ± 4.77**
Unbiased Sample	77.75 ± 2.27	22.16 ± 6.18

Table 9.4 – Prediction scores on *Adult* dataset, bias on *Education* (100 runs average)

Figure 9.5 – Impact of *Age* on salary



(a) Salary $\geq$ 50k$ w.r.t. age



(b) Salary $\geq$ 50k$ w.r.t. education by age group

	LogReg	RF
Standard ERM	78.74 $\pm$ 1.24	83.52 $\pm$ 0.52
Debiased ERM	**80.90 $\pm$ 1.09**	**84.09 $\pm$ 0.62**
Unbiased Sample	77.96 $\pm$ 1.96	80.46 $\pm$ 1.17

Table 9.5 – Prediction scores on the *Adult* dataset, bias on *Age* (100 runs average)

9.6 Conclusion

In this chapter, we have provided a sound methodology to address bias selection issues in statistical learning. We have extended the paradigmatic ERM approach to the situation where learning is based on biased training samples. In contrast to alternative techniques documented in the literature, the method proposed is very general and can be possibly applied to any ERM-like learning algorithm straightforwardly. It relies on a preliminary debiasing of the raw empirical risk functional in the spirit of the procedure introduced in Vardi (1985) for cumulative density function estimation in biased sampling models. The theoretical analysis carried out under mild assumptions reveals that the learning rate thus achieved is the same as that attained in absence of any selection bias phenomenon. This is also empirically confirmed by the illustrative examples displayed in Section 9.5.

Conclusion and Perspectives

The contribution presented in this manuscript is twofold.

First, in Part I, we have introduced and studied a new hypothesis set: the composition of functions from vector-valued Reproducing Kernel Hilbert Spaces. Inspired from Deep Learning architectures, it also benefits from the thorough theoretical understanding and the rigorous complexity control we have on kernel functions. But the main advantage of this architecture surely lies in its ability to deal with infinite dimensional inputs and outputs. A kernelization step then allows to cope with any complex structured data, as soon as a kernel can be defined on them. The theoretical guarantees, stated in terms of excess risk, however require a trace class assumption, which is not granted for the useful identity decomposable kernel for instance. A possible alternative could be found in stability approaches, long neglected because of the alternate descent scheme promoted to optimize our model.

Indeed, Chapter 5's duality approach shows that the last layer of the architecture can be finitely parametrized, even when outputs are infinite dimensional. This notably opens the door to a full Gradient Descent approach, made possible by a Representer Theorem dedicated to compositional kernel architectures. This technique should improve the convergence process, so far made difficult by the non-convexity of the objective and the tendency of alternate approaches to find local minima. Another optimization related research direction is the introduction of kernel approximations. They can be used both for the inputs and the outputs, and should drastically reduce the computation time, especially if they are combined with a doubly stochastic scheme.

These speed-ups would result in making the approach more practical on large unlabeled structured datasets, for which Kernel Autoencoders are of particular interest. Indeed, beyond providing a deep extension of Kernel Principal Component Analysis, Kernel Autoencoders can be used in semi-supervised settings, where they can help learning a relevant output embedding for instance. Finally, and as an answer to the initial objective which consisted in taking advantage from both deep and kernel methods, one of the most promising model seems to me the hybrid architecture. With first and last layers only as operator-valued kernel mappings, it would be able to handle complex data, while benefiting from the deep machinery at its core. This could definitely be considered as taking the best of the two worlds.

Second, in Part II, we have studied some alternatives to the standard Empirical Risk Minimization framework. Indeed, when training data are biased, contain outliers, or come from heavy-tailed distributions, the empirical mean may not be the best substitute to the expectation. An estimator of particular interest is the Median-of-Means. It can be shown to be sub-Gaussian, on the sole requirement that the targeted distribution has a finite second order moment. Interestingly, these strong concentration properties have been extended to randomization and to the case of U-statistics. Nevertheless, the choice of randomization is limited to the sampling without replacement for technical reasons, and the particular case of Median-of-Incomplete-U-Statistics is still unsolved. A tighter analysis of V-statistics concentration might be the key to remedy these situations.

Median-of-Means-like estimators can be used to perform learning in two ways. The first one is the closest to Empirical Risk Minimization, and advocates minimizing a Median-of-Means estimator of the risk. This paradigm has been shown to nicely extend to randomized and U-statistics settings, although it suffers from slower rates, compared to the second learning approach. Solutions are approximately computed through an adaptation of Gradient Descent, that often converges toward local solutions. In order to avoid this problem, one usually adds an artificial randomization at each step. This, however, is naturally incorporated in the randomized version of Median-of-Means we have introduced, making the analysis easier. The second way to use Median-of-Means estimators is to perform tournament procedures. This approach has been adapted to the pairwise setting, and benefits from strong guarantees, even for heavy-tailed data. Its computation for infinite hypothesis sets remains nonetheless a challenge. Addressing this issue, by leveraging ε-coverages for instance, would make it a credible and serious contender to Empirical Risk Minimization in many unfriendly situations.

Finally, the sample bias issue was tackled by proposing a general reweighting adaptation of Empirical Risk Minimization. The minimizer of the debiased risk estimate has been proven to satisfy guarantees of the same order as that of an unbiased risk minimizer. If requiring some knowledge about the biasing mechanism at work seems reasonable, assuming the biasing functions to be known might be unrealistic in practice. A key research direction would then consist in studying how an approximation of the biasing functions affects the guarantees. If the latter are preserved, the debiased Empirical Risk Minimization framework we have proposed would constitute a valuable asset to address all types of dataset shift scenarios, a crucial concern in modern Machine Learning.

Appendices

A The Bounded Differences Inequality

Proposition. *(The Bounded Differences Inequality, McDiarmid (1989)) Let* $\mathcal{S}_n = \{Z_i\}_{i\leq n}$ *be* n *i.i.d. realizations of a* $\mathcal{Z}$*-valued random variable* Z*, and* $f : \mathcal{Z}^n \to \mathbb{R}$ *such that there exists* $(c_i)_{i\leq n}$ *such that:* $\forall i \leq n, \forall (z_1, \ldots, z_n, z_{i'}) \in \mathcal{Z}^{n+1}$

$$\left| f(z_1, \ldots, z_i, \ldots, z_n) - f(z_1, \ldots, z_{i'}, \ldots, z_n) \right| \leq c_i.$$

Then, with the notation $f(\mathcal{S}_n) = f(Z_1, \ldots, Z_n)$*, it holds for every* $t > 0$

$$\mathbb{P}\left\{ f(\mathcal{S}_n) - \mathbb{E}\left[f(\mathcal{S}_n) \right] > t \right\} \leq \exp\left(-\frac{2t^2}{\sum_{i=1}^n c_i^2} \right),$$
$$\mathbb{P}\left\{ f(\mathcal{S}_n) - \mathbb{E}\left[f(\mathcal{S}_n) \right] < -t \right\} \leq \exp\left(-\frac{2t^2}{\sum_{i=1}^n c_i^2} \right).$$

Notice that Hoeffding's Inequality is a particular case of this proposition, with $f(\mathcal{S}_n) = \frac{1}{n}\sum_{i=1}^n Z_i$*, and* $c_i = (b_i - a_i)/n$*.*

B Probabilities Upper-Bounding

B.1 MoRM

By virtue of Chebyshev's inequality, one gets:

$$\bar{p}_t \leq \frac{\mathbb{E}\left[(\bar{\theta}_1 - \theta)^2 \right]}{t^2} = \frac{\mathbb{E}_{\mathcal{S}_n}\left[\mathbb{E}\left[(\bar{\theta}_1 - \theta)^2 \mid \mathcal{S}_n \right] \right]}{t^2}.$$

Observing that $\mathbb{E}[\bar{\theta}_1 | \mathcal{S}_n] = \hat{\theta}_n$ and that

$$\mathbb{E}\left[(\bar{\theta}_1 - \theta)^2 | \mathcal{S}_n \right] = \mathrm{Var}\left(\bar{\theta}_1 \mid \mathcal{S}_n \right) + (\hat{\theta}_n - \theta)^2 = (\hat{\theta}_n - \theta)^2 + \frac{1}{B}\frac{n-B}{n}\hat{\sigma}_n^2,$$

where $\hat{\sigma}_n^2 = \frac{1}{n-1}\sum_{i=1}^n (Z_i - \hat{\theta}_n)^2$, we deduce that

$$\bar{p}_t \leq \left(\frac{1}{n} + \frac{n-B}{nB} \right) \frac{\sigma^2}{t^2} = \frac{\sigma^2}{Bt^2}.$$

□

MoRU

Observe first that

$$\mathrm{Var}\left(\bar{U}_1(h) \right) = \mathbb{E}\left[\mathrm{Var}(\bar{U}_1(h) \mid \mathcal{S}_n) \right] + \mathrm{Var}\left(\mathbb{E}\left[\bar{U}_1(h) \mid \mathcal{S}_n \right] \right). \tag{13}$$

Recall that $\mathbb{E}[\bar{U}_1(h) \mid \mathcal{S}_n] = U_n(h)$, so that

$$\mathrm{Var}\left(\mathbb{E}\left[\bar{U}_1(h) \mid \mathcal{S}_n \right] \right) = \frac{4\sigma_1^2(h)}{n} + \frac{2\sigma_2^2(h)}{n(n-1)}. \tag{14}$$

In addition, we have, for $B \geq 4$,

$$\begin{aligned}\mathrm{Var}\left(\bar{U}_1(h) \mid \mathcal{S}_n\right) =& \frac{4}{B^2(B-1)^2}\sum_{i<j} h^2(X_i, X_j)\mathrm{Var}(\epsilon_{1,i}\epsilon_{1,j}) \\ &+ \sum_{\substack{i<j,\, k<l \\ (i,j)\neq(k,l)}} \mathrm{Cov}(\epsilon_{1,i}\epsilon_{1,j},\ \epsilon_{1,l}\epsilon_{1,k}) h(X_i, X_j) h(X_k, X_l).\end{aligned}$$

Let $i \neq j$, one may check that

$$\mathrm{Var}(\epsilon_{1,i}\epsilon_{1,j}) = \frac{B(B-1)(n-B)(n+B-1)}{n^2(n-1)^2}.$$

And, for any $k \neq l$, we have

$$\mathrm{Cov}(\epsilon_{1,i}\epsilon_{1,j},\ \epsilon_{1,l}\epsilon_{1,k}) = -\frac{B(B-1)}{n(n-1)}\frac{(n-B)(4nB-6n-6B+6)}{n(n-1)(n-2)(n-3)}$$

when $\{i,j\} \cap \{k,l\} = \emptyset$, as well as

$$\mathrm{Cov}(\epsilon_{1,i}\epsilon_{1,j},\ \epsilon_{1,i}\epsilon_{1,k}) = \frac{B(B-1)}{n(n-1)}\frac{(n-B)(nB-2n-2B+2)}{n(n-1)(n-2)}$$

when $k \neq j$ and $k \neq i$. Hence, observing that $\mathbb{E}[h(X_1, X_2)h(X_1, X_3)] = \sigma_1^2(h) + \theta^2(h)$, we obtain:

$$\begin{aligned}\mathbb{E}\left[\mathrm{Var}(\bar{U}_1(h) \mid \mathcal{S}_n)\right] =& \frac{2(n-B)(n+B-1)}{n(n-1)B(B-1)}\left(\sigma^2(h) + \theta^2(h)\right) \\ &- \frac{(n-B)(4nB-6n-6B+6)}{n(n-1)B(B-1)}\theta^2(h) \\ &+ \frac{4(n-B)(nB-2n-2B+2)}{n(n-1)B(B-1)}(\sigma_1^2(h) + \theta^2(h)). \qquad (15)\end{aligned}$$

Combining (13), (14) and (15), we get:

$$\begin{aligned}\mathrm{Var}\left(\bar{U}_1(h)\right) =& \frac{4\sigma_1^2(h)}{n} + \frac{2\sigma_2^2(h)}{n(n-1)} + \frac{2(n-B)(n+B-1)}{n(n-1)B(B-1)}\left(2\sigma_1^2(h) + \sigma_2^2 + \theta^2(h)\right) \\ &- \frac{(n-B)(4nB-6n-6B+6)}{n(n-1)B(B-1)}\theta^2(h) \\ &+ \frac{4(n-B)(nB-2n-2B+2)}{n(n-1)B(B-1)}(\sigma_1^2(h) + \theta^2(h)), \\ \mathrm{Var}\left(\bar{U}_1(h)\right) =& \frac{4\sigma_1^2(h)}{B} + \frac{2\sigma_2^2(h)}{B(B-1)}.\end{aligned}$$

Chebyshev's inequality permits to conclude. □

C Useful Lemma

Lemma. *Let $d \in \mathbb{N}^*$. Then, for any $n, K \in \mathbb{N}^2$ it holds*

$$K \leq \frac{n}{2d-1} \quad \Rightarrow \quad \left\lfloor \frac{n}{K} \right\rfloor - d + 1 \geq \frac{n}{2K}.$$

Proof. First case: $K \leq \frac{n}{2d}$.

$$\begin{aligned} K &\leq \frac{n}{2d}, \\ d &\leq \frac{n}{2K}, \\ \frac{n}{2K} &\leq \frac{n}{K} - d, \\ \frac{n}{2K} &\leq \left\lfloor \frac{n}{K} \right\rfloor - d + 1. \end{aligned}$$

Second case: $\frac{n}{2d} \leq K \leq \frac{n}{2d-1}$.

$$\begin{aligned} \frac{n}{2d} \leq K &\leq \frac{n}{2d-1}, \\ 2d - 1 &\leq \left\lfloor \frac{n}{K} \right\rfloor \leq 2d, \\ \frac{n}{2K} &\leq d \leq \left\lfloor \frac{n}{K} \right\rfloor - d + 1. \end{aligned}$$

□

D Details on Incomplete U-Statistic Bounded Difference

$$\begin{aligned} \frac{\binom{(n-1)(n-2)/2}{M}}{\binom{n(n-1)/2}{M}} &= \frac{\left(\frac{n(n-1)}{2} - M\right) \dots \left(\frac{n(n-1)}{2} - M - (n-2)\right)}{\left(\frac{n(n-1)}{2}\right) \dots \left(\frac{n(n-1)}{2} - (n-2)\right)}, \\ &= \prod_{k=\frac{n(n-1)}{2}-(n-2)}^{\frac{n(n-1)}{2}} \left(1 - \frac{M}{k}\right), \\ &\geq \left(1 - \frac{2M}{(n-1)(n-2)}\right)^{n-1}, \\ &\geq 1 - \frac{2M}{(n-2)}, \end{aligned}$$

$$1 - \frac{\binom{(n-1)(n-2)/2}{M}}{\binom{n(n-1)/2}{M}} \leq \frac{2M}{(n-2)}.$$

Bibliography

Alain, G. and Bengio, Y. (2014). What regularized auto-encoders learn from the data-generating distribution. *J. Mach. Learn. Res.*, 15(1):3563–3593. Page 34.

Alon, N., Matias, Y., and Szegedy, M. (1999). The space complexity of approximating the frequency moments. *Journal of Computer and system sciences*, 58(1):137–147. Page 115.

Álvarez, M. A., Rosasco, L., and Lawrence, N. D. (2012). Kernels for vector-valued functions: A review. *Foundations and Trends in Machine Learning*, 4(3):195–266. Page 25.

Arcones, M. A. and Gine, E. (1993). Limit theorems for u-processes. *The Annals of Probability*, pages 1494–1542. Page 161.

Aronszajn, N. (1950). Theory of reproducing kernels. *Transactions of the American Mathematical Society*, pages 337–404. Pages 10, 21, 22.

Audibert, J.-Y. and Catoni, O. (2011). Robust linear least squares regression. *The Annals of Statistics*, 39(5):2766–2794. Page 116.

Audiffren, J. and Kadri, H. (2013). Stability of multi-task kernel regression algorithms. In *Asian Conference on Machine Learning*, pages 1–16. Page 51.

Ausset, G., Clémençon, S., and Portier, F. (2019). Empirical Risk Minimization under Random Censorship: Theory and Practice. *Submitted, available at https://arxiv.org/abs/1906.01908*. Pages 166, 167.

Bach, F. R. and Jordan, M. I. (2002). Kernel independent component analysis. *Journal of machine learning research*, 3(Jul):1–48. Page 24.

Baldi, P. (2012). Autoencoders, unsupervised learning, and deep architectures. In *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, pages 37–49. Page 35.

Bartlett, P. L. and Mendelson, S. (2002). Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482. Page 12.

Bauschke, H. H., Combettes, P. L., et al. (2011). *Convex analysis and monotone operator theory in Hilbert spaces*, volume 408. Springer. Pages 42, 78, 79.

Bellet, A. and Habrard, A. (2015). Robustness and Generalization for Metric Learning. *Neurocomputing*, 151(1):259–267. Page 107.

Bellet, A., Habrard, A., and Sebban, M. (2012). Similarity Learning for Provably Accurate Sparse Linear Classification. In *ICML*. Page 107.

Bellet, A., Habrard, A., and Sebban, M. (2013). A Survey on Metric Learning for Feature Vectors and Structured Data. *ArXiv e-prints*. Page 107.

Bellet, A., Habrard, A., and Sebban, M. (2015). *Metric Learning*. Morgan & Claypool Publishers. Page 107.

Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., and Wortman Vaughan, J. (2010). A theory of learning from different domains. *Machine Learning*, 79(1). Pages 32, 166.

Bengio, Y., Courville, A., Vincent, P., and Umanità, V. (2013a). Representation learning: a review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828. Pages 32, 71, 72.

Bengio, Y., Lamblin, P., Popovici, D., and Larochelle, H. (2007). Greedy layer-wise training of deep networks. In *Advances in neural information processing systems*, pages 153–160. Page 33.

Bengio, Y., Yao, L., Alain, G., and Vincent, P. (2013b). Generalized denoising auto-encoders as generative models. In *Advances in neural information processing systems*, pages 899–907. Page 34.

Berlinet, A. and Thomas-Agnan, C. (2011). *Reproducing kernel Hilbert spaces in probability and statistics*. Springer Science & Business Media. Page 25.

Bertail, P. and Tressou, J. (2006). Incomplete generalized u-statistics for food risk assessment. *Biometrics*, 62(1):66–74. Page 111.

Blom, G. (1976). Some properties of incomplete U-statistics. *Biometrika*, 63(3):573–580. Page 110.

Bohn, B., Rieger, C., and Griebel, M. (2019). A representer theorem for deep kernel learning. *Journal of Machine Learning Research*, 20(64):1–32. Page 56.

Bolukbasi, T., Chang, K., Zou, J., Saligrama, V., and Kalai, A. (2016). Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Advances in Neural Information Processing Systems (NIPS)*, page 4349–4357. Page 166.

Bottou, L. (1998). Online learning and stochastic approximations. *On-line learning in neural networks*, 17(9):142. Page 154.

Boucheron, S., Bousquet, O., and Lugosi, G. (2005). Theory of classification : a survey of some recent advances. *ESAIM: Probability and Statistics*, 9:323–375. Page 166.

Boucheron, S., Lugosi, G., and Massart, P. (2013). *Concentration Inequalities: A Nonasymptotic Theory of Independence*. OUP Oxford. Pages 11, 166.

Bourlard, H. and Kamp, Y. (1988). Auto-association by multilayer perceptrons and singular value decomposition. *Biological cybernetics*, 59(4):291–294. Pages 34, 39.

Bousquet, O. and Elisseeff, A. (2002). Stability and generalization. *Journal of Machine Learning Research*, 2(Mar):499–526. Page 51.

Boyd, S. and Vandenberghe, L. (2004). *Convex optimization*. Cambridge university press. Page 78.

Brault, R., Heinonen, M., and Buc, F. (2016). Random fourier features for operator-valued kernels. In *Asian Conference on Machine Learning*, pages 110–125. Pages 60, 101.

Brault, R., Lambert, A., Szabo, Z., Sangnier, M., and d'Alché-Buc, F. (2019). Infinite task learning in rkhss. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1294–1302. Pages 28, 94.

Brouard, C., d'Alché-Buc, F., and Szafranski, M. (2011). Semi-supervised penalized output kernel regression for link prediction. In *International Conference on Machine Learning (ICML)*, pages 593–600. Page 29.

Brouard, C., Shen, H., Dührkop, K., d'Alché-Buc, F., Böcker, S., and Rousu, J. (2016a). Fast metabolite identification with input output kernel regression. *Bioinformatics*, 32(12):28–36. Pages 30, 73, 99.

Brouard, C., Szafranski, M., and d'Alché Buc, F. (2016b). Input output kernel regression: supervised and semi-supervised structured output prediction with operator-valued kernels. *The Journal of Machine Learning Research*, 17(1):6105–6152. Pages 10, 21, 30, 32, 51, 80, 82, 90, 92, 164.

Brownlees, C., Joly, E., Lugosi, G., et al. (2015). Empirical risk minimization for heavy-tailed losses. *The Annals of Statistics*, 43(6):2507–2536. Page 140.

Bruna, J., Zaremba, W., Szlam, A., and LeCun, Y. (2013). Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*. Page 35.

Bubeck, S., Cesa-Bianchi, N., and Lugosi, G. (2013). Bandits with heavy tail. *IEEE Transactions on Information Theory*, 59(11):7711–7717. Page 140.

Burns, K., Hendricks, L., Saenko, K., Darrell, T., and Rohrbach, A. (2018). Women also snowboard: Overcoming bias in captioning models. *arXiv preprint arXiv:1803.09797*. Pages 15, 166.

Callaert, H., Janssen, P., et al. (1978). The Berry-Esseen theorem for U-statistics. *The Annals of Statistics*, 6(2):417–421. Page 159.

Caponnetto, A., Micchelli, C. A., , M., and Ying, Y. (2008). Universal multitask kernels. *Journal of Machine Learning Research*, 9:1615–1646. Page 25.

Carmeli, C., De Vito, E., and Toigo, A. (2006). Vector valued reproducing kernel hilbert spaces of integrable functions and mercer theorem. *Analysis and Applications*, 4(04):377–408. Page 25.

Carmeli, C., De Vito, E., Toigo, A., and Umanitá, V. (2010). Vector valued reproducing kernel hilbert spaces and universality. *Analysis and Applications*, 8(01):19–61. Pages 25, 83, 95, 96.

Catoni, O. (2012). Challenging the empirical mean and empirical variance: a deviation study. In *Annales de l'Institut Henri Poincaré, Probabilités et Statistiques*, volume 48, pages 1148–1185. Institut Henri Poincaré. Pages 116, 136.

Chatelin, F. (2011). *Spectral approximation of linear operators*. SIAM. Page 98.

Chollet, F. et al. (2015). Keras, https://keras.io. Page 73.

Ciliberto, C., Rosasco, L., and Rudi, A. (2016). A consistent regularization approach for structured prediction. In *Advances in Neural Information Processing Systems (NIPS) 29*, pages 4412–4420. Page 29.

Clémençon, S. (2014). A statistical view of clustering performance through the theory of U-processes. *Journal of Multivariate Analysis*, 124:42–56. Page 106.

Clémençon, S., Bertail, P., and Chautru, E. (2017). Sampling and empirical risk minimization. *Statistics*, 51(1):30–42. Page 167.

Clémençon, S., Colin, I., and Bellet, A. (2016). Scaling-up Empirical Risk Minimization: Optimization of Incomplete U-statistics. *Journal of Machine Learning Research*, 17:1–36. Page 111.

Clémençon, S., Lugosi, G., and Vayatis, N. (2005). Ranking and scoring using empirical risk minimization. In *Proceedings of COLT*. Page 107.

Clémençon, S., Lugosi, G., and Vayatis, N. (2008). Ranking and empirical risk minimization of U-statistics. *The Annals of Statistics*, 36(2):844–874. Pages 147, 151, 161.

Clémençon, S., Robbiano, S., and Tressou, J. (2013). Maximal deviations of incomplete u-statistics with applications to empirical risk sampling. In *Proceedings of the 2013 SIAM International Conference on Data Mining*, pages 19–27. SIAM. Page 147.

Cortes, C., Mohri, M., and Weston, J. (2005). A general regression technique for learning transductions. In *International Conference on Machine Learning (ICML)*, pages 153–160. Page 29.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3):273–297. Pages 10, 21, 24, 79.

Cuturi, M., Vert, J.-P., Birkenes, O., and Matsui, T. (2007). A kernel for time series based on global alignments. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, volume 2, pages II–413. IEEE. Pages 10, 24.

Dai, B., Xie, B., He, N., Liang, Y., Raj, A., Balcan, M.-F. F., and Song, L. (2014). Scalable kernel methods via doubly stochastic gradients. In *Advances in Neural Information Processing Systems*, pages 3041–3049. Page 60.

de la Peña, V. H. (1992). Decoupling and khintchine's inequalities for u-statistics. *The Annals of Probability*, pages 1877–1892. Page 161.

de la Peña, V. H. and Giné, E. (1999). *Decoupling: from dependence to independence*. Probability and its Applications. Springer-Verlag, New York. Pages 113, 161.

Devroye, L., Györfi, L., and Lugosi, G. (1996a). *A Probabilistic Theory of Pattern Recognition*. Springer. Page 166.

Devroye, L., Györfi, L., and Lugosi, G. (1996b). *A Probabislistic Theory of Pattern Recognition*. Springer. Page 11.

Devroye, L., Lerasle, M., Lugosi, G., Oliveira, R. I., et al. (2016). Sub-gaussian mean estimators. *The Annals of Statistics*, 44(6):2695–2725. Page 118.

Dhillon, I. S., Guan, Y., and Kulis, B. (2004). Kernel k-means: spectral clustering and normalized cuts. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 551–556. ACM. Page 24.

Dinuzzo, F., Ong, C., Gehler, P., and Pillonetto, G. (2011). Learning output kernels with block coordinate descent. In *International Conference on Machine Learning (ICML)*, pages 49–56. Page 27.

Drucker, H., Burges, C. J., Kaufman, L., Smola, A. J., and Vapnik, V. (1997). Support vector regression machines. In *Advances in neural information processing systems*, pages 155–161. Page 80.

Dubin, J. and Rivers, D. (1989). Selection bias in linear regression, logit and probit models. *Sociological Methods & Research*, 18(2-3):360–390. Page 167.

Dubois-Laforgue, D., Caillat-Zucman, S., Boitard, C., and Timsit, J. (2000). Clinical characteristics of type 2 diabetes in patients with mutations of hfe. *Diabetes & metabolism*, 26(1):65–68. Page 14.

Dudík, M., Phillips, S., and Schapire, R. (2006). Correcting sample selection bias in maximum entropy density estimation. In *Advances in neural information processing systems*, pages 323–330. Page 167.

Dudley, R. (1999). *Uniform Central Limit Theorems*. Cambridge University Press. Page 185.

Duvenaud, D. K., Maclaurin, D., Iparraguirre, J., Bombarell, R., Hirzel, T., Aspuru-Guzik, A., and Adams, R. P. (2015). Convolutional networks on graphs for learning molecular fingerprints. In *Advances in neural information processing systems*, pages 2224–2232. Page 35.

Erhan, D., Bengio, Y., Courville, A., Manzagol, P.-A., Vincent, P., and Bengio, S. (2010). Why does unsupervised pre-training help deep learning? *Journal of Machine Learning Research*, 11(Feb):625–660. Page 33.

Erhan, D., Bengio, Y., Courville, A., and Vincent, P. (2009). Visualizing higher-layer features of a deep network. *University of Montreal*, 1341(3):1. Pages 10, 20, 32.

Fei-Fei, L., Fergus, R., and Perona, P. (2006). One-shot learning of object categories. *IEEE transactions on pattern analysis and machine intelligence*, 28(4):594–611. Page 32.

Gärtner, T. (2008). *Kernels for Structured Data*, volume 72 of *Series in Machine Perception and Artificial Intelligence*. WorldScientific. Page 24.

Gholami, B. and Hajisami, A. (2016). Kernel autoencoder for semi-supervised hashing. In *Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on*, pages 1–8. IEEE. Page 35.

Gill, R., Vardi, Y., and Wellner, J. (1988). Large sample theory of empirical distributions in biased sampling models. *The Annals of Statistics*, 16(3):1069–1112. Pages 167, 170, 171, 173, 174, 179.

Giné, E., Latała, R., and Zinn, J. (2000). Exponential and moment inequalities for u-statistics. In *High Dimensional Probability II*, pages 13–38. Springer. Page 113.

Giné, E. and Zinn, J. (1984). Some limit theorems for empirical processes. *Ann. Probab.*, 12(4):929–998. With discussion. Page 143.

Glorot, X., Bordes, A., and Bengio, Y. (2011). Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 513–520. Page 33.

Godsil, C. and Royle, G. (2001). *Algebraic Graph Theory*. Springer-Verlag. Page 170.

Goodfellow, I. J., Courville, A., and Bengio, Y. (2012). Spike-and-slab sparse coding for unsupervised feature discovery. *arXiv preprint arXiv:1201.3382*. Page 32.

Gori, M., Monfardini, G., and Scarselli, F. (2005). A new model for learning in graph domains. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, volume 2, pages 729–734. IEEE. Page 35.

Grams, W. F., Serfling, R., et al. (1973). Convergence rates for u-statistics and related statistics. *The Annals of Statistics*, 1(1):153–160. Page 113.

Hajek, J. (1968). Asymptotic normality of simple linear rank statistics under alternatives. *Ann. Math. Stat.*, 39:325–346. Page 113.

Hardoon, D. R., Szedmak, S., and Shawe-Taylor, J. (2004). Canonical correlation analysis: An overview with application to learning methods. *Neural computation*, 16(12):2639–2664. Page 24.

Hardt, M., Recht, B., and Singer, Y. (2015). Train faster, generalize better: Stability of stochastic gradient descent. *arXiv preprint arXiv:1509.01240*. Page 51.

Heckman, J. (1979). Sample selection bias as a specification error. *Econometrica: Journal of the econometric society*, pages 153–161. Page 166.

Heckman, J. (1990). Varieties of selection bias. *The American Economic Review*, 80(2):313. Pages 15, 166.

Hinton, G. E., Osindero, S., and Teh, Y.-W. (2006). A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554. Page 33.

Hinton, G. E. and Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507. Pages 33, 34, 41.

Hinton, G. E. and Zemel, R. S. (1994). Autoencoders, minimum description length and helmholtz free energy. In *Advances in neural information processing systems*, pages 3–10. Page 34.

Hoeffding, W. (1948). A class of statistics with asymptotically normal distribution. *Ann. Math. Stat.*, 19:293–325. Pages 104, 107, 113.

Hoeffding, W. (1963). Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30. Pages 110, 113, 116, 168.

Hofmann, T., Schoelkopf, B., and Smola, A. J. (2008). Kernel methods in machine learning. *Ann. Statist.*, 36(3):1171–1220. Page 25.

Hopkins, S. B. (2018). Sub-gaussian mean estimation in polynomial time. *arXiv preprint arXiv:1809.07425*. Page 120.

Hsu, D. and Sabato, S. (2014). Heavy-tailed regression with a generalized median-of-means. In *International Conference on Machine Learning*, pages 37–45. Page 140.

Hsu, D. and Sabato, S. (2016). Loss minimization and parameter estimation with heavy tails. *The Journal of Machine Learning Research*, 17(1):543–582. Pages 118, 119, 140.

Huang, J., Gretton, A., Borgwardt, K., Schölkopf, B., and Smola, A. (2007). Correcting sample selection bias by unlabeled data. In *Advances in neural information processing systems*, pages 601–608. Page 167.

Huber, P. J. (1964). Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, pages 73–101. Pages 86, 90.

Ian J. Goodfellow, I., Bengio, Y., and Aaron C. Courville, A. C. (2016). *Deep Learning. Adaptive computation and machine learning*. MIT Press. Pages 31, 34.

Jerrum, M. R., Valiant, L. G., and Vazirani, V. V. (1986). Random generation of combinatorial structures from a uniform distribution. *Theoretical Computer Science*, 43:169–188. Page 115.

Joachims, T., Hofmann, T., Yue, Y., and Yu, C.-N. (2009). Predicting structured objects with support vector machines. *Commun. ACM*, 52(11):97–104. Page 29.

Joly, E. and Lugosi, G. (2016). Robust estimation of u-statistics. *Stochastic Processes and their Applications*, 126(12):3760–3773. Pages 129, 131.

Joly, E., Lugosi, G., Oliveira, R. I., et al. (2017). On the estimation of the mean of a random vector. *Electronic Journal of Statistics*, 11(1):440–451. Page 119.

Kadri, H., Duflos, E., Preux, P., Canu, S., Rakotomamonjy, A., and Audiffren, J. (2016). Operator-valued kernels for learning from functional response data. *Journal of Machine Learning Research*, 17(20):1–54. Pages 28, 95, 98.

Kampffmeyer, M., Løkse, S., Bianchi, F. M., Jenssen, R., and Livi, L. (2017). Deep kernelized autoencoders. In *Scandinavian Conference on Image Analysis*, pages 419–430. Springer. Page 35.

Kipf, T. N. and Welling, M. (2016a). Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*. Pages 20, 35.

Kipf, T. N. and Welling, M. (2016b). Variational graph autoencoders. *NIPS Workshop on Bayesian Deep Learning*. Page 35.

Koenker, R. (2005). *Quantile regression*. Cambridge university press. Pages 28, 95.

Kolmogorov, A. N. (1941). The local structure of turbulence in incompressible viscous fluid for very large reynolds numbers. *Cr Acad. Sci. URSS*, 30:301–305. Page 22.

Koltchinskii, V. and Mendelson, S. (2015). Bounding the smallest singular value of a random matrix without concentration. *International Mathematics Research Notices*, 2015(23):12991–13008. Page 141.

Krizhevsky, A. and Hinton, G. E. (2011). Using very deep autoencoders for content-based image retrieval. In *ESANN*, volume 1, page 2. Page 34.

Laforgue, **P**., Clémençon, S., and d'Alché-Buc, F. (2019a). Autoencoding any data through kernel autoencoders. In *Artificial Intelligence and Statistics*, pages 1061–1069. Pages 35, 56.

Laforgue, **P**. and Clémençon, S. (2019). Statistical learning from biased training samples. *arXiv preprint arXiv:1906.12304*. Page 171.

Laforgue, **P**., Clemencon, S., and Bertail, P. (2019b). On medians of (Randomized) pairwise means. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 1272–1281, Long Beach, California, USA. PMLR. Page 120.

Laforgue, **P**., Lambert, A., Motte, L., and d'Alché Buc, F. (2019c). On the dualization of operator-valued kernel machines. *arXiv preprint arXiv:1910.04621*. Page 82.

Lai, P. L. and Fyfe, C. (2000). Kernel and nonlinear canonical correlation analysis. *International Journal of Neural Systems*, 10(05):365–377. Page 24.

Larochelle, H. and Bengio, Y. (2008). Classification using discriminative restricted boltzmann machines. In *Proceedings of the 25th international conference on Machine learning*, pages 536–543. ACM. Page 32.

Lecué, G. and Lerasle, M. (2017). Robust machine learning by median-of-means: theory and practice. *arXiv preprint arXiv:1711.10306*. Pages 140, 164.

Lecué, G. and Lerasle, M. (2019). Learning from mom's principles: Le cam's approach. *Stochastic Processes and their applications*, 129(11):4385–4410. Page 164.

Lecué, G., Lerasle, M., and Mathieu, T. (2018). Robust classification via mom minimization. *arXiv preprint arXiv:1808.03106*. Pages 15, 121, 140, 141, 144, 152.

Lecué, G. and Mendelson, S. (2013). Learning subgaussian classes: Upper and minimax bounds. *arXiv preprint arXiv:1305.4825*. Pages 141, 155.

LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., et al. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324. Page 21.

Ledoux, M. and Talagrand, M. (1991). *Probability in Banach Spaces: Isoperimetry and Processes*. Springer Science & Business Media. Page 45.

Lee, A. J. (1990). *U-statistics: Theory and practice*. Marcel Dekker, Inc., New York. Page 113.

Lerasle, M. and Oliveira, R. I. (2011). Robust empirical mean estimators. *arXiv preprint arXiv:1112.3914*. Page 140.

Li, Y., Tarlow, D., Brockschmidt, M., and Zemel, R. (2015). Gated graph sequence neural networks. *arXiv preprint arXiv:1511.05493*. Page 35.

Lin, Y., Lee, Y., and Wahba, G. (2002). Support vector machines for classification in nonstandard situations. *Machine learning*, 46(1-3):191–202. Page 167.

Liu, Z., Yang, J., Liu, H., and Wang, W. (2016). Transfer learning by sample selection bias correction and its application in communication specific emitter identification. *JCM*, 11:417–427. Page 166.

Lugosi, G. (1992). Learning with an unreliable teacher. *Pattern Recognition*, 25(1):79–87. Page 166.

Lugosi, G. and Mendelson, S. (2016). Risk minimization by median-of-means tournaments. *arXiv preprint arXiv:1608.00757*. Pages 140, 154, 155, 156, 157, 159, 160, 161, 162.

Lugosi, G. and Mendelson, S. (2017). Sub-gaussian estimators of the mean of a random vector. *arXiv preprint arXiv:1702.00482*. Page 119.

Lugosi, G., Mendelson, S., et al. (2019). Regularization, sparse recovery, and median-of-means tournaments. *Bernoulli*, 25(3):2075–2106. Page 140.

Mahé, P. and Vert, J.-P. (2009). Graph kernels based on tree patterns for molecules. *Machine learning*, 75(1):3–35. Pages 10, 24.

Mairal, J. (2016). End-to-end kernel learning with supervised convolutional kernel networks. In *Advances in neural information processing systems*, pages 1399–1407. Pages 35, 52.

Mairal, J., Koniusz, P., Harchaoui, Z., and Schmid, C. (2014). Convolutional kernel networks. In *Advances in neural information processing systems*, pages 2627–2635. Pages 35, 52.

Manski, C. and Lerman, S. (1977). The estimation of choice probabilities from choice based samples. *Econometrica: Journal of the Econometric Society*, pages 1977–1988. Page 167.

Matsuda, S., Vert, J.-P., Saigo, H., Ueda, N., Toh, H., and Akutsu, T. (2005). A novel representation of protein sequences for prediction of subcellular location using support vector machines. *Protein Science*, 14(11):2804–2813. Page 14.

Maurer, A. (2014). A chain rule for the expected suprema of gaussian processes. In *Algorithmic Learning Theory: 25th International Conference, ALT 2014, Bled, Slovenia, October 8-10, 2014, Proceedings*, volume 8776, page 245. Springer. Pages 44, 45, 46.

Maurer, A. (2016). A vector-contraction inequality for rademacher complexities. In *International Conference on Algorithmic Learning Theory*, pages 3–17. Springer. Page 45.

Maurer, A. et al. (2019). A bernstein-type inequality for functions of bounded interaction. *Bernoulli*, 25(2):1451–1471. Page 110.

Maurer, A. and Pontil, M. (2016). Bounds for vector-valued function estimation. *arXiv preprint arXiv:1606.01487*. Pages 42, 43.

McClelland, J. L., Rumelhart, D. E., Group, P. R., et al. (1987). *Parallel distributed processing*, volume 2. MIT press Cambridge, MA:. Page 34.

McDiarmid, C. (1989). On the method of bounded differences. In *Surveys in combinatorics, 1989 (Norwich, 1989)*, volume 141 of *London Math. Soc. Lecture Note Ser.*, pages 148–188. Cambridge Univ. Press, Cambridge. Pages 125, 198.

Mendelson, S. (2014). Learning without concentration. In *Conference on Learning Theory*, pages 25–39. Page 141.

Mendelson, S. (2016). Upper bounds on product and multiplier empirical processes. *Stochastic Processes and their Applications*, 126(12):3652–3680. Page 141.

Mendelson, S. (2017). On aggregation for heavy-tailed classes. *Probability Theory and Related Fields*, 168(3-4):641–674. Pages 141, 159.

Mercer, J. (1909). Xvi. functions of positive and negative type, and their connection the theory of integral equations. *Philosophical transactions of the royal society of London. Series A, containing papers of a mathematical or physical character*, 209(441-458):415–446. Page 22.

Mesnil, G., Dauphin, Y., Glorot, X., Rifai, S., Bengio, Y., Goodfellow, I., Lavoie, E., Muller, X., Desjardins, G., Warde-Farley, D., et al. (2011). Unsupervised and transfer learning challenge: a deep learning approach. In *Proceedings of the 2011 International Conference on Unsupervised and Transfer Learning workshop-Volume 27*, pages 97–111. JMLR. org. Page 32.

Micchelli, C. A. and Pontil, M. (2005). On learning vector-valued functions. *Neural computation*, 17(1):177–204. Pages 25, 27, 48, 50, 64.

Mikolov, T., Le, Q. V., and Sutskever, I. (2013). Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168*. Page 33.

Minsker, S. et al. (2015). Geometric Median and Robust Estimation in Banach Spaces. *Bernoulli*, 21(4):2308–2335. Pages 119, 120, 126.

Minsker, S. and Wei, X. (2018). Robust modifications of u-statistics and applications to covariance estimation problems. *arXiv preprint arXiv:1801.05565*. Page 130.

Mohri, M., Rostamizadeh, A., and Talwalkar, A. (2012). *Foundations of Machine Learning*. MIT press. Pages 43, 44.

Moreau, J. J. (1962). Fonctions convexes duales et points proximaux dans un espace hilbertien. *Comptes rendus hebdomadaires des séances de l'Académie des sciences*, 255:2897–2899. Page 86.

Nemirovsky, A. S. and Yudin, D. B. (1983). *Problem Complexity and Method Efficiency in Optimization*. John Wiley & Sons Ltd. Pages 114, 115.

Nocedal, J. and Wright, S. (2006). *Numerical optimization*. Springer Science & Business Media. Page 78.

Nowozin, S. and Lampert, C. H. (2011). Structured learning and prediction in computer vision. *Foundations and Trends in Computer Graphics and Vision*, 6(3-4):185–365. Page 29.

Obozinski, G., Taskar, B., and Jordan, M. I. (2010). Joint covariate selection and joint subspace selection for multiple classification problems. *Statistics and Computing*, 20(2):231–252. Page 89.

Palatucci, M., Pomerleau, D., Hinton, G. E., and Mitchell, T. M. (2009). Zero-shot learning with semantic output codes. In *Advances in neural information processing systems*, pages 1410–1418. Page 32.

Papa, G., Clémençon, S., and Bertail, P. (2016). Learning from Survey Training Samples: Rate Bounds for Horvitz-Thompson Risk Minimizers. In *Proceedings of ACML*. Page 166.

Pedregosa, F. et al. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830. Pages 11, 73, 164, 173.

Pisier, G. (1986). Probabilistic methods in the geometry of banach spaces. In *Probability and analysis*, pages 167–241. Springer. Page 45.

Quionero-Candela, J., Sugiyama, M., Schwaighofer, A., and Lawrence, N. (2009). *Dataset shift in machine learning*. The MIT Press. Pages 15, 167.

Rahimi, A. and Recht, B. (2008). Random features for large-scale kernel machines. In *Advances in neural information processing systems*, pages 1177–1184. Pages 60, 101.

Ramsay, J. O. and Silverman, B. W. (2007). *Applied functional data analysis: methods and case studies*. Springer. Pages 27, 94, 100.

Ranzato, M., Boureau, Y.-L., and Cun, Y. L. (2008). Sparse feature learning for deep belief networks. In *Advances in neural information processing systems*, pages 1185–1192. Page 34.

Ranzato, M., Poultney, C., Chopra, S., and Cun, Y. L. (2007). Efficient learning of sparse representations with an energy-based model. In *Advances in neural information processing systems*, pages 1137–1144. Pages 33, 34.

Rifai, S., Vincent, P., Muller, X., Glorot, X., and Bengio, Y. (2011). Contractive auto-encoders: Explicit invariance during feature extraction. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, pages 833–840. Omnipress. Page 34.

Robbins, H. and Monro, S. (1951). A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407. Page 174.

Rockafellar, R. T. (1970). *Convex analysis*, volume 28. Princeton university press. Page 78.

Rosset, S., Zhu, J., Zou, H., and Hastie, T. (2005). A method for inferring label sampling mechanisms in semi-supervised learning. In *Advances in neural information processing systems*, pages 1161–1168. Page 167.

Saigo, H., Vert, J.-P., Ueda, N., and Akutsu, T. (2004). Protein homology detection using string alignment kernels. *Bioinformatics*, 20(11):1682–1689. Pages 21, 24.

Salakhutdinov, R. and Hinton, G. (2009a). Deep boltzmann machines. In van Dyk, D. and Welling, M., editors, *Proceedings of the Twelth International Conference on Artificial Intelligence and Statistics*, volume 5 of *Proceedings of Machine Learning Research*, pages 448–455. PMLR. Page 34.

Salakhutdinov, R. and Hinton, G. (2009b). Semantic hashing. *International Journal of Approximate Reasoning*, 50(7):969–978. Page 34.

Sangnier, M., Fercoq, O., and d'Alché-Buc, F. (2017). Data sparse nonparametric regression with ϵ-insensitive losses. In *Asian Conference on Machine Learning*, pages 192–207. Pages 80, 82, 86, 88.

Saunders, C., Gammerman, A., and Vovk, V. (1998). Ridge regression learning algorithm in dual variables. In *Proc. of the 15th International Conference on Machine Learning*. Page 80.

Schölkopf, B., Smola, A., and Müller, K.-R. (1997). Kernel principal component analysis. In *International conference on artificial neural networks*, pages 583–588. Springer. Pages 24, 39, 40.

Schölkopf, B., Smola, A., and Müller, K.-R. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural computation*, 10(5):1299–1319. Pages 24, 39, 40, 51.

Schölkopf, B., Smola, A. J., et al. (2002). Learning with kernels: Support vector machines, regularization. *Optimization, and Beyond. MIT press*, 1(2). Page 25.

Schölkopf, B., Tsuda, K., and Vert, J.-P. (2004). *Support vector machine applications in computational biology*. MIT press. Pages 21, 25.

Senkene, E. and Tempel'man, A. (1973). Hilbert spaces of operator-valued functions. *Lithuanian Mathematical Journal*, 13(4):665–670. Page 25.

Serfling, R. (1980). *Approximation Theorems of Mathematical Statistics*. Wiley Series in Probability and Statistics. John Wiley & Sons. Page 113.

Serfling, R. J. (1974). Probability inequalities for the sum in sampling without replacement. *The Annals of Statistics*, pages 39–48. Page 111.

Shawe-Taylor, J., Cristianini, N., et al. (2004). *Kernel methods for pattern analysis*. Cambridge university press. Page 25.

Shehzadex (2017). https://commons.wikimedia.org/wiki/file:kernel_yontemi_ile_veriyi_daha_fazla_dimensiyonlu_uzaya_tasima_islemi.png. Page 24.

Shimodaira, H. (2000). Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 90(2):227–244. Page 167.

Socher, R., Ganjoo, M., Manning, C. D., and Ng, A. (2013). Zero-shot learning through cross-modal transfer. In *Advances in neural information processing systems*, pages 935–943. Page 32.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958. Page 33.

Steinwart, I. and Christmann, A. (2008). *Support vector machines*. Springer Science & Business Media. Pages 25, 95.

Su, H., Heinonen, M., and Rousu, J. (2010). Structured output prediction of anti-cancer drug activity. In Dijkstra, T., Tsivtsivadze, E., Marchiori, E., and Heskes, T., editors, *Pattern Recognition in Bioinformatics - 5th IAPR International Conference, PRIB 2010, Proceedings*, volume 6282 of *Lecture Notes in Computer Science*, pages 38–49. Springer. Page 73.

Sugiyama, M. and Kawanabe, M. (2012). *Machine learning in non-stationary environments: Introduction to covariate shift adaptation*. MIT press. Page 167.

Sugiyama, M. and Müller, K. (2005). Input-dependent estimation of generalization error under covariate shift. *Statistics & Decisions*, 23(4/2005):249–279. Page 167.

Suykens, J. A. et al. (2002). *Least squares support vector machines*. World Scientific. Page 80.

Tseng, P. (2001). Convergence of a block coordinate descent method for nondifferentiable minimization. *J. Optim. Theory Appl.*, 109(3):475–494. Page 89.

Tseng, P. and Yun, S. (2009). Block-coordinate gradient descent method for linearly constrained nonsmooth separable optimization. *J. Optim. Theory Appl.*, 140(3):513. Page 89.

Valsesia, D., Fracastoro, G., and Magli, E. (2018). Learning localized generative models for 3d point clouds via graph convolution. Page 35.

van Belle, V., Pelckmans, K., Suykens, J., and Van Huffel, S. (2011). Learning transformation models for ranking and survival analysis. *Journal of machine learning research*, page 44. Page 166.

van der Vaart, A. (1998). *Asymptotic Statistics*. Cambridge University Press. Pages 113, 168.

van der Vaart, A. and Wellner, J. (1996). *Weak convergence and empirical processes*. Springer-Verlag. Page 176.

van Miltenburg, E. (2016). Stereotyping and bias in the flickr30k dataset. In *Workshop on Multi-modal Corpora: Computer vision and language processing*. Page 166.

Vapnik, V. (1998). Statistical learning theory. Pages 10, 21.

Vardi, Y. (1985). Empirical distributions in selection bias models. *Ann. Statist.*, 13:178–203. Pages 167, 168, 170, 173, 174, 194.

Vella, F. (1998). Estimating models with sample selection bias: a survey. *Journal of Human Resources*, pages 127–169. Page 166.

Vert, J.-P. (2002). A tree kernel to analyse phylogenetic profiles. *Bioinformatics*, 18(suppl_1):S276–S284. Page 24.

Vert, R. and Vert, J.-P. (2006). Consistency and convergence rates of one-class svms and related algorithms. *Journal of Machine Learning Research*, 7(May):817–854. Page 24.

Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., and Manzagol, P.-A. (2010). Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *J. Mach. Learn. Res.*, 11:3371–3408. Pages 34, 35.

Vogel, R., Clémençon, S., and Bellet, A. (2018). A Probabilistic Theory of Supervised Similarity Learning: Pairwise Bipartite Ranking and Pointwise ROC Curve Optimization. In *International Conference in Machine Learning*. Page 107.

Weiss, Y., Torralba, A., and Fergus, R. (2009). Spectral hashing. In *Advances in neural information processing systems*, pages 1753–1760. Page 34.

Williams, C. K. and Seeger, M. (2001). Using the nyström method to speed up kernel machines. In *Advances in neural information processing systems*, pages 682–688. Pages 60, 101.

Winship, C. and Mare, R. (1992). Models for sample selection bias. *Annual review of sociology*, 18(1):327–350. Page 167.

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., and Yu, P. S. (2019). A comprehensive survey on graph neural networks. *arXiv preprint arXiv:1901.00596*. Page 35.

Yamanishi, Y., Vert, J.-P., Nakaya, A., and Kanehisa, M. (2003). Extraction of correlated gene clusters from multiple genomic data by generalized kernel canonical correlation analysis. *Bioinformatics*, 19(suppl_1):i323–i330. Page 24.

Zadrozny, B. (2004). Learning and evaluating classifiers under sample selection bias. In *Proceedings of the twenty-first international conference on Machine learning*, page 114. ACM. Page 167.

Zadrozny, B. and Elkan, C. (2001). Learning and making decisions when costs and probabilities are both unknown. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 204–213. Pages 28, 95.

Zeiler, M. D. and Fergus, R. (2014). Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer. Page 32.

Zhao, J., Wang, T., Yatskar, M., Ordonez, V., and Chang, K. (2017). Men also like shopping: Reducing gender bias amplification using corpus-level constraints. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Page 166.

ECOLE DOCTORALE

Titre : Apprentissage de Représentations par Méthodes à Noyaux Profondes pour les Données Complexes et Problèmes de Fiabilité

Mots clés : Noyaux à Valeurs Opérateurs, Autoencodeurs, Fonctions de perte Robustes, Médiane-des-Moyennes, Bias de Sélection

Résumé : Cette thèse débute par l'étude d'architectures profondes à noyaux pour les données complexes. L'une des clefs du succès des algorithmes d'apprentissage profond est la capacité des réseaux de neurones à extraire des représentations pertinentes. Cependant, les raisons théoriques de ce succès nous sont encore largement inconnues, et ces approches sont presque exclusivement réservées aux données vectorielles. D'autre part, les méthodes à noyaux engendrent des espaces fonctionnels étudiés de longue date, les Espaces de Hilbert à Noyau Reproduisant (Reproducing Kernel Hilbert Spaces, RKHSs), dont la complexité est facilement contrôlée par le noyau ou la pénalisation, tout en autorisant les prédictions dans les espaces structurés complexes via les RKHSs à valeurs vectorielles (vv-RKHSs).

L'architecture proposée consiste à remplacer les blocs élémentaires des réseaux usuels par des fonctions appartenant à des vv-RKHSs. Bien que très différents à première vue, les espaces fonctionnels ainsi définis sont en réalité très similaires, ne différant que par l'ordre dans lequel les fonctions linéaires/non-linéaires sont appliquées. En plus du contrôle théorique sur les couches, considérer des fonctions à noyau permet de traiter des données structurées, en entrée comme en sortie, étendant le champ d'application des réseaux aux données complexes. Nous conclurons cette partie en montrant que ces architectures admettent la plupart du temps une paramétrisation finie-dimensionnelle, ouvrant la voie à des méthodes d'optimisation efficaces pour une large gamme de fonctions de perte.

La seconde partie de cette thèse étudie des alternatives à la moyenne empirique comme substitut de l'espérance dans le cadre de la Minimisation du Risque Empirique (Empirical Risk Minimization, ERM). En effet, l'ERM suppose de manière implicite que la moyenne empirique est un bon estimateur. Cependant, dans de nombreux cas pratiques (e.g. données à queue lourde, présence d'anomalies, biais de sélection), ce n'est pas le cas.

La Médiane-des-Moyennes (Median-of-Means, MoM) est un estimateur robuste de l'espérance construit comme suit : des moyennes empiriques sont calculées sur des sous-échantillons disjoints de l'échantillon initial, puis est choisie la médiane de ces moyennes. Nous proposons et analysons deux extensions de MoM, via des sous-échantillons aléatoires et/ou pour les U-statistiques. Par construction, les estimateurs MoM présentent des propriétés de robustesse, qui sont exploitées plus avant pour la construction de méthodes d'apprentissage robustes. Il est ainsi prouvé que la minimisation d'un estimateur MoM (aléatoire) est robuste aux anomalies, tandis que les méthodes de tournoi MoM sont étendues au cas de l'apprentissage sur les paires.

Enfin, nous proposons une méthode d'apprentissage permettant de résister au biais de sélection. Si les données d'entraînement proviennent d'échantillons biaisés, la connaissance des fonctions de biais permet une repondération non-triviale des observations, afin de construire un estimateur non biaisé du risque. Nous avons alors démontré des garanties non-asymptotiques vérifiées par les minimiseurs de ce dernier, tout en supportant empiriquement l'analyse.

Title : Deep Kernel Representation Learning for Complex Data and Reliability Issues

Keywords : Operator-Valued Kernels, Autoencoders, Robust Losses, Median-of-Means, Selection Bias

Abstract : The first part of this thesis aims at exploring deep kernel architectures for complex data. One of the known keys to the success of deep learning algorithms is the ability of neural networks to extract meaningful internal representations. However, the theoretical understanding of why these compositional architectures are so successful remains limited, and deep approaches are almost restricted to vectorial data. On the other hand, kernel methods provide with functional spaces whose geometry are well studied and understood. Their complexity can be easily controlled, by the choice of kernel or penalization. In addition, vector-valued kernel methods can be used to predict kernelized data. It then allows to make predictions in complex structured spaces, as soon as a kernel can be defined on it.

The deep kernel architecture we propose consists in replacing the basic neural mappings by functions from vector-valued Reproducing Kernel Hilbert Spaces (vv-RKHSs). Although very different at first glance, the two functional spaces are actually very similar, and differ only by the order in which linear/nonlinear functions are applied. Apart from gaining understanding and theoretical control on layers, considering kernel mappings allows for dealing with structured data, both in input and output, broadening the applicability scope of networks. We finally expose works that ensure a finite dimensional parametrization of the model, opening the door to efficient optimization procedures for a wide range of losses.

The second part of this thesis investigates alternatives to the sample mean as substitutes to the expectation in the Empirical Risk Minimization (ERM) paradigm. Indeed, ERM implicitly assumes that the empirical mean is a good estimate of the expectation. However, in many practical use cases (e.g. heavy-tailed distribution, presence of outliers, biased training data), this is not the case.

The Median-of-Means (MoM) is a robust mean estimator constructed as follows : the original dataset is split into disjoint blocks, empirical means on each block are computed, and the median of these means is finally returned. We propose two extensions of MoM, both to randomized blocks and/or U-statistics, with provable guarantees. By construction, MoM-like estimators exhibit interesting robustness properties. This is further exploited by the design of robust learning strategies. The (randomized) MoM minimizers are shown to be robust to outliers, while MoM tournament procedure are extended to the pairwise setting.

We close this thesis by proposing an ERM procedure tailored to the sample bias issue. If training data comes from several biased samples, computing blindly the empirical mean yields a biased estimate of the risk. Alternatively, from the knowledge of the biasing functions, it is possible to reweight observations so as to build an unbiased estimate of the test distribution. We have then derived non-asymptotic guarantees for the minimizers of the debiased risk estimate thus created. The soundness of the approach is also empirically endorsed.

Institut Polytechnique de Paris
91120 Palaiseau, France