NNT: 201XSACLSXXX

A thesis submitted to attain the degree of

Doctor Of Université Paris-Saclay

Prepared at

Université d'Évry-Val-d'Essonne & Télécom ParisTech

Presented by

M. Romain Brault*

Entitled

Data Are Not Reals!

About

Large-Scale Learning On Structured Input-Output Data With Operator-Valued Kernels

Doctoral School — 580
Sciences et Technologies de l'Information et de la Communication (STIC)
Laboratoire Informatique, Biologie Intégrative et Systèmes Complexes (IBISC)
Specialty Area of Computer Science

Under supervision of Professor (Prof.) Florence d'Alché-Buc†

Presented and defended at Université d'Évry-Val-d'Essonne, the 1st of March 2017 Jury

Prof.	Florence	d'Alché-Buc	Télécom-ParisTech	Director,
Dr.	Aurélien	Bellet	INRIA, Lille	Examinator
Prof.	Jean-Marc	Delosme	Université d'Évry-Val-d'Essonne	Examinator,
Prof.	Paul	Honeine	Université de Technologique de Troyes	Reporter,
Assoc. Prof.	Hachem	Kadri	Université de Aix-Marseille	Invited,
Prof.	Liva	Ralaivola	Université de Aix-Marseille	Reporter,
Assoc. Prof.	Zoltán	Szabó	École Polytechnique	Examinator,
Assoc. Prof.	Marie	Szafranski	Université d'Évry-Val-d'Essonne	Examinator.

* Email: romain.brault@ibisc.fr
† Email: florence.dalche@telecom-paristech.fr

May 20, 2017 — version 0.1

Engineer (Eng.) ENSIIE, Romain Brault: *Data Are Not Reals!* Large-Scale Learning On Structured Input-Output Data With Operator-Valued Kernels,

SUPERVISOR:
Professor (Prof.) Florence d'Alché-Buc

LOCATION:
46, Rue Barrault, 75013 — Paris, France

DIGITALY SIGNED DOCUMENT:

```
gpg --keyserver keys.gnupg.net \
    --recv-keys A276D73294A106E2544FFF9E3E5B5D0B181C5E04
gpg --verify ThesisRomainBrault.pdf.asc ThesisRomainBrault.pdf
```

ABSTRACT

In this thesis we study scalable methods to perform regression with *Operator-Valued Kernels* in order to learn *vector-valued functions*.

When data present structure, or relations between them or their different components, a common approach is to treat the data as a vector living in an appropriate vector space rather than a collection of real number. This representation allows to take into account the structure of the data by defining an appropriate space embbeding the underlying structure. Thus many problems in machine learning can be cast into learning vector-valued functions. Operator-Valued Kernels *Operator-Valued Kernels* and *vector-valued Reproducing Kernel Hilbert Spaces* provide a theoretical and practical framework to address that issue, naturally extending the well-known framework of scalar-valued kernels. In the context of scalar-valued function learning, a scalar-valued kernel can be seen a a similarity measure between two data point. A solution of the learning problem has the form of a linear combination of theses similarities with respect to weights to determine in order to have the best "fit" of the data. When dealing with Operator-Valued Kernels, the evalution of the kernel is no longer a scalar similarity, but a function acting on vectors. A solution is then a linear combination of operators with respect to vector weights.

Although Operator-Valued Kernels generalize strictly scalar-valued kernels, large scale applications are usually not affordable with these tools that require an important computational power along with a large memory capacity. In this thesis, we propose and study scalable methods to perform regression with *Operator-Valued Kernels*. To achieve this goal, we extend Random Fourier Features, an approximation technique originally introduced for scalar-valued kernels, to *Operator-Valued Kernels*. The idea is to take advantage of an approximated operator-valued feature map in order to come up with a linear model in a finite dimensional space.

First we develop a general framework devoted to the approximation of shift-invariant Mercer kernels on Locally Compact Abelian groups and study their properties along with the complexity of the algorithms based on them. Second we show theoretical guarantees by bounding the error due to the approximation, with high probability. Third, we study various applications of Operator Random Fourier Features to different tasks of Machine learning such as multi-class classification, multi-task learning, time series modeling, functional regression and anomaly detection. We also compare the proposed framework with other state of the art methods. Fourth, we conclude by drawing short-term and mid-term perspectives.

PUBLICATIONS

Some ideas and figures have appeared previously in the following publications:

[23] R. Brault and F. d'Alché–Buc. "Borne sur l'approximation de noyaux à valeurs opérateurs à l'aide de transformées de Fourier." In: *Société Francaise des Statistiques*. SFDS. 2016 (cit. on pp. 4, 79).

[24] R. Brault, M. Heinonen, and F. d'Alché-Buc. "Random Fourier Features For Operator-Valued Kernels." In: *Proceedings of The 8th Asian Conference on Machine Learning*. 2016, pp. 110–125 (cit. on pp. 4, 52, 79, 84, 155, 157).

[25] R. Brault, N. Lim, and F. d'Alché-Buc. "Scaling up Vector Autoregressive Models With Operator-Valued Random Fourier Features." In: *Proceedings of AALTD 2016: Second ECML/PKDD International Workshop on Advanced Analytics and Learning on Temporal Data*. 2016, p. 3 (cit. on pp. 5, 109, 116, 135, 139).

[64] N. Goix, R. Brault, N. Drougard, and M. Chiapino. "One Class Splitting Criteria for Random Forests." In: *arXiv preprint arXiv:1611.01971* (2016) (cit. on p. 193).

"We have at our command computers with adequate data-handling ability and with sufficient computational speed to make use of machine-learning techniques, but our knowledge of the basic principles of these techniques is still rudimentary. Lacking such knowledge, it is necessary to specify methods of problem solution in minute and exact detail,
a time-consuming and costly procedure.
Programming computers to learn from experience should eventually eliminate the need for much of this detailed programming effort."
— Arthur Samuel [127]

ACKNOWLEDGEMENTS

First of all I would like to express my sincere gratitude to my advisor Prof. Florence d'Alché-Buc for guiding me, the continuous support during this Ph. D thesis and her insightful remarks.

Besides my advisor, I would also like to thank my thesis jury Prof. Liva Ralaivola and Prof. Paul Honeine for their time.

This thesis would not have been possible without my collaborators Dr. Markus Heinonen and Dr. Maxime Sangnier, whose help have been precious to developp fully my ideas and publish. I would like to thank Dr. Nicolas Goix, Dr. Nicolas Drougard and Maël Chiapino with whom it has been a pleasure to collaborate on a paper.

I would also like to acknowledge the Université d'Évry-val-d'Essonne for hosting and funding[1] me for the first part of my thesis and Télécom PariTech for the second part.

1 PhD grant numbered 76391

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

ACML	Asian Conference in Machine Learning
AUC	Area Under the Curve
c. f.	confer
cum.	cumulative
ECML	European Conference in Machine Learning
e. g.	exempli gratia
FT	Fourier Transform
i. e.	id est
IForest	Isolation Forest
i. i. d.	independent identically distributed
KDD	The Assotiation for Computing Machinery's Spetial Interest Group on Knowledge Discovery and Data Mining
L-BFGS-B	Limited-memory Broyden–Fletcher–Goldfarb–Shanno algorithm for Bound constraind optimization
LCA	Locally Compact Abelian
LOF	Local Outlier Factor
LSAD	Least Squares Anomaly Detection
MGF	Moment Generating Function
N. A.	Not Available
NORMA	Naive Online regularized Risk Minimization Algorithm
OCRFsampling	One-Class Random Forest Sampling
OCSM	One-Class Support Vector Machine
OKVAR	Operator-Valued Kernel-Based Vector Autoregressive
OneClassRF	One-Class Random Forest
ONORMA	Operator-valued Naive Online regularized Risk Minimization Algorithm
ORFF	Operator-valued Random Fourier Feature
ORFFVAR	Operator-valued Random Fourier Feature Vector Autoregressive
OVK	Operator-Valued Kernel

p. d. f	probibility density function
POVM	Positive Operator-Valued Measure
PR	Precision Recall
RF	Random Forest
RFC	Random Forest Clustering
RFF	Random Fourier Feature
RKHS	Reproducing Kernel Hilbert Space
ROC	Receiver Operating Characteristic
R. V.	random variable
SCV	Sequential cross-validation
SCV-MSE	Sequential cross-validation Mean Squared Error
SVM	Support Vector Machine
UCI	University of California Irvine
VAR	Vector Autoregressive
VC-dimension	Vapnik-Chernonenkis dimension
VV-RKHS	Vector Valued Reproducing Kernel Hilbert Space
w. r. t.	with respect to

Part I

INTRODUCTION

1

OUTLINE AND MOTIVATIONS

1.1 MOTIVATIONS

This thesis is dedicated to the definition of a general and flexible approach to learn vector-valued functions and implement efficiently the learning algorithms, while allowing couplings between the outputs or learning function valued functions. To achieve this goal, we turn to shallow architectures, namely the product of a (nonlinear) operator-valued feature $\tilde{\Phi}(x)$ and a parameter vector θ such that $\tilde{f}(x) = \tilde{\Phi}(x)^*\theta$, and combine two appealing methodologies: Operator-Valued Kernel Regression and Random Fourier Features.

Operator-Valued Kernels [4, 34, 98] extend the classic scalar-valued kernels to vector-valued functions. As in the scalar case, Operator-Valued Kernels (OVKs) are used to build Reproducing Kernel Hilbert Spaces (RKHS) in which representer theorems apply as for ridge regression or other appropriate loss functional. In these cases, learning a model in the RKHS boils down to learning a function of the form $f(x) = \sum_{i=1}^{N} K(x, x_i)\alpha_i$ where $x_1, \dots, x_N$ are the training input data and each $\alpha_i, i = 1, \dots, N$ is a vector of the output space $\mathcal{Y}$ and each $K(x, x_i)$, an operator on vectors of $\mathcal{Y}$.

However, OVKs suffer from the same drawbacks as classic (scalar-valued) kernel machines: they scale poorly to very large datasets because they are very demanding in terms of memory and computation. We propose to approximate OVKs by extending a methodology called Random Fourier Features (RFFs) [**Alacarte**, 9, 79, 121, 126, 145, 148] so far developed to speed up scalar-valued kernel machines. The RFF approach linearizes a shift-invariant kernel model by generating explicitly an approximated feature map $\tilde{\varphi}$. RFFs has been shown to be efficient on large datasets and has been further improved by efficient matrix computations such as [79, "FastFood"] and [53, "SORF"], which are considered as the best large scale implementations of kernel methods, along with Nyström approaches proposed in Drineas and Mahoney [47]. Moreover thanks to RFFs, kernel methods have been prooved to be competitive with deep architectures [43, 93, 171].

1.2 CONTRIBUTIONS

1.3 OUTLINE

Chapter 2.
Chapter 3.
Chapter 4.
Chapter 5. In this contribution chapter we refine the bound on the OVK approximation with ORFF we first proposed in [24] and presented in [23]. It generalizes the proof technique of Rahimi and Recht [121] to OVK on LCA groups thanks to the recent results of Koltchinskii et al.

[77], Minsker [104], Sutherland and Schneider [148], and Tropp et al. [155]. As a Bernstein bound it depends on the variance of the estimator for which we derive an "upper bound".

Chapter 6.

Chapter 7. This contribution chapter deals with a generalization bound for the a regression problem with ORFF based on the results of Maurer [97] and Rahimi and Recht [122]. We also discuss the case of Ridge regression presented in **??**.

Chapter 8. This contribution chapter shows how to use the ORFF methodology to non-linear vector autoregression. It is an instanciation of the ORFF framework to $\mathcal{X} = \mathcal{Y} = (\mathbb{R}^d, +)$. We also give a generalization of a stochastic gradient descent [43] to ORFF. This is a joint work with Néhémy Lim and Florence d'Alché-Buc and has been published at a workshop of ECML. It is based on the previous work Lim et al. [85] for time series vector autoregression with operator-valued kernels [25].

Chapter 9.

2

ON LEARNING EFFICENTLY SCALAR-VALUED FUNCTIONS

"For such a model there is no need to ask the question "Is the model true?".
If "truth" is to be the "whole truth" the answer must be "No".
The only question of interest is "Is the model illuminating and useful?"".
— George Box [22]

Contents

2.1 ABOUT STATISTICAL LEARNING

We place ourself in the context of supervised learning. In this context we suppose we are given a sequence of experiments composed of events and their respective outcome. We note $x_i \in \mathcal{X}$ the i-th event and $y_i \in \mathcal{Y}$ its respective outcome and we suppose that an experiment (x_i, y_i) occurs with probability $\mathbf{Pr}(x_i, y_i)$. We call the sequence of all the experiments available "training data" and a realization of an experiment is refered to as data and is noted $\mathbf{s} = (x_i, y_i)_{i=1}^N$ where N is the number of data. To simplify the problem further we suppose that all the events follow the same probability distribution and each experiment is idependent from all the others. We say that the experiments are independent identically distributed (i. i. d.).

In machine learning we are often interested in finding the most simple function f belonging to a class of function $\mathcal{F}$ that is able to find the best the relation between an event $x \in \mathcal{X}$ and an outcome $y \in \mathcal{Y}$ from the training data. To do so we suppose we are given a loss function $L : \mathcal{X} \times \mathcal{F} \times \mathcal{Y} \to \mathbb{R}_+$ that evaluates the capacity of a function f to predict the outcome y from an event x. In statistical learning the goal is often to find the model that minimize the average loss over all the training samples. This aforemention quantity is called empirical risk.

$$\mathfrak{R}\mathrm{emp}\,(f, \mathbf{s})\,\frac{1}{N}\sum_{i=1}^{N} L(x_i, f, y_i)$$

Since the training data are i. i. d. the celebrated strong law of large numbers tells us that for any given function f in $\mathcal{F}$ converges almost surely to a quantity called true risk (or simply risk).

$$\mathfrak{R}\,(f) = \int_{\mathcal{X}\times\mathcal{Y}} L(x, f, y) d\mathbf{Pr}(x, y)$$

Intuitively the empirical risk measure the performances of a model on the training data, while the risk measure the performances of a model with respect to all the possible experiments (event the one not present in the training set).

We call learning algorithm a function S that takes a class of function $\mathcal{F}$, a loss L and some training data $\mathbf{s}$ and return a function $f_{\mathbf{s}}$ in $\mathcal{F}$. Although the convergence of the empirical risk to the true risk is guaranteed by the strong law of large number, we usually require that a learning algorithm generalizes well. That is the empirical risk must converges to the true risk uniformly for all $f_{\mathbf{s}}$ returned by a learning algorithm. In other words we want that given a class of function $\mathcal{F}$ and a loss L, the bound

$$\mathfrak{R}\,(f_{\mathbf{s}}) \le \mathfrak{R}\mathrm{emp}\,(f_{\mathbf{s}}, \mathbf{s}) + C(\delta, \mathbf{s}, L, \{f_{\mathbf{s}}\} \subseteq \mathcal{F})$$

holds with probability $1 - \delta$, for all $\delta \in (0, 1)$ and $C(\delta, \mathbf{s}, \mathcal{F}) \to 0$ when the number of training data N in $\mathbf{s}$ goes to infinity. This type of bound

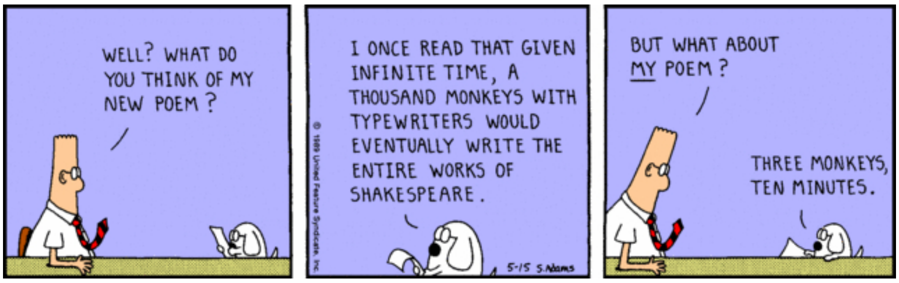

Figure 2.1: Borel's strong law of large numbers.

There is a crutial difference between the strong law of large numbers and the generalization property of a learning algorithm. The strong law of large number holds *after* a model f has been selected and fixed in $\mathcal{F}$. Thus minimizing the empirical risk doesn't yield *ipso facto* a model that minimize the true risk (which measure the adequation of the model on unseen data). This can be illustrated by an intuitive example adapted from Cornuéjols and Miclet [39, page 64] and the infinite monkey theorem.

Example 2.1 *Suppose we have a recruiter (a learning algorithm) whose task is to select the best students from a pool of candidates (the class of functions). Given ten students the recruiter make them pass a test with N questions. If the exam is well constructed and there are enough questions the recruiter should be able to retrieve the best student.*

Now suppose that ten millions monkeys $\gg N$ take the test and answer randomly to the questions. Then with high probability a monkey will score better or as well as the best student (strong law of large number). Can we say then that the recruiter has identified the best student?

Intuitively we see that when the capacity of the class of function grows (the number of students and random monkeys), the performance of the best element a posteriori (minimizing the empirical risk) is not linked to the future performance (minimizing the true risk).

On the contrary the generalization property ensure that the difference between the empirical risk and the true risk is controled because the bound does not depend on a single fixed model, but on the whole class of functions. In this case if there are too many random monkey, $C(\delta, \mathbf{s}, L, \mathcal{F})$ will blow-up, giving a poor generalization property.

A slightly stronger requirement is the consistency of learning algorithm. Given a loss function L and a class of function $\mathcal{F}$ there exists an optimal solution that minimizes the true risk.

$$f_* = \arg\min_{f \in \mathcal{F}} \mathfrak{R}\left(f\right).$$

The excess risk is defined as the difference between the empirical risk of a model returned by a learning algorithm and f_*. A learning is said to be consistent when it is possible to bound the excess risk uniformly over all the solutions returned by a learning algorithm. In other words we look for a bound such that given a class of function $\mathcal{F}$ and a loss L,

$$\begin{aligned}\mathfrak{R}\left(f_{\mathbf{s}}\right) &\leq \inf_{f\in\mathcal{F}} \mathfrak{R}\left(f\right) + C(\delta, \mathbf{s}, L, \{f_{\mathbf{s}}\}\\ &\subseteq \mathcal{F}),\end{aligned}$$

holds with probability $1-\delta$, for all $\delta \in (0,1)$ and $C(\delta, \mathbf{s}, \mathcal{F}) \to 0$ when the number of training data N in $\mathbf{s}$ goes to infinity.

To identify the best model in $\mathcal{F}$ an intuitive loss function would be the $0-1$ loss defined as

$$L(x, f, y) = \begin{cases} 1 & \text{if } yf(x) \leq 0 \\ 0 & \text{otherwise.}\end{cases}$$

This loss returns 0 if the model $f(x)$ and y have the same sign and 1 otherwise. In this simple setting Hoffgen, Simon, and Vanhorn [70] showed that finding an approximate to the empirical risk minimization with the $0-1$ loss is NP-Hard. However by "relaxing" a loss such that it becomes a convex in $f(x)$ functions yields a convex optimization problem which can then be solved in polynomial time. For instance, a convex surrogate of the $0-1$ loss is the Hinge loss

$$L(x, f, y) = \begin{cases} f(x) & \text{if } (2y-1)f(x) \leq 0 \\ 0 & \text{otherwise.}\end{cases}$$

or the logistic loss

$$L(x, f, y) = \frac{1}{\ln(2)} \ln(1 + \exp(-yf(x)))\,.$$

For regression, a common choice is the least square loss

$$L(x, f, y) = \frac{1}{2}(f(x) - y)^2$$

In the next section we discuss the choice of the class of functions $\mathcal{F}$.

2.1.1 *Introduction to kernel methods*

A fair simple choice for $\mathcal{F}$ is the set of all linear functions where we focus on defining learning algorithm picking up the "best" function in the class

$$\mathcal{F}_{lin.} = \left\{ f \,\middle|\, f(x) = \langle w, x\rangle + b,\ \ \forall w \in \mathbb{R}^d,\ \ \forall x \in \mathbb{R}^d,\ \ \forall b \in \mathbb{R} \right\}.$$

Although this class of function has been well studied and has good generalization propertis (as long as the norm of w is not too big), it has a rather

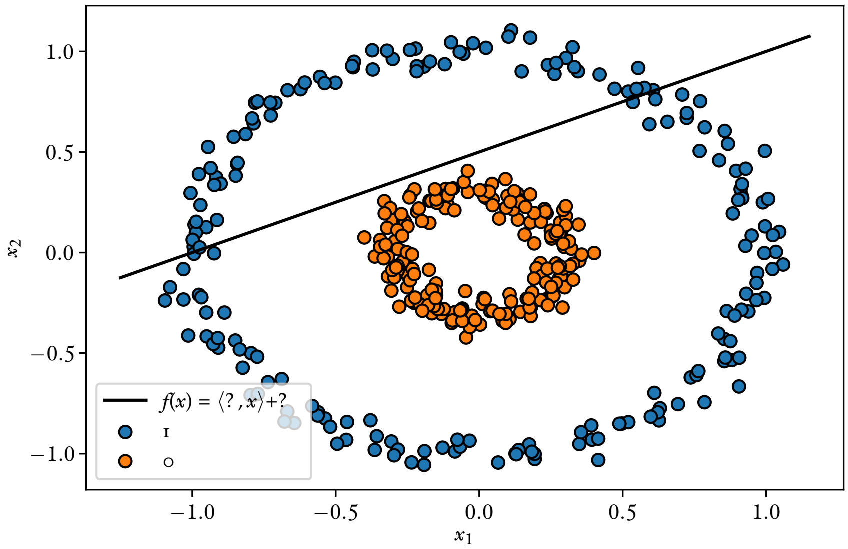

Figure 2.2: It is impossible to find a linear classifier that split perfectly two nested circles.

low capacity. For instance in $\mathbb{R}^2$ it is impossible to separate two nested circle with a line (see figure 2.2). On the other hand if one consider the class of function of all functions $\{ f \mid f \colon \mathcal{X} \to \mathbb{R} \}$, this space contains too many functions for any algorithm to be able to find a solution to the minimization of the empirical risk. The idea of kernel methods [6] is to consider a clever subset of the set of all functions, namely the set of all function belonging to a Hilbert space such that the evaluation of the function at x is bounded for all $x \in \mathcal{X}$.

$$\mathcal{F}_{\text{b. eval.}} = \{ f \mid |f(x)| \leq C(x) \|f\|_{\mathcal{H}} \leq \infty, \;\; \forall f \in \mathcal{H}, \forall x \in \mathcal{X}, \;\; C(x) \in \mathbb{R}_+ \} .$$

With this hypothesis it is legitimate to construct an evaluation operator for all $x \in \mathcal{X}$ which is bounded.

$$\mathrm{ev}_x : \begin{cases} \mathcal{F}_{\text{b. eval.}} & \to \mathbb{R} \\ f & \mapsto f(x). \end{cases}$$

From this evaluation operator, since $\mathcal{F}_{\text{b. eval.}}$ is a subspace of a Hilbert space by Riesz representation theorem, it is possible to find a function k_x of $\mathcal{F}$ such that

$$f(x) = \mathrm{ev}_x f = \langle k_x, f \rangle_{\mathcal{F}_{\text{b. eval.}}} .$$

Then we can define a function $k(x, z) := \langle k_x, k_z \rangle_{\mathcal{F}_{\text{b. eval.}}} = \overline{k_x(z)}$ and we have $f(x) = \langle k(\cdot, x), f \rangle_{\mathcal{F}_{\text{b. eval.}}}$. We call such functions k reproducing kernel because they reproduces the value of any function of the space $\mathcal{F}_{\text{b. eval.}}$ at any point $x \in \mathcal{X}$. We call the Hilbert space of function with locally bounded evaluation maps a Reproducing Kernel Hilbert Space (RKHS) and we note $\mathcal{F}_{\text{b. eval.}} = \mathcal{H}_k$. Moreover notice that the function k is by construction symmetric and positive definite.

One of the most important result of the theory of the reproducing kernels is the fact that there exists a bijection between the set of positive semi-definite functions and and the set of Reproducing Kernel Hilbert Space. In other words a symmetric positive definite function defines a unique RKHS and *vice-versa*. Thus we identify positive definite kernels with reproducing kernels and RKHSs.

Theorem 2.1 (Aronszajn [6]). *Suppose k is a symmetric, positive definite kernel on a set $\mathcal{X}$. Then there is a unique Hilbert space of functions on $\mathcal{X}$ for which k is a reproducing kernel.*

Now, back to learning and minimizing the empirical risk, a fair question is how do I find a function in an infinite dimensional set $\mathcal{H}_K$ in polynomial time? The answer comes from the regularization and interpolation theory. To limit the size of the space in which we search of the function minimizing the empirical risk we add a regularization term to the empirical risk.

$$\mathcal{J}_\lambda(f) = \frac{1}{N}\sum_{i=1}^{N} L\left(x_i, f, y_i\right) + \frac{\lambda}{2}\|f\|^2_{\mathcal{H}_K}$$

and we minimize $\mathcal{J}_\lambda$ instead of $\mathfrak{R}_{\text{emp}}$. Then the representer theorem (also called minimal norm interpolation theorem) states the following.

Theorem 2.2 (Representer theorem, Wahba [162]). *If $f_{\mathbf{s}}$ is a solution of*

$$\operatorname*{arg\,min}_{f\in\mathcal{H}_K} \mathcal{J}_\lambda(f),$$

where $\lambda > 0$ then $f_{\mathbf{s}} = \sum_{i=1}^N k(\cdot, x_i)\alpha_i$.

We note the vector $\alpha = (\alpha_i)_{i=1}^N$ and the matrix $\mathbf{K} = (k(x_i, x_k))_{i,k=1}^N$. Then we can rewrite

$$\mathcal{J}_\lambda(\alpha) = \frac{1}{N}\sum_{i=1}^{N} L(x_i, \alpha, y_i) + \lambda\langle\alpha, \mathbf{K}\alpha\rangle_2/2$$

where $f(x_i) = (K\alpha)_i$ for any $x_i \in \mathbf{s}$. If we suppose that L is convex in $f(x)$, then it is possible to derive a polynomial time (in N) algorithm minimizing $\mathcal{J}_\lambda$. For intance if we choose L to be the least square loss, then

$$\mathcal{J}_\lambda(\alpha) = \frac{1}{2N}\left\|\mathbf{K}\alpha - (y_i)_{i=1}^N\right\|_2^2 + \lambda\langle\alpha, \mathbf{K}\alpha\rangle_2/2. \tag{2.1}$$

This problem is called *Ridge regression*. By strict convexity and coercivity of $\mathcal{J}_\lambda$, and because $K + \lambda I_N$ is invertible for any $\lambda > 0$ the unique solution is $\alpha_{\mathbf{s}} = \operatorname{arg\,min}_{\alpha\in\mathbb{R}^N} \mathcal{J}_\lambda(\alpha) = (\mathbf{K}/N + \lambda I_N)^{-1}(y_i)_{i=1}^N$. This is an $O\left(N^3\right)$ algorithm.

Another way of describing positive definite kernels and RKHS consists in defining an approriate feature map.

Theorem 2.3 (Aronszajn [6]). *Given $\varphi : \mathcal{X} \to \mathcal{H}$ a function that maps a space $\mathcal{X}$ to a Hilbert space $\mathcal{H}$. Then $k(x, z) = \langle \varphi(x), \varphi(z) \rangle_{\mathcal{H}}$ is a positive definite kernel.*

Given k a positive definite kernel then there exist infinitely many functions $\varphi : \mathcal{X} \to \mathcal{H}$ that map a space $\mathcal{X}$ to some Hilbert space $\mathcal{H}$ such that $\langle \varphi(x), \varphi(z) \rangle_{\mathcal{H}} = k(x, z)$. We call the function φ a feature map.

Then any functions in $\mathcal{H}_K$ can be written $f(x) = \langle \varphi(x), \theta \rangle_{\mathcal{H}}$ In a nutshell the function φ is called feature map because it "extracts charateristic elements from a vector". Usually a feature map takes a vector in an input space with low dimension and maps it to a higher dimensional space. Put it differently, any function in $\mathcal{H}_K$ is the composition of linear functional θ^T with a non linear feature map φ. Thus is the feature map φ is fixed (which is equivalent to fixing the kernel), it is possible to "learn" with a linear class of function $\theta \in \mathcal{H}$ (see figure 2.3). If we note

$$\boldsymbol{\varphi} = \begin{pmatrix} \varphi(x_1) & \dots & \varphi(x_N) \end{pmatrix}$$

the "matrix" where each column represents the feature map evaluated at the point x_i with $1 \leq i \leq N$, the the regularized risk minimization with the least square loss reads

$$\mathcal{J}_\lambda = \frac{1}{2N} \left\| \boldsymbol{\varphi}^\mathsf{T} \theta - (y_i)_{i=1}^N \right\|_2^2 + \frac{\lambda}{2} \|\theta\|_2^2.$$

and the unique solution is $\theta_\mathbf{s} = \left(\boldsymbol{\varphi}\boldsymbol{\varphi}^\mathsf{T} / N + \lambda I_{\mathcal{H}} \right)^{-1} \boldsymbol{\varphi}(y_i)_{i=1}^N$. This is an $O\left(\dim(\mathcal{H})^2 (N + \dim \mathcal{H}) \right)$. This algorithm seems more appealing than its kernel counterpart when many data are given since one the space $\mathcal{H}$ has been fixed, the algorithm is linear in the number of training points. However many questions remains. First although it is possible to design a feature map *ex nihilo*, can we design systematically a feature map from a kernel? For some kernels (e. g. the gaussian kernel) it is well known that the Hilbert space corresponding to it has dimension $\dim(\mathcal{H}) = \infty$. Is it possible to find an approximation of the kernel such that $\dim(\mathcal{H}) < \infty$? If such a construction is possible and we know that N data are present in the training set, is it possible to have a sufficiently good approximation with $\dim(\mathcal{H}) \ll N$[1]?

2.1.2 *Random Fourier Features, Mercer Theorem, Nyström method and others*

In this subsection we answer the question whether it is possible to construct feature maps from a given kernel, that approximate a given kernel

1 When $\dim(\mathcal{H}) \geq N$ then is it is better to use the kernel algorithm than the feature algorithm. This is called the kernel trick.

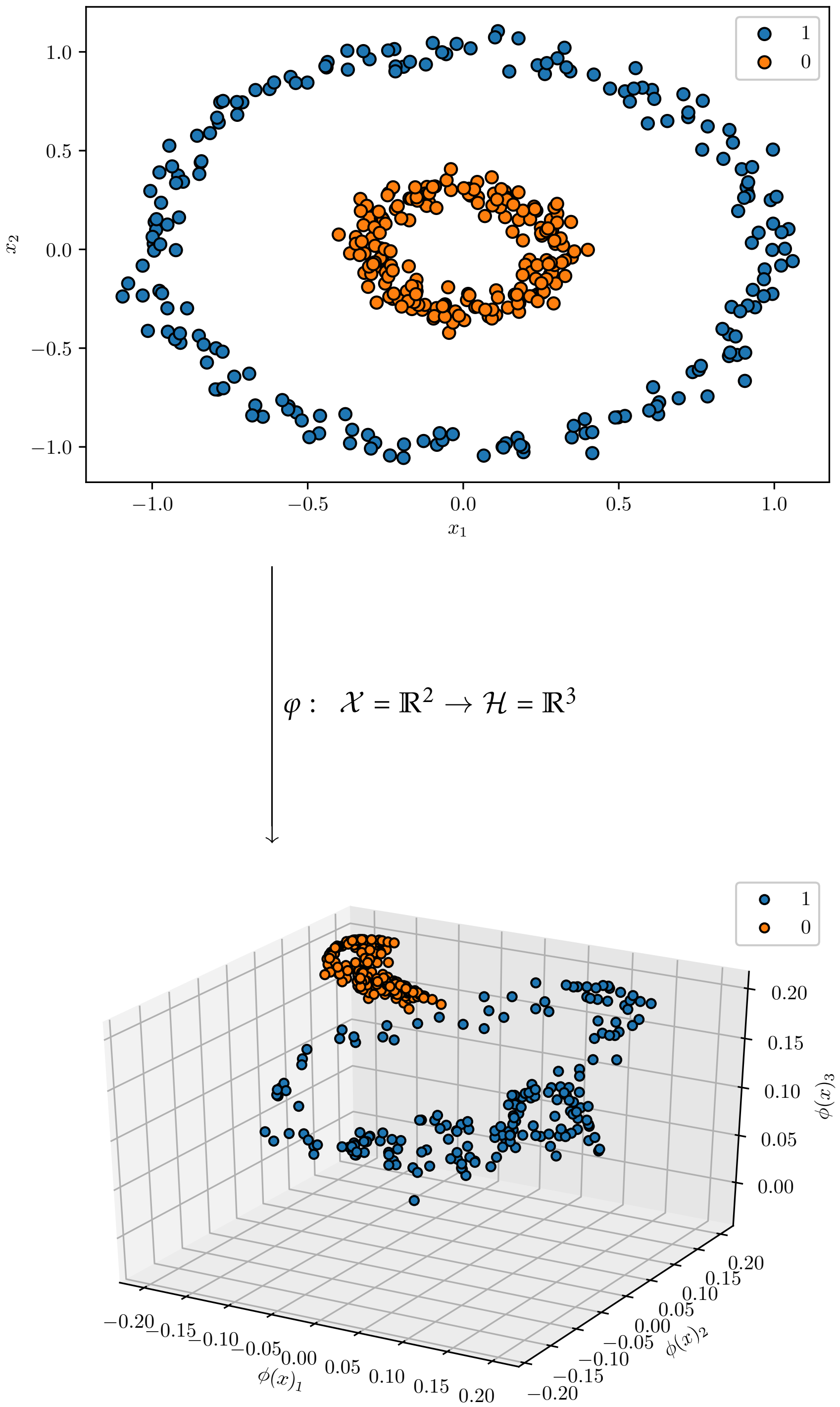

Figure 2.3: We map the two circles in $\mathbb{R}^2$ to $\mathbb{R}^3$. In $\mathbb{R}^3$ it is now possible to separate the circles with a linear functional: a plane. We used the feature map

$$\varphi(x) = 3.46 \begin{pmatrix} \cos(1.76x_1 + 2.24x_2 + 2.75) \\ \cos(0.40x_1 + 1.87x_2 + 5.6) \\ \cos(0.98x_1 - 0.98x_2 + 6.05) \end{pmatrix}.$$

k, such the the dimension of the space $\mathcal{H} < \infty$. We start with the seminal work of Rahimi and Recht [121] who show that given a continuous shift-invariant kernel ($\forall x, z, t \in \mathcal{X}, k(x+t, z+t) = k(x, z)$), it is possible to obtain a feature map called RFF that approximate the given kernel.

2.1.2.1 *Random Fourier Feature maps*

Random Fourier Feature methodology introduced by Rahimi and Recht [121] provides a way to scale up kernel methods when kernels are Mercer and *translation-invariant*. We view the input space $\mathcal{X}$ as a group endowed with the addition. Extensions to other group laws such as [83] are described in subsection 4.2.2.2 within the general framework of operator-valued kernels.

Denote $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ a positive definite kernel on $\mathbb{R}^d$. A kernel k is said to be *shift-invariant* or *translation-invariant* for the addition if for any $a \in \mathbb{R}^d$, and for all $(x, z, t) \in \left(\mathbb{R}^d\right)^3$ we have $k(x+t, z+t) = k(x, z)$. Then, we define $k_0 : \mathbb{R}^d \to \mathbb{R}$ the function such that $k(x, z) = k_0(x - z)$. k_0 is called the *signature* of kernel k. Bochner's theorem [57] is the theoretical result that leads to the Random Fourier Features.

Theorem 2.4 **(Bochner's theorem).** *Any continuous positive definite complex function is the Fourier Transform of a non-negative measure.*

It implies that any positive definite, continuous and shift-invariant kernel k, have a continuous and positive definite signature k_0, which is the Fourier Transform $\mathcal{F}$ of a non-negative measure μ. We therefore have the following corollary.

Corollary 2.1 *With the previous notations and assumptions on k,*

$$\begin{aligned} k(x, z) = k_0(x - z) &= \int_{\mathbb{R}^d} e^{-\mathrm{i}\langle \omega, x - z \rangle} d\mu(\omega) \\ &= \mathcal{F}\left[k_0\right](\omega). \end{aligned} \tag{2.2}$$

Moreover $\mu = \mathcal{F}^{-1}\left[k_0\right]$. Without loss of generality, we assume that μ is a probability measure, i. e. $\int_{\mathbb{R}^d} d\mu(\omega) = 1$ by renormalizing the kernel since

$$\int_{\mathbb{R}^d} d\mu(\omega) = \int_{\mathbb{R}^d} \exp -\mathrm{i}\langle \omega, 0 \rangle d\mu(\omega) = k_0(0).$$

and we can write equation 2.2 as an expectation over μ. For all $x, z \in \mathbb{R}^d$

$$k_0(x - z) = \mathbf{E}_\mu\left[e^{-\mathrm{i}\langle \omega, x - z \rangle}\right].$$

Eventuallt, if k is real valued we only write the real part,

$$\begin{aligned} k(x, z) &= \mathbf{E}_\mu[\cos\langle \omega, x - z \rangle] \\ &= \mathbf{E}_\mu[\cos\langle \omega, z \rangle \cos\langle \omega, x \rangle + \sin\langle \omega, z \rangle \sin\langle \omega, x \rangle]. \end{aligned}$$

Let $\bigoplus_{j=1}^{D} x_j$ denote the Dd-length column vector obtained by stacking vectors $x_j \in \mathbb{R}^d$. The feature map $\tilde{\varphi} : \mathbb{R}^d \to \mathbb{R}^{2D}$ defined as

$$\tilde{\varphi}(x) = \frac{1}{\sqrt{D}} \bigoplus_{j=1}^{D} \begin{pmatrix} \cos \langle x, \omega_j \rangle \\ \sin \langle x, \omega_j \rangle \end{pmatrix}, \quad \omega_j \sim \mathcal{F}^{-1}[k_0] \text{ i. i. d.} \tag{2.3}$$

is called a *Random Fourier Feature* (map). Each $\omega_j, j = 1, \dots, D$ is independently and identically sampled from the inverse Fourier transform μ of k_0. This Random Fourier Feature map provides the following Monte-Carlo estimator of the kernel: $\tilde{k}(x, z) = \tilde{\varphi}(x)^* \tilde{\varphi}(z)$. Using trigonometric identities, Rahimi and Recht [121] showed that the same feature map can also be written

$$\tilde{\varphi}(x) = \frac{2}{\sqrt{D}} \bigoplus_{j=1}^{D} \left(\cos \langle x, \omega_j + b_j \rangle \right), \tag{2.4}$$

where $\omega_j \sim \mathcal{F}^{-1}[k_0]$, $b_j \sim \mathcal{U}(0, 2\pi)$ i. i. d.. The feature map defined by equation 2.3 and equation 2.4 have been compared in Sutherland and Schneider [148] where they give the condition under wich equation 2.3 has lower variance than equation 2.4. For instance for the gaussian kernel, equation 2.3 has always lower variance. In practice, equation 2.4 is easier to program. In this manuscript we focus on random Fourier feature of the form equation 2.3.

The dimension D governs the precision of this approximation, whose uniform convergence towards the target kernel (as defined in equation 2.2) can be found in Rahimi and Recht [121] and in more recent papers with some refinements proposed in Sutherland and Schneider [148] and Sriperumbudur and Szabo [145]. Finally, it is important to notice that Random Fourier Feature approach *only* requires two steps before the application of a learning algorithm: (1) define the inverse Fourier transform of the given shift-invariant kernel, (2) compute the randomized feature map using the spectral distribution μ. Rahimi and Recht [121] show that fGor the Gaussian kernel $k_0(x - z) = \exp(-\gamma \|x - z\|_2^2)$, the spectral distribution μ is a Gaussian distribution. For the Laplacian kernel $k_0(x - z) = exp(-\gamma \|x - z\|_1)$, the spectral distribution is a Cauchy distribution.

We now focus on another famous way of obtaining feature maps for any scalar valued kernel called the Nyström method.

2.1.2.2 *Nyström approximation*

To overcome the bottleneck of Gram matrix computations in kernel methods, Williams and Seeger [165] have proposed to generate a low-rank matrix approximation of the Gram matrix using a subset of its columns. Since this feature map is based on a decomposition of the Gram matrix, the feature map resulting from the Nyström method is data dependent. Let $k : \mathcal{X}^2 \to \mathbb{R}$ be any scalar-valued kernel and let

$$\mathbf{s} = (x_i)_{i=1}^N$$

be the training data. We note a subsample fo the training data

$$\mathbf{s}_M = (x_i)_{i=1}^M$$

where $M \leq N$ and $\mathbf{s}_M$ is a subsequence of $\mathbf{s}$. Then construct the gram matrix $\mathbf{K}_M$ on the subsequence $\mathbf{s}_M$. Namely

$$\mathbf{K}_M = \left(k(x_i, x_j)\right)_{i,j=1}^M.$$

Then perform the singular-valued decomposition $\mathbf{K}_M = U\Lambda U^\top$. The Nyström feature map is given by

$$\tilde{\varphi}(x) = \Lambda^{-1/2}U^\top \left(\bigoplus_{i=1}^M k(x, x_i)\right).$$

Here M plays the same role than D in the RFF case: it controls the quality of the approximation. Let $\mathbf{K}$ be the full Gram matrix on the traing data $\mathbf{s}$, let

$$\mathbf{K}_b = \left(k(x_i, x_j)\right)_{i=1,j=1}^{i=N,j=M}.$$

Then it is easy to verify that $\boldsymbol{\varphi}^\top\boldsymbol{\varphi} = \mathbf{K}_b\mathbf{K}_M^\dagger\mathbf{K}_b^\top \approx \mathbf{K}$, where $\mathbf{K}_M^\dagger$ is the pseudo-inverse of $\mathbf{K}_M$ and the quantity $\mathbf{K}_b\mathbf{K}_M^\dagger\mathbf{K}_b^\top$ is a low rank approximation of the Gram matrix $\mathbf{K}$.

2.1.2.3 *Random features vs Nyström method*

The main conceptual difference between the Nyström features and the Random Fourier Feature is that the Nyström construction is data dependent, while the RFF is not. The advantage of random fourier feature lies in their fast construction. For N data in $\mathbb{R}^d$, it costs $O(NDd)$ to featurize all the data. For the Nyström features it costs $O(M^2(M + d)$. Moreover if one desire to add a new feature, the RFF methodology is as simple as drawing a new random vector $\omega \sim \mathcal{F}^{-1}[k_0]$, compute $\cos(\langle\omega, x\rangle + b)$, where $b \sim \mathcal{U}(0, 2\pi)$ and concatenate it the the existing feature. For the Nyström features one needs to recompute the singular value decomposition of the new augmented Gram matrix $\mathbf{K}_{M+1}$.

To analyse the RFF and Nyström features authors usually study the approximation error of the approximate Gram matrix and the targer kernel $\left\|\boldsymbol{\varphi}^\top\boldsymbol{\varphi} - \mathbf{K}\right\|$ (see [47, 124, 170]) or the supremum of the error between the approximated kernel and the true kernel over a compact subset $\mathcal{X}$ of the support if k: $\sup_{(x,z)\in\mathcal{C}\subseteq\mathcal{X}^2}\left|\tilde{\varphi}(x)^\top\tilde{\varphi}(z) - k(x,z)\right|$ (see [9, 121, 126, 148]). Because Bartlett and Mendelson [14] showed that for generalization error to be below $\epsilon \in \mathbb{R}_{>0}$ for kernel methods is $O(N^{-1/2})$, the number of samples M or D require to reach somme approximation error below ϵ should be not grow faster than $O(M^{-1/2})$ for the Nyström method or $O(D^{-1/2})$ for the RFF method to match kernel learning. Concerning the Nyström method, Yang et al. [170] suggest that the number of samples M is reduced to $O(M^{-1})$ to reach an error below ϵ when the gap between the eigenvalues of $\mathbf{K}$ is large enough. As a result in this specific case, one should sample $M = O(\sqrt{N})$ Nyström features to ensure good generalization. On the other hand Rahimi and Recht [122] reported that the generalization performance of RFF learning is $O(N^{-1/2} + D^{-1/2})$, which indicates that $D = O(N)$ features should be sampled to generalize well. As a result the complexity of learning with the RFF seems not to deacrease. However the bounds of Rahimi and Recht [122] are suboptimal and very recently (end of 2016) Rudi, Camoriano, and Rosasco [126] proved that in the case of ridge regression (equation 2.1), the generalization error is $O(N^{-1/2} + D^{-1})$ meaning that $D = O(\sqrt{N})$ random features are required for good generalization with RFFs. We refer the interrested reader to Yang et al. [170] for an empirical comparison between the Nyström method and the RFF method.

2.1.2.4 *Extensions of the RFF method*

The seminal idea of Rahimi and Recht [121] has open a large litterature on random features. Noway, many classes of kernels other than translation invariant are now proved to have an efficient random feature representation. Kar and Karnick [74] proposed random feature maps for dot product kernels (rotation invariant) and Hamid et al. [67] improved the rate of convergence of the approximation error for such kernels by noticing that feature maps for dot product kernels are usually low ranki and may not utilize the capacity of the projected feature space effectively. Pham and Pagh [117] proposed fast random feature maps for polynomial kernels.

To speed-up the convergence rate of the random features approximation, Yang et al. [168] proposed to sample the random variable from a quasi Monte-Carlo sequence instead of i. i. d. random variables. Le, Sarlós, and Smola [79] proposed the "Fastfood" algorithm to reduce the complexity of computing a RFF, using structured matrices and a fast Walsh-Hadarmard transform, from $O(Dd)$ to $O(D\log(d))$. More recently Felix et al. [53] proposed also an algorithm "SORF" to compute Gaussian RFF in $O(D\log(d))$ but with better convergence rates than "Fastfood" [79]. Mukuta and Harada [106] proposed data dependent features (comparable to the Nyström method) by estimating the distribution of the input data, and then finding the eigenfunction decomposition of Mercer's integral operator associated to the kernel.

Li, Ionescu, and Sminchisescu [83] generalized the original RFF of Rahimi and Recht [121]. Instead of computing feature maps for shift-invariant kenrels on the additive group $(\mathbb{R}^d, +)$, they used the generalized Fourier transform on any locally compact abelian group to derive random features on the multiplicative group $(\mathbb{R}^d, *)$. In the same spirit Yang et al. [169] noticed that an theorem equivalent to Bochner's theorem exists on the semi-group $(\mathbb{R}_+^d, +)$. From this they derived "Random Laplace" features and used them to approximate kernels adapted to learn on histograms.

In the context of large scale learning and deep learning, Lu et al. [93] showed that RFFs can achieve performances comparable to deep-learning methods by combining multiple kernel learning and composition of kernels along with a scalable parallel implementation. Dai et al. [43] and Xie, Liang, and Song [166] combined RFFs and stochastic gradient descent to defined an online learning algorithm called "Doubly stochastic gradient descent" adapted to large scale learning. Yang et al. [171] proposed an studied the idea of replacing the last fully interconnected layer of a deep convolutional neural network [80] by the "Fastfood" implementation of RFFs.

Eventually Yang et al. [172] introduced the algorithm "À la Carte", based on "Fastfood" which is able to learn the spectral distribution corresponding to a kernel rather than defining it from the kernel.

3

BACKGROUND

Contents

3.1 NOTATIONS

In this section we summarize briefly important notions used throughout this document. It is mainly based on books and lecture notes of Cotaescu [41] and Kurdila and Zabarankin [78].

3.1.1 *Algebraic structures*

[1] *Commutative.*

We note $\mathbb{K}$ any Abelian[1] field and call its elements scalars. $\mathbb{R}$ is the Abelian field of real numbers and $\mathbb{C}$ is the Abelian field of complex numbers. The unit pure imaginary number $\sqrt{-1} \in \mathbb{C}$ is denoted i and the Euler constant $\exp(1) \in \mathbb{R}$ is denoted e. $\mathbb{N}$ represents the set of natural numbers and $\mathbb{N}_n$, $n \in \mathbb{N}$ the set of natural numbers smaller or equal to n. For any space $\mathcal{S}$, $\mathcal{S}^d$, $d \in \mathbb{N}$ represents the Cartesian product space $\mathcal{S}^d = \mathcal{S} \times \cdots \times \mathcal{S}$. For any two algebraic structures $\mathcal{S}$ and $\mathcal{S}'$ we write $\mathcal{S} \cong \mathcal{S}'$ is there exist an isomorphism between these two structures. If $a + \mathrm{i}b = x \in \mathbb{C}$ then $\overline{x} = a - \mathrm{i}b \in \mathbb{C}$ denote the complex conjugate. By extension if $x \in \mathbb{R}$, $\overline{x} = x \in \mathbb{R}$.

3.1.2 *Topology and continuity*

In order to define a proper notion of continuity, we focus on topological spaces. A topological space is a pair of sets $(\mathcal{X}, \mathcal{T}_x)$ where $\mathcal{X}$ describes the points considered, and $\mathcal{T}_x$ describes the possible neighbourhoods. The standards axioms of topology suppose that $\mathcal{T}_x \subseteq \mathcal{P}(\mathcal{X})$ is a collection of subsets of $\mathcal{X}$ such that the empty set and $\mathcal{X}$ itself belongs to $\mathcal{T}_x$, any (finite or infinite) union of members of $\mathcal{T}_x$ still belongs to $\mathcal{T}_x$ and the intersection of any finite number of members of $\mathcal{T}_x$ still belongs to $\mathcal{T}_x$. The elements of $\mathcal{T}_x$ are called open sets and the collection $\mathcal{T}_x$ is a topology on $\mathcal{X}$. If $(\mathcal{X}, \mathcal{T}_x)$ and $(\mathcal{Y}, \mathcal{T}_y)$ are topological spaces, a function f is said to be continuous if for every open set $\mathcal{V} \in \mathcal{T}_y$, the inverse image $f^{-1}(\mathcal{V}) = \{ x \in \mathcal{X} \mid f(x) \in \mathcal{V} \}$ is an open subset of $\mathcal{T}_x$. Since the notion of continuity depends on open sets, it depends on the topology of the spaces $\mathcal{X}$ and $\mathcal{Y}$.

If $\mathcal{X}$ is a topological space and x is a point in $\mathcal{X}$, a neighbourhood of x is a subset $\mathcal{V}$ of $\mathcal{X}$ that includes an *open* set $\mathcal{U}$ containing x. A topological space $\mathcal{X}$ is said to be Hausdorff (T2) when all distinct points in $\mathcal{X}$ are pairwise neighborhood-separable. i. e. if there exists a neighbourhood $\mathcal{U}$ of x and a neighbourhood $\mathcal{V}$ of y such that $\mathcal{U}$ and $\mathcal{V}$ are disjoint. It implies the uniqueness of limits of sequences and existence of nets used throughout this thesis. Therefore in the whole document we always assume that a topological space $\mathcal{X}$ is Haussdorff.

[2] *Every open cover has a countable subcover.*

A topological space is said to be second countable if it has a countable base. Every second-countable space is separable and Lindelöf[2] (The reverse implications do not hold). A space is metrisable if and only if it is second countable.

A topological space is said to be separable if there exists a sequence $(x_n)_{n\in\mathbb{N}^*}$ of elements of $\mathcal{X}$ such that every nonempty open subsets of the space contains at least one element of the sequence. Separability plays an important role in numerical analysis because many theorems have only constructive proofs for separable spaces. Such constructive proofs can be turned into algorithms which is the primary goal of this work. In this document we also assume that any topological space is separable if there is no specific mention of the contrary. Moreover we recall that a Hilbert space is separable if and only if it has a countable orthonormal basis (Hence separable Hilbert spaces are second countable). Hence an operator between two separable Hilbert spaces can be written as an infinite dimensional matrix. In some cases we also introduce ***Polish spaces*** which are separable topological spaces $\mathcal{X}$ that pocess at least one d metric such that $(\mathcal{X}, d)$ is complete. Then d induces the topology $\mathcal{T}_x$ of $\mathcal{X}$. As metrisable spaces, Polish spaces are always second countable. Moreover ervery second countable locally compact Hausdorff space is a Polish space and every separable Banach space is a Polish space.

If $\mathcal{X}$ and $\mathcal{Y}$ are two topological spaces, we denote by $\mathcal{F}(\mathcal{X};\mathcal{Y})$ the topological vector space of functions $f : \mathcal{X} \to \mathcal{Y}$ and $\mathcal{C}(\mathcal{X};\mathcal{Y}) \subset \mathcal{F}(\mathcal{X};\mathcal{Y})$ the subspace of continuous functions, endowed with the product topology (topology of pointwise convergence).

3.1.3 *Measure theory*

A σ-algebra on $\mathcal{X}$ is a set $\mathcal{M} \subseteq \mathcal{P}(\mathcal{X})$ of subsets of $\mathcal{X}$, containing the empty set, which is closed under taking complements and countable unions. A pair $(\mathcal{X}, \mathcal{M})$ where $\mathcal{X}$ is a set and $\mathcal{M}$ is a σ-algebra is called a measure space. The Borel σ-algebra $\mathcal{B}(\mathcal{X})$ is a σ-algebra generated by the open sets of $\mathcal{X}$. A measure on a measurable space $(\mathcal{X}, \mathcal{B}(\mathcal{X}))$ is a map $\mu : \mathcal{B}(X) \to \overline{\mathbb{R}}_+$ which is zero on the empty set and countably additive, i. e. for any subset $(\mathcal{Z}_n)_{n\in\mathbb{N}}$ is a sequence of pairwise disjoint measurable sets,

$$\mu\left(\bigcup_{n\in\mathbb{N}} \mathcal{Z}_n\right) = \sum_{n\in\mathbb{N}} \mu(\mathcal{Z}_n).$$

We note $\mathcal{N}(m, \sigma)$ the Gaussian distribution with mean $m \in \mathbb{R}$ and variance $\sigma^2 \in \mathbb{R}$. $\mathcal{U}(a, b)$ is the uniform distribution with support (a, b) and $\mathcal{S}(m, \sigma)$ is the hyperbolic secant distribution with mean m and variance σ^2.

3.1.4 *Vector spaces, linear operators and matrices*

Given any vector space $\mathcal{H}$ over an Abelian field $\mathbb{K}$, the (continuous) dual space[1] $\mathcal{H}^*$ is defined as the set of all *continuous* linear functionals $x^* : \mathcal{H} \to \mathbb{K}$. When $\mathcal{H}$ is a vector space, there is a natural duality pairing between $\mathcal{H}^*$ and $\mathcal{H}$ defined for all $x^* \in \mathcal{H}^*$ and all $z \in \mathcal{H}$ as $(x^*, z)_{\mathcal{H}^*,\mathcal{H}} = x^*(z) = x^* z$. The duality paring $(\cdot,\cdot)_{\mathcal{H}^*,\mathcal{H}}$ is then a bilinear form.

Let $\mathcal{H}_1$ and $\mathcal{H}_2$ be two vector spaces. We call operator any linear function from $\mathcal{H}_1$ to $\mathcal{H}_2$ The transpose (or dual) of an operator $W : \mathcal{H}_1 \to \mathcal{H}_2$ is defined as $W^\mathsf{T} : \mathcal{H}_2^* \to \mathcal{H}_1^*$ such that $W^\mathsf{T} : x^* \mapsto x^*(W)$. It is characterized by the relation $(x^*, Wz)_{\mathcal{H}_2^*,\mathcal{H}_2} = \left(W^\mathsf{T} x^*, z\right)_{\mathcal{H}_1^*,\mathcal{H}_1}$ for all $x^* \in \mathcal{H}_2^*$ and all $z \in \mathcal{H}_1$. An operator is called self-dual when $W^\mathsf{T} = W$.

Let $\mathcal{H}_1$ and $\mathcal{H}_2$ be two vector space. We set $\mathcal{L}(\mathcal{H}_1;\mathcal{H}_2)$ to be the space of *bounded* (linear) operators from $\mathcal{H}_1$ to $\mathcal{H}_2$. The vector space $\mathcal{H}_1$ is called the domain, noted Dom and $\mathcal{H}_2$ the codomain. We use the shortcut notation $\mathcal{L}(\mathcal{H}) = \mathcal{L}(\mathcal{H},\mathcal{H})$. Interestingly if $\mathcal{H}_1$ and $\mathcal{H}_2$ are normed vector spaces, they can be viewed as topological vector spaces, and the notion of continuity coincides with that of boundedness. We recall the the norm of a linear operator is given by

$$\|W\|_{\mathcal{H}_1,\mathcal{H}_2} = \sup_{x \neq 0} \frac{\|Wx\|_{\mathcal{H}_2}}{\|x\|_{\mathcal{H}_1}}.$$

If $W \in \mathcal{L}(\mathcal{H}_1, \mathcal{H}_2)$

$$\operatorname{Ker} W = \{\, x \in \operatorname{Dom}(W) \mid Wx = 0 \,\}$$

denotes the kernel (nullspace), which is a vector subspace of the domain and

$$\operatorname{Im} W = \{\, y \in \mathcal{H}_2 \mid y = Wx, \;\; x \in \operatorname{Dom}(W) \,\}$$

the image (range) which is a vector subspace of the codomain $\mathcal{H}_2$.

If $\mathcal{H}$ is an Hilbert space on a field $\mathbb{K}$ we denote its scalar product by $\langle\cdot,\cdot\rangle_{\mathcal{H}}$ and its norm by $\|\cdot\|_{\mathcal{H}}$. When the base field of $\mathcal{H}$ is $\mathbb{R}$, $\langle\cdot,\cdot\rangle_{\mathcal{H}}$ is a *bilinear* form. When the base field of $\mathcal{H}$ is $\mathbb{C}$, $\langle\cdot,\cdot\rangle_{\mathcal{H}}$ is a *sesquilinear* form.

Let $\mathcal{H}$ be a Hilbert space. From Riesz's representation theorem, there is a unique isometric isomorphism $\iota_R : \mathcal{H} \to \mathcal{H}^*$ such that for any x and $y \in \mathcal{H}$, $(\iota_R(x), y)_{\mathcal{H}^*,\mathcal{H}} = \langle x, y\rangle_{\mathcal{H}}$ and $\|\iota_R(x)\|_{\mathcal{H}^*} = \|x\|_{\mathcal{H}}$. The Riesz map ι_R is self-dual, thus if $\mathcal{H}$ is a Hilbert space, $\mathcal{H}$ is reflexive. i. e. $\mathcal{H}^{**} \cong \mathcal{H}$.

1 The continuous dual space is also called topological dual space. This must be differentiate from the *algebraic* dual space, which is the space of linear functionals from the original vector-space to its base field. Hence the continuous dual space is a subset of the algebraic dual space. The continuous and the algebraic dual space only match when considering finite dimensional vector-spaces

When the base field of $\mathcal{H}$ is $\mathbb{C}$, then ι_R is an *anti-linear* form since $\langle\cdot,\cdot\rangle_{\mathcal{H}}$ is sesquilinear and $(\cdot,\cdot)_{\mathcal{H}^*,\mathcal{H}}$ is bilinear. In the same way when the base field of $\mathcal{H}$ is $\mathbb{R}$ then ι_R is *linear* since both $\langle\cdot,\cdot\rangle_{\mathcal{H}}$ and $(\cdot,\cdot)_{\mathcal{H}^*,\mathcal{H}}$ are bilinear. If $\mathcal{H}$ is a Hilbert space we make the dual space $\mathcal{H}^*$ a Hilbert space by endowing it with the inner product $\langle x^*, z^*\rangle_{\mathcal{H}^*} = \langle \iota_R^{-1}(x^*), \iota_R^{-1}(z^*)\rangle_{\mathcal{H}}$ for all $x^*, z^* \in \mathcal{H}^*$.

Let $\mathcal{H}_1$ and $\mathcal{H}_2$ be two Hilbert spaces. The adjoint of an operator $W : \mathcal{H}_1 \to \mathcal{H}_2$ is the unique mapping $W^* : \mathcal{H}_2 \to \mathcal{H}_1$ such that $\langle W^*x, z\rangle_{\mathcal{H}_1} = \langle x, Wz\rangle_{\mathcal{H}_2}$ for all $x \in \operatorname{Dom}(W^*)$, $z \in \operatorname{Dom}(W)$. Its existence is guaranteed by Riesz's representation theorem. An operator $W : \operatorname{Dom}(W) \subseteq \mathcal{H} \to \mathcal{H}$ is said to be symmetric when $W^* = W$ and self-adjoint when W is bounded, symmetric, $\operatorname{Dom}(W^*) = \operatorname{Dom}(W)$ and $\operatorname{Dom}(W)$ is dense in $\mathcal{H}$. If W is bounded, symmetric and $\operatorname{Dom}(W) = \mathcal{H}$ then W is self-adjoint. Notice that the transpose is linked to the adjoint by the relation $W^* = \iota_R^{-1} W^\top \iota_R$. When $\mathcal{H}$ is a Hilbert space, if $x \in \mathcal{H}$, we always define $x^* \in \mathcal{H}^*$ to be

$$x^* = \iota_R(x) = \langle x, \cdot\rangle_{\mathcal{H}}.$$

$$\begin{tikzcd}
\mathcal{H}_2 \arrow[r, "W^*"] \arrow[d, "\iota_R"] & \mathcal{H}_1 \arrow[d, bend left, "\iota_R"] \\
\mathcal{H}_2^* \arrow[r, "W^\top"] & \mathcal{H}_1^* \arrow[u, bend left, "\iota_R^{-1}"]
\end{tikzcd}$$

Figure 3.1: Riesz map, dual spaces and adjoints.

Let $\mathcal{H}$ be a *separable* Hilbert space and let $(e_i)_{i\in\mathbb{N}^*}$ be a basis of $\mathcal{H}$. We call $(e_i^*)_{i\in\mathbb{N}^*}$ the dual basis of $\mathcal{H}$, the basis of $\mathcal{H}^*$ such that for all $i, j \in \mathbb{N}^*$, $e_i^*(e_j) = \langle e_i, e_j\rangle_{\mathcal{H}} = \delta_{ij}$. In the whole document we consider that $\mathcal{H}^*$ is always equipped with the dual basis of $\mathcal{H}$. For a vector $x \in \mathcal{H}$ with a basis $(e_i)_{i\in\mathbb{N}^*}$ we write $x_i = e_i^*(x)$. For a linear operator $W : \mathcal{H}_1 \to \mathcal{H}_2$ where $\mathcal{H}_1$ and $\mathcal{H}_2$ are Hilbert spaces with respective basis $(e_i)_{i\in\mathbb{N}^*}$ and $(e'_j)_{j\in\mathbb{N}^*}$, we note $W_i = We_i$ and $W_{ij} = e_j^*(We_i)$. Eventually given two separable Hilbert spaces $\mathcal{H}_1$ and $\mathcal{H}_2$, an operator $W : \mathcal{H}_1 \to \mathcal{H}_2$, $(e_i)_{i\in\mathbb{N}^*}$ a basis of $\mathcal{H}_1$ and $(e'_i)_{i\in\mathbb{N}^*}$ a basis of $\mathcal{H}_2$ we have

$$(W^\top)_{ij} = e_j^{**} W^\top {e'}_i^* = e_j^{**} {e'}_i^* W = {e'}_i^* W e_j = W_{ji}.$$

We call matrix M of size $(m, n) \in \mathbb{N}^2$ on an Abelian field $\mathbb{K}$ a collection of elements $M = (m_{ij})_{1\le i\le m, 1\le j\le n}$, $m_{ij} \in \mathbb{K}$. We note $\mathcal{M}_{m,n}(\mathbb{K})$ the vector space of all matrices. If $\mathcal{H}_1$ and $\mathcal{H}_2$ are two separable Hilbert spaces on an Abelian field $\mathbb{K}$, any linear operator $L \in \mathcal{L}(\mathcal{H}_1; \mathcal{H}_2)$ can be viewed as a (potentially infinite) matrix. Let $n = \dim(\mathcal{H}_1)$, $m = \dim(\mathcal{H}_2)$

and let $B = (e_i)_{i=1}^n$ and $C = (e'_i)_{i=1}^m$ be the respective bases of $\mathcal{H}_1$ and $\mathcal{H}_2$. We note $\operatorname{mat}_{B,C} : \mathcal{L}(\mathcal{H}_1; \mathcal{H}_2) \to \mathcal{M}_{m,n}(\mathbb{K})$ such that $M = \operatorname{mat}_{B,C}(L) = (e'^*_j L e_i)_{1 \le i \le n, 1 \le j \le m} \in \mathcal{M}_{m,n}(\mathbb{K})$. Let $M_1 \in \mathcal{M}_{m,n}(\mathbb{K})$ and $M_2 \in \mathcal{M}_{n,l}(\mathbb{K})$. The product between two matrices is written $M_1 M_2 \in \mathcal{M}_{m,l}(\mathbb{K})$ and obey $(M_1 M_2)_{ij} = \sum_{k=1}^n M_{ik} M_{kj}$. Given two linear operator $L_1 \in \mathcal{L}(\mathcal{H}_1; \mathcal{H}_2)$ and $L2 \in \mathcal{L}(\mathcal{H}_2; \mathcal{H}_3)$ we have $L_1 L_2 \in \mathcal{L}(\mathcal{H}_1; \mathcal{H}_3)$ and i

$$\operatorname{mat}_{B,D}(L_1 L_2) = \operatorname{mat}_{B,C}(L_1) \operatorname{mat}_{C,D}(L_2).$$

The operator $\operatorname{mat}_{B,C}$ is a vector space isomorphism allowing us to identify $\mathcal{L}(\mathcal{H}_1; \mathcal{H}_2)$ with $\mathcal{M}_{mn}(\mathbb{K})$ where $n = \dim(\mathcal{H}_1)$ and $m = \dim(\mathcal{H}_2)$. All these notations are summarized in tables 3.1 and 3.3.

3.2 ELEMENTS OF ABSTRACT HARMONIC ANALYSIS

3.2.1 *Locally compact Abelian groups*

Definition 3.1 (Locally Compact Abelian (LCA) group.). *A group $\mathcal{X}$ endowed with a binary operation $\star$ is said to be a Locally Compact Abelian group if $\mathcal{X}$ is a topological commutative group w. r. t. $\star$ for which every point has a compact neighborhood and is Hausdorff (T2).*

Moreover given a element z of a LCA group $\mathcal{X}$, we define the set $z \star \mathcal{X} = \mathcal{X} \star z = \{ z \star x \mid \forall x \in \mathcal{X} \}$ and the set $\mathcal{X}^{-1} = \{ x^{-1} \mid \forall x \in \mathcal{X} \}$. We also note e the neutral element of $\mathcal{X}$ such that $x \star e = e \star x = e$ for all $x \in \mathcal{X}$. Throughout this thesis we focus on positive definite function. Let $\mathcal{Y}$ be a complex separable Hilbert space. A function $f : \mathcal{X} \to \mathcal{Y}$ is positive definite if for all $N \in \mathbb{N}$ and all $y \in \mathcal{Y}$,

$$\sum_{i,j=1}^{N} \left\langle y_i, f\left(x_j^{-1} \star x_i\right) y_j \right\rangle_{\mathcal{Y}} \geq 0 \tag{3.1}$$

for all sequences $(y_i)_{i \in \mathbb{N}_N^*} \in \mathcal{Y}^N$ and all sequences $(x_i)_{i \in \mathbb{N}_N^*} \in \mathcal{X}^N$. If $\mathcal{Y}$ is real we add the assumption that $f(x^{-1}) = f(x)^*$ for all $x \in \mathcal{X}$. A consequence is that a positive definite function is bounded, as shown by Falb [52], $\|f(x)\|_{\mathcal{Y},\mathcal{Y}} \le 2\|f(e)\|_{\mathcal{Y},\mathcal{Y}}$ for all $x \in \mathcal{X}$, however positive definite functions are not necessarily continuous. This motivates the introduction of functions of positive type which are nothing but continuous positive definite function.

3.2.2 *The Haar measure*

Measures on topological spaces which appear in practice often satisfy the following regularity properties.

Definition 3.2 (Radon measure). *A Radon measure $\mu =$ **Rad** on a topological measurable space $\mathcal{X}$ is a measure on $(\mathcal{X}, \mathcal{B}(\mathcal{X}))$ which satisfies the following properties.*

Table 3.1: Mathematical symbols and their signification (part 1).

Symbol	Meaning
$:=$	Equal by definition.
$\mathbb{N}$	The semi-group of natural numbers.
$\mathbb{K}$	Any non-discrete Abelian field endowed with an absolute value. Elements of $\mathbb{K}$ are called scalars.
$\mathbb{R}$	The Abelian field of real numbers.
$\mathbb{C}$	The Abelian field of complex numbers.
$\mathbb{U}$	The circle group of complex numbers with unit module.
$\mathrm{i} \in \mathbb{C}$	Unit pure imaginary number $\mathrm{i}^2 := -1$.
$\mathrm{e} \in \mathbb{R}$	Euler constant.
$e \in \mathcal{X}$	The neutral element of the group $\mathcal{X}$.
δ_{ij}	Kronecker delta function. $\delta_{ij} = 0$ if $i \neq j$, 1 otherwise.
$\langle \cdot, \cdot \rangle_2$	Euclidean inner product.
$\lVert \cdot \rVert_2$	Euclidean norm.
$\mathcal{X}$	Input space.
$\widehat{\mathcal{X}}$	The Pontryagin dual of $\mathcal{X}$ when $\mathcal{X}$ is a LCA group.
$\mathcal{Y}$	Output space (Hilbert space).
$\mathcal{H}$	Feature space (Hilbert space).
$\langle \cdot, \cdot \rangle_{\mathcal{Y}}$	The canonical inner product of the Hilbert space $\mathcal{Y}$.
$\lVert \cdot \rVert_{\mathcal{Y}}$	The canonical norm induced by the inner product of the Hilbert space $\mathcal{Y}$.
$\mathcal{F}(\mathcal{X};\mathcal{Y})$	Topological vector space of functions from $\mathcal{X}$ to $\mathcal{Y}$.
$\mathcal{C}(\mathcal{X};\mathcal{Y})$	The topological vector subspace of $\mathcal{F}$ of continuous functions from $\mathcal{X}$ to $\mathcal{Y}$.
$\mathcal{L}(\mathcal{H};\mathcal{Y})$	The set of bounded linear operator from a Hilbert space $\mathcal{H}$ to a Hilbert space $\mathcal{Y}$.
$\lVert \cdot \rVert_{\mathcal{Y},\mathcal{Y}'}$	The operator norm $\lVert \Gamma \rVert_{\mathcal{Y},\mathcal{Y}'} = \sup_{\lVert y \rVert_{\mathcal{Y}}=1} \lVert \Gamma y \rVert_{\mathcal{Y}'}$ for all $\Gamma \in \mathcal{L}(\mathcal{Y},\mathcal{Y}')$
$\mathcal{M}_{m,n}(\mathbb{K})$	The set of matrices of size (m,n).
$\mathcal{L}(\mathcal{Y})$	The set of bounded linear operator from a Hilbert space $\mathcal{Y}$ to itself.
$\mathcal{L}_+(\mathcal{Y})$	The set of non-negative bounded linear operator from a Hilbert space $\mathcal{H}$ to itself.
$\mathcal{B}(\mathcal{X})$	Borel σ-algebra on a topological space $\mathcal{X}$.
$\mu(\mathcal{X})$	A scalar positive measure of $\mathcal{X}$.
$\mathbf{Leb}(\mathcal{X})$	The Lebesgue measure of $\mathcal{X}$.
$\mathbf{Haar}(\mathcal{X})$	A Haar measure of $\mathcal{X}$.

Table 3.3: Mathematical symbols and their signification (part 2).

Symbol	Meaning
$\mathbf{Pr}_{\mu,\rho}(\mathcal{X})$	A probability measure of $\mathcal{X}$ whose Radon-Nikodym derivative (density) with respect to the measure μ is ρ.
$\mathcal{F}[\cdot]$	The Fourier Transform operator.
$\mathcal{F}^{-1}[\cdot]$	The Inverse Fourier Transform operator.
$\operatorname{ess\,sup}$	The essential supremum.
$L^p(\mathcal{X},\mu)$	The Banach space of $\lvert\cdot\rvert^p$-integrable function from $(\mathcal{X},\mathcal{B}(\mathcal{X}),\mu)$ to $\mathbb{C}$ for $p\in\mathbb{R}_+$.
$L^p(\mathcal{X},\mu;\mathcal{Y})$	The Banach space of $\lVert\cdot\rVert_{\mathcal{Y}}^p$ (Bochner)-integrable function from $(\mathcal{X},\mathcal{B}(\mathcal{X}),\mu)$ to $\mathcal{Y}$ for $p\in\mathbb{R}_+$. $L^p(\mathcal{X},\mu,\mathbb{R}):=L^p(\mathcal{X},\mu)$.
$\bigoplus_{j=1}^D x_i$	The direct sum of $D\in\mathbb{N}$ vectors x_i's in the Hilbert spaces $\mathcal{H}_i$. By definition $\langle\bigoplus_{j=1}^D x_j,\bigoplus_{j=1}^D z_j\rangle=\sum_{j=1}^D\langle x_j,z_j\rangle_{\mathcal{H}_i}$ [8].
$\lVert\cdot\rVert_p$	The $L^p(\mathcal{X},\mu,\mathcal{Y})$ norm. $\lVert f\rVert_p^p:=\int_{\mathcal{X}}\lVert f(x)\rVert_{\mathcal{Y}}^p d\mu(x)$. When $\mathcal{X}=\mathbb{N}^*$, $\mathcal{Y}\subseteq\mathbb{R}$ and μ is the counting measure and $p=2$ it coincide with the Euclidean norm $\lVert\cdot\rVert_2$ for finite dimensional vectors.
$\lVert\cdot\rVert_\infty$	The uniform norm $\lVert f\rVert_\infty=\operatorname{ess\,sup}\{\,\lVert f(x)\rVert_{\mathcal{Y}}\mid x\in\mathcal{X}\,\}=\lim_{p\to\infty}\lVert f\rVert_p$.
$^\top$	The transpose operator of a linear operator.
*	The adjoint operator of a linear operator.
$\lvert\Gamma\rvert$	The absolute value of the linear operator $\Gamma\in\mathcal{L}(\mathcal{Y})$, i. e. $\lvert\Gamma\rvert^2=\Gamma^*\Gamma$.
$\mathbf{Tr}[\Gamma]$	The trace of a linear operator $\Gamma\in\mathcal{L}(\mathcal{Y})$.
$\sigma(\Gamma)$	The spectrum of the bounded linear operator *Gamma* $\in\mathcal{L}(\mathcal{Y})$ where $\mathcal{Y}$ is a Hilbert space, i. e. $\sigma(\Gamma)=\{\,\lambda\in\mathbb{C}\mid\nexists s, s(\lambda e-\Gamma)=e\,\}$.
$\lambda_i(\Gamma)$	The i-th eigenvalue of $\Gamma\in\mathcal{L}(\mathcal{Y})$, ranked by increasing modulus, where $\mathcal{Y}$ is a *separable* Hilbert space and $i\in\mathbb{N}^*$.
$\rho(\Gamma)$	The spectral radius of the linear operator Γ i. e. $\rho(\Gamma)=\sup\{\,\lvert\lambda\rvert\mid\lambda\in\sigma(\Gamma)\,\}$.
$\lVert\cdot\rVert_{\sigma,p}$	The Schatten p-norm, $\lVert\Gamma\rVert_{\sigma,p}^p=\mathbf{Tr}\left[\lvert\Gamma\rvert^p\right]$ for $\Gamma\in\mathcal{L}(\mathcal{Y})$, where $\mathcal{Y}$ is a Hilbert space. Note that $\lVert\Gamma\rVert_{\sigma,\infty}=\rho(\Gamma)\le\lVert\Gamma\rVert_{\mathcal{Y},\mathcal{Y}}$.
$\succcurlyeq$	"Greater than" in the Loewner partial order of operators. $\Gamma_1\succcurlyeq\Gamma_2$ if $\sigma(\Gamma_1-\Gamma_2)\subseteq\mathbb{R}_+$.
$\bar{\mathbb{R}}$	The one point compacification of the real line $\mathbb{R}\cup\{\,\infty\,\}$.
$\cong$	Given two sets $\mathcal{X}$ and $\mathcal{Y}$, $\mathcal{X}\cong\mathcal{Y}$ if there exists an isomorphism $\varphi:\mathcal{X}\to\mathcal{Y}$.

1. *The measure* **Rad** *is finite on every compact set.*

$$\mathbf{Rad}(K) < \infty, \quad \textit{for any compact set } K \in \mathcal{B}(\mathcal{X}).$$

2. *The measure* **Rad** *is outer regular on any Borel sets E.*

$$\mathbf{Rad}(E) = \inf \{ \mathbf{Rad}(U) \mid E \subseteq U \}, \quad \textit{for any open set } U.$$

3. *The measure* **Rad** *is inner regular on open sets E.*

$$\mathbf{Rad}(E) = \sup \{ \mathbf{Rad}(K) \mid K \subseteq E \}, \quad \textit{for any compact set } K.$$

When dealing with topological groups it is natural to look for measures which are invariant under translation. There exist, up to a positive multiplicative constant, a unique countably additive, nontrivial measure **Haar** on *any* LCA group. For more details and constructive proofs see Alfsen [3], Conway [38], and Folland [57].

Definition 3.3 (The Haar measure). *A Haar measure* μ = **Haar** *on a LCA group* $\mathcal{X} = (G, \star)$ *is a Radon measure on* $(\mathcal{X}, \mathcal{B}(X))$ *which is non-zero on non-empty open sets and is invariant under translation. Namely*

1. *if* $\mathcal{Z} \subseteq \mathcal{X}$ *is open, then* $\mathbf{Haar}(\mathcal{X}) > 0$.
2. *For all* $\mathcal{Z} \in \mathcal{B}(\mathcal{X})$ *and* $x \in \mathcal{X}$, $\mathbf{Haar}(x \star \mathcal{Z}) = \mathbf{Haar}(\mathcal{Z})$.

Such a measure on a LCA group $\mathcal{X}$ is called a Haar measure[3] . An immediate consequence of the invariance is that for any $s \in \mathcal{X}$,

$$\int_{\mathcal{X}} f(s \star x) d\mathbf{Haar}(x) = \int_{\mathcal{X}} f(x) d\mathbf{Haar}(x).$$

It can be shown that $\mathbf{Haar}(U) > 0$ for every non-empty open subset U. In particular, if $\mathcal{X}$ is compact then $\mathbf{Haar}(\mathcal{X})$ is finite and positive, so we can uniquely specify a Haar measure on $\mathcal{X}$ by adding the normalization condition $\mathbf{Haar}(\mathcal{X}) = 1$. We call measured space the space $(\mathcal{X}, \mathcal{B}(\mathcal{X}), \mathbf{Haar})$ the space $\mathcal{X}$ endowed with its Borel σ-algebra and some measure **Haar**. If $\mathbf{Haar}(\mathcal{X}) = 1$ then the space $(\mathcal{X}, \mathcal{B}(\mathcal{X}), \mathbf{Haar})$ is called a probability space. Last but not least, on the additive group $(\mathbb{R}, +)$, the Lebesgue measure noted **Leb** is a valid Haar measure. For a concise introduction and important properties we refer the reader to the lecture of Tornier [153].

[3] *If $\mathcal{X}$ was not supposed to be Abelian, we should have defined a left Haar measure and a right Haar measure. In our case both measure are the same, so we refer to both of them as Haar measure*

3.2.3 *Even and odd functions*

Let $\mathcal{X}$ be a LCA group and $\mathbb{K}$ be a field viewed as an additive group. We say that a function $f : \mathcal{X} \to \mathbb{K}$ is even if for all $x \in \mathcal{X}, f(x) = f\left(x^{-1}\right)$ and odd if $f(x) = -f\left(x^{-1}\right)$. The definition can be extended to operator-valued functions.

Definition 3.4 (Even and odd operator-valued function on a LCA group). *Let $\mathcal{X}$ be a measured LCA group and $\mathcal{Y}$ be a Hilbert space, and $\mathcal{L}(\mathcal{Y})$ the space of bounded linear operators from $\mathcal{Y}$ to itself viewed as an additive group. A function $f : \mathcal{X} \to \mathcal{L}(\mathcal{Y})$ is (weakly) even if for all $x \in \mathcal{X}$ and all $y, y' \in \mathcal{Y}$,*

$$\langle y, f\left(x^{-1}\right) y' \rangle_{\mathcal{Y}} = \langle y, f(x) y' \rangle_{\mathcal{Y}} \tag{3.2}$$

and (weakly) odd if

$$\langle y, f\left(x^{-1}\right) y' \rangle_{\mathcal{Y}} = -\langle y, f(x) y' \rangle_{\mathcal{Y}} \tag{3.3}$$

It is easy to check that if f is odd then $\int_{\mathcal{X}} \langle y, f(x) y' \rangle_{\mathcal{Y}} d\mathbf{Haar}(x) = 0$.

Proof

$$\begin{aligned}
&\int_{\mathcal{X}} \langle y, f(x) y' \rangle_{\mathcal{Y}} d\mathbf{Haar}(x) \\
&= \int_{\mathcal{X}} \left\langle y, \left(\frac{f\left(x^{-1}\right) + f(x)}{2}\right) - \left(\frac{f\left(x^{-1}\right) - f(x)}{2}\right) y' \right\rangle_{\mathcal{Y}} d\mathbf{Haar}(x) \\
&= \frac{1}{2}\left(- \int_{\mathcal{X}} \langle y, f(x) y' \rangle_{\mathcal{Y}} d\mathbf{Haar}(x) + \int_{\mathcal{X}} \langle y, f(x) y' \rangle_{\mathcal{Y}} d\mathbf{Haar}(x) \right) \\
&= 0.
\end{aligned}$$

Besides the product of an even and an odd function is odd. Indeed for all $f, g \in \mathcal{F}(\mathcal{X}; \mathcal{L}(\mathcal{Y}))$, where f is even and g odd. Define $h(x) = \langle y, f(x) g(x) y' \rangle$. Then we have

$$\begin{aligned}
h\left(x^{-1}\right) &= \langle y, f\left(x^{-1}\right) g\left(x^{-1}\right) y' \rangle_{\mathcal{Y}} = \langle y, f(x) \left(-g(x)\right) y' \rangle_{\mathcal{Y}} \\
&= -h(x).
\end{aligned} \tag{3.4}$$

3.2.4 *Characters*

Locally Compact Abelian (LCA) groups are central to the general definition of Fourier Transform which is related to the concept of Pontryagin duality [57]. Let $(\mathcal{X}, \star)$ be a LCA group with e its neutral element and the notation, x^{-1}, for the inverse of $x \in \mathcal{X}$. A *character* is a complex continuous homomorphism $\omega : \mathcal{X} \to \mathbb{U}$ from $\mathcal{X}$ to the set of complex numbers of unit module $\mathbb{U}$. The set of all characters of $\mathcal{X}$ forms the Pontryagin *dual group* $\widehat{\mathcal{X}}$. The dual group of an LCA group is an LCA group so that we can endow $\widehat{\mathcal{X}}$ with a "dual" Haar measure noted $\widehat{\mathbf{Haar}}$. Then the dual group operation is defined by

$$(\omega_1 \star \omega_2)(x) = \omega_1(x)\omega_2(x) \in \mathbb{U}.$$

The Pontryagin duality theorem states that $\widehat{\widehat{\mathcal{X}}} \cong \mathcal{X}$. i.e. there is a canonical isomorphism between any LCA group and its double dual. To emphasize this duality the following notation is usually adopted

$$\omega(x) = (x, \omega) = (\omega, x) = x(\omega), \tag{3.5}$$

where $x \in \mathcal{X} \cong \widehat{\widehat{\mathcal{X}}}$ and $\omega \in \widehat{\mathcal{X}}$. The form $(\cdot,\cdot)$ defined in equation 3.5 is called (duality) pairing. Another important property involves the complex conjugate of the pairing which is defined as

$$\overline{(x,\omega)} = \left(x^{-1},\omega\right) = \left(x,\omega^{-1}\right). \tag{3.6}$$

Table 3.5: Classification of Fourier Transforms in terms of their domain and transform domain.

$\mathcal{X} =$	$\widehat{\mathcal{X}} \cong$	Operation	Pairing
$\mathbb{R}^d$	$\mathbb{R}^d$	$+$	$(x,\omega) = \exp\left(\mathrm{i}\langle x,\omega\rangle_2\right)$
$\mathbb{R}^d_{*,+}$	$\mathbb{R}^d$	$\cdot$	$(x,\omega) = \exp\left(\mathrm{i}\langle \log(x),\omega\rangle_2\right)$
$(-c;+\infty)^d$	$\mathbb{R}^d$	$\odot$	$(x,\omega) = \exp\left(\mathrm{i}\langle \log(x+c),\omega\rangle_2\right)$

We notice that for any pairing depending of ω, there exists a function $h_\omega : \mathcal{X} \to \mathbb{R}$ such that $(x,\omega) = \exp(\mathrm{i}h_\omega(x))$ since any pairing maps into $\mathbb{U}$. Moreover,

$$\begin{aligned}\left(x \star z^{-1},\omega\right) &= \omega(x)\omega\left(z^{-1}\right)\\ &= \exp\left(+\mathrm{i}h_\omega(x)\right)\exp\left(+\mathrm{i}h_\omega\left(z^{-1}\right)\right)\\ &= \exp\left(+\mathrm{i}h_\omega(x)\right)\exp\left(-\mathrm{i}h_\omega(z)\right).\end{aligned}$$

The following example shows how to determine the (Pontryagin) dual of a LCA group.

Example 3.1 *On the additive group $\mathcal{X} = (\mathbb{R},+)$ we have $\widehat{\mathbb{R}} \cong \mathbb{R}$ with the duality pairing $(x,\omega) = \exp(\mathrm{i}x\omega)$ for all $x \in \mathbb{R}$ and all $\omega \in \mathbb{R}$. The Haar measure on $\mathcal{X}$ is the Lebesgue measure.*

Proof *If $\omega \in \widehat{\mathbb{R}}$ then $\omega(0) = 1$ since ω is an homeomorphism from $\mathbb{R}$ to $\mathbb{U}$. Therefore there exists $a > 0$ such that $\int_0^a \omega(t)d\mathbf{Leb}(t) \neq 0$. Setting $A\omega = \int_0^a \omega(t)d\mathbf{Leb}(t)$ we have*

$$(A\omega)(x) = \int_0^a \omega(x+t)d\mathbf{Leb}(t) = \int_x^{a+x} \omega(t)d\mathbf{Leb}(t).$$

so ω is differentiable and

$$\omega'(x) = A^{-1}(\omega(a+x) - \omega(x)) = c\omega(x) \quad \textit{where} \quad c = A^{-1}(\omega(a) - 1).$$

It follow that $\omega(x) = e^{cx}$, and since $|\omega| = 1$, one can take $c = i\xi$ for some $\xi \in \mathbb{R}$. Hence we can identify ω with ξ and $\widehat{\mathbb{R}}$ with $\mathbb{R}$ since ξ uniquely determines ω, thus we identify $\omega = \xi$. □

We also especially mention the duality pairing associated to the skewed multiplicative LCA product group. This group together with the operation $\odot$ has been proposed by Li, Ionescu, and Sminchisescu [83] to handle histograms features especially useful in image recognition applications. Let $\mathcal{X} = (-c_k; +\infty)_{k=1}^d$, where $c_k \in \mathbb{R}_+$, endowed with the group operation $\odot$ defined component-wise for all $x, z \in \mathcal{X}$ as follow.

$$x \odot z := ((x_k + c_k)(z_k + c_k) - c_k)_{k=1}^d.$$

Example 3.2 (Li, Ionescu, and Sminchisescu [83]). *On the skewed multiplicative group* $\mathcal{X} = ((-c, +\infty), \odot)$ *we have* $\widehat{(-, +\infty)} \cong \mathbb{R}$, *with duality pairing* $(x, \omega) = \exp(\mathrm{i}\log(x + c)\omega)$ *for all* $x \in \mathcal{X}$ *and all* $\omega \in \widehat{\mathcal{X}}$. *The Haar measure on* $\mathcal{X}$ *is given for all* $\mathcal{Z} \in \mathcal{B}(\mathcal{X})$ *by* $\mathbf{Haar}(\mathcal{X}) = \int_{\mathcal{Z}}(z + c)^{-1} d\mathbf{Leb}(z)$.

Proof *Let* $a, b \in (-c, +\infty)$ *and* $\mu([a, b]) = \int_a^b (z + c)^{-1} d\mathbf{Leb}(z)$. *Then for all* $d \in (-c, +\infty)$

$$\begin{aligned}
\mu([d \odot a, d \odot b]) &= \int_{(d+c)(a+c)-c}^{(d+c)(b+c)-c} (z + c)^{-1} d\mathbf{Leb}(z) \\
&= \log(d + c)(b + c) - \log(d + c)(a + c) \\
&= \log(b + c) - \log(a + c) \\
&= \int_a^b (z + c)^{-1} d\mathbf{Leb}(z) = \mu([a, b]).
\end{aligned}$$

Thus μ *is translation invariant, making* $\mathbf{Haar} = \lambda\mu$ *a valid Haar measure on* $\mathcal{X}$ *for any multiplicative constant* $\lambda \in \mathbb{R}_*$. *Let* $(x, \omega) = \exp(\mathrm{i}\log(x + c)\omega)$ *for all* $x \in \mathcal{X}$ *and all* $\omega \in \widehat{\mathcal{X}}$. *We have for all* $z \in \mathcal{X}$

$$\begin{aligned}
(x \odot z, \omega) &= \exp(\mathrm{i}\log((x + c)(z + c))\omega) \\
&= \exp(\mathrm{i}\log(x + c)\omega)\exp(\mathrm{i}\log(z + c)\omega) \\
&= (x, \omega)(z, \omega)
\end{aligned}$$

Thus $\omega(x \odot z) = \omega(x)\omega(z)$, *which defines a valid pairing, therefore we can identify* $\widehat{\mathcal{X}} = \widehat{(-c, +\infty)} \cong \mathbb{R}$ *where* $\mathbb{R}$ *is the additive group endowed with the Haar measure being the Lebesgue measure.* □

It is easy to extend the pontryagin dual of groups to dual groups, as well as defining the pairing on the dual group using the following proposition [57]

Proposition 3.1 *Let* $(\mathcal{X}_i)_{i \in \mathbb{N}}$ *be a collection of LCA groups. Then*

$$\widehat{\left(\prod_{i \in \mathbb{N}} \mathcal{X}_i\right)} \cong \prod_{i \in \mathbb{N}} \widehat{\mathcal{X}}_i$$

Proof *Each* $\omega = (\omega_1, \ldots, \omega_N) \in \prod_{i=1}^N \mathcal{X}_i$ *defines a character on* $\prod_{i=1}^N \mathcal{X}_i$ *by*

$$((x_1, \ldots, x_N), (\omega_1, \ldots, \omega_N)) = (x_1, \omega_1) \cdots (x_N, \omega_N).$$

Moreover, every character ω *on* $\prod_{i=1}^N \mathcal{X}_i$ *is of this form, where* ω_i *is defined by*

$$(x_i, \omega_i) = ((e_1, \ldots, e_{i-1}, x_j, e_{i+1}, \ldots, e_N), \omega),$$

where e_i*'s denotes the neutral elements of the LCA group* $\mathcal{X}_i$. □

Hence $\widehat{\mathbb{R}^d} \cong \mathbb{R}^d$ with duality pairing

$$(x, \omega) = \exp\left(\mathrm{i}\sum_{k=1}^{d} x_k\omega_k\right),$$

hence $h_\omega(x) = \sum_{k=1}^{d} \omega_k x_k = \langle x, \omega\rangle_2$. For the skewed multiplicative group $\widehat{(-c_k; +\infty)_{k=1}^d} \cong \mathbb{R}^d$ and the duality pairing is defined by

$$(x, \omega) = \exp\left(\mathrm{i}\sum_{k=1}^{d} \log(x_k + c_k)\omega_k\right).$$

Hence $h_\omega(x) = \sum_{k=1}^{d} \log(x_k + c_k)\omega_k = \langle \log(x + c), \omega\rangle_2$. Eventually the natural Haar measure on a product group is the product measure. e. g. for $\mathcal{X} = \mathbb{R}^d$, the Haar measure on $\mathbb{R}^d$ is the d-th power of the Lebesgue measure on $\mathbb{R}$. Table 3.5 provides an explicit list of pairings for various groups based on $\mathbb{R}^d$ or its subsets. The interested reader can refer to Folland [57] for a more detailed construction of LCA, Pontryagin duality and Fourier Transforms on LCA.

3.2.5 *The Fourier Transform*

For a function with values in a separable Hilbert space $f \in L^1(\mathcal{X}, \mathbf{Haar}; \mathcal{Y})$, we denote $\mathcal{F}[f]$ its Fourier Transform (FT) which is defined by

$$\forall \omega \in \widehat{\mathcal{X}}, \ \ \mathcal{F}[f](\omega) = \int_{\mathcal{X}} \overline{(x, \omega)} f(x) d\mathbf{Haar}(x).$$

The Inverse Fourier Transform (IFT) of a function $g \in L^1(\widehat{\mathcal{X}}, \widehat{\mathbf{Haar}}; \mathcal{Y})$ is noted $\mathcal{F}^{-1}[g]$ defined by

$$\forall x \in \mathcal{X}, \ \ \mathcal{F}^{-1}[g](x) = \int_{\widehat{\mathcal{X}}} (x, \omega) g(\omega) d\widehat{\mathbf{Haar}}(\omega),$$

We also define the flip operator $\mathcal{R}$ by $(\mathcal{R}f)(x) := f\left(x^{-1}\right)$.

Theorem 3.1 (Fourier inversion). *Given a measure* **Haar** *defined on* $\mathcal{X}$, *there exists a unique suitably normalized dual measure* $\widehat{\mathbf{Haar}}$ *on* $\widehat{\mathcal{X}}$ *such that for all* $f \in L^1(\mathcal{X}, \mathbf{Haar}; \mathcal{Y})$ *and if* $\mathcal{F}[f] \in L^1(\widehat{\mathcal{X}}, \widehat{\mathbf{Haar}}; \mathcal{Y})$ *we have*

$$f(x) = \int_{\widehat{\mathcal{X}}} (x, \omega) \mathcal{F}[f](\omega) d\widehat{\mathbf{Haar}}(\omega), \quad \textit{for } \mathbf{Haar}\textit{-almost all } x \in \mathcal{X}. \tag{3.7}$$

i. e. such that $(\mathcal{R}\mathcal{F}\mathcal{F}[f])(x) = \mathcal{F}^{-1}\mathcal{F}[f](x) = f(x)$ *for* **Haar**-*almost all* $x \in \mathcal{X}$. *If f is continuous this relation holds for all* $x \in \mathcal{X}$.

Proof *The proof is based on Bochner's theorem and the Pontryagin duality theorem. We refer the reader to Folland [57, theorem 4.22 page 105 and theorem 4.33 page 111] for the full proof.* □

Thus when a Haar measure **Haar** on $\mathcal{X}$ is given, the measure on $\widehat{\mathcal{X}}$ that makes theorem 3.1 true is called the dual measure of **Haar**, noted $\widehat{\mathbf{Haar}}$. Let $c \in \mathbb{R}_*$ If $c\mathbf{Haar}$ is the measure on $\mathcal{X}$, then $c^{-1}\widehat{\mathbf{Haar}}$ is the dual measure on $\widehat{\mathcal{X}}$. Hence one must replace $\widehat{\mathbf{Haar}}$ by $c^{-1}\widehat{\mathbf{Haar}}$ in the inversion formula to compensate. Therefore, *we always take the Haar measure* $\widehat{\mathbf{Haar}}$ *on* $\widehat{\mathcal{X}}$ *to be the dual of the given Haar measure* **Haar** *on* $\mathcal{X}$. Whenever $\widehat{\mathbf{Haar}} = \mathbf{Haar}$ we say that the Haar measure is self-dual. Moreover if $\widehat{\mathbf{Haar}}$ is normalized, the Fourier Transform on

$$L^1(\mathcal{X}, \mathbf{Haar}; \mathcal{Y}) \cap L^2(\mathcal{X}, \mathbf{Haar}; \mathcal{Y})$$

extends uniquely to a unitary isomorphism from $L^2(\mathcal{X}, \mathbf{Haar}, \mathcal{Y})$ onto $L^2(\widehat{\mathcal{X}}, \widehat{\mathbf{Haar}}; \mathcal{Y})$ (Plancherel theorem). For the familiar case of a scalar-valued function f on the LCA group $(\mathbb{R}^d, +)$, we have for all $\omega \in \widehat{\mathcal{X}} = \mathbb{R}^d$

$$\begin{aligned} \mathcal{F}\left[f\right](\omega) &= \int_{\mathcal{X}} \overline{(x, \omega)} f(x) d\mathbf{Haar}(x) \\ &= \int_{\mathbb{R}^d} \exp(-\mathrm{i}\langle x, \omega\rangle_2) f(x) d\mathbf{Leb}(x), \end{aligned} \tag{3.8}$$

the Haar measure being here the Lebesgue measure. Notice that the normalization factor of $\widehat{\mathbf{Haar}}$ on $\widehat{\mathcal{X}}$ depends on the measure **Haar** on $\mathcal{X}$ *and* the duality pairing. For instance let $\mathcal{X} = (\mathbb{R}^d, +)$. In example 3.1 we showed that $\widehat{\mathcal{X}} \cong \mathbb{R}^d$ with pairing $(x, \omega) = \exp(\mathrm{i}x\omega)$, for all $x \in \mathcal{X}$ and $\omega \in \widehat{\mathcal{X}}$. If one endow $\mathcal{X}$ with the Lebesgue measure as the Haar measure, the Haar measure on the dual is defined for all $\mathcal{Z} \in \mathcal{B}(\mathbb{R}^d)$ by

$$\mathbf{Haar}(\mathcal{Z}) = \mathbf{Leb}(\mathcal{Z}), \quad \text{and} \quad \widehat{\mathbf{Haar}}(\mathcal{Z}) = \frac{1}{(2\pi)^d}\mathbf{Leb}(\mathcal{Z}),$$

in order to have $\mathcal{F}^{-1}\mathcal{F}\left[f\right] = f$. If one use the cleaner equivalent pairing $(x, \omega) = \exp(2\mathrm{i}\pi x\omega)$ rather than $(x, \omega) = \exp(\mathrm{i}x\omega)$, then

$$\widehat{\mathbf{Haar}}(\mathcal{Z}) = \mathbf{Leb}(\mathcal{Z}).$$

The pairing $(x, \omega) = \exp(2\mathrm{i}\pi x\omega)$ looks more attractive in theory since it limits the messy factor outside the integral sign and make the Haar measure self-dual. However it is of lesser use in practice since it yields additional unnecessary computation when evaluating the pairing. Hence for symmetry reason on $(\mathbb{R}^d, +)$ and reduce computations we settle with the Haar measure on $\mathbb{R}^d$ groups (additive and multiplicative) defined as

$$\widehat{\mathbf{Haar}}(\mathcal{Z}) = \mathbf{Haar}(\mathcal{Z}) = \frac{1}{\sqrt{2\pi}^d}\mathbf{Leb}(\mathcal{Z}).$$

We conclude this subsection by recalling the injectevity property of the Fourier Transform.

Corollary 3.1 (Fourier Transform injectivity). *Given μ and ν two measures, if $\mathcal{F}\left[\mu\right] = \mathcal{F}\left[\nu\right]$ then $\mu = \nu$. Moreover given two functions f and $g \in L^1(\mathcal{X}, \mathbf{Haar}; \mathcal{Y})$ if $\mathcal{F}\left[f\right] = \mathcal{F}\left[g\right]$ then $f = g$*

Proof *We refer the reader to the proof of Folland [57, corollary 4.34 page 112].* □

3.2.6 *Representations of Groups*

Representations of groups are convenient tools that allows group-theoretic problems to be replaced by linear algebra problems. Let $Gl(\mathcal{H})$ be the group of continuous isomorphism of $\mathcal{H}$, a Hilbert space, onto itself. A representation π of a LCA group $\mathcal{X}$ in $\mathcal{H}$ is an homomorphism π:

$$\pi : \mathcal{X} \to Gl(\mathcal{H})$$

for which all the maps $\mathcal{X} \to \mathcal{H}$ defined for all $v \in \mathcal{H}$ as $x \mapsto \pi(x)v$, are continuous. The space $\mathcal{H}$ in which the representation takes place is called the representation space of π. A representation π of a group $\mathcal{X}$ in a vector space $\mathcal{H}$ defines an action defined for all $x \in \mathcal{X}$ by

$$\pi_x : \begin{cases} \mathcal{H} & \to \mathcal{H} \\ v & \mapsto \pi(x)v. \end{cases}$$

If for all $x \in \mathcal{X}$, $\pi(x)$ is a unitary operator, then the group representation π is said to be unitary (i. e. $\forall x \in \mathcal{X}$, $\pi(x)$ is isometric and surjective). Thus π is unitary when for all $x \in \mathcal{X}$

$$\pi(x)^* = \pi(x)^{-1} = \pi\left(x^{-1}\right).$$

The representation π of $\mathcal{X}$ in $\mathcal{H}$ is said to be irreducible when $\mathcal{H} \neq \{ 0 \}$ and $\{ 0 \}$ and $\mathcal{H}$ are the only two stable invariant subspaces under all operators $\pi(x)$ for all $x \in \mathcal{X}$. i. e. for all $U \subset \mathcal{H}$, $U \neq \{ 0 \}$,

$$\{ \pi(x)v \mid \forall x \in \mathcal{X}, \forall v \in \mathcal{U} \} \neq \mathcal{U}.$$

To study LCA groups we also introduce the left regular representation of $\mathcal{X}$ acting on a Hilbert space of function $\mathcal{H} \subset \mathcal{F}(\mathcal{X};\mathcal{Y})$. For all $x, z \in \mathcal{X}$ and for all $f \in \mathcal{H}$,

$$(\lambda_z f)(x) := f(z^{-1} \star x).$$

The representation λ of $\mathcal{X}$ defines an action λ_x on $\mathcal{H}$ which is the translation of $f(x)$ by z^{-1}. With this definition one has for all $x, z \in \mathcal{X}$, $\lambda_x\lambda_z = \lambda_{x^{-1}\star z}$. Such representations are faithful, that is $\lambda_x = 1 \iff x = e$.

3.3 ON OPERATOR-VALUED KERNELS

We now introduce the theory of Vector Valued Reproducing Kernel Hilbert Space (VV-RKHS) that provides a flexible framework to study and learn vector-valued functions. The fundations of the general theory of scalar kernel is mostly due to Aronszajn [6] and provides a unifying point of view for the study of an important class of Hilbert spaces of real or complex valued functions. It has been first applied in the theory of partial differential equation. The theory of Operator-Valued Kernels (OVKs) which extends the scalar-valued kernel was first developped by Pedrick [115] in his Ph. D Thesis. Since then it has been successfully applied to machine learning by many authors. In particular we introduce the notion of Operator-Valued Kernels following the propositions of Carmeli, De Vito, and Toigo [33], Carmeli et al. [34], and Micchelli and Pontil [98].

3.3.1 *Definitions and properties*

In machine learning the goal is often to find a function f belonging to a space of function $\mathcal{F}(\mathcal{X};\mathcal{Y})$ that minimizes some criterion. The class of functions we consider are functions living in a Hilbert space $\mathcal{H} \subset \mathcal{F}(\mathcal{X};\mathcal{Y})$. The completeness allows to consider sequences of functions $f_n \in \mathcal{H}$ where the limit $f_n \to f$ is in $\mathcal{H}$. Moreover the existence of an inner product gives rise to a norm and also makes $\mathcal{H}$ a metric space.

Among all these functions $f \in \mathcal{H}$, we consider a subset of functions $f \in \mathcal{H}_K \subset \mathcal{H}$ such that the evaluation map $\operatorname{ev}_x : f \mapsto f(x)$ is bounded for all x. i. e. such that $\|\operatorname{ev}_x\|_K \le C_x$ for all x. For scalar valued kernel the evaluation map is a linear functional. Thus by Riesz's representation theorem there is an isomorphism between evaluating a function at a point and an inner product: $f(x) = \operatorname{ev}_x f = \langle K_x, f\rangle_K$. From this we deduce the reproducing property $K(x, z) = \langle K_x, K_z\rangle_K$ which is the cornerstone of many proofs in machine learning and functional analysis. When dealing with vector-valued functions, the evaluation map ev_x is no longer a linear functional, since it is vector-valued. However, inspired by the theory of scalar valued kernel, many authors showed that if the evaluation map of functions with values in a Hilbert space $\mathcal{Y}$ is bounded, a similar reproducing property can be obtained; namely $\langle y', K(x, z)y\rangle = \langle K_x y', K_z y\rangle_K$ for all $y, y' \in \mathcal{Y}$. This motivates the following definition of a Vector Valued Reproducing Kernel Hilbert Space (VV-RKHS).

Definition 3.5 (Vector Valued Reproducing Kernel Hilbert Space [33, 98]). *Let $\mathcal{Y}$ be a (real or complex) Hilbert space. A Vector Valued Reproducing Kernel Hilbert Space on a locally compact second countable topological space $\mathcal{X}$ is a Hilbert space $\mathcal{H}$ such that*

1. *the elements of $\mathcal{H}$ are functions from $\mathcal{X}$ to $\mathcal{Y}$ (i. e. $\mathcal{H} \subset \mathcal{F}(\mathcal{X}, \mathcal{Y})$);*
2. *for all $x \in \mathcal{X}$, there exists a positive constant C_x such that for all $f \in \mathcal{H}$*

$$\|f(x)\|_{\mathcal{Y}} \le C_x \|f\|_{\mathcal{H}}. \tag{3.9}$$

Throughout this section we show that a VV-RKHS defines a unique positive-definite function called Operator-Valued Kernel (OVK) and conversely an OVK uniquely defines a VV-RKHS. The bijection between OVKss and VV-RKHSss has been first proved by Senkene and Tempel'man [134] in 1973. In this introduction to OVKs we follow the definitions and most recent proofs of Carmeli et al. [34].

Definition 3.6 (positive-definite Operator-Valued Kernel acting on a complex Hilbert space). *Given $\mathcal{X}$ a locally compact second countable topological space and $\mathcal{Y}$ a complex Hilbert Space, a map $K : \mathcal{X} \times \mathcal{X} \to \mathcal{L}(\mathcal{Y})$ is called an positive-definite Operator-Valued Kernel kernel if*

$$\sum_{i,j=1}^{N} \langle K(x_i, x_j) y_j, y_i \rangle_{\mathcal{Y}} \geq 0, \tag{3.10}$$

for all $N \in \mathbb{N}$, for all sequences of points $(x_i)_{i=1}^N$ in $\mathcal{X}^N$ and all sequences of points $(y_i)_{i=1}^N$ in $\mathcal{Y}^N$.

If $\mathcal{Y}$ is a complex Hilbert space, a positive-definite Operator-Valued Kernel is always self-adjoint, i. e. $K(x, z) = K(z, x)^*$. This gives rise to the following definition of positive definite Operator-Valued Kernel acting on a real Hilbert space.

Definition 3.7 (positive-definite Operator-Valued Kernel acting on a real Hilbert space). *Given $\mathcal{X}$ a locally compact second countable topological space and $\mathcal{Y}$ a real Hilbert Space, a map $K : \mathcal{X} \times \mathcal{X} \to \mathcal{L}(\mathcal{Y})$ is called a positive-definite Operator-Valued Kernel kernel if*

$$K(x, z) = K(z, x)^* \tag{3.11}$$

and

$$\sum_{i,j=1}^{N} \langle K(x_i, x_j) y_j, y_i \rangle_{\mathcal{Y}} \geq 0, \tag{3.12}$$

for all $N \in \mathbb{N}$, for all sequences of points $(x_i)_{i=1}^N$ in $\mathcal{X}^N$, and all sequences of points $(y_i)_{i=1}^N$ in $\mathcal{Y}^N$.

As in the scalar case any Vector Valued Reproducing Kernel Hilbert Space defines a unique positive-definite Operator-Valued Kernel and conversely a positive-definite Operator-Valued Kerneldefines a unique Vector Valued Reproducing Kernel Hilbert Space.

Proposition 3.2 ([33]). *Given a Vector Valued Reproducing Kernel Hilbert Space there is a unique positive-definite Operator-Valued Kernel $K : \mathcal{X} \times \mathcal{X} \to \mathcal{L}(\mathcal{Y})$.*

Proof *Given $x \in \mathcal{X}$, equation 3.9 ensure that the evaluation map at x defined as*

$$ev_x : \begin{cases} \mathcal{H} \to \mathcal{Y} \\ f \mapsto f(x) \end{cases}$$

is a bounded operator and the Operator-Valued Kernel K associated to $\mathcal{H}$ is defined as

$$K : \mathcal{X} \times \mathcal{X} \to \mathcal{L}(\mathcal{Y}) \qquad K(x, z) = ev_x ev_z^*.$$

Since for all $(x_i)_{i=1}^N$ *in* $\mathcal{X}^N$ *and all* $(y_i)_{i=1}^N$ *in* $\mathcal{Y}^N$,

$$\begin{aligned}\sum_{i,j=1}^N \langle K(x_i, x_j)y_j, y_i\rangle_{\mathcal{Y}} &= \sum_{i,j=1}^N \langle ev_{x_j}^* y_j, ev_{x_i}^* y_i\rangle_{\mathcal{Y}} \\ &= \left\langle \sum_{i=1}^N ev_{x_i}^* y_i, \sum_{i=1}^N ev_{x_i}^* y_i \right\rangle_{\mathcal{Y}} \\ &= \left\| \sum_{i=1}^N ev_{x_i}^* y_i \right\|_{\mathcal{Y}} \geq 0,\end{aligned}$$

the map K is positive-definite. □

Given $x \in \mathcal{X}$, $K_x : \mathcal{Y} \to \mathcal{F}(\mathcal{X};\mathcal{Y})$ denotes the linear operator whose action on a vector y is the function $K_x y \in \mathcal{F}(\mathcal{X};\mathcal{Y})$ defined for all $z \in \mathcal{X}$ by $K_x = \mathrm{ev}_x^*$. As a consequence we have that

$$K(x, z)y = \mathrm{ev}_x \mathrm{ev}_z^* y = K_x^* K_z y = (K_z y)(x). \tag{3.13}$$

Some direct consequences follow from the definition.

1. The kernel reproduces the value of a function $f \in \mathcal{H}$ at a point $x \in \mathcal{X}$ since for all $y \in \mathcal{Y}$ and $x \in \mathcal{X}$, $\mathrm{ev}_x^* y = K_x y = K(\cdot, x)y$ so that

$$\langle f(x), y\rangle_{\mathcal{Y}} = \langle f, K(\cdot, x)y\rangle_{\mathcal{H}} = \langle K_x^* f, y\rangle_{\mathcal{Y}}. \tag{3.14}$$

2. The set $\{ K_x y \mid \forall x \in \mathcal{X}, \forall y \in \mathcal{Y} \}$ is total in $\mathcal{H}$. Namely,

$$\left(\bigcup_{x\in\mathcal{X}} \operatorname{Im} K_x\right)^{\perp} = \{ 0 \}.$$

If $f \in (\cup_{x\in\mathcal{X}} \operatorname{Im} K_x)^{\perp}$, then for all $x \in \mathcal{X}$, $f \in (\operatorname{Im} K_x)^{\perp} = \operatorname{Ker} K_x^*$, hence $f(x) = 0$ for all $x \in \mathcal{X}$ that is $f = 0$.

3. Finally for all $x \in \mathcal{X}$ and all $f \in \mathcal{H}$, $\|f(x)\|_{\mathcal{Y}} \leq \sqrt{\|K(x, x)\|_{\mathcal{Y},\mathcal{Y}}} \|f\|_{\mathcal{H}}$. This comes from the fact that $\|K_x\|_{\mathcal{Y},\mathcal{H}} = \|K_x^*\|_{\mathcal{H},\mathcal{Y}} = \sqrt{\|K(x, x)\|_{\mathcal{Y},\mathcal{Y}}}$ and the operator norm is sub-multiplicative.

Additionally given a positive-definite Operator-Valued Kernel, it defines a unique VV-RKHS.

Proposition 3.3 ([33]). *Given a positive-definite Operator-Valued Kernel* $K : \mathcal{X} \times \mathcal{X} \to \mathcal{L}(\mathcal{Y})$*, there is a unique Vector Valued Reproducing Kernel Hilbert Space* $\mathcal{H}$ *on* $\mathcal{X}$ *with reproducing kernel K.*

Proof *Let* $K_{x,y} = K(\cdot, x)y \in \mathcal{F}(\mathcal{X};\mathcal{Y})$ *and let*

$$\mathcal{H}_0 = \operatorname{span} \left\{ K_{x,y} \;\middle|\; \forall x \in \mathcal{X}, \forall y \in \mathcal{Y} \right\} \subset \mathcal{F}(\mathcal{X};\mathcal{Y}).$$

If $f = \sum_{i=1}^{N} c_i K_{x_i,y_i}$ and $g = \sum_{i=1}^{N} d_i K_{z_i,y_i'}$ are elements of $\mathcal{H}_0$ we have that

$$\sum_{j=1}^{N} \overline{d_j} \langle f(z_j), y_j' \rangle_{\mathcal{Y}} = \sum_{i,j=1}^{N} c_i \overline{d_j} \langle K(z_j, x_i) y_i, y_j' \rangle_{\mathcal{Y}} = \sum_{i=1}^{N} c_i \langle y_i, g(x_i) \rangle_{\mathcal{Y}},$$

so that the sesquilinear form on $\mathcal{H}_0 \times \mathcal{H}_0$

$$\langle f, g \rangle_{\mathcal{H}_0} = \sum_{i,j=1}^{N} c_i \overline{d_j} \langle K(z_j, x_i) y_i, y_j' \rangle_{\mathcal{Y}}$$

is well defined. Since K is a positive-definite Operator-Valued Kernel, we have that $\langle f, f \rangle_{\mathcal{H}_0} \geq 0$ for all $f \in \mathcal{H}_0$. Because the sesquilinear form is positive if $\mathcal{Y}$ is a complex Hilbert space, it is also Hermitian. If $\mathcal{Y}$ is a real Hilbert space, by assumption $K(x,z) = K(z,x)^*$, making $\langle \cdot, \cdot \rangle_{\mathcal{H}_0}$ an Hermitian sesquilinear form. Choosing $g = K_{x,y}$ in the above definition yields for all $x \in \mathcal{X}$, all $f \in \mathcal{H}_0$ and all $y \in \mathcal{Y}$

$$\langle f, K_{x,y} \rangle_{\mathcal{H}_0} = \langle f(x), y \rangle_{\mathcal{Y}}.$$

Besides if $f \in \mathcal{H}_0$ for all unitary vector $y \in \mathcal{Y}$, by the Cauchy-Schwartz inequality we have

$$\begin{aligned} \left| \langle f(x), y \rangle_{\mathcal{Y}} \right| &= \left| \langle f, K_{x,y} \rangle_{\mathcal{H}_0} \right| \leq \sqrt{\langle f, f \rangle_{\mathcal{H}_0}} \sqrt{\langle K_{x,y}, K_{x,y} \rangle_{\mathcal{Y}}} \\ &= \sqrt{\langle f, f \rangle_{\mathcal{H}_0}} \sqrt{\langle K(x,x) y, y \rangle_{\mathcal{Y}}} \leq \sqrt{\langle f, f \rangle_{\mathcal{H}_0}} \sqrt{\| K(x,x) \|_{\mathcal{Y},\mathcal{Y}}}, \end{aligned}$$

which implies that

$$\| f(x) \|_{\mathcal{Y}} \leq \| f \|_{\mathcal{H}_0} \sqrt{\| K(x,x) \|_{\mathcal{Y},\mathcal{Y}}}$$

Therefore if $\langle f, f \rangle_{\mathcal{H}_0} = 0$ then $f = 0$. Eventually we deduce that $\langle \cdot, \cdot \rangle_{\mathcal{H}_0}$ is an inner product on $\mathcal{H}_0$. Hence $\mathcal{H}_0$ is a pre-Hilbert space. To make it a (complete) Hilbert space we need to take the completion of this space. Let $\mathcal{H}$ be the completion of $\mathcal{H}_0$. Moreover let $K_x : \mathcal{Y} \to \mathcal{H}$ where $K_x y = K_{x,y}$. By construction K_x is bounded. Let $W : \mathcal{H} \to \mathcal{F}(\mathcal{X}; \mathcal{Y})$ where $(Wf)(x) = K_x^* f$. The operator W is injective. Indeed if $Wf = 0$ then for all $x \in \mathcal{X}, f \in Ker\ K_x^* = (Im\ f)^{\perp}$. Since the set $\cup_{x \in \mathcal{X}} Im\ K_x = \{ K_x y \mid \forall x \in \mathcal{X}, \forall y \in \mathcal{Y} \}$ generates by definition $\mathcal{H}_0$, we have $f = 0$. Besides, as W is injective, we have for all $f_1, f_2 \in \mathcal{H}_0$ $(Wf_1)(x) = (Wf_2)(x) \Longrightarrow f_1(x) = f_2(x)$ pointwise in $\mathcal{H}$ so that we can identify $\mathcal{H}$ with a subspace of $\mathcal{F}(\mathcal{X}; \mathcal{Y})$. Hence $K_x^* f = (Wf)(x) = f(x) = ev_x f$, showing that $\mathcal{H}$ is a Vector Valued Reproducing Kernel Hilbert Space with reproducing kernel

$$K_{\mathcal{H}}(x,z) y = (ev_z^* y)(x) = K(x,z) y.$$

The uniqueness of $\mathcal{H}$ comes from the uniqueness of the completion of $\mathcal{H}_0$ up to an isometry. □

The above theorem also holds if $\mathcal{Y}$ is a real Hilbert space provided we add the assumption that $K(x,z)$ is self-adjoint i. e. $K(x,z) = K(z,x)^*$ for all x, $z \in \mathcal{X}$. Then $K(x,z)$ still defines a valid symmetric bilinear form on $\mathcal{Y}$ when $\mathcal{Y}$ is a *real* Hilbert space.

Since an positive-definite Operator-Valued Kernel defines a unique Vector Valued Reproducing Kernel Hilbert Space (VV-RKHS) and conversely a VV-RKHS defines a unique Operator-Valued Kernel, we denotes the Hilbert space $\mathcal{H}$ endowed with the scalar product $\langle \cdot, \cdot \rangle$ respectively $\mathcal{H}_K$ and $\langle \cdot, \cdot \rangle_K$. From now we refer to positive-definite Operator-Valued Kernel or reproducing Operator-Valued Kernel as Operator-Valued Kernel whether they act on complex or real Hilbert spaces. As a consequence, given K an Operator-Valued Kernel, define $K_x = K(\cdot, x)$ we have

$$K(x,z) = K_x^* K_z \quad \forall x, z \in \mathcal{X}, \tag{3.15a}$$

$$\mathcal{H}_K = \overline{\operatorname{span}} \left\{ K_x y \mid \forall x \in \mathcal{X}, \ \forall y \in \mathcal{Y} \right\}. \tag{3.15b}$$

Another way to describe functions of $\mathcal{H}_K$ consists in using a suitable feature map.

Proposition 3.4 (Feature Operator [34]). *Let $\mathcal{H}$ be any Hilbert space and $\Phi : \mathcal{X} \to \mathcal{L}(\mathcal{Y}; \mathcal{H})$, with $\Phi_x := \Phi(x)$. Then the operator $W : \mathcal{H} \to \mathcal{F}(\mathcal{X}; \mathcal{Y})$ defined for all $g \in \mathcal{H}$, and for all $x \in \mathcal{X}$ by $(Wg)(x) = \Phi_x^* g$ is a partial isometry from $\mathcal{H}$ onto the VV-RKHS $\mathcal{H}_K$ with reproducing kernel*

$$K(x,z) = \Phi_x^* \Phi_z, \quad \forall x, z \in \mathcal{X}.$$

$W^* W$ is the orthogonal projection onto

$$(Ker\ W)^{\perp} = \overline{\operatorname{span}} \left\{ \Phi_x y \mid \forall x \in \mathcal{X}, \ \forall y \in \mathcal{Y} \right\}.$$

Then

$$\|f\|_K = \inf \left\{ \|g\|_{\mathcal{H}} \mid \forall g \in \mathcal{H}, \ Wg = f \right\}. \tag{3.16}$$

Proof *The operator $(Wg)(x) = \Phi(x)^* g$ ensures that the nullspace of W is $\mathcal{N} = Ker\ W = \cap_{x \in \mathcal{X}} Ker\ \Phi(x)^*$. Since $\Phi(x)$ is bounded, $\Phi(x)$ is a continuous operator, thus for all $x \in \mathcal{X}$, $Ker\ \Phi(x)^*$ is closed so that $\mathcal{N}$ is closed. Moreover,*

$$\mathcal{N} = Ker\ W = \bigcap_{x \in \mathcal{X}} Ker\ \Phi(x)^* = \bigcap_{x \in \mathcal{X}} (Im\ \Phi(x))^{\perp} = \left(\bigcup_{x \in \mathcal{X}} Im\ \Phi(x) \right)^{\perp}$$

So that $\mathcal{N}^{\perp} = \cup_{x \in \mathcal{X}} Im\ \Phi(x)$ and the restriction of W to $\mathcal{N}^{\perp}$ is injective.

Let $\mathcal{H}_K = Im\ W$ be a vector space. Define the unique inner product on $\mathcal{H}_K$ such that W becomes a partial isometry from $\mathcal{H}$ onto $\mathcal{H}_K$. We call this new partial isometry (again) W. We show that $\mathcal{H}_K$ is a Vector Valued Reproducing Kernel Hilbert Space. Since $W^* W$ is a projection on $\mathcal{N}^{\perp}$, given $f \in \mathcal{H}_K$, where $f = Wg$ and $g \in \mathcal{N}^{\perp}$ we have for all $x \in \mathcal{X}$

$$f(x) = (Wg)(x) = \Phi(x)^* g = \Phi(x)^* W^* W g = (W \Phi(x))^* f.$$

Since $Ker\ W$ is closed, W is bounded, and $\Phi(x)$ is bounded by definition so that the evaluation map

$$ev_x = (W \Phi(x))^*$$

is bounded so continuous. Then the reproducing kernel is given for all $x, z \in \mathcal{X}$ by

$$K(x,z) = ev_x ev_z^* = (W\Phi(x))^*(W\Phi(z)) = \Phi(x)^* W^* W\Phi(z) = \Phi(x)^*\Phi(z),$$

Since W^*W is the identity on Im $\Phi(z)$. Hence $\mathcal{H}_K$ is a VV-RKHS (see proof of proposition 3.2). □

We call Φ a *feature map*, W a *feature operator* and $\mathcal{H}$ a *feature space*. Since W is an isometry from $(\operatorname{Ker} W)^\perp$ onto $\mathcal{H}_K$, the map W allows us to identify $\mathcal{H}_K$ with the closed subspace $(\operatorname{Ker} W)^\perp$ of $\mathcal{H}$. Notice that W is a partial isometry, meaning that there can exist multiple functions $g \in \mathcal{H}$, the redescription space, such that $Wg = f$ where f is a function of the VV-RKHS $\mathcal{H}_K$. However equation 3.16 shows that there is a unique function $g \in \mathcal{H}$ such that $Wg = f$, and $\|g\|_{\mathcal{H}} = \|f\|_{\mathcal{H}_K}$. Among all functions $g \in \mathcal{H}$ such that $Wg = f$, the only one making the norm in the VV-RKHS and the redescription space is the one with minimal norm.

In this work we mainly focus on the class kernels inducing a VV-RKHS of continuous functions. Such kernels are named $\mathcal{Y}$-Mercer kernels.

Definition 3.8 ($\mathcal{Y}$-Mercer kernel [34]). *A reproducing kernel $K : \mathcal{X} \times \mathcal{X} \to \mathcal{L}(\mathcal{Y})$ is called $\mathcal{Y}$-Mercer kernel if $\mathcal{H}_K$ is a subspace of $\mathcal{C}(\mathcal{X};\mathcal{Y})$.*

The following proposition characterize $\mathcal{Y}$-Mercer kernel in terms of the properties of a kernel rather than properties of the VV-RKHS.

Proposition 3.5 (Characterization of $\mathcal{Y}$-Mercer kernel [34]). *Let K be a reproducing kernel. The kernel K is Mercer if and only if the function $x \mapsto \|K(x,x)\|$ is locally bounded and for all $x \in \mathcal{X}$ and all $y \in \mathcal{Y}$, $K_x y \in \mathcal{C}(\mathcal{X};\mathcal{Y})$.*

Proof *If $\mathcal{H}_K \subset \mathcal{C}(\mathcal{X};\mathcal{Y})$, then for all $x \in \mathcal{X}$ and all $y \in \mathcal{Y}$, $K_x y$ is an element of $\mathcal{C}(\mathcal{X};\mathcal{Y})$ (see equation 3.15b). In addition for all $f \in \mathcal{H}_K$, $\|K_x^* f\| = \|f(x)\| \leq \|f\|_\infty$. Hence there exist a constant $M \in \mathbb{R}_+$ such that for all $x \in \mathcal{X}$, $\|K_x\| \leq M$. Therefore from equation 3.15a, for all $x \in \mathcal{X}$, $\|K(x,x)\| = \|K_x^*\|^2 \leq M^2$. Conversely assume that the function $x \mapsto \|K(x,x)\|$ is locally bounded and $K_x y \in \mathcal{C}(\mathcal{X};\mathcal{Y})$. For all $f \in \mathcal{H}_K$ and all $x \in \mathcal{X}$,*

$$\|f(x)\| = \|f\|_K \sqrt{\|K(x,x)\|} \leq M\|f\|_K.$$

Thus convergence in $\mathcal{H}_K$ implies uniform convergence. Since by assumption

$$\{\, K_x t \mid \forall x \in \mathcal{X}, \forall y \in \mathcal{Y} \,\} \subset \mathcal{C}(\mathcal{X};\mathcal{Y}),$$

then the Vector Valued Reproducing Kernel Hilbert Space

$$\mathcal{H}_K = \overline{\operatorname{span}}\ \{\, K_x y \mid \forall x \in \mathcal{X}, \forall y \in \mathcal{Y} \,\} \subset \mathcal{C}$$

is also a subset of $\mathcal{C}(\mathcal{X};\mathcal{Y})$ by the uniform convergence theorem. □

The next lemma shows that when $\mathcal{X}$ and $\mathcal{Y}$ are separable and $\mathcal{H}_K$ is a space of continuous functions then $\mathcal{H}_K$ is separable. It is worth mentioning that when the Hilbert space $\mathcal{H}_K$ is separable, it admits a countable orthonormal basis.

Lemma 3.1 (Separable VV-RKHS [33]). *Let $\mathcal{H}_K$ be a Vector Valued Reproducing Kernel Hilbert Space of continuous function $f : \mathcal{X} \to \mathcal{Y}$. If $\mathcal{X}$ and $\mathcal{Y}$ are separable then $\mathcal{H}_K$ is separable.*

Proof *The separability of $\mathcal{X}$ assure that there exist a countable dense subset $\mathcal{X}_0 \subseteq \mathcal{X}$. Since $\mathcal{Y}$ is separable,*

$$\mathcal{S} = \bigcup_{x \in \mathcal{X}_0} Im\, K_x = \{ K_x y \mid \forall x \in \mathcal{X}_0, \forall y \in \mathcal{Y} \} \subset \mathcal{H}_K$$

is separable too. We show that $\mathcal{S}$ is total in $\mathcal{H}_K$ so that $\mathcal{H}_K$ is separable. If for all $x \in \mathcal{X}_0, f \in \mathcal{S}^\perp$, then $f \in Ker\, K_x^*$. Namely $f(x) = ev_x f = 0$. Since f is continuous and $\mathcal{X}_0$ is dense in $\mathcal{X}$, for all $x \in \mathcal{X}, f(x) = 0$ so $f = 0$. □

Since a $\mathcal{Y}$-Mercer kernel K defines a VV-RKHS $\mathcal{H}_K$ of continuous functions, $\mathcal{H}_K$ is separable when $\mathcal{X}$ and $\mathcal{Y}$ are separables.

Proposition 3.6 (Separable VV-RKHS for $\mathcal{Y}$-Mercer kernel [33]). *Let $K : \mathcal{X} \times \mathcal{X} \to \mathcal{Y}$ be a reproducing kernel where $\mathcal{X}$ and $\mathcal{Y}$ are separable spaces. If K is a $\mathcal{Y}$-Mercer kernel then $\mathcal{H}_K$ is separable.*

Proof *From proposition 3.5 K is a $\mathcal{Y}$-Mercer kernel if and only if $\mathcal{H}_K \subset \mathcal{C}(\mathcal{X}; \mathcal{Y})$. Applying lemma 3.1 of Carmeli, De Vito, and Toigo [33], we have that $\mathcal{H}_K$ is separable.* □

Thus since $\mathcal{H}_K$ is also a Hilbert space and is separable it is second countable (i. e. it has a countable orthonormal basis). An important consequence is that if K is a $\mathcal{Y}$-Mercer and $\mathcal{X}$ and $\mathcal{Y}$ are separable then $\mathcal{H}_K$ and any resdescription is isometrically isomorphic to ℓ^2.

3.3.2 *Shift-Invariant OVK on LCA groups*

The main subjects of interest of chapter 4 are shift-invariant Operator-Valued Kernel. When referring to a shift-invariant OVK $K : \mathcal{X} \times \mathcal{X} \to \mathcal{L}(\mathcal{Y})$ we assume that $\mathcal{X}$ is a locally compact second countable topological group with identity e.

Definition 3.9 (Shift-invariant OVK). *A reproducing Operator-Valued Kernel $K : \mathcal{X} \times \mathcal{X} \to \mathcal{L}(\mathcal{Y})$ is called shift-invariant[4] if for all $x, z, t \in \mathcal{X}$,*

$$K(x \star t, z \star t) = K(x, z). \tag{3.17}$$

[4] *Also referred to as translation-invariant OVK.*

A shift-invariant kernel can be characterized by a function of one variable K_e called the signature of K. Here e denotes the neutral element of the LCA group $\mathcal{X}$ endowed with the binary group operation $\star$.

We recall the definition of left regular representation of $\mathcal{X}$ acting on $\mathcal{H}_K$ which is useful to study LCA groups. For all $x, z \in \mathcal{X}$ and for all $f \in \mathcal{H}_K$,

$$(\lambda_z f)(x)G := f(z^{-1} \star x).$$

A group representation λ_z describe the group by making it act on a vector space (here $\mathcal{H}_K$) in a linear manner. In other word, the group representation let us see a group as a linear operator which are well studied mathematical object.

Proposition 3.7 (Kernel signature [34]). *Let $K : \mathcal{X} \times \mathcal{X} \to \mathcal{L}(\mathcal{Y})$ be a reproducing kernel. The following conditions are equivalents.*

1. *K is a positive-definite shift-invariant Operator-Valued Kernel.*
2. *There is a positive-definite function $K_e : \mathcal{X} \to \mathcal{L}(\mathcal{Y})$ such that $K(x, z) = K_e(z^{-1} \star x)$.*

If one of the above conditions is satisfied, then the representation λ leaves invariant $\mathcal{H}_K$, its action on $\mathcal{H}_K$ is unitary and

$$K(x, z) = K_e^* \lambda_{x^{-1} \star z} K_e \qquad \forall (x, z) \in \mathcal{X}^2. \tag{3.18a}$$

$$\|K(x, x)\| = \|K_e(e)\| \qquad \forall x \in \mathcal{X} \tag{3.18b}$$

Proof *Assume proposition 3.7 item 1 holds true. Given $x, z \in \mathcal{X}$, equation 3.13 and equation 3.17 yields*

$$K_e(z^{-1} \star x) = K(z^{-1} \star x, e) = K(x, z).$$

Since K is a reproducing kernel, K_e is of completely positive type, so that proposition 3.7 item 2 holds true. Besides if proposition 3.7 item 2 holds true obviously the definition of a reproducing kernel (definition 3.6) is fulfilled so that proposition 3.7 item 1 holds true.

Suppose that K is a shift-invariant reproducing kernel. Given $t \in \mathcal{X}$ and $y \in \mathcal{Y}$, for all $x, z \in \mathcal{X}$,

$$(\lambda_x K_t y)(z) = (K_t y)(x^{-1} \star z) = K(x^{-1} \star z, t) = K(z, x \star t) = (K_{x \star t} y) z,$$

that is $\lambda_x K_t = K_{x \star t}$. Besides for all $y, y' \in \mathcal{Y}$ and all $x, z, t, t' \in \mathcal{X}$,

$$\begin{aligned} \langle \lambda_x K_t y, \lambda_x K_{t'} y' \rangle_K &= \langle K_{x \star t} y, K_{x \star t'} y' \rangle_K = \langle K(x \star t', x \star t) y, y' \rangle \\ &= \langle K(t', t) y, y' \rangle = \langle K_t y, K_{t'} y' \rangle_K \end{aligned}$$

This means that λ leaves the set $\{ K_x y \mid \forall x \in \mathcal{X}, \forall y \in \mathcal{Y} \}$ invariant. Since

$$\{ K_x y \mid \forall x \in \mathcal{X}, \forall y \in \mathcal{Y} \}$$

is total in $\mathcal{H}_K$ (see equation 3.15b), λ is surjective and because it also leaves the inner product invariant, the first two claims follow. □

The notation K_e for the function of completely positive type associated with the reproducing kernel K is consistent with the definition given by equation 3.13 since for all $x \in \mathcal{X}$ and all $y \in \mathcal{Y}$

$$(K_e y)(x) = K_e(x) y.$$

Moreover notice that shift-invariant $\mathcal{Y}$-Mercer kernels are directly linked to functions of positive type (see equation 3.1), since shift-invariant $\mathcal{Y}$-Mercer kernels are nothing but functions whose signature is of positive type (continuous positive definite functions).

3.3.3 *Examples of Operator-Valued Kernels*

In this subsection we list some operator-valued kernel that have been used successfully in the litterature. We do not recall the proof that the following kernels are well defined are refer the interrested reader to the respective authors original work.

Operator-valued kernels have been first introduced in Machine Learning to solve multi-task regression problems. Multi-task regression is encountered in many fields such as structured classification when classes belong to a hierarchy for instance. Instead of solving independently p single output regression task, one would like to take advantage of the relationships between output variables when learning and making a decision.

Definition 3.10 (Decomposable kernel [99]). *Let Γ be a non-negative operator of $\mathcal{L}_+(\mathcal{Y})$. K is said to be a decomposable kernel[5] if for all $(x,z) \in \mathcal{X}^2$,*

$$K(x,z) := k(x,z)\Gamma,$$

where k is a scalar kernel.

[5] Some authors also refer to as separable kernels.

When $\mathcal{Y} = \mathbb{R}^p$, the matrix Γ is interpreted as encoding the relationships between the outputs coordinates. If a graph coding for the proximity between tasks is known, then it is shown in Álvarez, Rosasco, and Lawrence [4], Baldassarre et al. [11], and Evgeniou, Micchelli, and Pontil [51] that Γ can be chosen equal to the pseudo inverse $L^\dagger$ of the graph Laplacian such that the norm in $\mathcal{H}_K$ is a graph-regularizing penalty for the outputs (tasks). When no prior knowledge is available, Γ can be set to the empirical covariance of the output training data or learned with one of the algorithms proposed in the literature [45, 85, 139]. Another interesting property of the decomposable kernel is its universality (a kernel which may approximate an arbitrary continuous target function uniformly on any compact subset of the input space). A reproducing kernel K is said *universal* if the associated VV-RKHS $\mathcal{H}_K$ is dense in the space of continuous functions $\mathcal{C}(\mathcal{X},\mathcal{Y})$. The conditions for a kernel to be universal have been discussed in Caponnetto et al. [32] and Carmeli et al. [34]. In particular they show that a decomposable kernel is universal provided that the scalar kernel k is universal and the operator Γ is injective. Given $(e_k)_{k=1}^p$ a basis of $\mathcal{Y}$, we recall here how the matrix Γ act as a regularizer between the components of the outputs $f_k = \langle f(\cdot), e_k\rangle_{\mathcal{Y}}$ of a function $f \in \mathcal{H}_K$.

Proposition 3.8 (Kernels and Regularizers [4]). *Let $K(x,z) := k(x,z)\Gamma$ for all $x, z \in \mathcal{X}$ be a decomposable kernel where Γ is a matrix of size $p \times p$. Then for all $f \in \mathcal{H}_K$,*

$$\|f\|_K = \sum_{i,j=1}^{p} \left(\Gamma^\dagger\right)_{ij} \langle f_i, f_j\rangle_k \tag{3.19}$$

where $f_i = \langle f, e_i\rangle$ (resp $f_j = \langle f, e_j\rangle$), denotes the i-th (resp j-th) component of f.

We prove a generalized version of proposition 3.8 to any Operator-Valued Kernel in subsection 4.3.4.

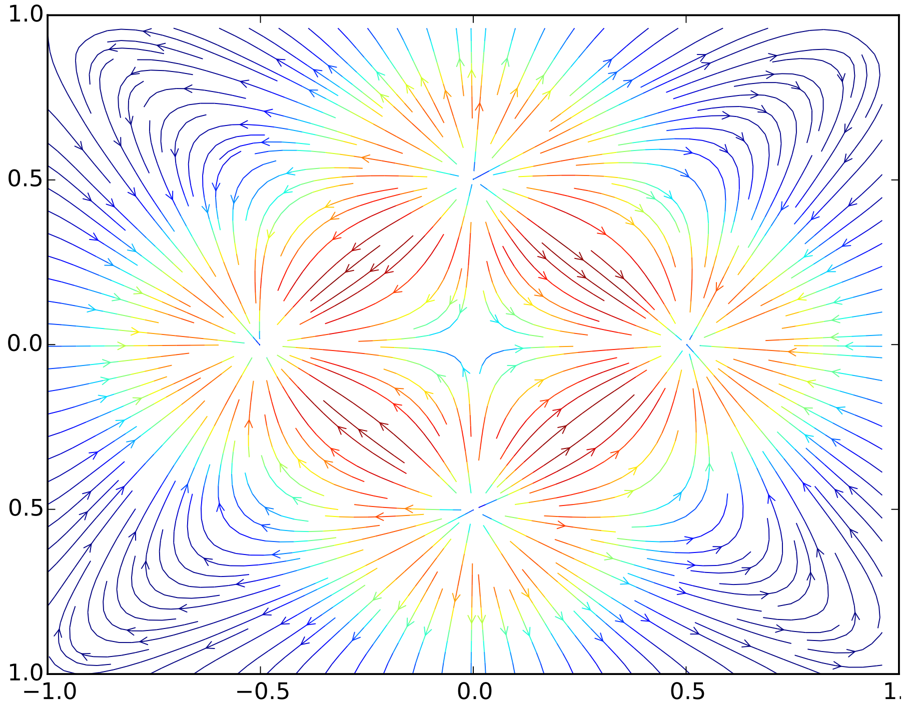

Figure 3.2: Synthetic $2D$ curl-free field .

Curl-free and divergence-free kernels provide an interesting application of operator-valued kernels [12, 94, 99] to *vector field* learning, for which input and output spaces have the same dimensions ($d = p$). Applications cover shape deformation analysis [99] and magnetic fields approximations [163]. These kernels discussed in [59] allow encoding input-dependent similarities between vector-fields. An illustration of a synthetic $2D$ curl-free and divergence free fields are given respectively in figure 3.2 and figure 3.3. To obain the curl-free field we took the gradient of a mixture of five two dimensional gaussians (since the gradient of a potential is always curl-free). We generated the divergence-free field by taking the orthogonal of the curl-free field.

Definition 3.11 (Curl-free and Div-free kernel [94]). *Assume* $\mathcal{X} = (\mathbb{R}^d, +)$ *and* $\mathcal{Y} = \mathbb{R}^p$ *with* $d = p$*. The divergence-free kernel is defined as*

$$K^{div}(x, z) = K_0^{div}(\delta) = (\nabla\nabla^\top - \Delta I)k_0(\delta)$$

and the curl-free kernel as

$$K^{curl}(x, z) = K_0^{curl}(\delta) = -\nabla\nabla^\top k_0(\delta),$$

where ∇ *is the gradient operator*[6] , $\nabla\nabla^\top$ *is the Hessian operator and* Δ *is the Laplacian operator.*

[6] See *subsection 6.2.1 for a formal definition of the operator* ∇.

Although taken separately these kernels are not universal, a convex combination of the curl-free and divergence-free kernels allows to learn any vector field that satisfies the Helmholtz decomposition theorem [12, 94].

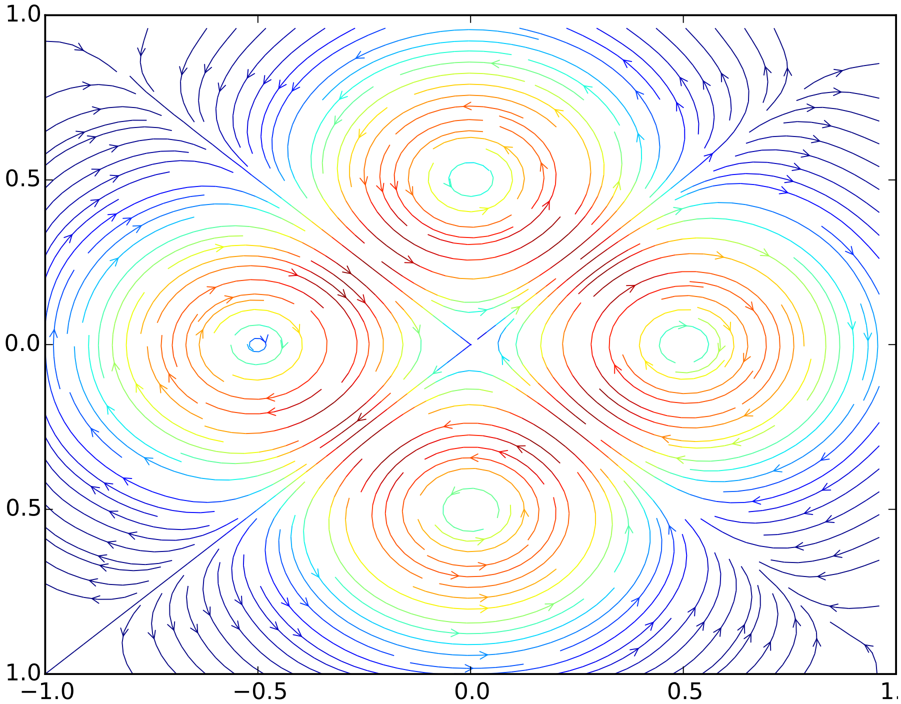

Figure 3.3: Synthetic $2D$ divergence-free field .

The next class of kernel we present are transformable kernels, whose action on each coordinate of an outputs vector is determined by a "views" of an input data.

Definition 3.12 (Transformable kernel[32]). *Let $k : \mathcal{X}' \times \mathcal{X}' \to \mathbb{R}$ be a scalar-valued kernels and $\psi_1, \dots, \psi_p$ be a collection functions from $\mathcal{X} \to \mathcal{X}'$. Then the transformable kernel is defined for all $(i, j) \in (\mathbb{N}_p^*)^2$ as*

$$K(x,z)_{ij} = \langle e_i, K(x,z)e_j \rangle_{\mathcal{Y}} = k(\psi_i(x), \psi_j(z)),$$

for all $x, z \in \mathcal{X}$.

Transformable kernels have successfully used for network inference from time series by means of autoregressive models (Lim et al. [87, 88]), and by Vazquez and Walter [159] for cokriging the multi-output version of kriging[7] , which takes into account the correlations between the outputs.

[7] *Gaussian process regression.*

We also introduce an example of Operator-Valued Kernel acting on a function space which found applications in Kadri et al. [73].

Definition 3.13 (Hilbert Schmidt Integral kernel [73]). *Let $k_{\mathcal{X}} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a scalar valued kernel acting on the inputs and $k_{\mathcal{T}} : \mathcal{T} \times \mathcal{T} \to \mathbb{R}$ be a scalar valued kernel acting on the outputs. Define the integral operator $L_{\mathcal{T}} g = \int_{\mathcal{Y}} k_{\mathcal{T}}(\cdot, t) g(t) d\mu(t)$. Then the Hilbert Schmidt Integral kernel is defined as*

$$K : \begin{cases} \mathcal{X} \times \mathcal{X} & \to \mathcal{L}(\mathcal{Y}) \\ (x,z) & \mapsto k_{\mathcal{X}}(x,z) L_{\mathcal{T}}. \end{cases}$$

This kernel is useful to learn functions f that are function valued. In other words, $f \in \mathcal{F}(\mathcal{X}; \mathcal{F}(\mathcal{T}; \mathbb{R}))$ and the Operator-Valued Kernel K act on a function g in the following way.

$$K(x,z)g = k_{\mathcal{X}}(x,z) \int_{\mathcal{Y}} k_{\mathcal{Y}}(\cdot, t) g(t) d\mu(t). \tag{3.20}$$

In Kadri et al. [73], the author studied the case where $\mathcal{T} = \mathbb{R}$ with $k_{\mathcal{Y}}(t,s) = \exp(-|t-s|)$ and applied it to speech inversion (Which and how Human articulators are activated from an audible speech signal). Notice that the Hilbert Schmidt integral kernel is a particular case of decomposable kernel where $\mathcal{Y} = \mathcal{F}(\mathcal{T}; \mathbb{R})$.

Part II

CONTRIBUTIONS

4

OPERATOR-VALUED RANDOM FOURIER FEATURES

Contents

4.1 MOTIVATION

Random Fourier Features have been proved useful to implement efficiently kernel methods in the scalar case. In this work, we propose to extend Random Fourier Feature methodology in order to approximate Operator-Valued Kernels. As in the scalar case, we are mainly interested on explicit approximated feature maps because they open the door to learning linear models. Our final goal is to come up with the definition of $\tilde{\Phi} : \mathcal{X} \to \mathcal{L}(\mathcal{Y}, \mathcal{H})$ for some Hilbert spaces $\mathcal{Y}$ and $\mathcal{H}$[1], a feature map of some approximation $\tilde{K}$ of a given OVK $\mathcal{K}$. This chapter is devoted to the construction of these approximations based on Random Fourier Feature principles. It is followed by a non asymptotical study of the error of approximation (chapter 4) and the development of learning tools based on Operator Random Fourier Feature maps with practical and theoretical insights.

We present in this chapter a construction methodology devoted to shift-invariant $\mathcal{Y}$-Mercer operator-valued kernels defined on any Locally Compact Abelian (LCA) group, noted $(\mathcal{X}, \star)$, for some operation noted $\star$. This allows us to use the general context of Pontryagin duality for Fourier Transform of functions on LCA groups. Building upon a generalization of the celebrated Bochner's theorem for operator-valued measures, an operator-valued kernel is seen as the *Fourier Transform* of an operator-valued positive measure. From that result, we extend the principle of RFF for scalar-valued kernels and derive a general methodology to build Operator-valued Random Fourier Feature (ORFF) when operator-valued kernels are shift-invariant according to the chosen group operation. Elements of this chapter have been developped in Brault, Heinonen, and d'Alché-Buc [24].

We present a construction of feature maps called Operator-valued Random Fourier Feature (ORFF), such that $f : x \mapsto \tilde{\Phi}(x)^*\theta$ is a continuous function that maps an arbitrary LCA group $\mathcal{X}$ as input space to an arbitrary output Hilbert space $\mathcal{Y}$. First we define a functional *Fourier feature map*, and then propose a Monte-Carlo sampling from this feature map to construct an approximation of a shift-invariant $\mathcal{Y}$-Mercer kernel. Then, we prove the convergence of the kernel approximation $\tilde{K}(x, z) = \tilde{\Phi}(x)^*\tilde{\Phi}(z)$ with high probability on *compact* subsets of the LCA $\mathcal{X}$, when $\mathcal{Y}$ is *finite dimensional*. Eventually we conclude with some numerical experiments.

4.2 THEORETICAL STUDY

The following proposition of Carmeli et al. [34] and Zhang, Xu, and Zhang [173] extends Bochner's theorem to any shift-invariant $\mathcal{Y}$-Mercer kernel.

Proposition 4.1 (Operator-valued Bochner's theorem [108, 173]). *If a function K from $\mathcal{X} \times \mathcal{X}$ to $\mathcal{Y}$ is a shift-invariant $\mathcal{Y}$-Mercer kernel on $\mathcal{X}$, then*

1 In the finite dimension case, $\mathcal{H} = \mathbb{R}^p$.

there exists a unique positive projection-valued measure $\widehat{\mathcal{Q}}: \mathcal{B}(\mathcal{X}) \to \mathcal{L}_+(\mathcal{Y})$ such that for all $x, z \in \mathcal{X}$,

$$K(x,z) = \int_{\widehat{\mathcal{X}}} \overline{(x \star z^{-1}, \omega)} d\widehat{\mathcal{Q}}(\omega), \tag{4.1}$$

where $\widehat{\mathcal{Q}}$ belongs to the set of all the projection-valued measures of bounded variation on the σ-algebra of Borel subsets of $\widehat{\mathcal{X}}$. Conversely, from any positive operator-valued measure M, a shift-invariant kernel K can be defined by equation 4.1.

Although this theorem is central to the spectral decomposition of shift-invariant $\mathcal{Y}$-Mercer OVK, the following results proved by Carmeli et al. [34] provides insights about this decomposition that are more relevant in practice. It first gives the necessary conditions to build shift-invariant $\mathcal{Y}$-Mercer kernel with a pair $(A, \widehat{\mu})$ where A is an operator-valued function on $\widehat{\mathcal{X}}$ and $\widehat{\mu}$ is a real-valued positive measure on $\widehat{\mathcal{X}}$. Note that obviously such a pair is not unique and the choice of this paper may have an impact on theoretical properties as well as practical computations. Secondly it also states that any OVK have such a spectral decomposition when $\mathcal{Y}$ is finite dimensional or $\mathcal{X}$.

Proposition 4.2 (Carmeli et al. [34]). *Let $\widehat{\mu}$ be a positive measure on $\mathcal{B}(\widehat{\mathcal{X}})$ and $A : \widehat{\mathcal{X}} \to \mathcal{L}(\mathcal{Y})$ such that $\langle A(\cdot)y, y'\rangle \in L^1(\mathcal{X}, \widehat{\mu})$ for all $y, y' \in \mathcal{Y}$ and $A(\omega) \succcurlyeq 0$ for $\widehat{\mu}$-almost all $\omega \in \widehat{\mathcal{X}}$. Then, for all $\delta \in \mathcal{X}$,*

$$K_e(\delta) = \int_{\widehat{\mathcal{X}}} \overline{(\delta, \omega)} A(\omega) d\widehat{\mu}(\omega) \tag{4.2}$$

is the kernel signature of a shift-invariant $\mathcal{Y}$-Mercer kernel K such that $K(x,z) = K_e(x \star z^{-1})$. The VV-RKHS $\mathcal{H}_K$ is embed in $L^2(\widehat{\mathcal{X}}, \widehat{\mu}; \mathcal{Y}')$ by means of the feature operator

$$(Wg)(x) = \int_{\widehat{\mathcal{X}}} \overline{(x, \omega)} B(\omega) g(\omega) d\widehat{\mu}(\omega), \tag{4.3}$$

Where $B(\omega)B(\omega)^* = A(\omega)$ and both integrals converge in the weak sense. If $\mathcal{Y}$ is finite dimensional or $\mathcal{X}$ is compact, any shift-invariant kernel is of the above form for some pair $(A, \widehat{\mu})$.

When $p = 1$ one can always assume A is reduced to the scalar 1, $\widehat{\mu}$ is still a bounded positive measure and we retrieve the Bochner theorem applied to the scalar case (theorem 2.4).

Proposition 4.2 shows that a pair $(A, \widehat{\mu})$ entirely characterize an OVK. Namely a given measure $\widehat{\mu}$ and a function A such that $\langle y', A(.)y\rangle \in L^1(\mathcal{X}, \widehat{\mu})$ for all $y, y' \in \mathcal{Y}$ and $A(\omega) \succcurlyeq 0$ for $\widehat{\mu}$-almost all ω, give rise to an OVK. Since $(A, \widehat{\mu})$ determine a unique kernel we can write $\mathcal{H}_{(A,\widehat{\mu})} \Longrightarrow \mathcal{H}_K$ where K is defined as in equation 4.2. However the converse is not true: Given a $\mathcal{Y}$-Mercer shift invariant Operator-Valued Kernel, there exist infinitely many pairs $(A, \widehat{\mu})$ that characterize an OVK.

The main difference between equation 4.1 and equation 4.2 is that the first one characterizes an OVK by a unique Positive Operator-Valued Measure (POVM), while the second one shows that the POVM that uniquely characterize a $\mathcal{Y}$-Mercer OVK has an operator-valued density with respect to a *scalar* measure $\widehat{\mu}$; and that this operator-valued density is not unique.

Finally proposition 4.2 does not provide any *constructive* way to obtain the pair $(A, \widehat{\mu})$ that characterizes an OVK. The following subsection 4.2.1 is based on another proposition of Carmeli, De Vito, and Toigo and shows that if the kernel signature $K_e(\delta)$ of an OVK is in L^1 then it is possible to construct *explicitly* a pair $(C, \widehat{\mathbf{Haar}})$ from it. Additionally, we show that we can always extract a scalar-valued *probability* density function from C such that we obtain a pair $(A, \mathbf{Pr}_{\widehat{\mu},\rho})$ where $\mathbf{Pr}_{\widehat{\mu},\rho}$ is a *probability* distribution absolutely continuous with respect to $\widehat{\mu}$ and with associated probibility density function (p. d. f) ρ. Thus for all $\mathcal{Z} \subset \mathcal{B}(\widehat{\mathcal{X}})$,

$$\mathbf{Pr}_{\widehat{\mu},\rho}(\mathcal{Z}) = \int_{\mathcal{Z}} \rho(\omega) d\widehat{\mu}(\omega).$$

When the reference measure $\widehat{\mu}$ is the Lebesgue measure, we note $\mathbf{Pr}_{\widehat{\mu},\rho} = \mathbf{Pr}_{\rho}$.

4.2.1 *Sufficient conditions of existence*

While proposition 4.2 gives some insights on how to build an approximation of a $\mathcal{Y}$-Mercer kernel, we need a theorem that provides an explicit construction of the pair $(A, \mathbf{Pr}_{\widehat{\mu},\rho})$ from the kernel signature K_e. Proposition 14 in Carmeli et al. [34] gives the solution, and also provides a sufficient condition for proposition 4.2 to apply.

Proposition 4.3 (Carmeli et al. [34]). *Let K be a shift-invariant $\mathcal{Y}$-Mercer kernel of signature K_e. Suppose that for all $z \in \mathcal{X}$ and for all $y, y' \in \mathcal{Y}$, the function*

$$\langle K_e(.)y, y' \rangle_{\mathcal{Y}} \in L^1(\mathcal{X}, \mathbf{Haar})$$

where $\mathcal{X}$ is endowed with the group law $\star$. Denote $C : \widehat{X} \to \mathcal{L}(\Upsilon)$, the function defined for all $\omega \in \widehat{\mathcal{X}}$ that satisfies for all y, y' in $\mathcal{Y}$:

$$\begin{aligned} \langle y', C(\omega)y \rangle_{\mathcal{Y}} &= \int_{\mathcal{X}} \overline{(\delta, \omega)} \langle y', K_e(\delta)y \rangle_{\mathcal{Y}} d\mathbf{Haar}(\delta) \\ &= \mathcal{F}^{-1}\left[\langle y', K_e(\cdot)y \rangle \right]_{\mathcal{Y}}(\omega). \end{aligned} \tag{4.4}$$

Then

1. *$C(\omega)$ is a bounded non-negative operator for all $\omega \in \widehat{\mathcal{X}}$,*
2. *$\langle y, C(\cdot)y' \rangle_{\mathcal{Y}} \in L^1\left(\widehat{\mathcal{X}}, \widehat{\mathbf{Haar}}\right)$ for all $y, y' \in \mathcal{X}$,*

3. *for all $\delta \in \mathcal{X}$ and for all y, y' in $\mathcal{Y}$,*

$$\begin{aligned}\langle y', K_e(\delta)y\rangle_{\mathcal{Y}} &= \int_{\widehat{\mathcal{X}}} \overline{(\delta,\omega)}\langle y', C(\omega)y\rangle_{\mathcal{Y}} d\widehat{\mathbf{Haar}}(\omega)\\ &= \mathcal{F}\left[\langle y', C(\cdot)y\rangle_{\mathcal{Y}}\right](\delta).\end{aligned}$$

There has been a lot of confusion in the literature whether a kernel is the Fourier Transform or Inverse Fourier Transform of a measure. However lemma 4.1 clarifies the relation between the Fourier Transform and Inverse Fourier Transform for a translation invariant Operator-Valued Kernel. Notice that in the real scalar case the Fourier Transform and Inverse Fourier Transform of a shift-invariant kernel are the same, while the difference is significant for OVK.

The following lemma is a direct consequence of the definition of $C(\omega)$ as the Fourier Transform of the adjoint of K_e and also helps to simplify the definition of ORFF.

Lemma 4.1 *Let K_e be the signature of a shift-invariant $\mathcal{Y}$-Mercer kernel such that for all $y, y' \in \mathcal{Y}$, $\langle y', K_e(\cdot)y\rangle_{\mathcal{Y}} \in L^1(\mathcal{X}, \mathbf{Haar})$ and let*

$$\langle y', C(\cdot)y\rangle_{\mathcal{Y}} = \mathcal{F}^{-1}\left[\langle y', K_e(\cdot)y\rangle_{\mathcal{Y}}\right].$$

Then

1. *$C(\omega)$ is self-adjoint and C is even.*
2. $\mathcal{F}^{-1}\left[\langle y', K_e(\cdot)y\rangle_{\mathcal{Y}}\right] = \mathcal{F}\left[\langle y', K_e(\cdot)y\rangle_{\mathcal{Y}}\right]$.
3. *$K_e(\delta)$ is self-adjoint and K_e is even.*

Proof *For any function f on $(\mathcal{X}, \star)$ define the flip operator $\mathcal{R}$ by*

$$(\mathcal{R}f)(x) := f\left(x^{-1}\right).$$

For any shift invariant $\mathcal{Y}$-Mercer kernel and for all $\delta \in \mathcal{X}$, $K_e(\delta) = K_e\left(\delta^{-1}\right)^*$. Indeed from the definition of a shift-invariant kernel,

$$K_e\left(\delta^{-1}\right) = K\left(\delta^{-1}, e\right) = K(e, \delta) = K(\delta, e)^* = K_e(\delta)^*.$$

Proposition 4.3 item 1: taking the Fourier Transform yields,

$$\begin{aligned}\langle y', C(\omega)y\rangle_{\mathcal{Y}} &= \mathcal{F}^{-1}\left[\langle y', K_e(\cdot)y\rangle_{\mathcal{Y}}\right](\omega)\\ &= \mathcal{F}^{-1}\left[\langle y', (\mathcal{R}K_e(\cdot))^* y\rangle_{\mathcal{Y}}\right](\omega)\\ &= \mathcal{F}^{-1}\left[\langle \mathcal{R}K_e(\cdot)y', y\rangle_{\mathcal{Y}}\right](\omega)\\ &= \mathcal{F}^{-1}\left[\mathcal{R}\langle K_e(\cdot)y', y\rangle_{\mathcal{Y}}\right](\omega)\\ &= \mathcal{R}\mathcal{F}^{-1}\left[\langle K_e(\cdot)y', y\rangle_{\mathcal{Y}}\right](\omega)\\ &= \mathcal{R}\langle C(\cdot)y', y\rangle_{\mathcal{Y}}(\omega)\\ &= \left\langle y', C\left(\omega^{-1}\right)^* y\right\rangle_{\mathcal{Y}}.\end{aligned}$$

Hence $C(\omega) = C\left(\omega^{-1}\right)^*$. Suppose that $\mathcal{Y}$ is a complex Hilbert space. Since for all $\omega \in \widehat{\mathcal{X}}$, $C(\omega)$ is bounded and non-negative so $C(\omega)$ is self-adjoint. Besides we have $C(\omega) = C\left(\omega^{-1}\right)^*$ so C must be even. Suppose that $\mathcal{Y}$ is a real Hilbert space. The Fourier Transform of a real valued function obeys $\mathcal{F}[f](\omega) = \overline{\mathcal{F}[f](\omega^{-1})}$. Therefore since $C(\omega)$ is non-negative for all $\omega \in \widehat{\mathcal{X}}$,

$$\begin{aligned}\langle y', C(\omega)y\rangle &= \overline{\langle y', C\left(\omega^{-1}\right)y\rangle} = \langle y, C\left(\omega^{-1}\right)^* y'\rangle \\ &= \langle y, C(\omega)y'\rangle.\end{aligned}$$

Hence $C(\omega)$ is self-adjoint and thus C is even. Proposition 4.3 item 2: simply, for all $y, y' \in \mathcal{Y}$, $\langle y, C(\omega^{-1})y'\rangle = \langle y', C(\omega)y\rangle$ thus

$$\begin{aligned}\mathcal{F}^{-1}\left[\langle y', K_e(\cdot)y\rangle_{\mathcal{Y}}\right](\omega) = \langle y', C(\omega)y\rangle &= \mathcal{R}\langle y', C(\cdot)y\rangle(\omega) \\ &= \mathcal{R}\mathcal{F}^{-1}\left[\langle y', K_e(\cdot)y\rangle_{\mathcal{Y}}\right](\omega) \\ &= \mathcal{F}\left[\langle y', K_e(\cdot)y\rangle_{\mathcal{Y}}\right](\omega).\end{aligned}$$

Proposition 4.3 item 3: from proposition 4.3 item 2 we have $\mathcal{F}^{-1}\left[\langle y', K_e(\cdot)y\rangle\right] = \mathcal{F}^{-1}\mathcal{R}\langle y', K_e(\cdot)y\rangle$. By injectivity of the Fourier Transform, K_e is even. Since $K_e(\delta) = K_e(\delta^{-1})^*$, we must have $K_e(\delta) = K_e(\delta)^*$. □

While proposition 4.3 gives an explicit form of the operator $C(\omega)$ defined as the Fourier Transform of the kernel K, it is not really convenient to work with the Haar measure $\widehat{\mathbf{Haar}}$ on $\mathcal{B}(\widehat{\mathcal{X}})$. However it is easily possible to turn $\widehat{\mathbf{Haar}}$ into a probability measure to allow efficient integration over an infinite domain.

The following proposition allows to build a spectral decomposition of a shift-invariant $\mathcal{Y}$-Mercer kernel on a LCA group $\mathcal{X}$ endowed with the group law $\star$ with respect to a scalar probability measure, by extracting a scalar probability density function from C.

Proposition 4.4 (Shift-invariant $\mathcal{Y}$-Mercer kernel spectral decomposition). *Let K_e be the signature of a shift-invariant $\mathcal{Y}$-Mercer kernel. If for all $y, y' \in \mathcal{Y}$, $\langle K_e(.)y, y'\rangle \in L^1(\mathcal{X}, \mathbf{Haar})$ then there exists a positive probability measure $\mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho}$ and an operator-valued function A an such that for all $y, y' \in \mathcal{Y}$,*

$$\langle y', K_e(\delta)y\rangle = \mathbf{E}_{\widehat{\mathbf{Haar}},\rho}\left[\overline{(\delta,\omega)}\langle y', A(\omega)y\rangle\right], \tag{4.5}$$

with

$$\langle y', A(\omega)y\rangle\rho(\omega) = \mathcal{F}\left[\langle y', K_e(\cdot)y\rangle\right](\omega). \tag{4.6}$$

Moreover

1. *for all $y, y' \in \mathcal{Y}$, $\langle A(.)y, y'\rangle \in L^1(\widehat{\mathcal{X}}, \mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho})$,*
2. *$A(\omega)$ is non-negative for $\mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho}$-almost all $\omega \in \widehat{\mathcal{X}}$,*
3. *$A(\cdot)$ and $\rho(\cdot)$ are even functions.*

Proof *This is a simple consequence of proposition 4.3 and lemma 4.1. By taking* $\langle y', C(\omega)y\rangle = \mathcal{F}^{-1}\left[\langle y', K_e(\cdot)y\rangle\right](\omega) = \mathcal{F}\left[\langle y', K_e(\cdot)y\rangle\right](\omega)$ *we can write the following equality concerning the OVK signature* K_e.

$$\begin{aligned}\langle y', K_e(\delta)y\rangle(\omega) &= \int_{\widehat{\mathcal{X}}} \overline{(\delta,\omega)}\langle y', C(\omega)y\rangle d\widehat{\mathbf{Haar}}(\omega)\\ &= \int_{\widehat{\mathcal{X}}} \overline{(\delta,\omega)}\left\langle y', \frac{1}{\rho(\omega)}C(\omega)y\right\rangle \rho(\omega)d\widehat{\mathbf{Haar}}(\omega).\end{aligned}$$

It is always possible to choose $\rho(\omega)$ *such that* $\int_{\widehat{\mathcal{X}}}\rho(\omega)d\widehat{\mathbf{Haar}}(\omega) = 1$. *For instance choose*

$$\rho(\omega) = \frac{\|C(\omega)\|_{\mathcal{Y},\mathcal{Y}}}{\int_{\widehat{\mathcal{X}}}\|C(\omega)\|_{\mathcal{Y},\mathcal{Y}} d\widehat{\mathbf{Haar}}(\omega)}$$

Since for all $y, y' \in \mathcal{Y}$, $\langle y', C(\cdot)y\rangle \in L^1(\widehat{\mathcal{X}}, \widehat{\mathbf{Haar}})$ *and* $\mathcal{Y}$ *is a separable Hilbert space, by pettis measurability theorem,* $\int_{\widehat{\mathcal{X}}}\|C(\omega)\|_{\mathcal{Y},\mathcal{Y}} d\widehat{\mathbf{Haar}}(\omega)$ *is finite and so is* $\|C(\omega)\|_{\mathcal{Y},\mathcal{Y}}$ *for all* $\omega \in \widehat{\mathcal{X}}$. *Therefore* $\rho(\omega)$ *is the density of a probability measure* $\mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho}$, *i. e. conclude by taking*

$$\mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho}(\mathcal{Z}) = \int_{\mathcal{Z}} \rho(\omega)d\widehat{\mathbf{Haar}}(\omega),$$

for all $\mathcal{Z} \in \mathcal{B}(\widehat{\mathcal{X}})$. □

In the case where $\mathcal{Y} = \mathbb{R}^p$, we rewrite equation 4.6 coefficient-wise by choosing an orthonormal basis $\{e_j\}_{j\in\mathbb{N}^*_p}$ of $\mathbb{R}^p$.

$$A(\omega)_{ij}\rho(\omega) = \mathcal{F}\left[K_e(\cdot)_{ij}\right](\omega). \tag{4.7}$$

It follows that for all i and j in $\mathbb{N}^*_p$,

$$K_e(x \star z^{-1})_{ij} = \mathcal{F}\left[A(\cdot)_{ij}\rho(\cdot)\right](x \star z^{-1}) \tag{4.8}$$

Remark 4.1 *Note that although the Fourier Transform of* K_e *yields a unique operator-valued function* $C(\cdot)$, *the decomposition of* $C(\cdot)$ *into* $A(\cdot)\rho(\cdot)$ *is again not unique. The choice of the decomposition may be justified by the computational cost.*

Another difficulty arises from the fact that the quantity $\sup_{\omega\in\widehat{\mathcal{X}}}\|A(\omega)\|_{\mathcal{Y},\mathcal{Y}}$ obtained in proposition 4.4 might not be bounded. Later, when we will focus on Monte-Carlo approximation of these integrals, we will have to take care of the unboundedness of $\|A(\cdot)\|_{\mathcal{Y},\mathcal{Y}}$ that forbids the use of the most simple concentrations inequalities that require the boundedness of the random variable to be controlled. Therefore in the context of Operator-Valued Kernel concentration inequalities for unbounded random operators should be used.

However, as pointed out by Minh [101], under some condition on the trace of $K_e(\delta)$, it is possible to turn $A(\cdot)$ into a bounded random operator for all ω in $\widehat{\mathcal{X}}$. The idea is to define a sum measure $\rho = \sum_{j\in\mathbb{N}^*}\rho_{e_j}$, which gives rise to a bounded operator $A(\omega)$ and is independent of the $\{ e_j \}_{j\in\mathbb{N}^*}$ base, instead of constructing a measure from the operator norm as in proposition 4.4. Additionally with such construction the measure associated to $A(\cdot)$ is *independent* from the basis of $\mathcal{Y}$. We present this result and in this proof we relax the assumptions of Minh [101] which requires $\int_{\mathcal{X}} |\operatorname{\mathbf{Tr}} K_e(\delta)| d\mathbf{Haar}(\delta)$ to be well defined. We only require $\operatorname{\mathbf{Tr}} K_e(e)$ to be well defined.

Proposition 4.5 (Bounded shift-invariant $\mathcal{Y}$-Mercer kernel spectral decomposition (adaptation of a result from Minh [101]). *Let K_e be the signature of a shift-invariant $\mathcal{Y}$-Mercer kernel. If for all y and y' in $\mathcal{Y}$, $\langle K_e(.)y, y'\rangle \in L^1(\mathcal{X}, \mathbf{Haar})$ and $\operatorname{\mathbf{Tr}} K_e(e) \in \mathbb{R}$, then*

$$\langle y', K_e(\delta)y\rangle = \mathbf{E}_{\widehat{\mathbf{Haar}},\rho_{\mathbf{Tr}}}\left[\overline{(\delta,\omega)}\langle y, A_{\mathbf{Tr}}(\omega)y'\rangle\right]. \tag{4.9}$$

with

$$\langle y', C(\cdot)y\rangle = \mathcal{F}\left[\langle y', K_e(\cdot)y\rangle\right] \tag{4.10a}$$

$$c_{\mathbf{Tr}} = \operatorname{\mathbf{Tr}}\left[K_e(e)\right] \tag{4.10b}$$

$$A_{\mathbf{Tr}}(\omega) = c_{\mathbf{Tr}} \operatorname{\mathbf{Tr}}\left[C(\omega)\right]^{-1} C(\omega) \tag{4.10c}$$

$$\rho_{\mathbf{Tr}}(\omega) = c_{\mathbf{Tr}}^{-1} \operatorname{\mathbf{Tr}}\left[C(\omega)\right]. \tag{4.10d}$$

Moreover

1. *For all $y, y' \in \mathcal{Y}$, $\langle y, A_{\mathbf{Tr}}(\cdot)y'\rangle \in L^1(\widehat{\mathcal{X}}, \mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho_{\mathbf{Tr}}})$.*
2. *$A_{\mathbf{Tr}}(\omega)$ is non-negative for all $\omega \in \widehat{\mathcal{X}}$,*
3. *$\operatorname{ess\,sup}_{\omega\in\widehat{\mathcal{X}}} \|A_{\mathbf{Tr}}(\omega)\|_{\mathcal{Y},\mathcal{Y}} \le c_p$,*
4. *$A_{\mathbf{Tr}}(\cdot)$ and $\rho_{\mathbf{Tr}}$ are even functions.*

Proof *Let $\{ e_j \}_{j\in\mathbb{N}^*}$ be an orthonormal basis of $\mathcal{Y}$. Notice that*

$$\begin{aligned}\int_{\widehat{\mathcal{X}}} \langle e_j, C(\omega)e_j\rangle d\widehat{\mathbf{Haar}}(\omega) &= \int_{\widehat{\mathcal{X}}} \underbrace{\overline{(e,\omega)}}_{=1}\langle e_j, C(\omega)e_j\rangle d\widehat{\mathbf{Haar}}(\omega) \\ &= \langle e_j, K_e(e)e_j\rangle.\end{aligned}$$

Since $C(\omega)$ is non-negative, all the $\langle e_j, C(\omega)e_j\rangle$. Thus using the monotone convergence theorem,

$$\begin{aligned}\int_{\widehat{\mathcal{X}}} \operatorname{\mathbf{Tr}}\left[C(\omega)\right] d\widehat{\mathbf{Haar}}(\omega) &= \int_{\widehat{\mathcal{X}}} \sum_{j\in\mathbb{N}^*} \langle e_j, C(\omega)e_j\rangle d\widehat{\mathbf{Haar}}(\omega) \\ &= \sum_{k\in\mathbb{N}^*} \langle e_j, K_e(e)e_j\rangle \\ &= \operatorname{\mathbf{Tr}}\left[K_e(e)\right] = c_{\mathbf{Tr}} < \infty.\end{aligned}$$

Let $A_{\mathbf{Tr}}(\omega)$ and $\rho_{\mathbf{Tr}}(\omega)$ be defined as in equation 4.10c and equation 4.10d, respectively. By definition, $\int_{\widehat{\mathcal{X}}} \rho_{\mathbf{Tr}}(\omega) d\widehat{\mathbf{Haar}}(\omega) = 1$ and $A_{\mathbf{Tr}}(\omega)\rho_{\mathbf{Tr}}(\omega) = C(\omega)$.

Now it remains to check the finiteness of $\mathbf{Tr}\left[C(\omega)\right]$ *for all* $\omega \in \widehat{\mathcal{X}}$. *Since for all* $\omega \in \widehat{\mathcal{X}}$, $\mathbf{Tr}\left[C(\omega)\right] \geq 0$,

$$\mathbf{Tr}\left[C(\omega)\right] \leq \int_{\widehat{\mathcal{X}}} \mathbf{Tr}\left[C(\omega)\right] d\widehat{\mathbf{Haar}}(\omega) = \mathbf{Tr}\left[K_e(e)\right] < \infty.$$

Since $\mathbf{Tr}\left[C(\omega)\right]$ *is positive and its integral is finite,* $\rho_{\mathbf{Tr}}$ *is a probability density function. In particular* $C(\omega)$ *is self-adjoint operator thus* $\|C(\omega)\|_{\sigma,\infty} = \|C(\omega)\|_{\mathcal{Y},\mathcal{Y}}$ *for all* $\omega \in \widehat{\mathcal{X}}$. *Thus the Schatten norms* $\|\cdot\|_{\sigma,p}$ *verifies* $\mathbf{Tr}\left[|\cdot|\right] = \|\cdot\|_{\sigma,1} \geq \|\cdot\|_{\sigma,p} \geq \|\cdot\|_{\sigma,q} \geq \|\cdot\|_{\sigma,\infty} = \|\cdot\|_{\mathcal{Y},\mathcal{Y}}$ *for all* $p, q \in \mathbb{N}^*$ *such that* $1 \leq p \leq q \leq \infty$. *Therefore since for all* $\omega \in \widehat{\mathcal{X}}$, $C(\omega)$ *is non-negative, we have for* $\mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho}$-*almost all* $\omega \in \widehat{\mathcal{X}}$,

$$\begin{aligned}
\|A_{\mathbf{Tr}}(\omega)\|_{\mathcal{Y},\mathcal{Y}} &= c_{\mathbf{Tr}} \mathbf{Tr}\left[C(\omega)\right]^{-1} \|C(\omega)\|_{\sigma,\infty} \\
&\leq c_{\mathbf{Tr}} \mathbf{Tr}\left[C(\omega)\right]^{-1} \|C(\omega)\|_{\sigma,1} \\
&= c_{\mathbf{Tr}} \mathbf{Tr}\left[C(\omega)\right]^{-1} \mathbf{Tr}\left[|C(\omega)|\right] \\
&= c_{\mathbf{Tr}} \mathbf{Tr}\left[C(\omega)\right]^{-1} \mathbf{Tr}\left[C(\omega)\right] \\
&\leq c_{\mathbf{Tr}} < \infty.
\end{aligned}$$

Thus $\operatorname{ess\,sup}_{\omega \in \widehat{\mathcal{X}}} \|A(\omega)\|_{\mathcal{Y},\mathcal{Y}} \leq c_{\mathbf{Tr}} < \infty$. *As* C *is an even function, so are* $A_{\mathbf{Tr}}$ *and* $\rho_{\mathbf{Tr}}$. *Eventually* $\langle y', C(\cdot)y\rangle$ *is in* $L^1(\widehat{\mathcal{X}}, \widehat{\mathbf{Haar}})$, *thus* $\langle y, A_{\mathbf{Tr}}(\cdot)\rho_{\mathbf{Tr}}(\cdot)y'\rangle$ *is in* $L^1(\widehat{\mathcal{X}}, \widehat{\mathbf{Haar}})$, *hence* $\langle y, A_{\mathbf{Tr}}(\cdot)y'\rangle \in L^1(\widehat{\mathcal{X}}, \mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho_{\mathbf{Tr}}})$. *Since the trace is independent of the basis of* $\mathcal{Y}$, *so is* $\rho_{\mathbf{Tr}}$. □

If $\mathcal{Y}$ is finite dimensional then $\mathbf{Tr}\left[K_e(e)\right]$ is well defined hence proposition 4.5 is valid as long as $K_e(\cdot)_{ij} \in L^1(\mathcal{X}, \mathbf{Haar})$ for all $i, j \in \mathbb{N}^*_p$, where p is the dimension of $\mathcal{Y}$.

4.2.2 *Examples of spectral decomposition*

In this section we give examples of spectral decomposition for various $\mathcal{Y}$-Mercer kernels, based on proposition 4.4 and proposition 4.5.

4.2.2.1 *Gaussian decomposable kernel*

Recall that a decomposable $\mathbb{R}^p$-Mercer introduced in the Background section has the form $K(x,z) = k(x,z)\Gamma$, where $k(x,z)$ is a scalar Mercer kernel and $\Gamma \in \mathcal{L}(\mathbb{R}^p)$ is a non-negative operator. Let us focus on $K_e^{dec,gauss}(\cdot) = k_e^{gauss}(\cdot)\Gamma$, the Gaussian decomposable kernel where $K_e^{dec,gauss}$ and k_e^{gauss} are respectively the signature of K and k on the additive group $\mathcal{X} = (\mathbb{R}^d, +)$ – *i.e.* $\delta = x - z$ and $e = 0$. The well known Gaussian kernel is defined for all $\delta \in \mathbb{R}^d$ as follows

$$k_0^{\text{gauss}}(\delta) = \exp\left(-\frac{1}{2\sigma^2}\|\delta\|_2^2\right)$$

where $\sigma \in \mathbb{R}_+$ is an hyperparameter corresponding to the bandwith of the kernel. The –Pontryagin– dual group of $\mathcal{X} = (\mathbb{R}^d, +)$ is $\widehat{\mathcal{X}} \cong (\mathbb{R}^d, +)$ with the pairing

$$(\delta, \omega) = \exp\left(i\langle\delta, \omega\rangle\right)$$

where δ and $\omega \in \mathbb{R}^d$. In this case the Haar measures on $\mathcal{X}$ and $\widehat{\mathcal{X}}$ are in both cases the Lebesgue measure. However in order to have the property that $\mathcal{F}^{-1}\left[\mathcal{F}\left[f\right]\right] = f$ and $\mathcal{F}^{-1}\left[f\right] = \mathcal{R}\mathcal{F}\left[f\right]$ one must normalize both measures by $\sqrt{2\pi}^{-d}$, i. e. for all $\mathcal{Z} \in \mathcal{B}\left(\mathbb{R}^d\right)$,

$$\begin{aligned}\sqrt{2\pi}^d\mathbf{Haar}(\mathcal{Z}) &= \mathbf{Leb}(\mathcal{Z}) \text{ and}\\ \sqrt{2\pi}^d\widehat{\mathbf{Haar}}(\mathcal{Z}) &= \mathbf{Leb}(\mathcal{Z}).\end{aligned}$$

Then the Fourier Transform on $(\mathbb{R}^d, +)$ is

$$\begin{aligned}\mathcal{F}\left[f\right](\omega) &= \int_{\mathbb{R}^d} \exp\left(-\mathrm{i}\langle\delta,\omega\rangle\right) f(x) d\mathbf{Haar}(\delta)\\ &= \int_{\mathbb{R}^d} \exp\left(-\mathrm{i}\langle\delta,\omega\rangle\right) f(x) \frac{d\mathbf{Leb}(\delta)}{\sqrt{2\pi}^d}.\end{aligned}$$

Since $k_0^{\text{gauss}} \in L^1$ and Γ is bounded, it is possible to apply proposition 4.4, and obtain for all y and $y' \in \mathcal{Y}$,

$$\begin{aligned}\left\langle y', C^{dec,gauss}(\omega)y\right\rangle &= \mathcal{F}\left[\left\langle y', K_0^{dec,gauss}(\cdot)y\right\rangle\right](\omega)\\ &= \mathcal{F}\left[k_0^{gauss}\right](\omega)\langle y', \Gamma y\rangle.\end{aligned}$$

Thus

$$C^{dec,gauss}(\omega) = \int_{\mathbb{R}^d} \exp\left(-\mathrm{i}\langle\omega, x\rangle - \frac{\|\delta\|_2^2}{2\sigma^2}\right)\frac{d\mathbf{Leb}(\delta)}{\sqrt{2\pi}^d}\Gamma.$$

Hence

$$C^{dec,gauss}(\omega) = \underbrace{\frac{1}{\sqrt{2\pi\frac{1}{\sigma^2}}^d}\exp\left(-\frac{\sigma^2}{2}\|\omega\|_2^2\right)\sqrt{2\pi}^d}_{\rho(\cdot)=\mathcal{N}(0,\sigma^{-2}I_d)\sqrt{2\pi}^d}\underbrace{\Gamma}_{A(\cdot)=\Gamma}.$$

Therefore the canonical decomposition of $C^{dec,gauss}$ is $A^{dec,gauss}(\omega) = \Gamma$ and $\rho^{dec,gauss} = \mathcal{N}(0, \sigma^{-2}I_d)\sqrt{2\pi}^d$, where $\mathcal{N}$ is the Gaussian probability distribution. Note that this decomposition is done with respect to the *normalized* Lebesgue measure $\widehat{\mathbf{Haar}}$, meaning that for all $\mathcal{Z} \in \mathcal{B}(\widehat{\mathcal{X}})$,

$$\begin{aligned}\mathbf{Pr}_{\widehat{\mathbf{Haar}},\mathcal{N}(0,\sigma^{-2}I_d)\sqrt{2\pi}^d}(\mathcal{Z}) &= \int_{\mathcal{Z}} \mathcal{N}(0,\sigma^{-2}I_d)\sqrt{2\pi}^d d\widehat{\mathbf{Haar}}(\omega)\\ &= \int_{\widehat{\mathcal{X}}} \mathcal{N}(0,\sigma^{-2}I_d) d\mathbf{Leb}(\omega)\\ &= \mathbf{Pr}_{\mathcal{N}(0,\sigma^{-2}I_d)}(\mathcal{Z}).\end{aligned}$$

Thus, the same decomposition with respect to the usual –non-normalized– Lebesgue measure $\mathbf{Leb}$ yields

$$A^{dec,gauss}(\cdot) = \Gamma \tag{4.11a}$$

$$\rho^{dec,gauss} = \mathcal{N}(0, \sigma^{-2}I_d). \tag{4.11b}$$

If Γ is a trace class operator, applying proposition 4.5 yields the same decomposition since $\mathbf{Tr}\left[K_0^{dec,gauss}(0)\right] = \mathbf{Tr}\left[\Gamma\right]$ and

$$\mathbf{Tr}\left[C^{dec,gauss}(\cdot)\right] = \mathcal{N}(0, \sigma^{-2}I_d)\sqrt{2\pi}^d\,\mathbf{Tr}\left[\Gamma\right].$$

4.2.2.2 *Skewed-χ^2 decomposable kernel*

The skewed-χ^2 scalar kernel [83], useful for image processing, is defined on the LCA group $\mathcal{X} = (-c_k; +\infty)_{k=1}^d$, with $c_k \in \mathbb{R}_+$ and endowed with the group operation $\odot$. Let $(e_k)_{k=1}^d$ be the standard basis of $\mathcal{X}$ and $_k : x \mapsto \langle x, e_k \rangle$. The operator $\odot : \mathcal{X} \times \mathcal{X} \to \mathcal{X}$ is defined by

$$x \odot z = ((x_k + c_k)(z_k + c_k) - c_k)_{k=1}^d .$$

The identity element e is $(1-c_k)_{k=1}^d$ since $(1-c) \odot x = x$. Thus the inverse element x^{-1} is $((x_k + c_k)^{-1} - c_k)_{k=1}^d$. The skewed-$\chi^2$ scalar kernel reads

$$k_{1-c}^{skewed}(\delta) = \prod_{k=1}^d \frac{2}{\sqrt{\delta_k + c_k} + \sqrt{\frac{1}{\delta_k + c_k}}}. \tag{4.12}$$

The dual of $\mathcal{X}$ is $\widehat{\mathcal{X}} \cong \mathbb{R}^d$ with the pairing

$$(\delta, \omega) = \prod_{k=1}^d \exp\left(\mathrm{i}\log(\delta_k + c_k)\omega_k\right) .$$

The Haar measure are defined for all $\mathcal{Z} \in \mathcal{B}((-c; +\infty)^d)$ and all $\widehat{\mathcal{Z}} \in \mathcal{B}(\mathbb{R}^d)$ by

$$\sqrt{2\pi}^d \mathbf{Haar}(\mathcal{Z}) = \int_{\mathcal{Z}} \prod_{k=1}^d \frac{1}{z_k + c_k} d\mathbf{Leb}(z)$$
$$\sqrt{2\pi}^d \widehat{\mathbf{Haar}}(\widehat{\mathcal{Z}}) = \mathbf{Leb}(\widehat{\mathcal{Z}}).$$

Thus the Fourier Transform is

$$\mathcal{F}\left[f\right](\omega) = \int_{(-c;+\infty)^d} \prod_{k=1}^d \frac{\exp\left(-\mathrm{i}\log(\delta_k + c_k)\omega_k\right)}{\delta_k + c_k} f(\delta) \frac{d\mathbf{Leb}(\delta)}{\sqrt{2\pi}^d}.$$

Then, applying Fubini's theorem over product space, and the fact that each dimension is independent

$$\mathcal{F}\left[k_0^{skewed}\right](\omega) = \prod_{k=1}^d \int_{-c_k}^{+\infty} \frac{2\exp\left(-\mathrm{i}\log(\delta_k + c_k)\omega_k\right)}{(\delta_k + c_k)\left(\sqrt{\delta_k + c_k} + \sqrt{\frac{1}{\delta_k + c_k}}\right)} \frac{d\mathbf{Leb}(\delta_k)}{\sqrt{2\pi}^d}.$$

Making the change of variable $t_k = (\delta_k + c_k)^{-1}$ yields

$$\begin{aligned}\mathcal{F}\left[k_0^{skewed}\right](\omega) &= \prod_{k=1}^d \int_{-\infty}^{+\infty} \frac{2\exp\left(-\mathrm{i}t_k\omega_k\right)}{\exp\left(\frac{1}{2}t_k\right) + \exp\left(-\frac{1}{2}t_k\right)} \frac{d\mathbf{Leb}(t_k)}{\sqrt{2\pi}^d} \\ &= \sqrt{2\pi}^d \prod_{k=1}^d \operatorname{sech}(\pi\omega_k).\end{aligned}$$

Since $k_{1-c}^{\text{skewed}} \in L^1$ and Γ is bounded, it is possible to apply proposition 4.4, and obtain

$$\begin{aligned}C^{dec,skewed}(\omega) &= \mathcal{F}\left[k_{1-c}^{skewed}\right](\omega)\Gamma \\ &= \underbrace{\sqrt{2\pi}^d \prod_{k=1}^d \operatorname{sech}(\pi\omega_k)}_{\rho(\cdot) = \mathcal{S}(0, 2^{-1})^d \sqrt{2\pi}^d} \underbrace{\Gamma}_{A(\cdot)}.\end{aligned}$$

Hence the decomposition with respect to the usual –non-normalized– Lebesgue measure **Leb** yields

$$A^{dec,skewed}(\cdot) = \Gamma \tag{4.13a}$$
$$\rho^{dec,skewed} = \mathcal{S}\left(0, 2^{-1}\right)^d. \tag{4.13b}$$

4.2.2.3 *Curl-free Gaussian kernel*

The curl-free Gaussian kernel is defined as $K_0^{curl,gauss} = -\nabla\nabla^\top k_0^{gauss}$. Here $\mathcal{X} = (\mathbb{R}^d, +)$ so the setting is the same than subsection 4.2.2.1.

$$\begin{aligned}
C^{curl,gauss}(\omega)_{ij} &= \mathcal{F}\left[K_{1-c}^{curl,gauss}(\cdot)_{ij}\right](\omega) \\
&= \mathcal{F}\left[-\frac{\partial^2}{\partial\delta_i\partial\delta_j}k_0^{gauss}\right](\omega) \\
&= -(\mathrm{i}\omega_i)(\mathrm{i}\omega_j)\mathcal{F}\left[k_0^{gauss}\right](\omega) \\
&= \omega_i\omega_j\mathcal{F}\left[k_0^{gauss}\right](\omega) \\
&= \sqrt{2\pi\frac{1}{\sigma^2}}^d \exp\left(-\frac{\sigma^2}{2}\|\omega\|_2^2\right)\sqrt{2\pi}^d\omega_i\omega_j.
\end{aligned}$$

Hence

$$C^{curl,gauss}(\omega) = \underbrace{\frac{1}{\sqrt{2\pi\frac{1}{\sigma^2}}^d}\exp\left(-\frac{\sigma^2}{2}\|\omega\|_2^2\right)\sqrt{2\pi}^d}_{\mu(\cdot)=\mathcal{N}(0,\sigma^{-2}I_d)\sqrt{2\pi}^d}\underbrace{\omega\omega^\top}_{A(\omega)=\omega\omega^\top}.$$

Here a canonical decomposition is $A^{curl,gauss}(\omega) = \omega\omega^\top$ for all $\omega \in \mathbb{R}^d$ and $\mu^{curl,gauss} = \mathcal{N}(0,\sigma^{-2}I_d)\sqrt{2\pi}^d$ with respect to the normalized Lebesgue measure $d\omega$. Again the decomposition with respect to the usual –non-normalized– Lebesgue measure is for all $\omega \in \mathbb{R}^d$

$$A^{curl,gauss}(\omega) = \omega\omega^\top \tag{4.14a}$$
$$\mu^{curl,gauss} = \mathcal{N}(0,\sigma^{-2}I_d). \tag{4.14b}$$

Notice that in this case $\left\|A^{curl,gauss}(\cdot)\right\|_{\mathbb{R}^d,\mathbb{R}^d}$ is not bounded. However applying proposition 4.5 yields a different decomposition where the quantity $\left\|A^{curl,gauss}_{\mathbf{Tr}}(\cdot)\right\|_{\mathbb{R}^d,\mathbb{R}^d}$ is bounded. First we have for all $\delta \in \mathbb{R}^d$ and for all i, $j \in \mathbb{N}_d^*$

$$\frac{\partial^2}{\partial\delta_i\partial\delta_j}k_0^{gauss}(\delta) = \frac{\exp\left(-\frac{1}{2\sigma^2}\|\delta\|_2^2\right)}{\sigma^2}\begin{cases}\frac{\delta_i\delta_j}{\sigma^2} & \text{if } i \neq j \\ \left(1-\frac{\delta_i\delta_j}{\sigma^2}\right) & \text{otherwise.}\end{cases}$$

Hence

$$-\nabla\nabla^\top k_0^{gauss}(\delta) = \left(I_d - \frac{\delta\delta^\top}{\sigma^2}\right)\frac{\exp\left(-\frac{1}{2\sigma^2}\|\delta\|_2^2\right)}{\sigma^2}.$$

Thus

$$\begin{aligned}\mathbf{Tr}\left[K_0^{curl,gauss}(0)\right] &= \mathbf{Tr}\left[\nabla\nabla^\top k_0^{gauss}(0)\right]\\ &= d\sigma^{-2}\end{aligned}$$

and

$$\mathbf{Tr}\left[C(\omega)\right] = \|\omega\|_2^2\mathcal{N}(0,\sigma^{-2}I_d)\sqrt{2\pi}^d.$$

Apply proposition 4.5 to obtain the decomposition $A_{\mathbf{Tr}}^{curl,gauss}(\omega) = \omega\omega^\top\|\omega\|_2^{-2}$ and the measure $\mu_{\mathbf{Tr}}^{curl,gauss}(\omega) = \sigma^2 d^{-1}\|\omega\|_2^2\mathcal{N}\left(0,\sigma^{-2}\right)\sqrt{2\pi}^d$ for all $\omega \in \mathbb{R}^d$, with respect to the normalized Lebesgue measure. Therefore the decomposition with respect to the usual non-normalized Lebesgue measure is

$$A_{\mathbf{Tr}}^{curl,gauss}(\omega) = \frac{\omega\omega^\top}{\|\omega\|_2^2} \tag{4.15a}$$

$$\mu_{\mathbf{Tr}}^{curl,gauss}(\omega) = \frac{\sigma^2}{d}\|\omega\|_2^2\mathcal{N}\left(0,\sigma^{-2}\right)(\omega). \tag{4.15b}$$

This example also illustrates that there exists many decompositions of $C(\omega)$ into $(A(\omega),\mu(\omega))$.

4.2.2.4 *Divergence-free kernel*

The divergence-free Gaussian kernel is defined as $K_0^{div,gauss} = (\nabla\nabla^\top - \Delta)k_0^{gauss}$ on the group $\mathcal{X} = (\mathbb{R}^d,+)$. The setting is the same than subsection 4.2.2.1. Hence

$$\begin{aligned}C^{div,gauss}(\omega)_{ij} &= \mathcal{F}\left[K_0^{div,gauss}(\cdot)_{ij}\right](\omega)\\ &= \mathcal{F}\left[\frac{\partial^2}{\partial\delta_i\partial\delta_j}k_0^{gauss} - \delta_{i=j}\sum_{k=1}^d\frac{\partial^2}{\partial\delta_k\partial\delta_k}k_0^{gauss}\right](\omega)\\ &= \left(-(\mathrm{i}\omega_i)(\mathrm{i}\omega_j) - \delta_{i=j}\sum_{k=1}^d(\mathrm{i}\omega_k)^2\right)\mathcal{F}\left[k_0^{gauss}\right]\\ &= \left(\delta_{i=j}\sum_{k=1}^d\omega_k^2 - \omega_i\omega_j\right)\mathcal{F}\left[k_0^{gauss}\right](\omega).\end{aligned}$$

Hence

$$C^{div,gauss}(\omega) = \underbrace{\frac{1}{\sqrt{2\pi\frac{1}{\sigma^2}}^d}\exp\left(-\frac{\sigma^2}{2}\|\omega\|_2^2\right)\sqrt{2\pi}^d}_{\rho(\cdot)=\mathcal{N}(0,\sigma^{-2}I_d)\sqrt{2\pi}^d}\underbrace{\left(I_d\|\omega\|_2^2 - \omega\omega^\top\right)}_{A(\omega)=I_d\|\omega\|_2^2-\omega\omega^\top}.$$

Thus the canonical decomposition with respect to the normalized Lebesgue measure is $A^{div,gauss}(\omega) = I_d\|\omega\|_2^2 - \omega\omega^\top$ and the measure

$$\rho^{div,gauss} = \mathcal{N}(0,\sigma^{-2}I_d)\sqrt{2\pi}^d.$$

The canonical decomposition with respect to the usual Lebesgue measure is

$$A^{div,gauss}(\omega) = I_d\|\omega\|_2^2 - \omega\omega^\top \tag{4.16a}$$

$$\rho^{div,gauss} = \mathcal{N}(0, \sigma^{-2}I_d). \tag{4.16b}$$

To obtain the bounded decomposition, again, apply proposition 4.5. For all $\delta \in \mathbb{R}^d$,

$$\sum_{k=1}^{d} \frac{\partial^2}{\partial\delta_k\partial\delta_k} k_0^{gauss}(\delta) = \left(d - \frac{\|\delta\|_2^2}{\sigma^2}\right) \frac{\exp\left(-\frac{1}{2\sigma^2}\|\delta\|_2^2\right)}{\sigma^2}.$$

Thus overall,

$$K_0^{div,gauss}(\delta) = \left(\frac{\delta\delta^\top}{\sigma^2} + \left((d-1) - \frac{\|\delta\|_2^2}{\sigma^2}\right) I_d\right) \frac{\exp\left(-\frac{1}{2\sigma^2}\|\delta\|_2^2\right)}{\sigma^2}.$$

Eventually $\mathbf{Tr}\left[K_0^{div,gauss}(0)\right] = \mathbf{Tr}\left[(\nabla\nabla^\top - \Delta)k_0^{gauss}(0)\right] = d(d-1)\sigma^{-2}$ and $\mathbf{Tr}\left[C(\omega)\right] = (d-1)\|\omega\|_2^2\mathcal{N}(0,\sigma^2 I_d)\sqrt{2\pi}^d$. As a result the decomposition with respect to the normalized Lebesgue measure is $A_{\mathbf{Tr}}^{div,gauss}(\omega) = (I_d - \omega\omega^\top\|\omega\|_2^{-2})$ and $\rho_{\mathbf{Tr}}^{div,gauss}(\omega) = d^{-1}\sigma^2\|\omega\|_2^2\mathcal{N}(0,\sigma^2 I_d)\sqrt{2\pi}^d$. The decomposition with respect to the normalized Lebesgue measure being

$$A_{\mathbf{Tr}}^{div,gauss}(\omega) = I_d - \frac{\omega\omega^\top}{\|\omega\|_2^2} \tag{4.17a}$$

$$\rho_{\mathbf{Tr}}^{div,gauss} = \frac{\sigma^2}{d}\|\omega\|_2^2\mathcal{N}(0,\sigma^{-2}I_d). \tag{4.17b}$$

4.2.3 *Functional Fourier feature map*

We introduce a *functional* feature map, we call *Fourier Feature map*, defined by the following proposition as a direct consequence of proposition 4.2.

Proposition 4.6 (Functional Fourier feature map). *Let $\mathcal{Y}$ and $\mathcal{Y}'$ be two Hilbert spaces. If there exist an operator-valued function $B : \widehat{\mathcal{X}} \to \mathcal{L}(\mathcal{Y}, \mathcal{Y}')$ such that for all $y, y' \in \mathcal{Y}$,*

$$\langle y, B(\omega)B(\omega)^* y'\rangle_{\mathcal{Y}} = \langle y', A(\omega)y\rangle_{\mathcal{Y}}$$

$\widehat{\mu}$-almost everywhere and $\langle y', A(\cdot)y\rangle \in L^1(\widehat{\mathcal{X}}, \widehat{\mu})$ then the operator Φ_x defined for all y in $\mathcal{Y}$ by

$$(\Phi_x y)(\omega) = (x, \omega)B(\omega)^* y, \tag{4.18}$$

is a feature map[8] *of some shift-invariant $\mathcal{Y}$-Mercer kernel K.*

[8] *i. e. it satisfies for all $x, z \in \mathcal{X}$, $\Phi_x^*\Phi_z = K(x,z)$ where K is a $\mathcal{Y}$-Mercer OVK.*

Proof *For all $y, y' \in \mathcal{Y}$ and $x, z \in \mathcal{X}$,*

$$
\begin{aligned}
\langle y, \Phi_x^* \Phi_z y' \rangle_{\mathcal{Y}} &= \langle \Phi_x y, \Phi_z y' \rangle_{L^2(\widehat{\mathcal{X}}, \widehat{\mu}; \mathcal{Y}')} \\
&= \int_{\widehat{\mathcal{X}}} \overline{(x, \omega)} \langle y, B(\omega)(z, \omega) B(\omega)^* y' \rangle d\widehat{\mu}(\omega) \\
&= \int_{\widehat{\mathcal{X}}} \overline{(x \star z^{-1}, \omega)} \langle y B(\omega) B(\omega)^* y' \rangle d\widehat{\mu}(\omega) \\
&= \int_{\widehat{\mathcal{X}}} \overline{(x \star z^{-1}, \omega)} \langle y, A(\omega) y' \rangle d\widehat{\mu}(\omega),
\end{aligned}
$$

which defines a $\mathcal{Y}$-Mercer according to proposition 4.2 of Carmeli et al. [34]. □

With this notation we have $\Phi : \mathcal{X} \to \mathcal{L}(\mathcal{Y}; L^2(\widehat{\mathcal{X}}, \widehat{\mu}; \mathcal{Y}'))$ such that $\Phi_x \in \mathcal{L}(\mathcal{Y}; L^2(\widehat{\mathcal{X}}, \widehat{\mu}; \mathcal{Y}'))$ where $\Phi_x := \Phi(x)$.

4.3 OPERATOR-VALUED RANDOM FOURIER FEATURES

4.3.1 *Building Operator-valued Random Fourier Features*

As shown in propositions 4.4 and 4.5 it is always possible to find a pair $(A, \mathbf{Pr}_{\widehat{\mathbf{Haar}}, \rho})$ from a shift invariant $\mathcal{Y}$-Mercer Operator-Valued Kernel K_e such that $\mathbf{Pr}_{\widehat{\mathbf{Haar}}, \rho}$ is a probability measure, i. e. $\int_{\widehat{\mathcal{X}}} \rho d\widehat{\mathbf{Haar}} = 1$ where ρ is the density of $\mathbf{Pr}_{\widehat{\mathbf{Haar}}, \rho}$ and $K_e(\delta) = \mathbf{E}_\rho \overline{(\delta, \omega)} A(\omega)$. In order to obtain an approximation of K from a decomposition $(A, \mathbf{Pr}_{\widehat{\mathbf{Haar}}, \rho})$ we turn our attention to a Monte-Carlo estimation of the expectations in equation 4.9 and equation 4.5 characterizing a $\mathcal{Y}$-Mercer shift-invariant Operator-Valued Kernel. In the following, for sake of simplicity, we will assume that $\mathcal{Y}$ and $\mathcal{Y}$ are finite dimensional Hilbert spaces.

Proposition 4.7 *Let $K(x, z)$ be a shift-invariant $\mathcal{Y}$-Mercer kernel with signature K_e such that for all $y, y' \in \mathcal{Y}$, $\langle y', K_e(\cdot) y \rangle \in L^1(\mathcal{X}, \mathbf{Haar})$. Assume one can find a pair $(A, \mathbf{Pr}_{\widehat{\mathbf{Haar}}, \rho})$ that satisfies proposition 4.4: for $\mathbf{Pr}_{\widehat{\mathbf{Haar}}, \rho}$-almost all ω, $y, y' \in \mathcal{Y}$, $\langle y, A(\omega) y' \rangle \rho(\omega) = \mathcal{F}\left[\langle y', K_e(\cdot) y \rangle\right](\omega)$. If $(\omega_j)_{j=1}^D$ be a sequence of $D \in \mathbb{N}^*$ i. i. d. realizations drawn from $\mathbf{Pr}_{\widehat{\mathbf{Haar}}, \rho}$ then the operator-valued function $\tilde{K}$ defined as follows, for $(x, z) \in \mathcal{X} \times \mathcal{X}$:*

$$
\tilde{K}(x, z) = \frac{1}{D} \sum_{j=1}^{D} \overline{(x \star z^{-1}, \omega_j)} A(\omega_j)
$$

is an approximation of K.

Proof *Let us first notice that for a given D, $\tilde{K}$ satisfies the properties of a shift-invariant $\mathcal{Y}$-Mercer kernel. Second, from the strong law of large numbers and proposition 4.2 with $\widehat{\mu} = \mathbf{Pr}_{\widehat{\mathbf{Haar}}, \rho}$,*

$$
\frac{1}{D} \sum_{j=1}^{D} \overline{(x \star z^{-1}, \omega_j)} A(\omega_j) \xrightarrow[D \to \infty]{a.\,s.} \mathbf{E}_\rho [\overline{(x \star z^{-1}, \omega_j)} A(\omega)] = K_e(x \star z^{-1})
$$

where the integral converges in the weak operator topology. □

Now, for efficient computations as motivated in the introduction, we are interested in finding an approximated *feature map* instead of a kernel approximation. Indeed, an approximated feature map will allow to build linear models in regression tasks. The following proposition deals with the feature map construction.

Proposition 4.8 *Assume the same conditions as proposition 4.7. Moreover, if one can define $B : \widehat{\mathcal{X}} \to \mathcal{L}(\mathcal{Y}', \mathcal{Y})$, such that for all $y \in \mathcal{Y}$ and all $y' \in \mathcal{Y}'$, $\langle y, B(\cdot)y' \rangle \in L^2(\widehat{\mathcal{X}}, \mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho})$, and for $\mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho}$-almost all ω, and all $u, v \in \mathcal{Y}$,*

$$\begin{aligned}\langle u, B(\omega)B(\omega)^* v\rangle \rho(\omega) &= \langle u, A(\omega)v\rangle \rho(\omega)\\ &= \mathcal{F}\left[\langle v, K_e(\cdot)u\rangle\right](\omega),\end{aligned}$$

then the function $\widetilde{\Phi} : \widehat{\mathcal{X}} \to \mathcal{L}(\mathcal{Y}, \bigoplus_{j=1}^D \mathcal{Y}')$ defined for all $y \in \mathcal{Y}$ as follows:

$$\widetilde{\Phi}(x)y = \frac{1}{\sqrt{D}} \bigoplus_{j=1}^{D} (x, \omega_j) B(\omega_j)^* y, \quad \omega_j \sim \mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho} \text{ i. i. d.},$$

is a feature map of $\tilde{K}$ defined in proposition 4.7 and an approximated feature map for kernel K.

Proof *Let $(\omega_j)_{j=1}^D$ be a sequence of $D \in \mathbb{N}^*$ i. i. d. realizations drawn from $\mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho}$. For all $x, z \in \mathcal{X}$ and all $u, v \in \mathcal{Y}$,*

$$\left\langle \widetilde{\Phi}(x)u, \widetilde{\Phi}(z)v \right\rangle_{\bigoplus_{j=1}^D \mathcal{Y}'} = \frac{1}{D}\left\langle \bigoplus_{j=1}^{D} \left((x, \omega_j) B(\omega_j)^* u\right), \bigoplus_{j=1}^{D} \left((z, \omega_j) B(\omega_j)^* v\right) \right\rangle$$

By definition of the inner product in direct sum of Hilbert spaces,

$$\begin{aligned}&\frac{1}{D}\left\langle \bigoplus_{j=1}^{D} \left((x, \omega_j) B(\omega_j)^* u\right), \bigoplus_{j=1}^{D} \left((z, \omega_j) B(\omega_j)^* v\right) \right\rangle\\ &= \frac{1}{D}\sum_{j=1}^{D} \left\langle u, \overline{(x, \omega_j)} B(\omega_j) (z, \omega_j) B(\omega_j)^* v \right\rangle_{\mathcal{Y}}\\ &= \left\langle u, \left(\frac{1}{D}\sum_{j=1}^{D} \overline{(x \star z^{-1}, \omega_j)} A(\omega_j)\right) v \right\rangle_{\mathcal{Y}},\end{aligned}$$

With similar reasoning about plug-in Monte-Carlo estimator, we get the proof. □

Remark 4.2 *We find a decomposition such that $A(\omega_j) = B(\omega_j)B(\omega_j)^*$ for all $j \in \mathbb{N}_D^*$ either by exhibiting a closed-form or using a numerical decomposition.*

Corollary 4.1 allows us to define algorithm 2 for constructing ORFF from an operator valued kernel.

Algorithm 1: Construction of ORFF from OVK

Input : $K(x,z) = K_e(\delta)$ a $\mathcal{Y}$-shift-invariant Mercer kernel such that $\forall y, y' \in \mathcal{Y}$, $\langle y', K_e(\cdot)y \rangle \in L^1(\mathbb{R}^d, \mathbf{Haar})$ and D the number of features.

Output : A random feature $\widetilde{\Phi}(x)$ such that $\widetilde{\Phi}(x)^* \widetilde{\Phi}(z) \approx K(x,z)$

1 Define the pairing (x, ω) from the LCA group $(\mathcal{X}, \star)$;

2 Find a decomposition $(A, \mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho})$ and B such that

$$B(\omega)B(\omega)^*\rho(\omega) = \mathcal{F}^{-1}\left[K_e\right](\omega);$$

3 Draw D i. i. d. realizations $(\omega_j)_{j=1}^D$ from the probability distribution $\mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho}$;

4 **return** $\begin{cases} \widetilde{\Phi}(x) \in \mathcal{L}(\mathcal{Y}, \widetilde{\mathcal{H}}) & : y \mapsto \frac{1}{\sqrt{D}} \bigoplus_{j=1}^D (x, \omega_j) B(\omega_j)^* y \\ \widetilde{\Phi}(x)^* \in \mathcal{L}(\widetilde{\mathcal{H}}, \mathcal{Y}) & : \theta \mapsto \frac{1}{\sqrt{D}} \sum_{j=1}^D (x, \omega_j) B(\omega_j) \theta_j \end{cases}$;

4.3.2 *From Operator Random Fourier Feature maps to OVK*

It is also interesting to notice that we can go the other way and define from the general form of an Operator-valued Random Fourier Feature, an operator-valeued kernel.

Proposition 4.9 (Operator Random Fourier Feature map). *Let $\mathcal{Y}$ and $\mathcal{Y}'$ be two Hilbert spaces. If one defines an operator-valued function on the dual of a LCA group $\mathcal{X}$, $B : \widehat{\mathcal{X}} \to \mathcal{L}(\mathcal{Y}, \mathcal{Y}')$, and a probability measure $\mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho}$ on $\mathcal{B}(\widehat{\mathcal{X}})$, such that for all $y \in \mathcal{Y}$ and all $y' \in \mathcal{Y}'$, $\langle y, B(\cdot)y' \rangle \in L^2(\widehat{\mathcal{X}}, \mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho})$, assuming $\omega = (\omega_1, \dots, \omega_D)$ is a i. i. d. D-sample drawn from $\mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho}$, then the operator-valued function $\widetilde{\Phi} : \mathcal{X} \to \mathcal{L}(\mathcal{Y}, \bigoplus_{j=1}^D \mathcal{Y}')$ defined for all $x \in \mathcal{X}$ and for all $y \in \mathcal{Y}$ by*

$$\widetilde{\Phi}(x)y = \frac{1}{\sqrt{D}} \bigoplus_{j=1}^D (x, \omega_j) B(\omega_j)^* y, \tag{4.19}$$

is an approximated feature map[9] .

[9] *i. e. it satisfies $\widetilde{\Phi}(x)^* \widetilde{\Phi}(z) \xrightarrow[D\to\infty]{a.\,s.} K(x,z)$ where K is a $\mathcal{Y}$-Mercer OVK*

Proof *Similarly to the proof of corollary 4.1, we get*

$$\frac{1}{D} \sum_{j=1}^D \overline{(x \star z^{-1}, \omega_j)A(\omega_j)} \xrightarrow[D\to\infty]{a.\,s.} \mathbf{E}_\rho[\overline{(x \star z^{-1}, \omega)}A(\omega)]$$

where the integral converges in the weak operator topology. Now from proposition 4.2, $\mathbf{E}_\rho[\overline{(x \star z^{-1}, \omega)}A(\omega)]$ is the image by K_e of $(x \star z^{-1})$, K_e being the kernel signature of some shift-invariant $\mathcal{Y} - $Mercer. □

Proposition 4.10 *Let $\omega \in \widehat{\mathcal{X}}^D$. If for all $y, y' \in \mathcal{Y}$*

$$\langle y', \widetilde{K}_e\left(x \star z^{-1}\right) y\rangle_{\mathcal{Y}} = \langle \widetilde{\Phi}(x)y', \widetilde{\Phi}(z)y\rangle_{\widetilde{\mathcal{H}}}$$
$$= \left\langle y', \frac{1}{D}\sum_{j=1}^{D} \overline{(x \star z^{-1}, \omega_j)} B(\omega_j)B(\omega_j)^* y \right\rangle_{\mathcal{Y}},$$

for all $x, z \in \mathcal{X}$, then $\widetilde{K}$ is a shift-invariant Operator-Valued Kernel.

Proof *Apply proposition 3.4 to $\widetilde{\Phi}$ considering the Hilbert space $\widetilde{\mathcal{H}}$ to show that $\widetilde{K}$ is an OVK. Then proposition 3.7 shows that $\widetilde{K}$ is shift-invariant since $\widetilde{K}(x, z) = \widetilde{K}_e\left(x \star z^{-1}\right)$.* □

We stress out that if $\omega = (\omega_j)_{j=1}^D \sim \mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho}$ i. i. d. is a *random sequence* then

$$\widetilde{K}_e\left(x \star z^{-1}\right) = \widetilde{\Phi}(x)^* \widetilde{\Phi}(z)$$

is not *sensus* stricto an Operator-Valued Kernel since it is a random variable (and $\widetilde{\mathcal{H}}$ is no longer a Hilbert space, since its inner product is a then random variable and not a scalar). However this is not a problem since any realization of the random sequence ω gives birth to a (different) Operator-Valued Kernel, and $\mathbf{E}_{\widehat{\mathbf{Haar}},\rho}\widetilde{K}$ is an OVK. This illustrated by figure 4.1 where we represented the same function for different realization of $\widetilde{K} \approx K$. We generated 250 points equally separated on the segment $(-1; 1)$. We

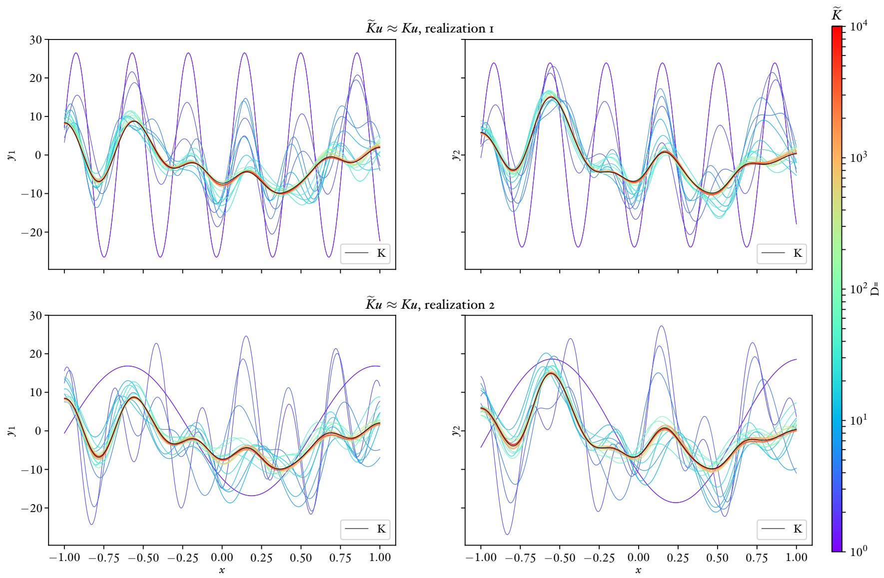

Figure 4.1: Different realizations of a Gaussian kernel approximation. Top row and bottom row correspond to two different realizations of $\widetilde{K}$, which are *different* Operator-Valued Kernel. However when D tends to infinity, the different realizations of $\widetilde{K}$ yield the samel OVK.

computed the Gram Matrix of the Gaussian decomposable kernel

$$K(x,z)_{ij} = \exp\left(-\frac{1}{2(0.1)^2(x_i - x_j)^2}\right)\Gamma, \quad \text{for } i,j \in \mathbb{N}^*_{250}.$$

We computed a reference function (black line) defined as $(y_1, y_2)^\top = f(x_i) = \sum_{j=1}^{250} K(x_i, x_j)u_j$ where $u_j \sim \mathcal{N}(0,1)$ i.i.d.. We took $\Gamma = .5I_2 + .51_2$ such that the outputs y_1 and y_2 share some similarities. Then we computed an approximate kernel matrix $\widetilde{K} \approx K$ for 25 increasing values of D ranging from 1 to 10^4. The two graphs on the top row shows that the more the number of features increase the closer the model $\widetilde{f}(x_i) = \sum_{j=1}^{250} \widetilde{K}(x_i, x_j)u_j$ is to f. The bottom row shows the same experiment but for a different realization of $\widetilde{K}$. When D is small the curves of the bottom and top rows are very dissimilar –and sine wave like– while they both converge to f when D increase.

In the same way we defined an ORFF, we can define an approximate feature operator $\widetilde{W}$ which maps $\widetilde{\mathcal{H}}$ onto $\mathcal{H}_{\widetilde{K}}$, where

$$\widetilde{K}(x,z) = \widetilde{\Phi}(x)^*\widetilde{\Phi}(z), \quad \text{for all } x, z \in \mathcal{X}.$$

Definition 4.1 (Random Fourier feature operator). *Let* $\omega = (\omega_j)_{j=1}^D \in \widehat{\mathcal{X}}^D$ *and let*

$$\widetilde{K}_e = \frac{1}{D}\sum_{j=1}^{D} \overline{(\cdot, \omega_j)} B(\omega_j) B(\omega_j)^*.$$

We call random Fourier feature operator the linear application $\widetilde{W} : \widetilde{\mathcal{H}} \to \mathcal{H}_{\widetilde{K}}$ *defined as*

$$\left(\widetilde{W}\theta\right)(x) := \widetilde{\Phi}(x)^*\theta = \frac{1}{\sqrt{D}}\sum_{j=1}^{D} \overline{(x, \omega_j)} B(\omega_j)\theta_j$$

where $\theta = \bigoplus_{j=1}^D \theta_j \in \widetilde{\mathcal{H}}$. *Then from proposition 3.4,*

$$\left(Ker\ \widetilde{W}\right)^\perp = \overline{\operatorname{span}}\left\{ \widetilde{\Phi}(x)y \;\middle|\; \forall x \in \mathcal{X},\ \forall y \in \mathcal{Y} \right\} \subseteq \widetilde{\mathcal{H}}.$$

The random Fourier feature operator is useful to show the relations between the random Fourier feature map with the functional feature map defined in proposition 4.6. The relationship between the generic feature map (defined for all Operator-Valued Kernel) the functional feature map (defining a shift-invariant $\mathcal{Y}$-Mercer Operator-Valued Kernel) and the random Fourier feature map is presented in figure 4.2.

Proposition 4.11 *For any* $g \in \mathcal{H} = L^2(\widehat{\mathcal{X}}, \mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho}; \mathcal{Y}')$, *let*

$$\theta := \frac{1}{\sqrt{D}}\bigoplus_{j=1}^{D} g(\omega_j), \quad \omega_j \sim \mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho} \textit{ i.i.d.}.$$

Then

1. $\left(\widetilde{W}\theta\right)(x) = \widetilde{\Phi}(x)^*\theta \xrightarrow[D\to\infty]{a.s.} \Phi_x^* g = (Wg)(x)$,
2. $\|\theta\|^2_{\widetilde{\mathcal{H}}} \xrightarrow[D\to\infty]{a.s.} \|g\|^2_{\mathcal{H}}$,

In other words $\widetilde{W}S_D g = \Phi_x^* S_D^* S_D g = \widetilde{W}\widetilde{g} \xrightarrow[D\to\infty]{\text{a.s.}} Wg$ and $\|S_D g\|_{\widetilde{\mathcal{H}}} \xrightarrow[D\to\infty]{\text{a.s.}} \|g\|_{\mathcal{H}}$. Notice that we can identify $\widetilde{W}$ with WS_D^* since $(\widetilde{W}\theta)(x) = \widetilde{\Phi}(x)^*\theta = (S_D\Phi_x)^*\theta = (\Phi_x^* S_D^*\theta) := (WS_D^*\theta)(x)$.

Proof (of proposition 4.11 item 1) *Since $(\omega_j)_{j=1}^D$ are i. i. d. random vectors, for all $y \in \mathcal{Y}$ and for all $y' \in \mathcal{Y}'$, $\langle y, B(\cdot)y'\rangle \in L^2(\widehat{\mathcal{X}}, \mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho})$ and $g \in L^2(\widehat{\mathcal{X}}, \mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho}; \mathcal{Y}')$, from the strong law of large numbers*

$$
\begin{aligned}
(\widetilde{W}\theta)(x) &= \widetilde{\Phi}(x)^*\theta \\
&= \frac{1}{D}\sum_{j=1}^D \overline{(x,\omega_j)}B(\omega_j)g(\omega_j), \qquad \omega_j \sim \mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho} \text{ i. i. d.} \\
&\xrightarrow[D\to\infty]{a.s.} \int_{\widehat{\mathcal{X}}} \overline{(x,\omega)}B(\omega)g(\omega)d\mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho}(\omega) \\
&= (Wg)(x) := \Phi_x^* g. \qquad \square
\end{aligned}
$$

Proof (of proposition 4.11 item 2) *Again, since $(\omega_j)_{j-1}^D$ are i. i. d. random vectors and*

$$
g \in L^2(\widehat{\mathcal{X}}, \mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho}; \mathcal{Y}'),
$$

from the strong law of large numbers

$$
\begin{aligned}
\|\theta\|^2_{\widetilde{\mathcal{H}}} &= \frac{1}{D}\sum_{j=1}^D \|g(\omega_j)\|^2_{\mathcal{Y}'}, \qquad \omega_j \sim \mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho} \text{ i. i. d.} \\
&\xrightarrow[D\to\infty]{a.s.} \int_{\widehat{\mathcal{X}}} \|g(\omega)\|^2_{\mathcal{Y}'} d\mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho}(\omega) \\
&= \|g\|^2_{L^2\left(\widehat{\mathcal{X}}, \mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho}; \mathcal{Y}'\right)}. \qquad \square
\end{aligned}
$$

We write $\widetilde{\Phi}(x)^*\widetilde{\Phi}(x) \approx K(x,z)$ when $\widetilde{\Phi}(x)^*\widetilde{\Phi}(x) \xrightarrow{\text{a.s.}} K(x,z)$ in the weak operator topology when D tends to infinity. With mild abuse of notation we say that $\widetilde{\Phi}(x)$ is an approximate feature map of Φ_x i. e. $\widetilde{\Phi}(x) \approx \Phi_x$, when for all $y', y \in \mathcal{Y}$,

$$
\begin{aligned}
\langle y, K(x,z)y'\rangle_{\mathcal{Y}} &= \langle \Phi_x y, \Phi_z y'\rangle_{L^2(\widehat{\mathcal{X}}, \mathbf{Pr}_{\widehat{\mathbf{Haar}},\subset}; \mathcal{Y}')} \\
&\approx \langle \widetilde{\Phi}(x)y, \widetilde{\Phi}(x)y'\rangle_{\widetilde{\mathcal{H}}} := \langle y, \widetilde{K}(x,z)y'\rangle_{\mathcal{Y}}
\end{aligned}
$$

where Φ_x is defined in the sense of proposition 4.6. Then corollary 4.1 exhibit a construction of an ORFF directly from an OVK.

Corollary 4.1 *If $K(x,z)$ is a shift-invariant $\mathcal{Y}$-Mercer kernel such that for all $y, y' \in \mathcal{Y}$, $\langle y', K_e(\cdot)y\rangle \in L^1(\mathcal{X}, \mathbf{Haar})$. Then*

$$\widetilde{\Phi}(x)y = \frac{1}{\sqrt{D}}\bigoplus_{j=1}^{D}(x,\omega_j)B(\omega_j)^* y, \quad \omega_j \sim \mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho} \text{ i. i. d.},$$

where $\langle y, B(\omega)B(\omega)^* y'\rangle \rho(\omega) = \mathcal{F}\left[\langle y', K_e(\cdot)y\rangle\right](\omega)$, is an approximated feature map of K.

Proof *Find $(A, \mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho})$ from proposition 4.4. Then find a decomposition of*

$$A(\omega) = B(\omega)B(\omega)^*$$

for $\mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho}$-almost all ω and apply proposition 4.9. □

Remark 4.3 *We find a decomposition such that $A(\omega_j) = B(\omega_j)B(\omega_j)^*$ for all $j \in \mathbb{N}^*_D$ either by exhibiting a closed-form or using a numerical decomposition.*

Corollary 4.1 allows us to define algorithm 2 for constructing ORFF from an operator valued kernel.

Algorithm 2: Construction of ORFF from OVK

Input : $K(x,z) = K_e(\delta)$ a $\mathcal{Y}$-shift-invariant Mercer kernel such that $\forall y, y' \in \mathcal{Y}$, $\langle y', K_e(\cdot)y\rangle \in L^1(\mathbb{R}^d, \mathbf{Haar})$ and D the number of features.

Output : A random feature $\widetilde{\Phi}(x)$ such that $\widetilde{\Phi}(x)^*\widetilde{\Phi}(z) \approx K(x,z)$

1 Define the pairing (x, ω) from the LCA group $(\mathcal{X}, \star)$;

2 Find a decomposition $(B(\omega), \mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho})$ such that

$$B(\omega)B(\omega)^*\rho(\omega) = \mathcal{F}^{-1}\left[K_e\right](\omega);$$

3 Draw D random vectors $(\omega_j)_{j=1}^D$ i. i. d. from the probability law $\mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho}$;

4 **return** $\begin{cases} \widetilde{\Phi}(x) \in \mathcal{L}(\mathcal{Y}, \widetilde{\mathcal{H}}) & : y \mapsto \frac{1}{\sqrt{D}}\bigoplus_{j=1}^{D}(x,\omega_j)B(\omega_j)^* y \\ \widetilde{\Phi}(x)^* \in \mathcal{L}(\widetilde{\mathcal{H}}, \mathcal{Y}) & : \theta \mapsto \frac{1}{\sqrt{D}}\sum_{j=1}^{D}(x,\omega_j)B(\omega_j)\theta_j \end{cases}$;

4.3.3 *Examples of Operator Random Fourier Feature maps*

We now give two examples of operator-valued random Fourier feature map. First we introduce the general form of an approximated feature map for a matrix-valued kernel on the additive group $(\mathbb{R}^d, +)$.

Example 4.1 **(Matrix-valued kernel on the additive group).** *In the following let $K(x,z) = K_0(x-z)$ be a $\mathcal{Y}$-Mercer matrix-valued kernel on $\mathcal{X} =$*

$$\Phi_x^*\Phi_z = K(x,z) \quad = \quad K_e\left(x\star z^{-1}\right) \quad \approx \quad \widetilde{K}_e\left(x\star z^{-1}\right) = \widetilde{\Phi}(x)^*\widetilde{\Phi}(x)$$

$$\begin{tikzcd}
\Phi_x \in \mathcal{L}(\mathcal{Y};\mathcal{H}) \arrow[r, "\Phi_x^* g"] & \mathcal{Y} \\
x \in \mathcal{X} \arrow[u, "\Phi"] \arrow[ur, "f"'] &
\end{tikzcd}
\qquad
\begin{tikzcd}
\Phi_x \in \mathcal{L}\left(\mathcal{Y};L^2\left(\widehat{\mathcal{X}},\mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho};\mathcal{Y}'\right)\right) \arrow[r, "\Phi_x^* g"] & \mathcal{Y} \\
x \in \mathcal{X} \arrow[u, "\Phi"] \arrow[ur, "f"'] &
\end{tikzcd}
\qquad
\begin{tikzcd}
\widetilde{\Phi}(x) \in \mathcal{L}\left(\mathcal{Y};\widetilde{\mathcal{H}}\right) \arrow[r, "\widetilde{\Phi}(x)^*\theta"] & \mathcal{Y} \\
x \in \mathcal{X} \arrow[u, "\widetilde{\Phi}"] \arrow[ur, "\widetilde{f}"'] &
\end{tikzcd}$$

Fourier,
$\Phi_x(\omega)y = (x,\omega)B(\omega)^*y$.

Monte-Carlo,
$\widetilde{\Phi}(x)y \ = \ \frac{1}{\sqrt{D}}\bigoplus_{j=1}^D(\Phi_x y)(\omega_j)$, $\omega_j \ \sim \ \mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho}$
i. i. d..

Figure 4.2: Relationships between feature-maps. For any realization of $\omega_j \sim \mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho}$ i. i. d., $\widetilde{\mathcal{H}} = \bigoplus_{j=1}^D \mathcal{Y}'$.

$\mathbb{R}^d$, *invariant w. r. t. the group operation* $+$. *Then the function* $\widetilde{\Phi}$ *defined as follow is an Operator-valued Random Fourier Feature of* K_0.

$$\widetilde{\Phi}(x)y = \frac{1}{\sqrt{D}} \bigoplus_{j=1}^{D} \begin{pmatrix} \cos \langle x, \omega_j \rangle_2 B(\omega_j)^* y \\ \sin \langle x, \omega_j \rangle_2 B(\omega_j)^* y \end{pmatrix}, \quad \omega_j \sim \mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho} \text{ i. i. d.}.$$

for all $y \in \mathcal{Y}$.

Proof *The (Pontryagin) dual of* $\mathcal{X} = \mathbb{R}^d$ *is* $\widehat{\mathcal{X}} \cong \mathbb{R}^d$, *and the duality pairing is* $(x - z, \omega) = \exp(i\langle x - z, \omega \rangle_2)$. *The kernel approximation yields*

$$\begin{aligned}
\tilde{K}(x, z) &= \widetilde{\Phi}(x)^* \widetilde{\Phi}(z) \\
&= \frac{1}{D} \sum_{j=1}^{D} \begin{pmatrix} \cos \langle x, \omega_j \rangle_2 & \sin \langle x, \omega_j \rangle_2 \end{pmatrix} \begin{pmatrix} \cos \langle z, \omega_j \rangle_2 \\ \sin \langle z, \omega_j \rangle_2 \end{pmatrix} A(\omega_j) \\
&= \frac{1}{D} \sum_{j=1}^{D} \cos \langle x - z, \omega_j \rangle_2 A(\omega_j) \\
&\xrightarrow[D \to \infty]{a.\,s.} \mathbf{E}_\rho \left[\cos \langle x - z, \omega \rangle_2 A(\omega) \right]
\end{aligned}$$

in the weak operator topology. Since for all $x \in \mathcal{X}$, $\sin\langle x, \cdot \rangle_2$ *is an odd function and* $A(\cdot)\rho(\cdot)$ *is even,*

$$\mathbf{E}_\rho \left[\cos \langle x - z, \omega \rangle_2 A(\omega) \right] = \mathbf{E}_\rho \left[\exp(-i\langle x - z, \omega \rangle_2) A(\omega) \right] = K(x, z).$$

Hence $\tilde{K}(x, z) \xrightarrow[D \to \infty]{a.\,s.} K(x, z)$. □

In particular we deduce the following features maps for the kernels proposed in subsection 4.2.2.

- For the decomposable gaussian kernel $K_0^{dec,gauss}(\delta) = k_0^{gauss}(\delta)\Gamma$ for all $\delta \in \mathbb{R}^d$, let $BB^* = \Gamma$. A bounded –and unbounded– ORFF map is

$$\begin{aligned}
\widetilde{\Phi}(x)y &= \frac{1}{\sqrt{D}} \bigoplus_{j=1}^{D} \begin{pmatrix} \cos \langle x, \omega_j \rangle_2 B^* y \\ \sin \langle x, \omega_j \rangle_2 B^* y \end{pmatrix} \\
&= (\widetilde{\varphi}(x) \otimes B^*) y,
\end{aligned}$$

  where $\omega_j \sim \mathbf{Pr}_{\mathcal{N}(0, \sigma^{-2} I_d)}$ i. i. d. and $\widetilde{\varphi}(x) = \frac{1}{\sqrt{D}} \bigoplus_{j=1}^{D} \begin{pmatrix} \cos \langle x, \omega_j \rangle_2 \\ \sin \langle x, \omega_j \rangle_2 \end{pmatrix}$ is a scalar RFF map [121].

- For the curl-free gaussian kernel, $K_0^{curl,gauss} = -\nabla\nabla^\intercal k_0^{gauss}$ an unbounded ORFF map is

$$\widetilde{\Phi}(x)y = \frac{1}{\sqrt{D}} \bigoplus_{j=1}^{D} \begin{pmatrix} \cos \langle x, \omega_j \rangle_2 \omega_j^\intercal y \\ \sin \langle x, \omega_j \rangle_2 \omega_j^\intercal y \end{pmatrix}, \tag{4.20}$$

$\omega_j \sim \mathbf{Pr}_{\mathcal{N}(0,\sigma^{-2}I_d)}$ i. i. d. and a bounded ORFF map is

$$\widetilde{\Phi}(x)y = \frac{1}{\sqrt{D}}\bigoplus_{j=1}^{D}\begin{pmatrix}\cos\langle x,\omega_j\rangle_2\frac{\omega_j^\mathsf{T}}{\|\omega_j\|}y\\ \sin\langle x,\omega_j\rangle_2\frac{\omega_j^\mathsf{T}}{\|\omega_j\|}y\end{pmatrix},\quad \omega_j \sim \mathbf{Pr}_\rho \text{ i. i. d..}$$

where $\rho(\omega) = \frac{\sigma^2\|\omega\|^2}{d}\mathcal{N}(0,\sigma^{-2}I_d)(\omega)$ for all $\omega \in \mathbb{R}^d$.

- For the divergence-free gaussian kernel $K_0^{div,gauss}(x,z) = (\nabla\nabla^\mathsf{T} - \Delta I_d)k_0^{gauss}(x,z)$ an unbounded ORFF map is

$$\widetilde{\Phi}(x)y = \frac{1}{\sqrt{D}}\bigoplus_{j=1}^{D}\begin{pmatrix}\cos\langle x,\omega_j\rangle_2 B(\omega_j)^\mathsf{T}y\\ \sin\langle x,\omega_j\rangle_2 B(\omega_j)^\mathsf{T}y\end{pmatrix} \tag{4.21}$$

where $\omega_j \sim \mathbf{Pr}_\rho$ i. i. d. and $B(\omega) = \left(\|\omega\|I_d - \omega\omega^\mathsf{T}\right)$ and $\rho = \mathcal{N}(0,\sigma^{-2}I_d)$ for all $\omega \in \mathbb{R}^d$. A bounded ORFF map is

$$\widetilde{\Phi}(x)y = \frac{1}{\sqrt{D}}\bigoplus_{j=1}^{D}\begin{pmatrix}\cos\langle x,\omega_j\rangle_2 B(\omega_j)^\mathsf{T}y\\ \sin\langle x,\omega_j\rangle_2 B(\omega_j)^\mathsf{T}y\end{pmatrix},\quad \omega_j \sim \mathbf{Pr}_\rho \text{ i. i. d.,}$$

where $B(\omega) = \left(I_d - \frac{\omega\omega^\mathsf{T}}{\|\omega\|^2}\right)$ and $\rho(\omega) = \frac{\sigma^2\|\omega\|^2}{d}\mathcal{N}(0,\sigma^{-2}I_d)$ for all $\omega \in \mathbb{R}^d$.

The second example extends scalar-valued Random Fourier Features on the skewed multiplicative group –described in subsection 3.2.4 and subsection 4.2.2.2– to the operator-valued case.

Example 4.2 (Matrix-valued kernel on the skewed multiplicative group). *In the following, $K(x,z) = K_{1-c}(x \odot z^{-1})$ is a $\mathcal{Y}$-Mercer matrix-valued kernel on $\mathcal{X} = (-c;+\infty)^d$ invariant w. r. t. the group operation[10] $\odot$. Then the function $\widetilde{\Phi}$ defined as follow is an Operator-valued Random Fourier Feature of K_{1-c}.*

[10] *The group operation $\odot$ is defined in subsection 4.2.2.2.*

$$\widetilde{\Phi}(x)y = \frac{1}{\sqrt{D}}\bigoplus_{j=1}^{D}\begin{pmatrix}\cos\langle\log(x+c),\omega_j\rangle_2 B(\omega_j)^* y\\ \sin\langle\log(x+c),\omega_j\rangle_2 B(\omega_j)^* y\end{pmatrix},$$

$\omega_j \sim \mathbf{Pr}_{\widehat{\text{Haar}},\rho}$ *i. i. d., for all $y \in \mathcal{Y}$.*

Proof *The dual of $\mathcal{X} = (-c;+\infty)^d$ is $\widehat{\mathcal{X}} \cong \mathbb{R}^d$, and the duality pairing is $(x \odot z^{-1},\omega) = \exp(i\langle\log(x\odot z^{-1}+c),\omega\rangle_2)$. Following the proof of example 4.1, we have*

$$\widetilde{K}(x,z) = \frac{1}{D}\sum_{j=1}^{D}\cos\left\langle\log\left(\frac{x+c}{z+c}\right),\omega_j\right\rangle_2 A(\omega_j).$$

which converges almost surely to

$$\mathbf{E}_\rho\left[\exp\left(-i\left\langle\log(x\odot z^{-1}+c)\right\rangle_2\right)A(\omega)\right] = \mathbf{E}_\rho[\overline{(x\odot z^{-1},\omega)}A(\omega)] = K(x,z)$$

when D tends to infinity, in the weak operator topology. □

- For the skewed-χ^2 decomposable kernel defined as $K_{1-c}^{dec,skewed}(\delta) = k_{1-c}^{skewed}(\delta)\Gamma$ for all $\delta \in \mathcal{X}$, let $BB^* = \Gamma$. A bounded –and unbounded– ORFF map is

$$\begin{aligned}\tilde{\Phi}(x)y &= \frac{1}{\sqrt{D}}\bigoplus_{j=1}^{D}\begin{pmatrix}\cos\langle\log(x+c),\omega_j\rangle_2 B^*y\\ \sin\langle\log(x+c),\omega_j\rangle_2 B^*y\end{pmatrix}, \quad \omega_j \sim \mathbf{Pr}_\rho \text{ i. i. d.}\\ &= (\tilde{\varphi}(x)\otimes B^*)y,\end{aligned}$$

where $\rho = \mathcal{S}(0, 2^{-1})$ and $\tilde{\varphi}(x) = \frac{1}{\sqrt{D}}\bigoplus_{j=1}^{D}\begin{pmatrix}\cos\langle\log(x+c),\omega_j\rangle_2\\ \sin\langle\log(x+c),\omega_j\rangle_2\end{pmatrix}$ is a scalar RFF map [83].

4.3.4 *Regularization property*

We have shown so far that it is always possible to construct a feature map that allows to approximate a shift-invariant $\mathcal{Y}$-Mercer kernel. However we could also propose a construction of such map by studying the regularization induced with respect to the Fourier Transform of a target function $f \in \mathcal{H}_K$. In other words, what is the norm in $L^2(\widehat{\mathcal{X}}, \widehat{\mathbf{Haar}}; \mathcal{Y}')$ induced by $\|\cdot\|_K$?

Proposition 4.12 *Let K be a shift-invariant $\mathcal{Y}$-Mercer Kernel such that for all y, y' in $\mathcal{Y}$, $\langle y', K_e(\cdot)y\rangle_{\mathcal{Y}} \in L^1(\mathcal{X}, \mathbf{Haar})$. Then for all $f \in \mathcal{H}_K$*

$$\|f\|_K^2 = \int_{\widehat{\mathcal{X}}} \frac{\left\langle \mathcal{F}\left[f\right](\omega), A(\omega)^\dagger \mathcal{F}\left[f\right](\omega)\right\rangle_{\mathcal{Y}}}{\rho(\omega)} d\widehat{\mathbf{Haar}}(\omega). \tag{4.22}$$

where $\langle y', A(\omega)y\rangle\rho(\omega) := \mathcal{F}\left[\langle y', K_e(\cdot)y\rangle\right](\omega)$.

Proof *We first show how the Fourier Transform relates to the feature operator. Since $\mathcal{H}_K$ is embed into $\mathcal{H} = L^2(\widehat{\mathcal{X}}, \mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho}; \mathcal{Y}')$ by means of the feature operator W, we have for all $f \in \mathcal{H}_k$, for all $f \in \mathcal{H}$ and for all $x \in \mathcal{X}$*

$$\begin{aligned}\mathcal{F}\left[\mathcal{F}^{-1}\left[f\right]\right](x) &= \int_{\widehat{\mathcal{X}}} \overline{(x,\omega)}\mathcal{F}^{-1}\left[f\right](\omega) d\widehat{\mathbf{Haar}}(\omega) = f(x)\\ (Wg)(x) &= \int_{\widehat{\mathcal{X}}} \overline{(x,\omega)}\rho(\omega)B(\omega)g(\omega)d\widehat{\mathbf{Haar}}(\omega) = f(x).\end{aligned}$$

By injectivity of the Fourier Transform, $\mathcal{F}^{-1}\left[f\right](\omega) = \rho(\omega)B(\omega)g(\omega)$. From proposition 3.4 we have

$$\begin{aligned}\|f\|_K^2 &= \inf\left\{ \|g\|_{\mathcal{H}}^2 \,\middle|\, \forall g \in \mathcal{H},\ \ Wg = f\right\}\\ &= \inf\left\{ \int_{\widehat{\mathcal{X}}} \|g(\omega)\|_{\mathcal{Y}'}^2 d\mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho}(\omega) \,\middle|\, \forall g \in \mathcal{H},\ \ \mathcal{F}^{-1}\left[f\right] = \rho(\cdot)B(\cdot)g(\cdot)\right\}.\end{aligned}$$

The pseudo inverse of the operator $B(\omega)$ – noted $B(\omega)^\dagger$ – is the unique solution of the system $\mathcal{F}^{-1}[f](\omega) = \rho(\omega)B(\omega)g(\omega)$ w.r.t. $g(\omega)$ with minimal norm[2]*. Eventually,*

$$\|f\|_K^2 = \int_{\widehat{\mathcal{X}}} \frac{\left\|B(\omega)^\dagger \mathcal{F}^{-1}[f](\omega)\right\|_{\mathcal{Y}}^2}{\rho(\omega)^2} d\mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho}(\omega)$$

Using the fact that $\mathcal{F}^{-1}[\cdot] = \mathcal{F}\mathcal{R}[\cdot]$ and $\mathcal{F}^2[\cdot] = \mathcal{R}[\cdot]$,

$$\begin{aligned}\|f\|_K^2 &= \int_{\widehat{\mathcal{X}}} \frac{\left\|\mathcal{R}\left[B(\cdot)^\dagger \rho(\cdot)\right](\omega)\mathcal{F}[f](\omega)\right\|_{\mathcal{Y}}^2}{\rho(\omega)^2} d\widehat{\mathbf{Haar}}(\omega) \\ &= \int_{\widehat{\mathcal{X}}} \frac{\left\|B(\omega)^\dagger \rho(\omega)\mathcal{F}[f](\omega)\right\|_{\mathcal{Y}}^2}{\rho(\omega)^2} d\widehat{\mathbf{Haar}}(\omega) \\ &= \int_{\widehat{\mathcal{X}}} \frac{\langle B(\omega)^\dagger \mathcal{F}[f](\omega), B(\omega)^\dagger \mathcal{F}[f](\omega)\rangle_{\mathcal{Y}}}{\rho(\omega)} d\widehat{\mathbf{Haar}}(\omega) \\ &= \int_{\widehat{\mathcal{X}}} \frac{\langle \mathcal{F}[f](\omega), A(\omega)^\dagger \mathcal{F}[f](\omega)\rangle_{\mathcal{Y}}}{\rho(\omega)} d\widehat{\mathbf{Haar}}(\omega)\end{aligned}$$

Note that if $K(x,z) = k(x,z)$ is a scalar kernel then for all ω in $\widehat{\mathcal{X}}$, $A(\omega) = 1$. Therefore we recover the well known result for kernels that is for any $f \in \mathcal{H}_k$ we have $\|f\|_k = \int_{\widehat{\mathcal{X}}} \mathcal{F}[k_e](\omega)^{-1}\mathcal{F}[f](\omega)^2 d\widehat{\mathbf{Haar}}(\omega)$ [143, 160, 170]. Eventually from this last equation we also recover proposition 3.8 for decomposable kernels. If $A(\omega) = \Gamma \in \mathcal{L}_+(\mathbb{R}^p)$,

$$\|f\|_K = \sum_{i,j=1}^{p} \left(\Gamma^\dagger\right)_{ij} \langle f_i, f_j\rangle_k \tag{4.23}$$

We also note that the regularization property in $\mathcal{H}_K$ does not depends (as expected) on the decomposition of $A(\omega)$ into $B(\omega)B(\omega)^*$. Therefore the decomposition should be chosen such that it optimizes the computation cost. For instance if $A(\omega) \in \mathcal{L}(\mathbb{R}^p)$ has rank r, one could find an operator $B(\omega) \in \mathcal{L}(\mathbb{R}^p, \mathbb{R}^r)$ such that $A(\omega) = B(\omega)B(\omega)^*$. Moreover, in light of equation 4.22 the regularization property of the kernel with respect to the Fourier Transform, it is also possible to define an approximate feature map of an Operator-Valued Kernel from its regularization properties in the VV-RKHS as proposed in algorithm 3.

4.4 CONCLUSIONS

---

2 Note that since $B(\omega)$ is bounded the pseudo inverse of $B(\omega)$ is well defined for $\widehat{\mathbf{Haar}}$-almost all ω. However if $B(\omega)$ is infinite dimensional, the pseudo inverse is continuous if and only if $B(\omega)$ has closed range in $\mathcal{Y}'$. This is always true if $B(\omega)$ is surjective, via the open mapping theorem.

Algorithm 3: Construction of ORFF

Input :

- The pairing (x, ω) of the LCA group $(\mathcal{X}, \star)$.
- A probability measure $\mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho}$ with density ρ w. r. t. the haar measure $\widehat{\mathbf{Haar}}$ on $\widehat{\mathcal{X}}$.
- An operator-valued function $B : \widehat{\mathcal{X}} \to \mathcal{L}(\mathcal{Y}, \mathcal{Y}')$ such that for all y $y' \in \mathcal{Y}$, $\langle y', B(\cdot)B(\cdot)^* y\rangle \in L^1(\widehat{\mathcal{X}}, \mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho})$.
- D the number of features.

Output : A random feature $\widetilde{\Phi}(x)$ such that $\widetilde{\Phi}(x)^* \widetilde{\Phi}(z) \approx K(x, z)$.

1 Draw D random vectors $(\omega_j)_{j=1}^D$ i. i. d. from the probability law $\mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho}$;

2 **return** $\begin{cases} \widetilde{\Phi}(x) \in \mathcal{L}(\mathcal{Y}, \widetilde{\mathcal{H}}) & : y \mapsto \frac{1}{\sqrt{D}} \bigoplus_{j=1}^D (x, \omega_j) B(\omega_j)^* y \\ \widetilde{\Phi}(x)^* \in \mathcal{L}(\widetilde{\mathcal{H}}, \mathcal{Y}) & : \theta \mapsto \frac{1}{\sqrt{D}} \sum_{j=1}^D (x, \omega_j) B(\omega_j) \theta_j \end{cases}$;

5

BOUNDING THE ERROR OF THE ORFF APPROXIMATION

In this chapter we refine the bound on the OVK approximation with ORFF we first proposed in [24] and presented in [23]. It generalizes the proof technique of Rahimi and Recht [121] to OVK on LCA groups thanks to the recent results of Koltchinskii et al. [77], Minsker [104], Sutherland and Schneider [148], and Tropp et al. [155]. As a Bernstein bound it depends on the variance of the estimator for which we derive an "upper bound".

Contents

5.1 CONVEGENCE WITH HIGH PROBABILITY OF THE ORFF ESTIMATOR

We are now interested in a non-asymptotic analysis of the ORFF approximation of shift-invariant $\mathcal{Y}$-Mercer kernels on LCA group $\mathcal{X}$ endowed with the operation group $\star$ where $\mathcal{X}$ is a Banach space (The more general case where $\mathcal{X}$ is a Polish space is discussed in the appendix appendix A.1). For a given D, we study how close is the approximation $\tilde{K}(x, z) = \tilde{\Phi}(x)^* \tilde{\Phi}(z)$ to the target kernel $K(x, z)$ for any x, z in $\mathcal{X}$.

If $A \in \mathcal{L}_+(\mathcal{Y})$ we denote $\|A\|_{\mathcal{Y},\mathcal{Y}}$ its operator norm, which amounts to the square root of the largest eigenvalue of A when $\mathcal{Y} = \mathbb{R}^p$ is finite dimensional. For x and z in some non-empty compact $\mathcal{C} \subset \mathbb{R}^d$, we consider: $F(x \star z^{-1}) = \tilde{K}(x, z) - K(x, z)$ and study how the uniform norm

$$\left\|\tilde{K} - K\right\|_{\mathcal{C}\times\mathcal{C}} = \sup_{(x,z)\in\mathcal{C}\times\mathcal{C}} \left\|\tilde{K}(x, z) - K(x, z)\right\|_{\mathcal{Y},\mathcal{Y}} \tag{5.1}$$

behaves according to D. All along this document we denote $\delta = x \star z^{-1}$ for all x and $z \in \mathcal{X}$. Figure 5.1 empirically shows convergence of three different OVK approximations for x, z sampled from the compact $[-1, 1]^4$ and using an increasing number of sample points D. The log-log plot shows that all three kernels have the same convergence rate, up to a multiplicative factor.

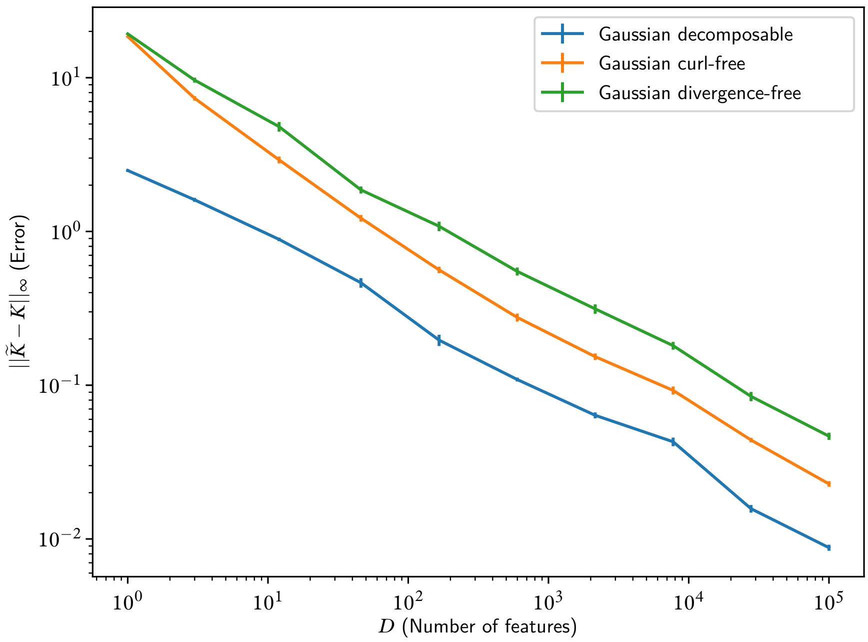

Figure 5.1: Error reconstructing the target operator-valued kernel K with ORFF approximation $\tilde{K}$ for the decomposable, curl-free and divergence-free kernel.

In order to bound the error with high probability, we turn to concentration inequalities devoted to random matrices [20]. The concentration phenomenon can be summarized in the following sentence of Ledoux [81]. "A random variable that depends (in a smooth way) on the influence of many random variables (but not too much on any of them) is essentially constant".

A typical application is the study of the deviation of the empirical mean of independent identically distributed random variables to their expectation. This means that given an error ϵ between the kernel approximation $\widetilde{K}$ and the true kernel K, if we are given enough samples to construct $\widetilde{K}$, the probability of measuring an error greater than ϵ is essentially zero (it drops at an exponential rate with respect to the number of samples D). To measure the error between the kernel approximation and the true kernel at a given point many metrics are possible. e.g. any matrix norm such as the Hilbert-Schmidt norm, trace norm, the operator norm or Schatten norms. In this work we focus on measuring the error in terms of operator norm. For all $x, z \in \mathcal{X}$ we look for a bound on

$$\begin{aligned}
&\mathbf{Pr}_\rho \left\{ (\omega_j)_{j=1}^D \;\middle|\; \left\|\widetilde{K}(x,z) - K(x,z)\right\|_{\mathcal{Y},\mathcal{Y}} \geq \epsilon \right\} \\
&= \mathbf{Pr}_\rho \left\{ (\omega_j)_{j=1}^D \;\middle|\; \sup_{0 \neq y \in \mathcal{Y}} \frac{\left\|(\widetilde{K}(x,z) - K(x,z))y\right\|_{\mathcal{Y}}}{\|y\|_{\mathcal{Y}}} \geq \epsilon \right\}
\end{aligned}$$

In other words, given any vector $y \in \mathcal{Y}$ we study how the residual operator $\widetilde{K} - K$ is able to send y to zero. We believe that this way of measuring the "error" to be more intuitive. Moreover, on contrary to an error measure with the Hilbert-Schmidt norm, the operator norm error does not grows linearly with the dimension of the output space as the Hilbert-Schmidt norm does. On the other hand the Hilbert-schmidt norm makes the studied random variables Hilbert space valued, for which it is much easier to derive concentration inequalities [107, 118, 142]. Note that in the scalar case ($A(\omega) = 1$) the Hilbert-Schmidt norm error and the operator norm are the same and measure the deviation between $\widetilde{K}$ and K as the absolute value of their difference.

A raw concentration inequality of the kernel estimator gives the error on one point. If one is interresting in bounding the maximum error over N points, applying a union bound on all the point would yoeld a bound that grows linearly with N. This would suggest that when the number of points increase, even if all of them are concentrated in a small subset of $\mathcal{X}$, we should draw increasingly more features to have an error below ϵ with high probability. However if we restrict ourselves to study the error on a compact subset of $\mathcal{X}$ (and in practice data points lies often in a closed bounded subset of $\mathbb{R}^d$), we can cover this compact subset by a finite number of closed balls and apply the concentration inequality and the union bound only on the center of each ball. Then if the function $\left\|\widetilde{K}_e - K_e\right\|$ is smooth enough on each ball (i. e. Lipschitz) we can guarantee with high probability that the error between the centers of the balls will not be too high. Eventually we obtain a bound in the worst case scenarion on all the points in a subset $\mathcal{C}$ of $\mathcal{X}$. This bound depends on the covering number $\mathcal{N}(\mathcal{C}, r)$ of $\mathcal{X}$ with ball of radius r. When $\mathcal{X}$ is a Banach space, the covering number is proportional to the diameter of the diameter of $\mathcal{C} \subseteq \mathcal{X}$.

Prior to the presentation of general results, we briefly recall the uniform convergence of RFF approximation for a scalar shift invariant kernel on the additive LCA group $\mathbb{R}^d$ and introduce a direct corollary about decomposable shift-invariant OVK on the LCA group $(\mathbb{R}^d, +)$.

5.1.1 *Random Fourier Features in the scalar case and decomposable OVK*

Rahimi and Recht [121] proved the uniform convergence of Random Fourier Feature (RFF) approximation for a scalar shift-invariant kernel on the LCA group $\mathbb{R}^d$ endowed with the group operation $\star = +$. In the case of the shift-invariant decomposable OVK, an upper bound on the error can be obtained as a direct consequence of the result in the scalar case obtained by Rahimi and Recht [121] and other authors [145, 148].

Theorem 5.1 (Uniform error bound for RFF, Rahimi and Recht [121]). *Let $\mathcal{C}$ be a compact of subset of $\mathbb{R}^d$ of diameter $|\mathcal{C}|$. Let k be a shift invariant kernel, differentiable with a bounded second derivative and $\mathbf{Pr}_\rho$ its normalized Inverse Fourier Transform such that it defines a probability measure. Let*

$$\widetilde{k} = \sum_{j=1}^{D} \cos\left\langle \cdot, \omega_j \right\rangle \approx k(x,z) \;\; \textit{and} \;\; \sigma^2 = \mathbf{E}_\rho \|\omega\|_2^2.$$

Then we have

$$\mathbf{Pr}_\rho \left\{ (\omega_j)_{j=1}^D \;\middle|\; \|\tilde{k} - k\|_{\mathcal{C}\times\mathcal{C}} \geq \epsilon \right\} \leq 2^8 \left(\frac{\sigma|\mathcal{C}|}{\epsilon} \right)^2 \exp\left(-\frac{\epsilon^2 D}{4(d+2)} \right)$$

From theorem 5.1, we can deduce the following corollary about the uniform convergence of the ORFF approximation of the decomposable kernel. We recall that for a given pair x, z in $\mathcal{C}$, $\widetilde{K}(x,z) = \widetilde{\Phi}(x)^* \widetilde{\Phi}(z) = \Gamma \tilde{k}(x,z)$ and $K_0(x-z) = \Gamma \mathbf{E}_{\widehat{\mathbf{Haar}},\rho}[\tilde{k}(x,z)]$.

Corollary 5.1 (Uniform error bound for decomposable ORFF). *Let $\mathcal{C}$ be a compact of subset of $\mathbb{R}^d$ of diameter $|\mathcal{C}|$. Let K be a decomposable kernel built from a positive operator self-adjoint Γ, and k a shift invariant kernel with bounded second derivative such that*

$$\widetilde{K} = \sum_{j=1}^{D} \cos\left\langle \cdot, \omega_j \right\rangle \Gamma \approx K \;\; \textit{and} \;\; \sigma^2 = \mathbf{E}_\rho \|\omega\|_2^2.$$

Then

$$\begin{aligned}&\mathbf{Pr}_\rho\left\{(\omega_j)_{j=1}^D \,\middle|\, \left\|\widetilde{K}-K\right\|_{\mathcal{C}\times\mathcal{C}}\geq\epsilon\right\}\\ &\leq 2^8\left(\frac{\sigma\|\Gamma\|_{\mathcal{Y},\mathcal{Y}}|\mathcal{C}|}{\epsilon}\right)^2\exp\left(-\frac{\epsilon^2 D}{4\|\Gamma\|_2^2(d+2)}\right)\end{aligned}$$

Proof *The proof directly extends theorem 5.1 given by [121]. Let $\tilde{k}$ the Random Fourier approximation for the scalar-valued kernel k. Since*

$$\sup_{(x,z)\,\in\mathcal{C}\times\mathcal{C}}\left\|\widetilde{K}(x,z)-K(x,z)\right\|_{\mathcal{Y},\mathcal{Y}}=\sup_{(x,z)\in\mathcal{C}\times\mathcal{C}}\|\Gamma\|_{\mathcal{Y},\mathcal{Y}}\left|\widetilde{K}(x,z)-k(x,z)\right|,$$

taking $\epsilon'=\|\Gamma\|_{\mathcal{Y},\mathcal{Y}}\epsilon$ gives the following result for all positive ϵ':

$$\begin{aligned}&\mathbf{Pr}_\rho\left\{(\omega_j)_{j=1}^D \,\middle|\, \sup_{x,z\in\mathcal{C}}\left\|\Gamma\left(\tilde{k}(x,z)-k(x,z)\right)\right\|_{\mathcal{Y},\mathcal{Y}}\geq\epsilon'\right\}\\ &\leq 2^8\left(\frac{\sigma\|\Gamma\|_{\mathcal{Y},\mathcal{Y}}|\mathcal{C}|}{\epsilon'}\right)^2\exp\left(-\frac{(\epsilon')^2 D}{4\|\Gamma\|_{\mathcal{Y},\mathcal{Y}}^2(d+2)}\right)\end{aligned}$$

which concludes the proof. □

Please note that a similar corollary could have been obtained for the recent result of Sutherland and Schneider [148] who refined the bound proposed by Rahimi and Recht by using a Bernstein concentration inequality instead of the Hoeffding inequality. More recently Sriperumbudur and Szabo [145] showed an optimal bound for Random Fourier Feature. The improvement of Sriperumbudur and Szabo [145] is mainly in the constant factors where the bound does not depend linearly on the diameter $|\mathcal{C}|$ of $\mathcal{C}$ but exhibit a logarithmic dependency $\log(|\mathcal{C}|)$, hence requiring significantly less random features to reach a desired uniform error with high probability. Moreover, Sutherland and Schneider [148] also considered a bound on the expected max error $\mathbf{E}_{\widehat{\mathbf{Haar}},\rho}\left\|\widetilde{K}-K\right\|_\infty$, which is obtained using Dudley's entropy integral [20, 48] as a bound on the supremum of an empirical process by the covering number of the indexing set. This useful theorem is also part of the proof of Sriperumbudur and Szabo [145].

5.1.2 *Uniform convergence of ORFF approximation on LCA groups*

In this analysis, we assume that $\mathcal{Y}$ is finite dimensional, in subsection 5.1.3, we discuss how the proof could be extended to infinite dimensional output Hilbert spaces. We propose a bound for Operator-valued Random Fourier Feature approximation in the general case. It relies on two main ideas:

1. a matrix-Bernstein concentration inequality for random matrices need to be used instead of concentration inequality for scalar random variables,

2. a general theorem valid for random matrices with bounded norms such as decomposable kernel ORFF approximation as well as unbounded norms such as the ORFF aproximation we proposed for curl and divergence-free kernels that behave as subexponential random variables.

Before introducing the new theorem, we give the definition of the Orlicz norm which gives a proxy-bound on the norm of subexponential random variables.

Definition 5.1 (Orlicz norm [156]). *Let $\psi : \mathbb{R}_+ \to \mathbb{R}_+$ be a non-decreasing convex function with $\psi(0) = 0$. For a random variable X on a measured space $(\Omega, \mathcal{T}(\Omega), \mu)$, the quantity*

$$\|X\|_\psi = \inf \left\{ C > 0 \;\middle|\; \mathbf{E}_\mu[\psi\left(|X|/C\right)] \leq 1 \right\}.$$

is called the Orlicz norm of X.

Here, the function ψ is chosen as $\psi(u) = \psi_\alpha(u)$ where $\psi_\alpha(u) := e^{u^\alpha} - 1$. When $\alpha = 1$, a random variable with finite Orlicz norm is called a *subexponential variable* because its tails decrease at an exponential rate. Let X be a self-adjoint random operator. Given a scalar-valued measure μ, we call *variance* of an operator X the quantity $\mathbf{Var}_\mu[X] = \mathbf{E}_\mu[X - \mathbf{E}_\mu[X]]^2$. With this convention if X is a $p \times p$ Hermitian matrix,

$$\mathbf{Var}_\mu[X]_{\ell m} = \sum_{r=1}^{p} \mathbf{Cov}_\mu[X_{\ell r}, X_{rm}].$$

Among the possible concentration inequalities adapted to random operators [77, 82, 104, 118, 155], we focus on the results of Minsker [104] and Tropp et al. [155], for their robustness to high or potentially infinite dimension of the output space $\mathcal{Y}$. To guarantee a good scaling with the dimension of $\mathcal{Y}$ we introduce the notion of intrinsic dimension (or effective rank) of an operator.

Definition 5.2 *Let A be a trace class operator acting on a Hilbert space $\mathcal{Y}$. We call intrinsic dimension the quantity*

$$IntDim(A) = \frac{\mathbf{Tr}\left[A\right]}{\|A\|_{\mathcal{Y},\mathcal{Y}}}.$$

Indeed the bound proposed in our first publication at ACML [24] based on Koltchinskii et al. [77] depends on p while the present bound depends on the intrinsic dimension of the variance of $A(\omega)$ which is always smaller than p when the operator $A(\omega)$ is Hilbert-Schmidt ($p \leq \infty$).

Corollary 5.2 *Let $K : \mathcal{X} \times \mathcal{X} \to \mathcal{L}(\mathcal{Y})$ be a shift-invariant $\mathcal{Y}$-Mercer kernel, where $\mathcal{Y}$ is a finite dimensional Hilbert space of dimension p and $\mathcal{X}$ a finite dimensional Banach space of dimension d. Moreover, let $\mathcal{C}$ be a closed ball of $\mathcal{X}$ centered at the origin of diameter $|\mathcal{C}|$, $A : \widehat{\mathcal{X}} \to \mathcal{L}(\mathcal{Y})$ and $\mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho}$ a pair such that*

$$\tilde{K}_e = \sum_{j=1}^{D} \cos\left(\cdot, \omega_j\right) A(\omega_j) \approx K_e, \quad \omega_j \sim \mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho} \text{ i. i. d..}$$

Let $\mathcal{D}_{\mathcal{C}} = \mathcal{C} \star \mathcal{C}^{-1}$ *and*

$$V(\delta) \succcurlyeq \mathbf{Var}_{\widehat{\mathbf{Haar}},\rho} \tilde{K}_e(\delta), \quad \textit{for all } \delta \in \mathcal{D}_{\mathcal{C}}$$

and H_ω *be the Lipschitz constant of the function* $h : x \mapsto (x, \omega)$. *If the three following constants exist*

$$m \geq \int_{\widehat{\mathcal{X}}} H_\omega \|A(\omega)\|_{\mathcal{Y},\mathcal{Y}} d\mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho} < \infty$$

and

$$u \geq 4 \left(\left\| \|A(\omega)\|_{\mathcal{Y},\mathcal{Y}} \right\|_{\psi_1} + \sup_{\delta \in \mathcal{D}_{\mathcal{C}}} \|K_e(\delta)\|_{\mathcal{Y},\mathcal{Y}} \right) < \infty$$

and

$$v \geq \sup_{\delta \in \mathcal{D}_{\mathcal{C}}} D \|V(\delta)\|_{\mathcal{Y},\mathcal{Y}} < \infty.$$

Define $p_{int} \geq \sup_{\delta \in \mathcal{D}_{\mathcal{C}}}$ *IntDim*$(V(\delta))$, *then for all* $0 < \epsilon \leq m|C|$,

$$\mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho} \left\{ (\omega_j)_{j=1}^D \;\middle|\; \|\tilde{K} - K\|_{\mathcal{C}\times\mathcal{C}} \geq \epsilon \right\}$$

$$\leq 8\sqrt{2} \left(\frac{m|\mathcal{C}|}{\epsilon} \right) \left(p_{int} r_{v/D}(\epsilon) \right)^{\frac{1}{d+1}} \begin{cases} \exp\left(-D \frac{\epsilon^2}{8v(d+1)\left(1+\frac{1}{p}\right)} \right), & \epsilon \leq \frac{v}{u} \frac{1+1/p}{K(v,p)} \\ \exp\left(-D \frac{\epsilon}{8u(d+1)K(v,p)} \right), & \textit{otherwise}, \end{cases}$$

where $K(v,p) = \log\left(16\sqrt{2}p\right) + \log\left(\frac{u^2}{v}\right)$ *and* $r_{v/D}(\epsilon) = 1 + \frac{3}{\epsilon^2 \log^2(1+D\epsilon/v)}$.

Sketch of proof *In the following, let* $F(\delta) = F(x \star z^{-1}) = \tilde{K}(x,z) - K(x,z)$. *Let* $\mathcal{D}_{\mathcal{C}} = \mathcal{C} \star \mathcal{C}^{-1} = \left\{ x \star z^{-1} \;\middle|\; x, z \in \mathcal{C} \right\}$. *Since* $\mathcal{C}$ *is supposed compact, so is* $\mathcal{D}_{\mathcal{C}}$. *Its diameter is at most* $2|\mathcal{C}|$ *where* $|\mathcal{C}|$ *is the diameter of* $\mathcal{C}$. *Since* $\mathcal{C}$ *is supposed to be a closed ball of a Bochner space tt is then possible to find an* ϵ*-net covering* $\mathcal{C}_\Delta$ *with at most* $T = (4|\mathcal{C}|/r)^d$ *balls of radius r [42]. We call* δ_i *for* $i \in \{1, \dots, T\}$ *the center of the i-th ball, called anchors of the* ϵ*-net. Denote* L_F *the Lipschitz constant of F. We introduce the following lemma proved in [121].*

Lemma 5.1 *For all* $\delta \in \mathcal{D}_{\mathcal{C}}$, *if*

$$L_F \leq \frac{\epsilon}{2r} \tag{5.2}$$

and

$$\|F(\delta_i)\|_{\mathcal{Y},\mathcal{Y}} \leq \frac{\epsilon}{2}, \quad \textit{for all } i \in \mathbb{N}_T^* \tag{5.3}$$

then $\|F(\delta)\|_{\mathcal{Y},\mathcal{Y}} \leq \epsilon$.

To apply the lemma, we must check assumptions equation 5.2 and equation 5.3.

Sketch of proof (equation 5.2) **Lemma 5.2** *Let $H_\omega \in \mathbb{R}_+$ be the Lipschitz constant of $h_\omega(\cdot)$ and assume that*

$$\int_{\widehat{\mathcal{X}}} H_\omega \|A(\omega)\|_{\mathcal{Y},\mathcal{Y}} d\mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho}(\omega) < \infty.$$

Then the operator-valued function $K_e : \mathcal{X} \to \mathcal{L}(\mathcal{Y})$ is Lipschitz with

$$\|K_e(x) - K_e(z)\|_{\mathcal{Y},\mathcal{Y}} \le d_{\mathcal{X}}(x,z) \int_{\widehat{\mathcal{X}}} H_\omega \|A(\omega)\|_{\mathcal{Y},\mathcal{Y}} d\mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho}(\omega). \quad (5.4)$$

In the same way, considering $\tilde{K}_e(\delta) = \frac{1}{D}\sum_{j=1}^{D} \cos h_{\omega_j}(\delta) A(\omega_j)$, where $\omega_j \sim \mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho}$, we can show that $\tilde{K}_e$ is Lipschitz with

$$\left\|\tilde{K}_e(x) - \tilde{K}_e(z)\right\|_{\mathcal{Y},\mathcal{Y}} \le d_{\mathcal{X}}(x,z) \frac{1}{D}\sum_{j=1}^{D} H_{\omega_j} \left\|A(\omega_j)\right\|_{\mathcal{Y},\mathcal{Y}}.$$

Taking the expectation yields

$$\mathbf{E}_{\widehat{\mathbf{Haar}},\rho}\left[L_F\right] = 2 \int_{\widehat{\mathcal{X}}} H_\omega \|A(\omega)\|_{\mathcal{Y},\mathcal{Y}} d\mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho}$$

Thus by Markov's inequality,

$$\begin{aligned}\mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho}\left\{ (\omega_j)_{j=1}^{D} \mid L_F \ge \epsilon \right\} &\le \frac{\mathbf{E}_{\widehat{\mathbf{Haar}},\rho}\left[L_F\right]}{\epsilon} \\ &\le \frac{2}{\epsilon} \int_{\widehat{\mathcal{X}}} H_\omega \|A(\omega)\|_{\mathcal{Y},\mathcal{Y}} d\mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho}.\end{aligned} \quad (5.5)$$

Sketch of proof (equation 5.3) *To obtain a bound on the anchors we apply theorem 4 of Koltchinskii et al. [77]. We suppose the existence of the two constants*

$$v_i = D\mathbf{Var}_{\widehat{\mathbf{Haar}},\rho}\left[\tilde{K}(\delta_i)\right]$$

and

$$u_i = 4\left(\left\|\|A(\omega)\|_{\mathcal{Y},\mathcal{Y}}\right\|_{\psi_1} + \|K_e(\delta_i)\|_{\mathcal{Y},\mathcal{Y}}\right)$$

Then $\forall i \in \{1, \dots, T\}$,

$$\begin{aligned}&\mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho}\left\{ (\omega_j)_{j=1}^{D} \;\middle|\; \|F(\delta_i)\|_{\mathcal{Y},\mathcal{Y}} \ge \epsilon \right\} \\ \le &\begin{cases} 4\mathit{IntDim}(v_i)\exp\left(-D\frac{\epsilon^2}{2\|v_i\|_{\mathcal{Y},\mathcal{Y}}\left(1+\frac{1}{p}\right)}\right) r_{v_i/D}(\epsilon), & \epsilon \le \frac{\|v_i\|_{\mathcal{Y},\mathcal{Y}}}{2u_i}\frac{1+1/p}{K(v_i,p)} \\ 4\mathit{IntDim}(v_i)\exp\left(-D\frac{\epsilon}{4u_iK(v_i,p)}\right) r_{v_i/D}(\epsilon), & \text{otherwise.}\end{cases}\end{aligned}$$

where

$$K(v_i,p) = \log\left(16\sqrt{2}p\right) + \log\left(\frac{u_i^2}{\|v_i\|_{\mathcal{Y},\mathcal{Y}}}\right)$$

and

$$r_{v_i/D} = 1 + \frac{3}{\epsilon^2\log^2(1 + D\epsilon/\|v_i\|_{\mathcal{Y},\mathcal{Y}})}.$$

Combining equation 5.2 and equation 5.3. Now applying the lemma and taking the union bound over the centers of the ϵ-net yields

$$\begin{aligned}
&\mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho}\left\{ (\omega_j)_{j=1}^D \;\middle|\; \left\|\tilde{K} - K\right\|_{\mathcal{C}\times\mathcal{C}} \geq \epsilon \right\} \\
&\leq 4 \left(\frac{rm}{\epsilon} + p_{int}\left(\frac{2|\mathcal{C}|}{r}\right)^d r_{v/D}(\epsilon) \right. \\
&\left. \begin{cases} \exp\left(-D\frac{\epsilon^2}{8v\left(1+\frac{1}{p}\right)}\right), & \epsilon \leq \frac{v}{u}\frac{1+1/p}{K(v,p)} \\ \exp\left(-D\frac{\epsilon}{8uK(v,p)}\right), & \textit{otherwise.} \end{cases} \right)
\end{aligned}$$

The right hand side of the equation has the form $ar + br^{-d}$ with

$$a = \frac{m}{\epsilon}$$

and

$$b = p_{int}(2|\mathcal{C}|)^d r_{v/D}(\epsilon) \begin{cases} \exp\left(-D\frac{\epsilon^2}{8v\left(1+\frac{1}{p}\right)}\right), & \epsilon \leq \frac{v}{u}\frac{1+1/p}{K(v,p)} \\ \exp\left(-D\frac{\epsilon}{8uK(v,p)}\right), & \textit{otherwise.} \end{cases}$$

Following [101, 121, 148], we optimize over r. It is a convex continuous function on $\mathbb{R}_+$ *and achieve the minimum value*

$$r_* = a^{\frac{d}{d+1}} b^{\frac{1}{d+1}} \left(d^{\frac{1}{d+1}} + d^{-\frac{d}{d+1}} \right),$$

hence

$$\begin{aligned}
&\mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho}\left\{ (\omega_j)_{j=1}^D \;\middle|\; \left\|\tilde{K} - K\right\|_{\mathcal{C}\times\mathcal{C}} \geq \epsilon \right\} \\
&\leq 8\sqrt{2}\left(\frac{m|\mathcal{C}|}{\epsilon}\right)\left(p_{int} r_{v/D}(\epsilon)\right)^{\frac{1}{d+1}} \begin{cases} \exp\left(-D\frac{\epsilon^2}{8v(d+1)\left(1+\frac{1}{p}\right)}\right), & \epsilon \leq \frac{v}{u}\frac{1+1/p}{K(v,p)} \\ \exp\left(-D\frac{\epsilon}{8u(d+1)K(v,p)}\right), & \textit{otherwise,} \end{cases}
\end{aligned}$$

We give a comprehensive full proof of the theorem in appendix A.1. It follows the usual scheme derived in Rahimi and Recht [121] and Sutherland and Schneider [148] and involves Bernstein concentration inequality for unbounded symmetric matrices (theorem A.3).

5.1.3 *Dealing with infinite dimensional operators*

We studied the concentration of ORFFs under the assumption that $\mathcal{Y}$ is finite dimensional. Indeed a d term characterizing the dimension of the input space $\mathcal{X}$ appears in the bound proposed in corollary 5.2, and when d tends to infinity, the exponential part goes to zero so that the probability is bounded by a constant greater than one. Unfornunately, considering unbounded random operators Minsker [104] doesn't give any tighter solution.

In our first bound presented at ACML, we presented a bound based on a matrix concentration inequality for unbounded random variable. Compared to this previous bound, corollary 5.2 does not depend on the dimensionality p of the output space $\mathcal{Y}$ but on the intrinsic dimension of the operator $A(\omega)$. However to remove the dependency in p in the exponential part, we must turn our attention to operator concentration inequalities for bounded random variable. To the best of our knowledge we are not aware of concentration inequalities working for "unbounded" operator- valued random variables. Following the same proof than corollary 5.2 we obtain

Corollary 5.3 *Let $K : \mathcal{X} \times \mathcal{X} \to \mathcal{L}(\mathcal{Y})$ be a shift-invariant $\mathcal{Y}$-Mercer kernel, where $\mathcal{Y}$ is a Hilbert space and $\mathcal{X}$ a finite dimensional Banach space of dimension D. Moreover, let $\mathcal{C}$ be a closed ball of $\mathcal{X}$ centered at the origin of diameter $|\mathcal{C}|$, subset of $\mathcal{X}$, $A : \widehat{\mathcal{X}} \to \mathcal{L}(\mathcal{Y})$ and $\mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho}$ a pair such that*

$$\tilde{K}_e = \sum_{j=1}^{D} \cos\left(\cdot, \omega_j\right) A(\omega_j) \approx K_e, \quad \omega_j \sim \mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho} \; i.\,i.\,d..$$

where $A(\omega_j)$ is a Hilbert-Schmidt operator for all $j \in \mathbb{N}^*_D$. Let $\mathcal{D}_{\mathcal{C}} = \mathcal{C} \star \mathcal{C}^{-1}$ and

$$V(\delta) \succcurlyeq \mathbf{Var}_{\widehat{\mathbf{Haar}},\rho} \tilde{K}_e(\delta), \quad \textit{for all } \delta \in \mathcal{D}_{\mathcal{C}}$$

and H_ω be the Lipschitz constant of the function $h : x \mapsto (x, \omega)$. If the three following constant exists

$$m \geq \int_{\widehat{\mathcal{X}}} H_\omega \|A(\omega)\|_{\mathcal{Y},\mathcal{Y}} d\mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho} < \infty$$

and

$$u \geq \operatorname*{ess\,sup}_{\omega \in \widehat{\mathcal{X}}} \|A(\omega)\|_{\mathcal{Y},\mathcal{Y}} + \sup_{\delta \in \mathcal{D}_{\mathcal{C}}} \|K_e(\delta)\|_{\mathcal{Y},\mathcal{Y}} < \infty$$

and

$$v \geq \sup_{\delta \in \mathcal{D}_{\mathcal{C}}} D \|V(\delta)\|_{\mathcal{Y},\mathcal{Y}} < \infty.$$

define $p_{int} \geq \sup_{\delta \in \mathcal{D}_{\mathcal{C}}} \mathit{IntDim}\left(V(\delta)\right)$ then for all $\sqrt{\frac{v}{D}} + \frac{u}{3D} < \epsilon < m|\mathcal{C}|$,

$$\begin{aligned} &\mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho} \left\{ (\omega_j)_{j=1}^D \;\middle|\; \sup_{\delta \in \mathcal{D}_{\mathcal{C}}} \|F(\delta)\|_{\mathcal{Y},\mathcal{Y}} \geq \epsilon \right\} \\ &\quad \leq 8\sqrt{2} \left(\frac{m|\mathcal{C}|}{\epsilon} \right) p_{int}^{\frac{1}{d+1}} \exp\left(-D\psi_{v,d,u}(\epsilon)\right) \end{aligned}$$

where $\psi_{v,d,u}(\epsilon) = \frac{\epsilon^2}{2(d+1)(v+u\epsilon/3)}$.

Again a full comprehensive proof is given in appendix A.1 of the appendix. Notice that in this result, The dimension $p = \dim \mathcal{Y}$ does not appear. Only the intrinsic dimension of the variance of the estimator. Moreover when d is large, the term $p_{int}^{\frac{1}{d+1}}$ goes to one, so that the impact of the inrinsic dimension on the bound is limited.

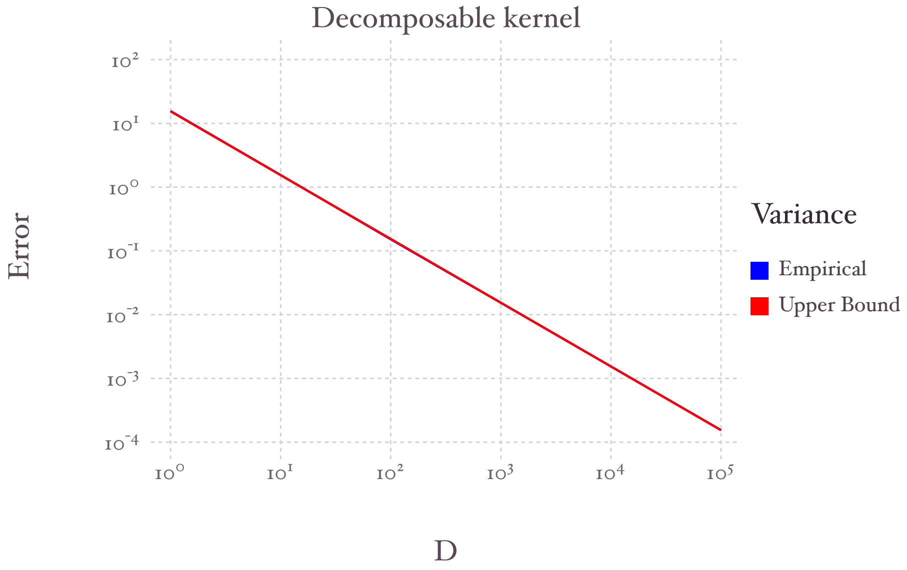

Figure 5.2: Comparison between an empirical bound on the norm of the variance of the decomposable ORFF obtained and the theoretical bound proposed in proposition A.3 versus D

5.1.4 *Variance of the ORFF approximation*

We now provide a bound on the norm of the variance of $\tilde{K}$, required to apply corollaries 5.2 and 5.3. This is an extension of the proof of Sutherland and Schneider [148] to the operator-valued case, and we recover their results in the scalar case when $A(\omega) = 1$. An illustration of the bound is provided in figure 5.3 for the decomposable and the curl-free OVK.

Proposition 5.1 (Bounding the *variance* of $\tilde{K}$). *Let K be a shift invariant $\mathcal{Y}$-Mercer kernel on a second countable LCA topological space $\mathcal{X}$. Let $A : \widehat{\mathcal{X}} \to \mathcal{L}(\mathcal{Y})$ and $\mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho}$ a pair such that*

$$\tilde{K}_e = \sum_{j=1}^{D} \cos\left(\cdot, \omega_j\right) A(\omega_j) \approx K_e, \quad \omega_j \sim \mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho} \text{ i. i. d..}$$

Then,

$$\begin{aligned}\mathbf{Var}_{\widehat{\mathbf{Haar}},\rho}\left[\tilde{K}_e(\delta)\right] \preccurlyeq \frac{1}{2D}\Big(&(K_e(2\delta) + K_e(e))\,\mathbf{E}_{\widehat{\mathbf{Haar}},\rho}\left[A(\omega)\right] \\ &- 2K_e(\delta)^2 + \mathbf{Var}_{\widehat{\mathbf{Haar}},\rho}\left[A(\omega)\right]\Big)\end{aligned}$$

Proof *It relies on the i. i. d. property of the random vectors ω_j and trigonometric identities (see the proof in proposition A.3 of the appendix).* □

5.1.5 *Application on decomposable, curl-free and divergence-free OVK*

First, the two following examples discuss the form of H_ω for the additive group and the skewed-multiplicative group. Here we view $\mathcal{X} = \mathbb{R}^d$ as a

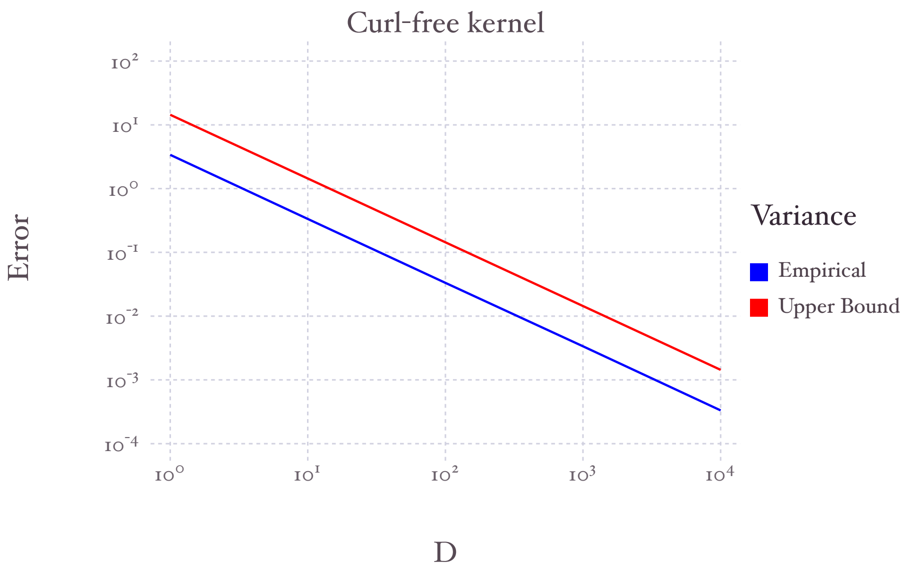

Figure 5.3: Comparison between an empirical bound on the norm of the variance of the curl-free ORFF obtained and the theoretical bound proposed in proposition A.3 versus D

Banach space endowed with the Euclidean norm. Thus the Lipschitz constant H_ω is bounded by the supremum of the norm of the gradient of h_ω.

Example 5.1 (Additive group). *On the additive group,* $h_\omega(\delta) = \langle \omega, \delta \rangle$. *Hence* $H_\omega = \|\omega\|_2$.

Example 5.2 (Skewed-multiplicative group). *On the skewed multiplicative group,* $h_\omega(\delta) = \langle \omega, \log(\delta + c)\rangle$. *Therefore*

$$\sup_{\delta \in \mathcal{C}} \|\nabla h_\omega(\delta)\|_2 = \sup_{\delta \in \mathcal{C}} \|\omega/(\delta + c)\|_2.$$

Eventually $\mathcal{C}$ *is compact subset of* $\mathcal{X}$ *and finite dimensional thus* $\mathcal{C}$ *is closed and bounded. Thus* $H_\omega = \|\omega\|_2/(\min_{\delta \in \mathcal{C}} \|\delta\|_2 + c)$.

Now we compute upper bounds on the norm of the variance and Orlicz norm of the three ORFFs we took as examples.

5.1.5.1 *Decomposable kernel*

notice that in the case of the Gaussian decomposable kernel, i. e. $A(\omega) = A$, $e = 0$, $K_0(\delta) = Ak_0(\delta)$, $k_0(\delta) \geq 0$ and $k_0(\delta) = 1$, then we have

$$D \left\| \mathbf{Var}_\mu \left[\tilde{K}_0(\delta) \right] \right\|_{\mathcal{Y},\mathcal{Y}} \leq (1 + k_0(2\delta)) \|A\|_{\mathcal{Y},\mathcal{Y}}/2 + k_0(\delta)^2.$$

5.1.5.2 *Curl-free and divergence-free kernels:*

recall that in this case $p = d$. For the (Gaussian) curl-free kernel, $A(\omega) = \omega\omega^*$ where $\omega \in \mathbb{R}^d \sim \mathcal{N}(0, \sigma^{-2} I_d)$ thus $\mathbf{E}_\mu[A(\omega)] = I_d/\sigma^2$ and $\mathbf{Var}_\mu[A(\omega)] = (d+1)I_d/\sigma^4$. Hence,

$$D\left\|\mathbf{Var}_\mu\left[\tilde{K}_0(\delta)\right]\right\|_{\mathcal{Y},\mathcal{Y}} \leq \frac{1}{2}\left\|\frac{1}{\sigma^2}K_0(2\delta) - 2K_0(\delta)^2\right\|_{\mathcal{Y},\mathcal{Y}} + \frac{(d+1)}{\sigma^4}.$$

This bound is illustrated by figure 5.1 B, for a given datapoint. Eventually for the Gaussian divergence-free kernel, $A(\omega) = I\|\omega\|_2^2 - \omega\omega^*$, thus $\mathbf{E}_\mu[A(\omega)] = I_d(d-1)/\sigma^2$ and $\mathbf{Var}_\mu[A(\omega)] = d(4d-3)I_d/\sigma^4$. Hence,

$$D\left\|\mathbf{Var}_\mu\left[\tilde{K}_0(\delta)\right]\right\|_{\mathcal{Y},\mathcal{Y}} \leq \frac{1}{2}\left\|\frac{(d-1)}{\sigma^2}K_0(2\delta) - 2K_0(\delta)^2\right\|_{\mathcal{Y},\mathcal{Y}} + \frac{d(4d-3)}{\sigma^4}.$$

To conclude, we ensure that the random variable $\|A(\omega)\|_{\mathcal{Y},\mathcal{Y}}$ has a finite Orlicz norm with $\psi = \psi_1$ in these three cases.

5.1.5.3 *Computing the Orlicz norm*

for a random variable with strictly monotonic moment generating function (MGF), one can characterize its inverse ψ_1 Orlicz norm by taking the functional inverse of the MGF evaluated at 2 (see lemma A.3 of the appendix). In other words $\|X\|_{\psi_1}^{-1} = \text{MGF}(x)_X^{-1}(2)$. For the Gaussian curl-free and divergence-free kernel,

$$\left\|A^{div}(\omega)\right\|_{\mathcal{Y},\mathcal{Y}} = \left\|A^{curl}(\omega)\right\|_{\mathcal{Y},\mathcal{Y}} = \|\omega\|_2^2,$$

where $\omega \sim \mathcal{N}(0, I_d/\sigma^2)$, hence $\|A(\omega)\|_2 \sim \Gamma(p/2, 2/\sigma^2)$. The MGF of this gamma distribution is $\text{MGF}(x)(t) = (1 - 2t/\sigma^2)^{-(p/2)}$. Eventually

$$\left\|\left\|A^{div}(\omega)\right\|_{\mathcal{Y},\mathcal{Y}}\right\|_{\psi_1}^{-1} = \left\|\left\|A^{curl}(\omega)\right\|_{\mathcal{Y},\mathcal{Y}}\right\|_{\psi_1}^{-1} = \frac{\sigma^2}{2}\left(1 - 4^{-\frac{1}{p}}\right).$$

5.2 CONCLUSIONS

In this chapter we have seen how to bound $\left\|\widetilde{K} - K\right\|$ in the operator norm with high probability (section 5.1). We studied the case of unbounded finite dimensional OVKs and bounded potentially infinite dimensional OVKs. The current lack of concentration inequalities working for both unbounded and infinite dimensional with the operator norm (Banach space) in the litterature provides us to unify these bounds.

6

LEARNING WITH FEATURE MAPS

Contents

6.1 LEARNING WITH ORFF

We now turn our attention to learning function with an ORFF model that approximate an OVK model.

6.1.1 *Supervised regression*

Let $\mathbf{s} = (x_i, y_i)_{i=1}^N \in (\mathcal{X} \times \mathcal{Y})^N$ be a sequence of training samples. Given a local loss function $L : \mathcal{X} \times \mathcal{F} \times \mathcal{Y} \to \overline{\mathbb{R}}$ such that L is proper, convex and lower semi-continous in $\mathcal{F}$, we are interested in finding a *vector-valued function* $f_{\mathbf{s}} : \mathcal{X} \to \mathcal{Y}$, that lives in a VV-RKHS and minimize a tradeoff between a data fitting term L and a regularization term to prevent from overfitting. Namely finding $f_{\mathbf{s}} \in \mathcal{H}_K$ such that

$$f_{\mathbf{s}} = \operatorname*{arg\,min}_{f \in \mathcal{H}_K} \frac{1}{N} \sum_{i=1}^N L(x_i, f, y_i) + \frac{\lambda}{2} \|f\|_K^2 \tag{6.1}$$

where $\lambda \in \mathbb{R}_+$ is a regularization[11] parameter. We call the quantity

[11] *Tychonov regularization.*

$$\mathcal{R}(f) = \frac{1}{N} \sum_{i=1}^N L(x_i, f, y_i), \quad \forall f \in \mathcal{H}_K, \forall \mathbf{s} \in (\mathcal{X} \times \mathcal{Y})^N.$$

is called the empirical risk of the model $f \in \mathcal{H}_K$ according the local loss L. A common choice of data fitting term for regression is $L : (x, f, y) \mapsto \|f(x) - y\|_{\mathcal{Y}}^2$. We introduce a corollary from Mazur and Schauder proposed in 1936 (see Górniewicz [65] and Kurdila and Zabarankin [78]) showing that equation 6.1 –and equation 6.3– attains a unique mimimizer.

Theorem 6.1 (Mazur-Schauder). *Let $\mathcal{H}$ be a Hilbert space and $\mathcal{J} : \mathcal{H} \to \overline{\mathbb{R}}$ be a proper, convex, lower semi-continuous and coercive function. Then $\mathcal{J}$ is bounded from below and attains a minimizer. Moreover if $\mathcal{J}$ is strictly convex the minimizer is unique.*

This is easily verified for Ridge regression. Define

$$\mathcal{J}_\lambda(f) = \frac{1}{N} \sum_{i=1}^N \|f(x_i) - y_i\|_{\mathcal{Y}}^2 + \frac{\lambda}{2} \|f\|_K^2, \tag{6.2}$$

where $f \in \mathcal{H}_K$ and $\lambda \in \mathbb{R}_{>0}$. $\mathcal{J}_\lambda$ is continuous[12] and strictly convex. Additionally $\mathcal{J}_\lambda$ is coercive since $\|f\|_K$ is coercive, $\lambda \in \mathbb{R}_{>0}$, and all the summands of $\mathcal{J}_\lambda$ are positive. Hence for all positive λ, $f_{\mathbf{s}} = \operatorname{arg\,min}_{f \in \mathcal{H}_K} \mathcal{J}_\lambda(f)$ exists, is unique and attained.

[12] *Reminder, if $f \in \mathcal{H}_k$, $ev_x : f \mapsto f(x)$ is continuous, see proposition 3.2.*

Remark 6.1 *We consider the optimization problem proposed in equation 6.2 where $L : (x_i, f, y_i) \mapsto \|f(x_i) - y_i\|_{\mathcal{Y}}^2$. If given a training sample $\mathbf{s}$, we have*

$$\frac{1}{N} \sum_{i=1}^N \|y_i\|_{\mathcal{Y}}^2 \leq \sigma_y^2,$$

then $\lambda \|f_s\|_K \le 2\sigma_y^2$. *Indeed, since* $\mathcal{H}_K$ *is a Hilbert space,* $0 \in \mathcal{H}_K$, *thus*

$$\begin{aligned}\frac{\lambda}{2}\|f_s\|_K^2 &\le \frac{1}{N}\sum_{i=1}^N L(x_i, f_s, y_i) + \frac{\lambda}{2}\|f_s\|_K^2 \\ &\le \frac{1}{N}\sum_{i=1}^N L(x_i, 0, y_i) \le \sigma_y^2, \quad \textit{by optimality of } f_s.\end{aligned}$$

Since for all $x \in \mathcal{X}$, $\|f(x)\|_{\mathcal{Y}} \le \sqrt{\|K(x,x)\|_{\mathcal{Y},\mathcal{Y}}}\|f\|_K$, *the maximum value that the solution* $\|f_s(x)\|_{\mathcal{Y}}$ *of* equation 6.2 *can reach is* $\sigma_y\sqrt{\frac{2\|K(x,x)\|_{\mathcal{Y},\mathcal{Y}}}{\lambda}}$. *Thus when solving a Ridge regression problem, given a shift-invariant kernel* K_e, *one should choose*

$$0 < \lambda \le 2\|K_e(e)\|_{\mathcal{Y},\mathcal{Y}}\frac{\sigma_y^2}{C^2}.$$

with $C \in \mathbb{R}_{>0}$ *to have a chance to fit all the* y_i *with norm* $\|y_i\|_{\mathcal{Y}} \le C$ *in the train set.*

6.1.2 *Representer theorem and Feature equivalence*

Regression in Vector Valued Reproducing Kernel Hilbert Space has been well studied [4, 30, 73, 98, 100, 103, 128], and a cornerstone of learning in VV-RKHS is the representer theorem[13] , which allows to replace the search of a minimizer in a infinite dimensional VV-RKHS by a finite number of paramaters $(u_i)_{i=1}^N$, $u_i \in \mathcal{Y}$.

[13] *Sometimes referred to as minimal norm interpolation theorem.*

In the following we suppose we are given a cost function $c : \mathcal{Y} \times \mathcal{Y} \to \overline{\mathbb{R}}$, such that $c(f(x), y)$ returns the error of the prediction $f(x)$ w. r. t. the ground truth y. A loss function of a model f with respect to an example $(x, y) \in \mathcal{X} \times \mathcal{Y}$ can be naturally defined from a cost function as $L(x, f, y) = c(f(x), y)$. Conceptually the function c evaluates the quality of the prediction versus its ground truth $y \in \mathcal{Y}$ while the loss function L evaluates the quality of the model f at a training point $(x, y) \in \mathcal{X} \times \mathcal{Y}$.

Theorem 6.2 (Representer theorem). *Let* K *be a* $\mathcal{Y}$*-Mercer Operator-Valued Kernel and* $\mathcal{H}_K$ *its corresponding* $\mathcal{Y}$*-Reproducing Kernel Hilbert space.*

Let $c : \mathcal{Y} \times \mathcal{Y} \to \overline{\mathbb{R}}$ *be a cost function such that* $L(x, f, y) = c(Vf(x), y)$ *is a proper convex lower semi-continuous function in* f *for all* $x \in \mathcal{X}$ *and all* $y \in \mathcal{Y}$.

Eventually let $\lambda \in \mathbb{R}_{>0}$ *be the Tychonov regularization hyperparameters The solution* $f_s \in \mathcal{H}_K$ *of the regularized optimization problem*

$$f_s = \operatorname*{arg\,min}_{f \in \mathcal{H}_K} \frac{1}{N}\sum_{i=1}^N c(f(x_i), y_i) + \frac{\lambda}{2}\|f\|_K^2 \tag{6.3}$$

has the form $f_s = \sum_{j=1}^N K(\cdot, x_j)u_{s,j}$ *where* $u_{s,j} \in \mathcal{Y}$ *and*

$$u_s = \operatorname*{arg\,min}_{u \in \bigoplus_{i=1}^N \mathcal{U}} \frac{1}{N}\sum_{i=1}^N c\left(\sum_{k=1}^N K(x_i, x_j)u_j, y_i\right) + \frac{\lambda}{2}\sum_{k=1}^N u_i^* K(x_i, x_k)u_k. \tag{6.4}$$

The first representer theorem was introduced by Wahba [162] in the case where $\mathcal{Y} = \mathbb{R}$. The extension to an arbitrary Hilbert space $\mathcal{Y}$ has been proved by many authors in different forms [29, 73, 98]. The idea behind the representer theorem is that eventhough we minimize over the whole space $\mathcal{H}_K$, when $\lambda > 0$, the solution of equation 6.3 falls inevitably into the set

$$\mathcal{H}_{K,\mathbf{s}} = \left\{ \sum_{j=1}^{N} K_{x_j} u_j \;\middle|\; \forall (u_i)_{i=1}^{N} \in \mathcal{Y}^N \right\}.$$

Therefore the result can be expressed as a finite linear combination of basis functions of the form $K(\cdot, x_k)$. Remark that we can perform the kernel expansion of $f_{\mathbf{s}} = \sum_{j=1}^{N} K(\cdot, x_j) u_{\mathbf{s},j}$ eventhough $\lambda = 0$. However $f_{\mathbf{s}}$ is no longer the solution of equation 6.3 over the whole space $\mathcal{H}_K$ but a projection on the subspace $\mathcal{H}_{K,\mathbf{s}}$. While this is in general not a problem for practical applications, it might raise issues for further theoretical investigations. In particular, it makes it difficult to perform theoretical comparison of the "exact" solution of equation 6.3 with respect to the ORFF approximation solution given in theorem 6.3.

Proof *Since $f(x) = K_x^* f$ (see equation 3.14), the optimization problem reads*

$$f_{\mathbf{s}} = \operatorname*{arg\,min}_{f \in \mathcal{H}_K} \frac{1}{N} \sum_{i=1}^{N} c(K_{x_i}^* f, y_i) + \frac{\lambda}{2} \|f\|_K^2$$

Let $W_{\mathbf{s}} : \mathcal{H}_K \to \bigoplus_{i=1}^{N} \mathcal{Y}$ be the restriction[1] *linear operator defined as*

$$W_{\mathbf{s}} f = \bigoplus_{i=1}^{N} K_{x_i}^* f,$$

with $K_{x_i}^* : \mathcal{H}_K \to \mathcal{Y}$ and $K_{x_i} : \mathcal{Y} \to \mathcal{H}_K$. Let $Y = \bigoplus_{i=1}^{N} y_i \in \mathcal{Y}^N$. We have

$$\langle Y, W_{\mathbf{s}} f \rangle_{\bigoplus_{i=1}^{N} \mathcal{Y}} = \sum_{i=1}^{N} \langle y_i, K_{x_i}^* f \rangle_{\mathcal{Y}} = \sum_{i=1}^{N} \langle K_{x_i} y_i, f \rangle_{\mathcal{H}_K}.$$

Thus the adjoint operator $W_{\mathbf{s}}^* : \bigoplus_{i=1}^{N} \mathcal{Y} \to \mathcal{H}_K$ is

$$W_{\mathbf{s}}^* Y = \sum_{i=1}^{N} K_{x_i} y_i,$$

and the operator $W_{\mathbf{s}}^* W_{\mathbf{s}} : \mathcal{H}_K \to \mathcal{H}_K$ is

$$W_{\mathbf{s}}^* W_{\mathbf{s}} f = \sum_{i=1}^{N} K_{x_i} K_{x_i}^* f.$$

1 $W_{\mathbf{s}}$ is sometimes called the sampling or evaluation operator as in Minh, Bazzani, and Murino [103]. However we prefer calling it "restriction operator" as in Rosasco, Belkin, and Vito [124] since $W_{\mathbf{s}} f$ is the restriction of f to the points in $\mathbf{s}$.

Let

$$\mathcal{J}_\lambda(f) = \underbrace{\frac{1}{N}\sum_{i=1}^{N} c(f(x_i), y_i)}_{=\mathcal{J}_c} + \frac{\lambda}{2}\|f\|_K^2$$

To ensure that $\mathcal{J}_\lambda$ has a global minimizer we need the following technical lemma (which is a consequence of the Hahn-Banach theorem for lower-semicontimuous functional, see Kurdila and Zabarankin [78]).

Lemma 6.1 *Let $\mathcal{J}$ be a proper, convex, lower semi-continuous functional, defined on a Hilbert space $\mathcal{H}$. If $\mathcal{J}$ is strongly convex, then $\mathcal{J}$ is coercive.*

Proof *Consider the convex function $G(f) := \mathcal{J}(f) - \lambda\|f\|^2$, for some $\lambda > 0$. Since $\mathcal{J}$ is by assumption proper, lower semi-continuous and strongly convex with parameter λ, G is proper, lower semi-continuous and convex. Thus Hahn-Banach theorem apply, stating that G is bounded by below by an affine functional. i. e. there exists f_0 and $f_1 \in \mathcal{H}$ such that*

$$G(f) \geq G(f_0) + \langle f - f_0, f_1\rangle, \quad \textit{for all } f \in \mathcal{H}.$$

Then substitute the definition of G to obtain

$$\mathcal{J}(f) \geq \mathcal{J}(f_0) + \lambda\left(\|f\| - \|f_0\|\right) + \langle f - f_0, f_1\rangle.$$

By the Cauchy-Schwartz inequality, $\langle f, f_1\rangle \geq -\|f\|\|f_1\|$, thus

$$\mathcal{J}(f) \geq \mathcal{J}(f_0) + \lambda\left(\|f\| - \|f_0\|\right) - \|f\|\|f_1\| - \langle f_0, f_1\rangle,$$

which tends to infinity as f tends to infinity. Hence $\mathcal{J}$ is coercive □

Since c is proper, lower semi-continuous and convex by assumption, thus the term $\mathcal{J}_c$ is also proper, lower semi-continuous and convex. Moreover the term $\mathcal{J}_M$ is always positive for any $f \in \mathcal{H}_K$ and $\frac{\lambda}{2}\|f\|_K^2$ is strongly convex. Thus $\mathcal{J}_\lambda$ is strongly convex. Apply lemma 6.1 to obtain the coercivity of $\mathcal{J}_\lambda$, and then theorem 6.1 to show that $\mathcal{J}_\lambda$ has a unique minimizer and is attained. Then let

$$\mathcal{H}_{K,\mathbf{s}} = \left\{ \sum_{j=1}^{N} K_{x_j} u_j \;\middle|\; \forall (u_i)_{i=1}^N \in \mathcal{Y}^N \right\}.$$

For $f \in \mathcal{H}_{K,\mathbf{s}}^{\perp}$ [14] *, the operator $W_\mathbf{s}$ satisfies*

[14] $\mathcal{H}_{K,\mathbf{s}}^{\perp} \oplus \mathcal{H}_{K,\mathbf{s}} = \mathcal{H}_{\mathcal{K}}$.

$$\langle \Upsilon, W_\mathbf{s} f\rangle_{\oplus_{i=1}^N \mathcal{Y}} = \langle \underbrace{f}_{\in \mathcal{H}_{K,\mathbf{s}}^{\perp}}, \underbrace{\sum_{i=1}^{N} K_{x_i} V^* y_i}_{\in \mathcal{H}_{K,\mathbf{s}}} \rangle_{\mathcal{H}_K} = 0$$

for all sequences $(y_i)_{i=1}^N$, since $y_i \in \mathcal{Y}$. Hence,

$$(f(x_i))_{i=1}^N = 0 \qquad (6.5)$$

In the same way,

$$\sum_{i=1}^N \langle K_{x_i}^* f, u_i\rangle_{\mathcal{Y}} = \langle \underbrace{f}_{\in \mathcal{H}_{K,\mathbf{s}}^\perp}, \underbrace{\sum_{j=1}^N K_{x_j} u_j}_{\in \mathcal{H}_{K,\mathbf{s}}} \rangle_{\mathcal{H}_K} = 0.$$

for all sequences $(u_i)_{i=1}^N \in \mathcal{Y}^N$. *As a result,*

$$(f(x_i))_{i=1}^N = 0. \tag{6.6}$$

Now for an arbitrary $f \in \mathcal{H}_K$, *consider the orthogonal decomposition* $f = f^\perp + f^\parallel$, *where* $f^\perp \in \mathcal{H}_{K,\mathbf{s}}^\perp$ *and* $f^\parallel \in \mathcal{H}_{K,\mathbf{s}}$. *Then since* $\left\| f^\perp + f^\parallel \right\|_{\mathcal{H}_K}^2 = \| f^\perp \|_{\mathcal{H}_K}^2 + \left\| f^\parallel \right\|_{\mathcal{H}_K}^2$, *equation 6.5 and equation 6.6 shows that if* $\lambda > 0$, *clearly then*

$$\mathcal{J}_\lambda(f) = \mathcal{J}_{\lambda_K}\left(f^\perp + f^\parallel\right) \geq \mathcal{J}_\lambda\left(f^\parallel\right)$$

The last inequality holds only when $\| f^\perp \|_{\mathcal{H}_K} = 0$, *that is when* $f^\perp = 0$. *As a result since the minimizer of* $\mathcal{J}_\lambda$ *is unique and attained, it must lies in* $\mathcal{H}_{K,\mathbf{s}}$. □

The representer theorem show that minimizing a functional in a VV-RKHS yields a solution which depends on all the points in the training set. Assuming that for all $x_i, x \in \mathcal{X}$ and for all $u_i \in \mathcal{Y}$ it takes time $O(P)$, to compute $K(x_i, x)u_i$, making a prediction using the representer theorem take $O(2P)$. Obviously If $\mathcal{Y} = \mathbb{R}^p$, Then $P = O(p^2)$ thus making a prediction cost $O(2p^2)$ operations.

Instead learning a model f that depends on all the points of the training set, we would like to learn a parametric model of the form $\widetilde{f}(x) = \widetilde{\Phi}(x)^*\theta$, where θ lives in some redescription space $\widetilde{\mathcal{H}}$. We are interested in finding a parameter vector $\theta_{\mathbf{s}}$ such that

$$\theta_{\mathbf{s}} = \operatorname*{arg\,min}_{\theta \in \widetilde{\mathcal{H}}} \frac{1}{N}\sum_{i=1}^N c\left(\widetilde{\Phi}(x_i)^*\theta, y_i\right) + \frac{\lambda}{2}\|\theta\|_{\widetilde{\mathcal{H}}}^2 \tag{6.7}$$

The following theorem states that when $\lambda > 0$ then learning with a feature map is equivalent to learn with a kernel. Moreover if $f_{\mathbf{s}} \in \mathcal{H}_K$ is a solution of equation 6.8 and $\theta_{\mathbf{s}} \in \mathcal{H}$ is the solution of equation 6.8, then $f_{\mathbf{s}} = \Phi(\cdot)^*\theta_{\mathbf{s}}$. To the best of our knowledge no such results exist in the litterature even for scalar-valued kernel. Usually the author suppose that the class of function $\mathcal{H}_K$ can be replace with $\mathcal{H}$ but do not study the link between these spaces.

Theorem 6.3 (Feature equivalence). *Let* $\widetilde{K}$ *be an Operator-Valued Kernel such that for all* $x, z \in \mathcal{X}$, $\widetilde{\Phi}(x)^*\widetilde{\Phi}(z) = \widetilde{K}(x,z)$ *where* $\widetilde{K}$ *is a* $\mathcal{Y}$*-Mercer OVK and* $\mathcal{H}_{\widetilde{K}}$ *its corresponding* $\mathcal{Y}$*-Reproducing kernel Hilbert space.*

Let $c : \mathcal{Y} \times \mathcal{Y} \to \overline{\mathbb{R}}$ *be a cost function such that* $L\left(x, \widetilde{f}, y\right) = c\left(\widetilde{f}(x), y\right)$ *is a proper convex lower semi-continuous function in* $\widetilde{f} \in \mathcal{H}_{\widetilde{K}}$ *for all* $x \in \mathcal{X}$ *and all* $y \in \mathcal{Y}$.

Eventually let $\lambda \in \mathbb{R}_{>0}\mathbb{R}_+$ *be the Tychonov regularization hyperparameter. The solution* $f_s \in \mathcal{H}_{\widetilde{K}}$ *of the regularized optimization problem*

$$\widetilde{f}_s = \underset{\widetilde{f} \in \mathcal{H}_{\widetilde{K}}}{\arg\min} \frac{1}{N} \sum_{i=1}^{N} c\left(\widetilde{f}(x_i), y_i\right) + \frac{\lambda}{2} \left\|\widetilde{f}\right\|_{\widetilde{K}}^2 \tag{6.8}$$

has the form $\widetilde{f}_s = \widetilde{\Phi}(\cdot)^* \theta_s$, *where* $\theta_s \in (Ker\ \widetilde{W})^\perp$ *and*

$$\theta_s = \underset{\theta \in \widetilde{\mathcal{H}}}{\arg\min} \frac{1}{N} \sum_{i=1}^{N} c\left(\widetilde{\Phi}(x_i)^* \theta, y_i\right) + \frac{\lambda}{2} \|\theta\|_{\widetilde{\mathcal{H}}}^2 \tag{6.9}$$

Proof *Since* $\widetilde{K}$ *is an operator-valued kernel, from theorem 6.2, equation 6.8 has a solution of the form*

$$\begin{aligned} \widetilde{f}_s &= \sum_{i=1}^{N} \widetilde{K}(\cdot, x_i) u_i, \quad u_i \in \mathcal{Y}, x_i \in \mathcal{X} \\ &= \sum_{i=1}^{N} \widetilde{\Phi}(\cdot)^* \widetilde{\Phi}(x_i) u_i = \widetilde{\Phi}(\cdot)^* \underbrace{\left(\sum_{i=1}^{N} \widetilde{\Phi}(x_i) u_i\right)}_{= \theta \in \left(Ker\ \widetilde{W}\right)^\perp \subset \widetilde{\mathcal{H}}}. \end{aligned}$$

Let

$$\theta_s = \underset{\theta \in \left(Ker\ \widetilde{W}\right)^\perp}{\arg\min} \frac{1}{N} \sum_{i=1}^{N} c\left(\widetilde{\Phi}(x_i)^* \theta, y_i\right) + \frac{\lambda}{2} \left\|\widetilde{\Phi}(\cdot)^* \theta\right\|_{\widetilde{K}}^2. \tag{6.10}$$

Since $\theta \in (Ker\ \widetilde{W})^\perp$ *and W is an isometry from* $(Ker\ \widetilde{W})^\perp \subset \widetilde{\mathcal{H}}$ *onto* $\mathcal{H}_{\widetilde{K}}$, *we have* $\left\|\widetilde{\Phi}(\cdot)^* \theta\right\|_{\widetilde{K}}^2 = \|\theta\|_{\widetilde{\mathcal{H}}}^2$. *Hence*

$$\theta_s = \underset{\theta \in \left(Ker\ \widetilde{W}\right)^\perp}{\arg\min} \frac{1}{N} \sum_{i=1}^{N} c\left(\widetilde{\Phi}(x_i)^* \theta, y_i\right) + \frac{\lambda}{2} \|\theta\|_{\widetilde{\mathcal{H}}}^2$$

Finding a minimizer θ_s *over* $\left(Ker\ \widetilde{W}\right)^\perp$ *is not the same as finding a minimizer over* $\widetilde{\mathcal{H}}$. *Although in both cases Mazur-Schauder's theorem guarantees that the respective minimizers are unique, they might not be the same. Since* $\widetilde{W}$ *is bounded, Ker* $\widetilde{W}$ *is closed, so that we can perform the decomposition* $\widetilde{\mathcal{H}} = \left(Ker\ \widetilde{W}\right)^\perp \oplus \left(Ker\ \widetilde{W}\right)$. *Then clearly by linearity of W and the fact that for all* $\theta^\parallel \in Ker\ \widetilde{W}$, $\widetilde{W}\theta^\parallel = 0$, *if* $\lambda > 0$ *we have*

$$\theta_s = \underset{\theta \in \widetilde{\mathcal{H}}}{\arg\min} \frac{1}{N} \sum_{i=1}^{N} c\left(\widetilde{\Phi}(x_i)^* \theta, y_i\right) + \frac{\lambda}{2} \|\theta\|_{\widetilde{\mathcal{H}}}^2$$

Thus

$$\theta_{\mathbf{s}} = \operatorname*{arg\,min}_{\substack{\theta^{\perp}\in\left(Ker\ \widetilde{W}\right)^{\perp},\\ \theta^{\|}\in Ker\ \widetilde{W}}} \frac{1}{N}\sum_{i=1}^{N} c\left(\left(\widetilde{W}\theta^{\perp}\right)(x) + \underbrace{\left(\widetilde{W}\theta^{\|}\right)(x)}_{=0 \text{ for all } \theta^{\|}}, y_i\right) + \frac{\lambda}{2}\left\|\theta^{\perp}\right\|^2_{\widetilde{\mathcal{H}}} + \underbrace{\frac{\lambda}{2}\left\|\theta^{\|}\right\|^2_{\widetilde{\mathcal{H}}}}_{=0 \text{ only if } \theta^{\|}=0}$$

Thus

$$\theta_{\mathbf{s}} = \operatorname*{arg\,min}_{\theta^{\perp}\in\left(Ker\ \widetilde{W}\right)^{\perp}} \frac{1}{N}\sum_{i=1}^{N} c\left(\left(\widetilde{W}\theta^{\perp}\right)(x), y_i\right) + \frac{\lambda}{2}\left\|\theta^{\perp}\right\|^2_{\widetilde{\mathcal{H}}}$$

Hence minimizing over $\left(Ker\ \widetilde{W}\right)^{\perp}$ or $\widetilde{\mathcal{H}}$ is the same when $\lambda > 0$. Eventually,

$$\theta_{\mathbf{s}} = \operatorname*{arg\,min}_{\theta\in\widetilde{\mathcal{H}}} \frac{1}{N}\sum_{i=1}^{N} c\left(\widetilde{\Phi}(x_i)^*\theta, y_i\right) + \frac{\lambda}{2}\|\theta\|^2_{\widetilde{\mathcal{H}}}.$$

In the aforemention theorem, we use the notation $\widetilde{K}$ and $\widetilde{\Phi}$ because our main subject of interest is the ORFF map. However this theorem works for *any* feature maps $\Phi(x) : \mathcal{L}(\mathcal{Y}, \mathcal{H})$ even when $\mathcal{H}$ is infinite dimensional.[2]. This shows that when $\lambda > 0$ the solution of equation 6.4 with the approximated kernel $K(x,z) \approx \widetilde{K}(x,z) = \widetilde{\Phi}(x)^*\widetilde{\Phi}(z)$ is the same than the solution of equation 6.10 up to a linear transformation. Namely, if $u_{\mathbf{s}}$ is the solution of equation 6.4, $\theta_{\mathbf{s}}$ is the solution of equation 6.10 and $\lambda > 0$ we have

$$\theta_{\mathbf{s}} = \sum_{i=1}^{N} \widetilde{\Phi}(x_i)(u_{\mathbf{s}})_i \in (\operatorname{Ker} W)^{\perp} \subseteq \widetilde{\mathcal{H}}.$$

If $\lambda_K = 0$ we can still find a solution $u_{\mathbf{s}}$ of equation 6.4. By construction of the kernel expansion, we have $u_{\mathbf{s}} \in (\operatorname{Ker} W)^{\perp}$. However looking at the proof of theorem 6.3 we see that $\theta_{\mathbf{s}}$ might *not* belong to $(\operatorname{Ker} W)^{\perp}$. We can compute a residual vector

$$r_{\mathbf{s}} = \sum_{i=1}^{N} \widetilde{\Phi}(x_i)(u_{\mathbf{s}})_i \in \widetilde{\mathcal{H}} - \theta_{\mathbf{s}}.$$

Since $\sum_{j=1}^{N} \widetilde{\Phi}(x_j) \in (\operatorname{Ker} W)^{\perp}$ by construction, if $r_{\mathbf{s}} = 0$, it means that λ_K is large enough for both representer theorem and ORFF representer theorem to apply. If $r_{\mathbf{s}} \neq 0$ but $\widetilde{\Phi}(\cdot)^* r_{\mathbf{s}} = 0$ it means that both $\theta_{\mathbf{s}}$ and $\sum_{j=1}^{N} \widetilde{\Phi}(x_j) u_{\mathbf{s}}$ are in $(\operatorname{Ker} W)^{\perp}$, thus the representer theorem fails to find the "true" solution over the whole space $\mathcal{H}_{\widetilde{K}}$ but returns a projection onto $\mathcal{H}_{\widetilde{K},\mathbf{s}}$ of the solution. If $r_{\mathbf{s}} \neq 0$ and $\widetilde{\Phi}(\cdot)^* r_{\mathbf{s}} \neq 0$ means that $\theta_{\mathbf{s}}$ is *not* in $(\operatorname{Ker} W)^{\perp}$, thus the feature equivalence theorem fails to apply. Since $r_{\mathbf{s}} = \sum_{i=1}^{N} \widetilde{\Phi}(x_i)(u_{\mathbf{s}})_i - \theta_{\mathbf{s}}^{\perp} - \theta_{\mathbf{s}}^{\|}$ and $\sum_{i=1}^{N} \widetilde{\Phi}(x_i)(u_{\mathbf{s}})_i$ is in $(\operatorname{Ker} W)^{\perp}$, with mild abuse of notation we write $r_{\mathbf{s}} = \theta^{\perp}$. This remark is illustrated in figure 6.2.

2 If $\Phi(x) : \mathcal{L}(\mathcal{Y}, \mathcal{H})$ and $\dim(\mathcal{H}) = \infty$, the decomposition $\mathcal{H} = (\operatorname{Ker} W) \oplus (\operatorname{Ker} W)^{\perp}$ holds since $\mathcal{H}$ is a Hilbert space and W is a closed operator.

6.2 SOLUTION OF THE EMPIRICAL RISK MINIMIZATION

In order to find a solution to equation 6.10, we turn our attention to gradient descent methods. In the following we let

$$\mathcal{J}_\lambda(\theta) = \frac{1}{N}\sum_{i=1}^{N} c\left(\widetilde{\Phi}(x_i)^*\theta, y_i\right) + \frac{\lambda}{2}\|\theta\|^2_{\widetilde{\mathcal{H}}} \tag{6.11}$$

6.2.1 *Gradient methods*

Since the solution of equation 6.10 is unique when $\lambda > 0$, a sufficient and necessary condition is that the gradient of $\mathcal{J}_\lambda$ at the minimizer θ_s is zero. We use the Frechet derivative, the strongest notion of derivative in Banach spaces[15] [38, 78] which directly generalizes the notion of gradient to Banach spaces. A function $f : \mathcal{H}_0 \to \mathcal{H}_1$ is call Frechet differentiable at $\theta_0 \in \mathcal{H}_0$ if there exist a bounded linear operator $A \in \mathcal{L}(\mathcal{H}_0, \mathcal{H}_1)$ such that

[15] *Here we view the Hilbert space $\mathcal{H}$ (feature space) as a reflexive Banach space.*

$$\lim_{\|h\|_{\mathcal{H}_0}\to 0} \frac{\|f(\theta_0 + h) - f(\theta_0) - Ah\|_{\mathcal{H}_1}}{\|h\|_{\mathcal{H}_0}} = 0$$

We write

$$(D_F f)(\theta_0) = \left.\frac{\partial f(\theta)}{\partial \theta}\right|_{\theta=\theta_0} = A$$

and call it Frechet derivative of f with respect to θ at θ_0. With mild abuse of notation we write

$$\left.\frac{\partial f(\theta)}{\partial \theta}\right|_{\theta=\theta_0} = \frac{\partial f(\theta_0)}{\partial \theta_0}.$$

The chain rule is valid in this context [78, theorem 4.1.1 page 140]. Namely, let $\mathcal{H}_0$, $\mathcal{H}_1$ and $\mathcal{H}_2$ be three Hilbert spaces. If a function $f : \mathcal{H}_0 \to \mathcal{H}_1$ is Frechet differentiable at θ and $g : \mathcal{H}_1 \to \mathcal{H}_2$ is Frechet differentiable at $f(\theta)$ then $g \circ f$ is Frechet differentiable at θ and for all $h \in \mathcal{H}_0$

$$\frac{\partial}{\partial \theta}(g \circ f)(\theta) \circ h = \frac{\partial g(f(\theta))}{\partial f(\theta)} \circ \frac{\partial f(\theta)}{\partial \theta} \circ h,$$

or equivalently,

$$D_F(g \circ f)(\theta) \circ h = (D_F g)(f(\theta)) \circ (D_F f)(\theta) \circ h.$$

If $f : \mathcal{H} \to \mathbb{R}$ then $(D_F f)(\theta_0) \in \mathcal{H}^*$ for all $\theta_0 \in \mathcal{H}$, and by Riesz's representation theorem we define the gradient of f noted $\nabla_\theta f(\theta) \in \mathcal{H}$ as the the vector in $\mathcal{H}$ such that

$$\langle \nabla_\theta f(\theta), h\rangle_{\mathcal{H}} = (D_F f)(\theta) \circ h = \frac{\partial f(\theta)}{\partial \theta} \circ h.$$

For a function $f : \mathcal{H}_0 \to \mathcal{H}_1$ we note the jacobian of f as $\mathbf{J}_\theta f(\theta) = \frac{\partial f(\theta)}{\partial \theta}$. In this context if $f : \mathcal{H}_0 \to \mathcal{H}_1$ and $g : \mathcal{H}_1 \to \mathbb{R}$ the chain rule reads for all $h \in \mathcal{H}_0$

$$\frac{\partial}{\partial \theta}(g \circ f)(\theta) \circ h = \frac{\partial g(f(\theta))}{\partial f(\theta)} \circ \mathbf{J}_\theta f(\theta) \circ h.$$

By Riesz's representation theorem,

$$\begin{aligned} \langle \nabla_\theta (g \circ f)(\theta), h \rangle_{\mathcal{H}_0} &= \langle \nabla_{f(\theta)} g(f(\theta)), \mathbf{J}_\theta f(\theta) h \rangle_{\mathcal{H}_0} \\ &= \left\langle \left(\mathbf{J}_\theta f(\theta)\right)^* \nabla_{f(\theta)} g(f(\theta)), h \right\rangle_{\mathcal{H}_0} \end{aligned}$$

Hence

$$\nabla_\theta (g \circ f)(\theta) = \left(\mathbf{J}_\theta f(\theta)\right)^* \nabla_{f(\theta)} g(f(\theta)).$$

Thus by linearity and applying the chaine rule to equation 6.10 we have

$$\begin{aligned} \nabla_\theta c\left(\widetilde{\Phi}(x_i)^* \theta, y_i\right) &= \widetilde{\Phi}(x_i) V^* \left(\frac{\partial}{\partial y} c\left(y, y_i\right) \Big|_{y = \widetilde{\Phi}(x_i)^* \theta} \right)^*, \\ \nabla_\theta \|\theta\|_{\widetilde{\mathcal{H}}}^2 &= 2\theta. \end{aligned}$$

Provided that $c(y, y_i)$ is Frechet differentiable w. r. t. y, for all y and $y_i \in \mathcal{Y}$ we have $\nabla_\theta \mathcal{J}_\lambda(\theta) \in \widetilde{\mathcal{H}}$ and

$$\nabla_\theta \mathcal{J}_\lambda(\theta) = \frac{1}{N} \sum_{i=1}^N \widetilde{\Phi}(x_i) \left(\frac{\partial}{\partial y} c\left(y, y_i\right) \Big|_{y = \widetilde{\Phi}(x_i)^* \theta} \right)^* + \lambda \theta$$

Example 6.1 (Naive closed form for the squared error cost). *Consider the cost function defined for all $y, y' \in \mathcal{Y}$ by $c(y, y') = \frac{1}{2} \|y - y\|_{\mathcal{Y}}^2$. Then*

$$\left(\frac{\partial}{\partial y} c\left(y, y_i\right) \Big|_{y = \widetilde{\Phi}(x_i)^* \theta} \right)^* = \left(\widetilde{\Phi}(x_i)^* \theta - y_i \right).$$

Thus, since the optimal solution $\theta_{\mathbf{s}}$ verifies $\nabla_{\theta_{\mathbf{s}}} \mathcal{J}_\lambda(\theta_{\mathbf{s}}) = 0$ we have

$$\frac{1}{N} \sum_{i=1}^N \widetilde{\Phi}(x_i) \left(\widetilde{\Phi}(x_i)^* \theta_{\mathbf{s}} - y_i \right) + \lambda \theta_{\mathbf{s}} = 0.$$

Therefore,

$$\left(\frac{1}{N} \sum_{i=1}^N \widetilde{\Phi}(x_i) \widetilde{\Phi}(x_i)^* + \lambda I_{\widetilde{\mathcal{H}}} \right) \theta_{\mathbf{s}} = \frac{1}{N} \sum_{i=1}^N \widetilde{\Phi}(x_i) y_i. \tag{6.12}$$

Suppose that $\mathcal{Y} \subseteq \mathbb{R}^p$, and for all $x \in \mathcal{X}$, $\widetilde{\Phi}(x) : \mathbb{R}^r \to \mathbb{R}^p$ where all spaces are endowed with the euclidean inner product. From this we can derive algorithm 4 which returns the closed form solution of equation 6.11 for $c(y, y') = \frac{1}{2} \|y - y'\|_2^2$.

If one considers a Mahalanobis inner product, evaluation of operators has to be done with extra care since the adjoint operator is *not* the classic conjugate transpose of the operator (see remark 6.2). Indeed let $x, z \in \mathcal{Y} = \mathbb{C}^p$ endowed with its standard basis B, and $\langle x, y\rangle_{\mathcal{Y}} = \langle x, \Sigma^{-1}z\rangle_2$ where Σ is some symmetric positive-definite operator w. r. t. the basis B. Some simple calculations shows that given an operator $A \in \mathcal{L}(\mathcal{Y})$,

$$(A^*)_{ij} := \langle e_j, \Sigma^{-1}A^*e_i\rangle_2 = \overline{\langle \Sigma^{-1}A^*e_i, e_j\rangle_2} = \overline{\langle e_i, A\Sigma^{-1}e_j\rangle_2} := \overline{(\Sigma A\Sigma^{-1})_{ji}}$$

Thus $A^* = \Sigma^{-1}\overline{A}^{\mathsf{T}}\Sigma$.

Remark 6.2 *Notice that the evaluation of each operator $\nabla_\theta \mathfrak{J}_\lambda(\theta)$, V^*, $\widetilde{\Phi}(x_i)^*$'s and M_{ik}'s depends on the inner product of the respective spaces in which they are defined. Namely $\mathcal{Y}$, and $\widetilde{\mathcal{H}}$. For instance if one chooses $\widetilde{\mathcal{H}} = \bigoplus_{j=1}^D \mathcal{Y}'$, $\mathcal{Y}' = \mathbb{R}^{u'}$ endowed with the Euclidean inner product $\langle \theta', \theta\rangle_{\mathcal{Y}'} = \langle \theta', \theta\rangle_2$, $\mathcal{Y}$ endowed with a Mahalanobis inner product $\langle u', u\rangle_{\mathcal{U}} = \langle u', \Sigma^{-1}u\rangle_2$ where Σ is some symmetric positive definite operator, then for all $x \in \mathcal{X}$,*

$$\widetilde{\Phi}(x)_{ij} = \langle e_j \widetilde{\Phi}(x) e_i\rangle_2 = \langle e_i, \Sigma^{-1}\Sigma\widetilde{\Phi}(x)^* e_j\rangle_2 = (\Sigma\widetilde{\Phi}(x)^*)_{ji}.$$

Thus $\widetilde{\Phi}(x)^* = \Sigma^{-1}\widetilde{\Phi}(x)^{\mathsf{T}}$ Tthen equation 6.12 reads

$$\left(\frac{1}{N}\sum_{i=1}^N \widetilde{\Phi}(x_i)\Sigma^{-1}\widetilde{\Phi}(x_i)^{\mathsf{T}} + \lambda I_{\widetilde{\mathcal{H}}}\right)\theta_s = \frac{1}{N}\sum_{i=1}^N \widetilde{\Phi}(x_i)y_i.$$

6.2.2 *Complexity analysis*

Algorithm 4 constitutes our first step toward large-scale learning with Operator-Valued Kernels. We can easily compute the time complexity of algorithm 4 when all the operators act on finite dimensional Hilbert spaces. Suppose that $p = \dim(\mathcal{Y}) < \infty$ and for all $x \in \mathcal{X}$, $\widetilde{\Phi}(x) : \mathcal{Y} \to \widetilde{\mathcal{H}}$ where $r = \dim(\widetilde{\mathcal{H}}) < \infty$ is the dimension of the redescription space $\widetilde{\mathcal{H}} = \mathbb{R}^r$. Since p and $r < \infty$, we view the operators $\widetilde{\Phi}(x)$ and $I_{\widetilde{\mathcal{H}}}$ as matrices. Step 1 costs $O_t(Nr^2p)$. Steps 2 costs $O_t(Nrp)$. For step 3, the naive inversion of the operator costs $O_t(r^3)$. Eventually the overall complexity of algorithm 4 is

$$O_t\left(r^2(Np + r)\right),$$

while the space complexity is $O_s(r^2)$. This complexity is to compare with the kernelized solution. Let

$$\mathbf{K} : \begin{cases} \mathcal{Y}^N \to \mathcal{Y}^N \\ u \mapsto \bigoplus_{i=1}^{N+U} \sum_{j=1}^{N+U} K(x_i, x_j)u_j \end{cases}$$

When $\mathcal{Y} = \mathbb{R}$,

$$\mathbf{K} = \begin{pmatrix} K(x_1, x_1) & \dots & K(x_1, x_{N+U}) \\ \vdots & \ddots & \vdots \\ K(x_{N+U}, x_1) & \dots & K(x_{N+U}, x_{N+U}) \end{pmatrix}$$

Algorithm 4: Naive closed form for the squared error cost.

Input :

- $\mathbf{s} = (x_i, y_i)_{i=1}^N \in \left(\mathcal{X} \times \mathbb{R}^p\right)^N$ a sequence of supervised training points,
- $\widetilde{\Phi}(x_i) \in \mathcal{L}\left(\mathbb{R}^p, \mathbb{R}^r\right)$ a feature map defined for all $x_i \in \mathcal{X}$,
- $\lambda \in \mathbb{R}_{>0}$ the Tychonov regularization term,

Output : A model

$$h : \begin{cases} \mathcal{X} \to \mathbb{R}^p \\ x \mapsto \widetilde{\Phi}(x)^\intercal \theta_{\mathbf{s}}, \end{cases}$$

such that $\theta_{\mathbf{s}}$ minimize equation 6.11, where $c(y, y') = \|y - y'\|_2^2$ and $\mathbb{R}^r$ and $\mathbb{R}^p$ are Hilbert spaces endowed with the euclidean inner product.

$\mathbf{P} \leftarrow \frac{1}{N}\sum_{i=1}^N \widetilde{\Phi}(x_i)\widetilde{\Phi}(x_i)^\intercal \in \mathcal{L}(\mathbb{R}^r, \mathbb{R}^r)$;
$\mathbf{Y} \leftarrow \frac{1}{N}\sum_{i=1}^N \widetilde{\Phi}(x_i)y_i \in \mathbb{R}^r$;
$\theta_{\mathbf{s}} \leftarrow \text{solve}_\theta\left((\mathbf{P} + \lambda I_r)\theta = \mathbf{Y}\right)$;
return $h : x \mapsto \widetilde{\Phi}(x)^\intercal \theta_{\mathbf{s}}$;

is called the Gram matrix of K. When $\mathcal{Y} = \mathbb{R}^p$, $\mathbf{K}$ is a matrix-valued Gram matrix of size $pN \times pN$ where each entry $\mathbf{K}_{ij} \in \mathcal{M}_{p,p}(\mathbb{R})$. Then the equivalent kernelized solution $u_{\mathbf{s}}$ of theorem 6.2 is

$$\left(\frac{1}{N}\mathbf{K} + \lambda I_{\bigoplus_{i=1}^N \mathcal{Y}}\right) u_{\mathbf{s}} = \bigoplus_{i=1}^N y_i.$$

which has time complexity $O\left(N^3 p^3\right)$ and space complexity $O_s\left(N^2 p^2\right)$. Suppose we are given a generic ORFF map (see subsection 4.3.3). Then $r = 2Dp$, where D is the number of samples. Hence algorithm 4 is better that its kernelized counterpart when $r = 2Dp$ is small compared to Np. Thus, roughly speaking it is better to use algorithm 4 when the number of features, r, required is small compared to the number of training points. Notice that algorithm 4 has a linear complexity with respect to the number of supervised training points N so it is better suited to large scale learning provided that D does not grows linearly with N.

Yet naive learning with algorithm 4 by viewing all the operators as matrices is still problematic. Indeed learning p independent models with scalar Random Fourier Features would cost $O_t\left(D^2 p^3 (N + D)\right)$ since $r = 2Dp$. This Means that learning vector-valued function has increased the (expected) complexity from p to p^3. However in some cases we can drastically reduce the complexity by viewing the feature-maps as linear operators rather than matrices.

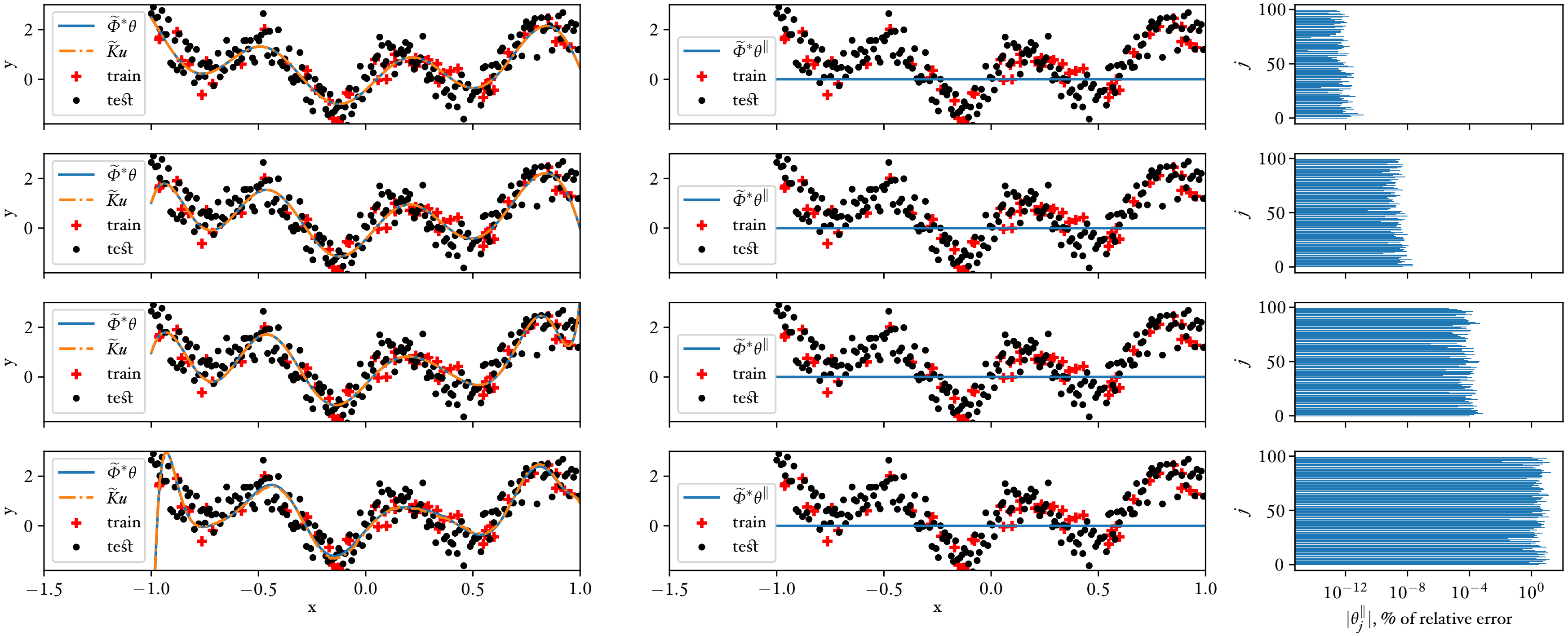

Figure 6.1: ORFF equivalence theorem. We trained a first model named $\widetilde{K}$ following

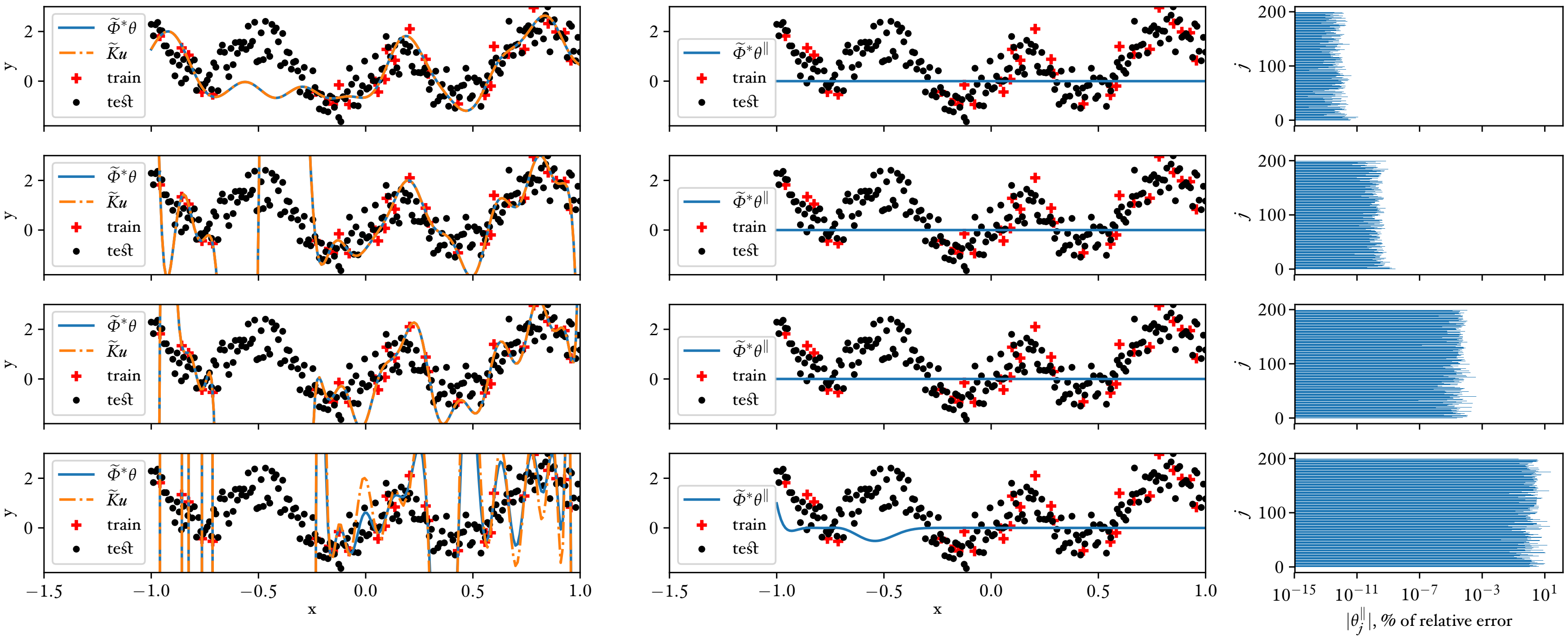

Figure 6.2: ORFF equivalence theorem. We trained a first model named $\widetilde{K}$ following

6.2.3 *Efficient matrix-free operators*

When developping algorithm 4 we considered that the feature map $\widetilde{\Phi}(x)$ was a matrix from $\mathbb{R}^p$ to $\mathbb{R}^r$ for all $x \in \mathcal{X}$, and therefore that computing $\widetilde{\Phi}(x)\widetilde{\varphi}(z)^\top$ has a time complexity of $O(r^2 p)$. While this holds true in the most generic senario, in many cases the feature maps present some structure or sparsity allowing to reduce the computational cost of evaluating the feature map. We focus on the Operator-valued Random Fourier Feature given by algorithm 2, developped in section 4.3 and subsection 4.3.3 and treat the decomposable kernel, the curl-free kernel and the divergence-free kernel as an example. We recall that if $\mathcal{Y}' = \mathbb{R}^{p'}$ and $\mathcal{Y} = \mathbb{R}^p$, then $\widetilde{\mathcal{H}} = \mathbb{R}^{2Dp'}$ thus the Operator-valued Random Fourier Features given in chapter 4 have the form

$$\begin{cases} \widetilde{\Phi}(x) \in \mathcal{L}\left(\mathbb{R}^p, \mathbb{R}^{2Dp'}\right) & : y \mapsto \frac{1}{\sqrt{D}} \bigoplus_{j=1}^{D} (x, \omega_j) B(\omega_j)^\top y \\ \widetilde{\Phi}(x)^\top \in \mathcal{L}\left(\mathbb{R}^{2Dp'}, \mathbb{R}^p\right) & : \theta \mapsto \frac{1}{\sqrt{D}} \sum_{j=1}^{D} (x, \omega_j) B(\omega_j) \theta_j \end{cases},$$

where $\omega_j \sim \mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho}$ i. i. d. and $B(\omega_j) \in \mathcal{L}\left(\mathbb{R}^p, \mathbb{R}^{p'}\right)$ for all $\omega_j \in \widehat{\mathcal{X}}$. Hence the Operator-valued Random Fourier Feature can be seen as the block matrix

$$\widetilde{\Phi}(x) = \begin{pmatrix} \cos\langle x, \omega_1 \rangle B(\omega_1)^\top \\ \sin\langle x, \omega_1 \rangle B(\omega_1)^\top \\ \vdots \\ \cos\langle x, \omega_D \rangle B(\omega_D)^\top \\ \sin\langle x, \omega_D \rangle B(\omega_D)^\top \end{pmatrix} \in \mathcal{M}_{2Dp',p}(\mathbb{R}), \tag{6.13}$$

$\omega_j \sim \mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho}$ i. i. d..

6.2.4 *Case of study: the decomposable kernel*

Troughout this section we show how the mathematical formulation relates to a concrete (Python) implementation. We propose a Python implementation based on NumPy [109], SciPy [72] and Scikit-learn [114]. Following equation 6.13, the feature map associated to the decomposable kernel would be

$$\widetilde{\Phi}(x) = \frac{1}{\sqrt{D}} \begin{pmatrix} \cos\langle x, \omega_1 \rangle B^\top \\ \sin\langle x, \omega_1 \rangle B^\top \\ \vdots \\ \cos\langle x, \omega_D \rangle B^\top \\ \sin\langle x, \omega_D \rangle B^\top \end{pmatrix} = \underbrace{\frac{1}{\sqrt{D}} \begin{pmatrix} \cos\langle x, \omega_1 \rangle \\ \sin\langle x, \omega_1 \rangle \\ \vdots \\ \cos\langle x, \omega_D \rangle \\ \sin\langle x, \omega_D \rangle \end{pmatrix}}_{\widetilde{\varphi}(x)} \otimes B^\top,$$

$\omega_j \sim \mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho}$ i. i. d., which would lead to the following naive python implementation for the Gaussian (RBF) kernel of parameter γ, whose associated spectral distribution is $\mathbf{Pr}_\rho = \mathcal{N}(0, 2\gamma)$.

```
def NaiveDecomposableGaussianORFF(X, A, gamma=1.,
                                  D=100, eps=1e-5, random_state=0):
    r"""Return the Naive ORFF map associated with the data X.

    Parameters
    ----------
    X : {array-like}, shape = [n_samples, n_features]
        Samples.
    A : {array-like}, shape = [n_targets, n_targets]
        Operator of the Decomposable kernel (positive semi-definite)
    gamma : {float},
        Gamma parameter of the RBF kernel.
    D : {integer}
        Number of random features.
    eps : {float}
        Cutoff threshold for the singular values of A.
    random_state : {integer}
        Seed of the generator.

    Returns
    -------
    \tilde{\Phi}(X) : array
    """
    # Decompose A=BB^\transpose
    u, s, v = svd(A, full_matrices=False, compute_uv=True)
    B = dot(diag(sqrt(s[s > eps])), v[s > eps, :])

    # Sample a RFF from the scalar Gaussian kernel
    phi_s = RBFSampler(gamma=gamma, n_components=D, random_state=random_state)
    phiX = phi_s.fit_transform(X)

    # Create the ORFF linear operator
    return matrix(kron(phiX, B))
```

Let $\theta \in \mathbb{R}^{2Dp'}$ and $y \in \mathbb{R}$. With such imlementation evaluating a matrix vector product such as $\tilde{\Phi}(x)^\intercal \theta$ or $\tilde{\Phi}(x)y$ have $O_t(2Dp'p)$ time complexity and $O_s(2Dp'p)$ of space complexity, which is utterly inefficient. Indeed, recall that if $B \in \mathcal{M}_{p,p'}\left(\mathbb{R}^{p'}\right)$ is matrix, the operator $\tilde{\Phi}(x)$ corresponding to the decomposable kernel is

$$\begin{aligned}\tilde{\Phi}(x)y &= \frac{1}{\sqrt{D}} \bigoplus_{j=1}^{D} \begin{pmatrix} \cos\langle x, \omega_j\rangle B^\intercal y \\ \sin\langle x, \omega_j\rangle B^\intercal y \end{pmatrix} \\ &= \left(\frac{1}{\sqrt{D}} \bigoplus_{j=1}^{D} \begin{pmatrix} \cos\langle x, \omega_j\rangle \\ \sin\langle x, \omega_j\rangle \end{pmatrix} \right) \otimes (B^\intercal y)\end{aligned}$$

and

$$\begin{aligned}\tilde{\Phi}(x)^\intercal \theta &= \frac{1}{\sqrt{D}} \sum_{j=1}^{D} \cos\langle x, \omega_j\rangle B\theta_j + \sin\langle x, \omega_j\rangle B\theta_j \\ &= B \left(\frac{1}{\sqrt{D}} \sum_{j=1}^{D} \left(\cos\langle x, \omega_j\rangle + \sin\langle x, \omega_j\rangle\right) \theta_j \right).\end{aligned} \tag{6.14}$$

Which requires only evaluation of B on y and can be implemented easily in Python thanks to SciPy's LinearOperator. Note that the computation of these expressions can be fully vectorized[16] using the vectorization property of the Kronecker product. In the following we consider $\Theta \in \mathcal{M}_{2D,u'}(\mathbb{R})$ and the operator $\mathbf{vec} : \mathcal{M}_{p',2D}(\mathbb{R}) \to \mathbb{R}^{2Dp'}$ which turns a matrix into a vector (i. e. $\theta_{p'i+j} = \mathbf{vec}(\Theta_{ij})$, $i \in \mathbb{N}_{2D}$ and $j \in \mathbb{N}^*_{p'}$). Then

$$\left(\widetilde{\varphi}(x) \otimes B^\mathsf{T}\right)^\mathsf{T} \theta = \left(\widetilde{\varphi}(x)^\mathsf{T} \otimes B\right) \mathbf{vec}(\Theta) = \mathbf{vec}\left(B\Theta\widetilde{\varphi}(x)\right).$$

with this trick, many authors [25, 124, 139] notice that the decomposable kernel usually yields a Stein equation [116]. Indeed rewriting step 3 of algorithm 4 gives a system to solve of the form

$$\widetilde{\varphi}(X)\widetilde{\varphi}(X)^\mathsf{T}\Theta B^\mathsf{T} B + \lambda\Theta - \Upsilon = 0. \Leftrightarrow \left(\widetilde{\varphi}X)\widetilde{\varphi}X)^\mathsf{T} \otimes B^\mathsf{T} B + \lambda I_{2Dp'}\right)\theta - \Upsilon = 0$$

Many solvers exists to solve efficiently this kind of systems[17] , but most of them share the particularity that they are not just restricted to handle Stein equations. Broadly speaking, iterative solvers (or matrix free solvers) are designed to solve any systems of equation of ther form $PX = C$, where P is a linear operator (not a matrix). This is exacly our case where $\widetilde{\varphi}(x) \otimes B^T$ is the matrix form of the operator $\Theta \mapsto \mathbf{vec}(B\Theta\widetilde{\varphi}X)$.

This leads us to the following (more efficient) Python implementation of the Decomposable ORFF "operator" to be feed to a matrix-free solvers.

```python
def EfficientDecomposableGaussianORFF(X, A, gamma=1.,
                                      D=100, eps=1e-5, random_state=0):
    r"""Return the efficient ORFF map associated with the data X.

    Parameters
    ----------
    X : {array-like}, shape = [n_samples, n_features]
        Samples.
    A : {array-like}, shape = [n_targets, n_targets]
        Operator of the Decomposable kernel (positive semi-definite)
    gamma : {float},
        Gamma parameter of the RBF kernel.
    D : {integer}
        Number of random features.
    eps : {float}
        Cutoff threshold for the singular values of A.
    random_state : {integer}
        Seed of the generator.

    Returns
    -------
    \tilde{\Phi}(X) : Linear Operator, callable
    """
    # Decompose A=BB^\transpose
    u, s, v = svd(A, full_matrices=False, compute_uv=True)
    B = dot(diag(sqrt(s[s > eps])), v[s > eps, :])

    # Sample a RFF from the scalar Gaussian kernel
    phi_s = RBFSampler(gamma=gamma, n_components=D, random_state=random_state)
    phiX = phi_s.fit_transform(X)

    # Create the ORFF linear operator
```

16 See Walt, Colbert, and Varoquaux [164].

17 for instance Sleijpen, Sonneveld, and Van Gijzen [141]

```
    cshape = (D, B.shape[0])
    rshape = (X.shape[0], B.shape[1])
    return LinearOperator((phiX.shape[0] * B.shape[1], D * B.shape[0]),
                          matvec=lambda b: dot(phiX, dot(b.reshape(cshape),
                                                         B)),
                          rmatvec=lambda r: dot(phiX.T, dot(r.reshape(rshape),
                                                          B.T)))
```

6.2.5 *Linear operators in matrix form*

For convenience we give the operators corresponding to the decomposable, curl-free and divergence-free kernels in matrix form. Let $(x_i)_{i=1}^N$, $N \in \mathbb{N}^*$, x_i's in $\mathbb{R}^d$, $d \leq \infty$ be a sequence of points in $\mathbb{R}^d$. We note

$$X = \begin{pmatrix} x_1 & \dots & x_N \end{pmatrix} \in \mathcal{M}_{d,N}$$

the data matrix where each column represents a data point[3]. Naturally if $\widetilde{\Phi}(x) : \mathbb{R}^u \to \mathbb{R}^{r_1}$ and $\widetilde{\varphi}(x) : \mathbb{R} \to \mathbb{R}^{r_2}$, for all $x \in \mathbb{R}^d$ we define

$$\widetilde{\Phi}(X) = \begin{pmatrix} \widetilde{\Phi}(x_1) & \dots & \widetilde{\Phi}(x_N) \end{pmatrix} \in \mathcal{M}_{r_1,Nu}$$

and

$$\widetilde{\varphi}(X) = \begin{pmatrix} \widetilde{\varphi}(x_1) & \dots & \widetilde{\varphi}(x_N) \end{pmatrix} \in \mathcal{M}_{r_2,N}$$

and

$$U = \begin{pmatrix} u_1 & \dots & u_N \end{pmatrix} \in \mathcal{M}_{u,N}.$$

Given a matrix $X \in \mathcal{M}_{m,n}(\mathbb{R})$, we note $X_{\bullet i}$ the *column* vector coresponding to the i-th column of the matrix X and $X_{i\bullet}$ the *row* vector (covector) corresponding to the i-th line of the matrix X. With these notations, if $X \in \mathcal{M}_{m,n}$ and $Z \in \mathcal{M}_{n,m'}$, $X_{i\bullet}Z_{\bullet j} \in \mathbb{R}$ is the inner product between the i-th row of X and the j-th column of Z and $X_{\bullet i}Z_{j\bullet} \in \mathcal{M}_{m,m'}(\mathbb{R})$ is the outer product between the i-th column of X and j-th row of X.

For the curl-free and divergence-free kernel given in subsection 4.3.3 we recall the unbounded ORFF maps are respectively for all $y \in \mathcal{Y}$

$$\widetilde{\Phi}(x)y = \frac{1}{\sqrt{D}} \bigoplus_{j=1}^{D} \begin{pmatrix} \cos \langle x, \omega_j \rangle_2 \omega_j^\top y \\ \sin \langle x, \omega_j \rangle_2 \omega_j^\top y \end{pmatrix},$$

3 In many programing language, such as Python, C, C++ or Java each data point is traditionally represented by a row in the data matrix (row major formulation). While this is more natural when parsing a data file, it is less common in mathematical formulations. In this document we adopt the *column major* formulation used by Matlab, Fortran or Julia. Moreover although C++ is commonly row major, some libraries such as Eigen are column major. When dealing with row major formulation, one should "transpose" all the equations given in table 6.3.

and

$$\widetilde{\Phi}(x)y = \frac{1}{\sqrt{D}} \bigoplus_{j=1}^{D} \begin{pmatrix} \cos \left\langle x, \omega_j \right\rangle_2 \left(\left\| \omega_j \right\|_2 I_d - \frac{\omega_j \omega_j^\top}{\left\| \omega_j \right\|_2} \right) y \\ \sin \left\langle x, \omega_j \right\rangle_2 \left(\left\| \omega_j \right\|_2 I_d - \frac{\omega_j \omega_j^\top}{\left\| \omega_j \right\|_2} \right) y \end{pmatrix},$$

where $\omega_j \sim \mathbf{Pr}_{\mathcal{N}(0, \sigma^{-2} I_d)}$. To avoid complex index notations we decompose the feature maps $\widetilde{\Phi}(X)$ into two sub feature maps $\widetilde{\Phi}^c$ and $\widetilde{\Phi}^s$ corresponding to the cosine part and the sine part of each feature map. Namely, for the curl-free kernel, for all $y \in \mathcal{Y}$

$$\widetilde{\Phi}(x)y = \begin{cases} \widetilde{\Phi}^c(x)y = \frac{1}{\sqrt{D}} \bigoplus_{j=1}^{D} \left(\cos \left\langle x, \omega_j \right\rangle_2 \omega_j^\top y \right), \\ \widetilde{\Phi}^s(x)y = \frac{1}{\sqrt{D}} \bigoplus_{j=1}^{D} \left(\sin \left\langle x, \omega_j \right\rangle_2 \omega_j^\top y \right). \end{cases}$$

In the same way, for the divergence-free kernel,

$$\widetilde{\Phi}(x)y = \begin{cases} \widetilde{\Phi}^c(x)y = \frac{1}{\sqrt{D}} \bigoplus_{j=1}^{D} \left(\cos \left\langle x, \omega_j \right\rangle_2 \left(\left\| \omega_j \right\|_2 I_d - \frac{\omega_j \omega_j^\top}{\left\| \omega_j \right\|_2} \right) y \right), \\ \widetilde{\Phi}^s(x)y = \frac{1}{\sqrt{D}} \bigoplus_{j=1}^{D} \left(\sin \left\langle x, \omega_j \right\rangle_2 \left(\left\| \omega_j \right\|_2 I_d - \frac{\omega_j \omega_j^\top}{\left\| \omega_j \right\|_2} \right) y \right). \end{cases}$$

We also introduce $\widetilde{\Phi}^e, e \in \{ s, c \}$ which denotes either $\widetilde{\Phi}^s$ or $\widetilde{\Phi}^c$. This equivalent formulation allows us to keep the notation "lighter" and closer to a proper Python/Matlab implementation with vectorization. With these notations, a summary of efficient linear operators in matrix form is given in table 6.3. The complexity of evaluating all this operators is given in table 6.1.

It is worth mentioning that the same strategy can be applied in many different language. For instance in C++, the libray Eigen [66] allows to wrap a sparse matrix with a custom type, where the user overloads the transpose and dot product operator (as in Python). Then the custom user operator behaves as a (sparse) matrix –see `https://eigen.tuxfamily.org/dox/group__MatrixfreeSolverExample.html`. With this implementation the time complexity of $\widetilde{\Phi}(x)^\top \theta$ and $\widetilde{\Phi}(x)y$ falls down to $O_t(2Dp' + p'[)$ and the same holds for space complexity.

Table 6.1: Complexity of efficient linear-operator (in matrix form) for different Feature maps given in table 6.3.

Kernel	$\widetilde{\Phi}(X)^*$	$\widetilde{\Phi}(X)$
Decomposable	$O\left((p'D + p'p)N\right)$	$O\left((pN + p'p)D\right)$
Curl-free	$O\left(pND\right)$	$O\left(pND\right)$
Divergence-free	$O\left((p^2 + pN)D\right)$	$O\left((p^2 + pN)D\right)$

Table 6.3: Efficient linear-operator (in matrix form) for different Feature maps.

Kernel	$\widetilde{\Phi}(X)^*$	$\widetilde{\Phi}(X)$
Decomposable[1]	$\Theta \mapsto B\left(\Theta\widetilde{\varphi}(X)\right)$	$\Upsilon \mapsto B^\intercal\left(\Upsilon\widetilde{\varphi}(X)^\intercal\right)$
Gaussian curl-free[2]	$\Theta^c, \Theta^s \mapsto \sum_{j=1}^{D} \omega_j \left(\Theta^c_j \widetilde{\varphi}^c(X)_{j\bullet} + \Theta^s_j \widetilde{\varphi}^s(X)_{j\bullet}\right)$	$\Upsilon \mapsto \Theta^e_j = \omega_j^\intercal \left(\Upsilon \widetilde{\varphi}^e(X)^\intercal_{\bullet j}\right)$
Gaussian divergence-free[2,3]	$\Theta^c, \Theta^s \mapsto \sum_{j=1}^{D} \left(B(\omega_j)\Theta^c_{\bullet j}\right) \widetilde{\varphi}^c(X)_{j\bullet} + \left(B(\omega_j)\Theta^s_{\bullet j}\right) \widetilde{\varphi}^s(X)_{j\bullet}$	$\Upsilon \mapsto \Theta^e_{\bullet j} = B(\omega_j) \left(\Upsilon \widetilde{\varphi}^e(X)^\intercal_{\bullet j}\right)$

[1] Where $\widetilde{\varphi}(X) = \begin{pmatrix} \widetilde{\varphi}(X_{\bullet 1}) & \dots & \widetilde{\varphi}(X_{\bullet N}) \end{pmatrix} \in \mathcal{M}_{r,N}$ is any design matrix, with scalar feature map $\widetilde{\varphi} : \mathbb{R}^d \to \mathbb{R}^r$ such that $\widetilde{\varphi}(x)^* \widetilde{\varphi}(z) = k(x, z) \in \mathbb{R}$ for all x, $z \in \mathcal{X}$. The input data $X \in \mathcal{M}_{d,N}(\mathbb{R})$, the output data $U \in \mathcal{M}_{p,N}(\mathbb{R})$, the parameter matrices Θ^c and $\Theta^s \in \mathcal{M}_{p',r}(\mathbb{R})$ and the decomposable operator $B \in \mathcal{M}_{p,p'}(\mathbb{R})$.

[2] Where $\widetilde{\varphi}^c(X)_{ji} = \cos\langle \omega_j, x_i \rangle$ and $\widetilde{\varphi}^s(X)_{ji} = \sin\langle \omega_j, x_i \rangle$, $j \in \mathbb{N}^*_D$ and $i \in \mathbb{N}^*_N$. Thus $\widetilde{\varphi}^c(X) \in \mathcal{M}_{D,N}(\mathbb{R})$ and $\widetilde{\varphi}^s(X) \in \mathcal{M}_{D,N}(\mathbb{R})$. The input data $X \in \mathcal{M}_{d,N}(\mathbb{R})$, the output data $U \in \mathcal{M}_{d,N}(\mathbb{R})$, the parameter matrices Θ^c and $\Theta^s \in \mathbb{R}^D$, $\omega_j \sim \mathbf{Pr}_{\mathcal{N}(0,\sigma^{-2}I_d)}$ i. i. d. for all $j \in \mathbb{N}^*_D$. Eventually $e \in \{ s, c \}$, namely $\Theta^c = \begin{pmatrix} \Theta^{e=c}_1 & \dots & \Theta^{e=c}_D \end{pmatrix}^\intercal$ and $\Theta^s = \begin{pmatrix} \Theta^{e=s}_1 & \dots & \Theta^{e=s}_D \end{pmatrix}^\intercal$.

[3] Here, Θ^c and $\Theta^s \in \mathcal{M}_{d,D}(\mathbb{R})$ thus $\Theta^c = \begin{pmatrix} \Theta^{e=c}_{\bullet 1} & \dots & \Theta^{e=c}_{\bullet D} \end{pmatrix}$, $\Theta^s = \begin{pmatrix} \Theta^{e=s}_{\bullet 1} & \dots & \Theta^{e=s}_{\bullet D} \end{pmatrix}$ and $B(\omega) = \left(\|\omega\|_2 I_d - \frac{\omega\omega^\intercal}{\|\omega\|_2} \right) \in \mathcal{M}_{d,d}$.

A quick experiment shows the advantage of seeing the decomposable kernel as a linear operator rather than a matrix. We draw $N = 100$ points $(x_i)_{i=1}^N$ in the interval $(0, 1)^{20}$ and use a decomposable kernel with matrix $\Gamma = BB^\top \in \mathcal{M}_{p,p}(\mathbb{R})$ where $B \in \mathcal{M}_{p,p}(\mathbb{R})$ is a random matrix with coefficients drawn uniformly in $(0, 1)$. We compute $\widetilde{\Phi}(x)^\top\theta$ for all x_i's, where $\theta \in \mathcal{M}_{2D,1}(\mathbb{R})$, $D = 100$, with the implementation `EfficientDecomposableGaussianORFF`, equation 6.14, and `NaiveDecomposableGaussianORFF`, equation 6.13. The coefficients of θ were drawn at random uniformly in $(0, 1)$. We report the execution time in figure 6.3 for different values of p, $1 \leq p \leq 100$. The left plot reports the execution time in seconds of

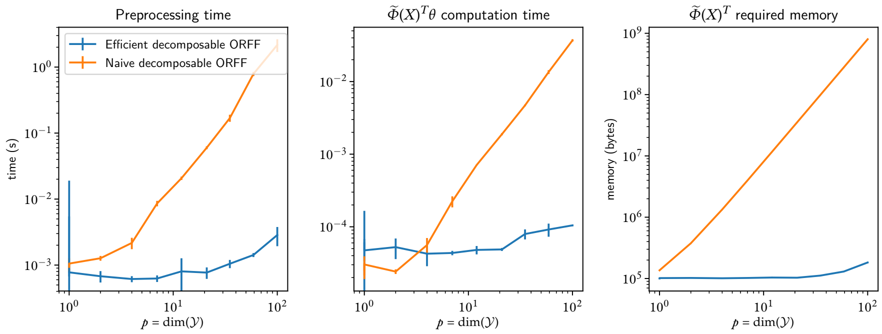

Figure 6.3: Efficient decomposable gaussian ORFF (lower is better).

the construction of the feature. The middle plot reports the execution time of $\widetilde{\Phi}(x)^\top\theta$, and the right plot the memory used in bytes to store $\widetilde{\Phi}(x)$ for all x_i's. We averaged the results over ten runs. Full code is given in appendix C.3.

6.2.6 *Curl-free kernel*

We use the unbounded ORFF map presented in equation 4.20. We draw $N = 1000$ points $(x_i)_{i=1}^N$ in the interval $(0, 1)^p$ and use a curl-free kernel. We compute $\widetilde{\Phi}(x)^\top\theta$ for all x_i's, where $\theta \in \mathcal{M}_{2D,1}(\mathbb{R})$, $D = 500$, with the matrix implementation and the `LinearOperator` implementation. The coefficients of θ were drawn at random uniformly in $(0, 1)$. We report the execution time in figure 6.4 for different values of p, $1 \leq p \leq 100$. The left plot reports the execution time in seconds of the construction of the features. The middle plot reports the execution time of $\widetilde{\Phi}(x)^\top\theta$, and the right plot the memory used in bytes to store $\widetilde{\Phi}(x)$ for all x_i's. We averaged the results over fifty runs. Full code is given in appendix C.4. As we can see the linear-operator implementation is one order of magnitude slower than its matrix counterpart. However it uses considerably less memory.

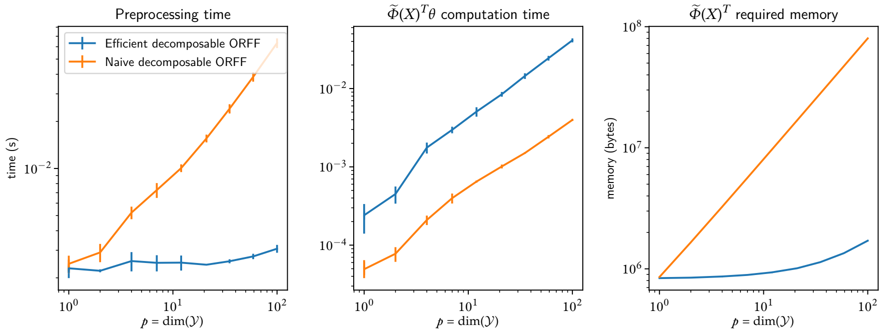

Figure 6.4: Efficient curl-free gaussian ORFF (lower is better).

6.2.7 *Divergence-free kernel*

We use the unbounded ORFF map presented in equation 4.21. We draw $N = 100$ points $(x_i)_{i=1}^N$ in the interval $(0, 1)^p$ and use a curl-free kernel. We compute $\widetilde{\Phi}(x)^\top\theta$ for all x_i's, where $\theta \in \mathcal{M}_{2Dp,1}(\mathbb{R})$, $D = 100$, with the matrix implementation and the `LinearOperator` implementation. The coefficients of θ were drawn at random uniformly in $(0, 1)$. We report the execution time in figure 6.4 for different values of p, $1 \leq p \leq 100$. The left plot reports the execution time in seconds of the construction of the feature. The middle plot reports the execution time of $\widetilde{\Phi}(x)^\top\theta$, and the right plot the memory used in bytes to store $\widetilde{\Phi}(x)$ for all x_i's. We averaged the results over ten runs. Full code is given in appendix C.5. We draw the same conclusions as the curl-free kernel.

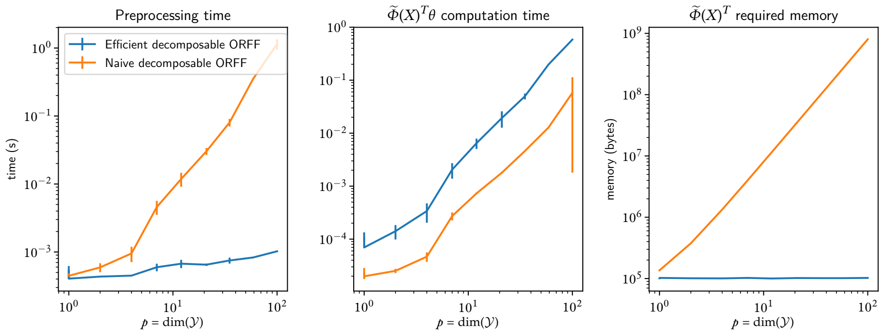

Figure 6.5: Efficient divergence-free gaussian ORFF (lower is better).

6.2.8 *Iterative (matrix-free) solvers*

We have shown in this section that viewing the Operator-valued Random Fourier Feature maps as linear operator rather than staking matrices leads to more efficient computations.

6.3 EXPERIMENTS

We present a set of experiments to complete the theoretical contribution and illustrate the behavior of ORFF-regression. First we study how well the ORFF regression recover the result of operator-valued kernel regression. Second we show the advantages of ORFF regression over independent RFF regression. A code implementing ORFF is available at `https://github.com/operalib/operalib` a framework for OVK Learning.

6.3.1 *Learning with ORFF vs learning with OVK*

6.3.1.1 *Datasets*

The *first dataset* considered is the handwritten digits recognition dataset MNIST available at `http://yann.lecun.com/exdb/mnist`. We select a training set of $12,000$ images and a test set of $10,000$ images. The inputs are images represented as a vector $x_i \in [0, 255]^{784}$ and the targets $y_i \in \mathbb{N}_9$are integers between 0 and 9.

First we scaled the inputs such that they take values in $[-1, 1]^{784}$. Then we binarize the targets such that each number is represented by a unique binary vector of dimension 10. The vector y_i is zero everywhere except on the dimension corresponding to the class where it is one. For instance the class 4 is encoded

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}^\intercal.$$

To predict classes, we use the simplex coding method presented in Mroueh et al. [105]. The intuition behind simplex coding is to project the binarized labels of dimension p onto the most separated vectors on the hypersphere of dimension $p-1$. For ORFF we can encode directly this projection in the B matrix of the decomposable kernel $K_0(\delta) = BB^* k_0(\delta)$ where k_0 is a Gaussian kernel. The matrix B is computed via the recursion

$$B_{p+1} = \begin{pmatrix} 1 & u^T \\ 0_{p-1} & \sqrt{1-p^{-2}} B_p \end{pmatrix}, \qquad B_2 = \begin{pmatrix} 1 & -1 \end{pmatrix},$$

where $u = \begin{pmatrix} -p^{-2} & \dots & -p^{-2} \end{pmatrix}^T \in \mathbb{R}^{p-1}$ and $0_{p-1} = \begin{pmatrix} 0 & \dots & 0 \end{pmatrix}^T \in \mathbb{R}^{p-1}$. For Operator-Valued Kernels we project the binarized targets on the simplex as a preprocessing step, before learning with the decomposable $K_0(\delta) = I_p k_0(\delta)$, where k_0 is a scalar Gaussian kernel.

The *second dataset* is a simulated five dimensional ($5D$) vector field with structure. We generate a scalar field as a random function $f : [-1, 1]^5 \to \mathbb{R}$, where $\widetilde{f}(x) = \widetilde{\varphi}(x)^* \theta$ where θ is a random matrix with each entry following a standard normal distribution, $\widetilde{\varphi}$ is a scalar Gaussian RFF with bandwidth $\sigma = 0.4$. The input data x are generated from a uniform probability distribution. We take the gradient of $\widetilde{f}$ to generate the curl-free $5D$ vector field.

The *third dataset* is a synthetic of data from $\mathbb{R}^{20} \to \mathbb{R}^4$ as described in Audiffren and Kadri [7]. In this dataset, inputs $(x_1, \dots, x_{20})$ are generated independently and uniformly over $[0, 1]$ and the different outputs are computed as follows. Let

$$\varphi(x) = (x_1^2, x_4^2, x_1x_2, x_3x_5, x_2, x_4, 1)$$

and (w_i) denotes the i. i. d. copies of a seven dimensional Gaussian distribution with zero mean and covariance $\Sigma \in \mathcal{M}_{7,7}(\mathbb{R})$ such that

$$\Sigma = \operatorname{Diag}\begin{pmatrix} 0.5 & 0.25 & 0.1 & 0.05 & 0.15 & 0.1 & 0.15 \end{pmatrix}$$

Then, the outputs of the different tasks are generated as $y_i = w_i\varphi(x)$. We use this dataset with $p = 4$, 10^5 instances and for the train set and also 10^5 instances for the test set.

6.3.1.2 *Results*

Performance of ORFF regression on the first dataset. We trained both ORFF and OVK models on MNIST dataset with a decomposable Gaussian kernel with signature

$$K_0(\delta) = \exp\left(-\|\delta\|/(2\sigma^2)\right)\Gamma.$$

To apply *algorithm* 4 after noticing that in the case of the decomposable kernel with $\lambda_M = 0$, it boilds down to a Stein equation [25, section 5.1], we use an off-the-shelf solver[4] able to handle Stein's equation. For both methods we choose $\sigma = 20$ and use a 2-fold cross validation on the training set to select the optimal λ. First, figure 6.6 compares the running time between OVK and ORFF models using $D = 1000$ Fourier features against the number of datapoints N. The log-log plot shows ORFF scaling better than the OVK w. r. t. the number of points. Second, figure 6.6 shows the test prediction error versus the number of ORFFs D, when using $N = 1000$ training points. As expected, the ORFF model converges toward the OVK model when the number of features increases.

Performance of ORFF regression on the second dataset. We perform a similar experiment on the second dataset (5D-vector field with structure). We use a Gaussian curl-free kernel with bandwidth equal to the median of the pairwise distances and tune the hyperparameter λ on a grid. Here we optimize equation 6.10 using Scipy's L-BFGS-B [31] solver[5] with the gradients given in **??** and the efficient linear operator described in

4 Available at `http://ta.twi.tudelft.nl/nw/users/gijzen/IDR.html`
5 Available at `http://docs.scipy.org/doc/scipy/reference/optimize.html`

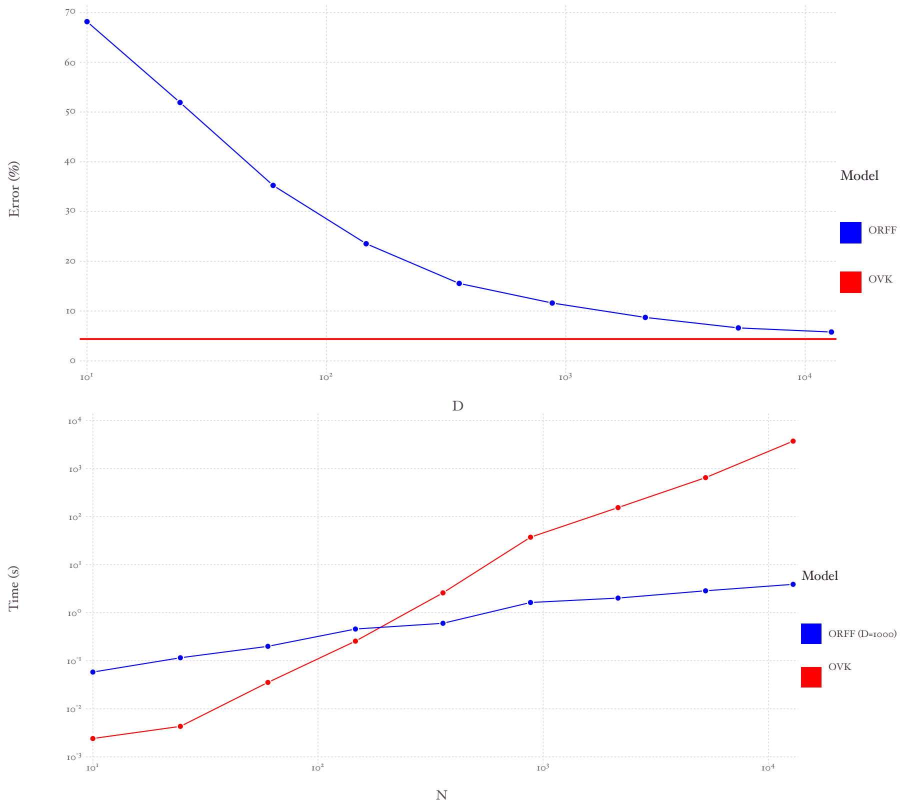

Figure 6.6: Empirical comparison of ORFF and OVK regression on MNIST dataset and empirical behavior of ORFF regression versus D and N.

subsection 6.2.5. Figure 6.7 (bottom row) reports the $R2$ score on the test set versus the number of curl-ORFF D with a comparison with curl-OVK. In this experiment, we see that curl-ORFF can even be better than curl-OVK, suggesting that ORFF might play an additional regularizing role. It also shows the computation time of curl-ORFF and curl-OVK. We see that OVK regression does not scale with large datasets, while ORFF regression does. When $N > 10^4$, OVK regression exceeds memory capacity.

Structured prediction vs Independent (RFF) prediction. On the second dataset, figure 6.7 (top row) compares R2 score and time of ORFF regression using the trivial identity decomposable kernel, e. g. independent RFFs, to curl-free ORFF regression. Curl-free ORFF outperforms independent RFFs, as expected, since the dataset involves structured outputs.

Impact of the number of random features (D). In this setting we solved the optimisation problem for both ORFF and OVK using a L-BFGS-B. Figure 6.8 top row shows that for a fixed number of instance in the train set, OVK performs better than ORFF in terms of accuracy ($R2$). However ORFF scales better than OVK w. r. t. the number of data. ORFF is able to process more data than OVK in the same time and thus

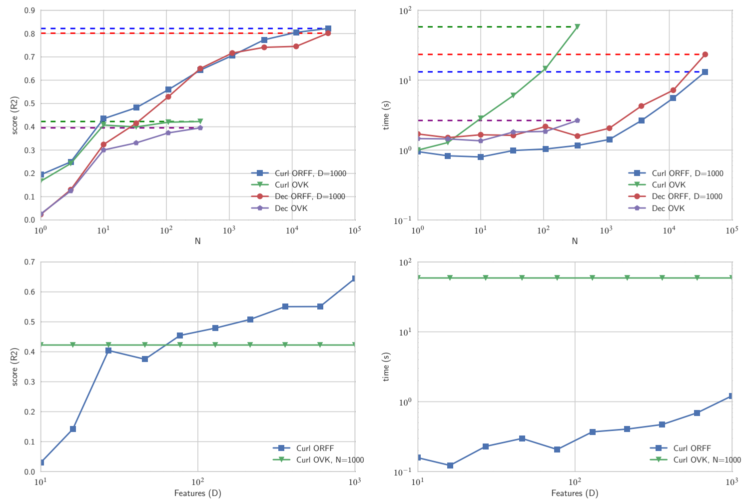

Figure 6.7: Empirical comparison between curl-free ORFF, curl-free OVK, independent ORFF, independent OVK on a synthetic vector field regression task.

reach a better accuracy for a given amount of time. Bottom row shows that ORFF tends to reach OVK's accuracy for a fixed number of data when the number of features increase.

Multitask learning. In this experiment we are interested in multitask learning with operator-valued random Fourier features, and see whether the approximation of a joint OVK performs better than an independent OVK. In this setting we assume that for each entry $x_i \in \mathbb{R}^d$ we only have access to one observation $y_i \in \mathbb{R}$ corresponding to a task t_i. We used the SARCOS dataset, taken from `http://www.gaussianprocess.org/gpml/data/` website. This is an inverse dynamics problem, i. e. we have to predict the 7 joint torques given the joint positions, velocities and accelerations. Hence, we have to solve a regression problem with 21 inputs and 7 outputs which is a very nonlinear function. It has $45K$ inputs data. Suppose that we are given a collection of inputs data $x_1, \dots, x_N \in \mathbb{R}^{21}$ and a collection of output data $((y_1, t_1) \dots, (y_N, t_N)) \in (\mathbb{R} \times \mathbb{N}_T)^N$ where T is the number of tasks. We consider the following multitask loss function

$$L(h(x), (y, t)) = \frac{1}{2}\left(\langle h(x), e_t \rangle_2 - y\right)^2,$$

This loss function is adapted to datasets where the number of data per tasks is unbalanced (i. e. for one input data we observe the value of only one task and not all the tasks.). We optimise the regularized risk

$$\frac{1}{N}\sum_{i=1}^{N} L\left(h(x_i), (y_i, t_i)\right) + \frac{\lambda}{2N}||h||^2_{\mathcal{H}} = \frac{1}{2N}\sum_{i=1}^{N}\left(\langle h(x_i), e_{t_i}\rangle - y_i\right)^2 + \frac{\lambda}{2N}||h||^2_{\mathcal{H}}$$

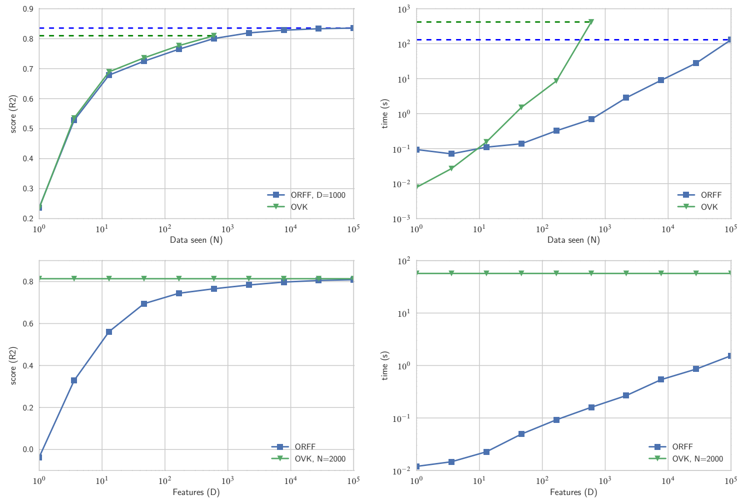

Figure 6.8: Decomposable kernel on the third dataset: $R2$ score vs number of data in the train set (N)

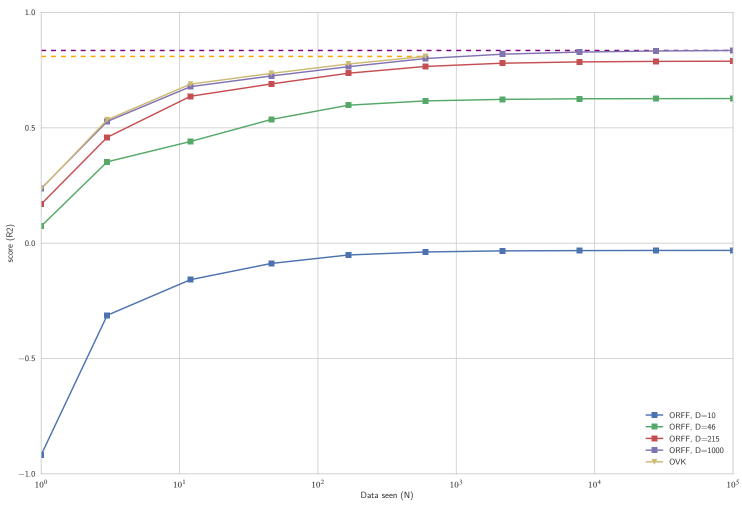

Figure 6.9: Decomposable kernel on the third dataset: $R2$ score vs number of data in the train set (N) for different number for different number of random samples (D).

We used a model h based on the decomposable kernel

$$h(x) = (\varphi(x)^T \otimes B)\theta \tag{6.15}$$

we chose B such that $BB^T = A$, where A is the inverse graph Laplacian L of the similarities between the tasks, parametrized by an hyperparameter $\gamma \in \mathbb{R}_+$.

$$L_{kl} = \exp\left(-\gamma\sqrt{\sum_{i=1}^{N}\left(y_i^k - y_i^l\right)^2}\right). \tag{6.16}$$

We draw N data randomly for each task, hence creating a dataset of $N \times 7$ data and computed the nMSE on the proposed test set (4.5K points). We repeated the experiments 80 times to avoid randomness. We choose $D = \frac{N \vee 500}{2}$ features, and optimized the problem with a second order batch gradient. Table 6.5 shows that using the ORFF approximation of

N	Independant (%)	Laplacian (%)	p-value	T
50×7	23.138 ± 0.577	22.254 ± 0.536	2.68%	4(s)
100×7	16.191 ± 0.221	15.568 ± 0.187	$<$ 0.1%	16(s)
150×7	13.821 ± 0.115	13.459 ± 0.106	$<$ 0.1%	13(s)
200×7	12.713 ± 0.0978	12.554 ± 0.0838	1.52%	12(s)
400×7	10.785 ± 0.0579	10.651 ± 0.0466	$<$ 0.1%	10(s)
800×7	7.512 ± 0.0344	7.512 ± 0.0344	100%	15(s)
1600×7	6.486 ± 0.0242	6.486 ± 0.0242	100%	20(s)
3200×7	5.658 ± 0.0187	5.658 ± 0.0187	100%	20(s)

Table 6.5: Error (% of nMSE) on SARCOS dataset.

an operator-valued kernel with a good prior on the data improves the performances w. r. t. the independent ORFF. However the advantage seems to be less important the more data are available.

6.4 CONCLUSIONS

We introduced ORFF, a general and versatile framework for shift-invariant OVK approximation. We proved the uniform convergence of the approximation error for bounded and unbounded ORFFs. The complexity in time of these approximations together with the linear learning algorithm make this implementation scalable with the data size and thus appealing compared to OVK regression as shown in numerical experiments. Further work concerns generalization bounds and consistency for ORFF-regression. Finally this work opens the door to building deeper architectures by stacking vector-valued functions while keeping a kernel view for large datasets.

7

CONSISTENCY AND GENERALIZATION BOUND FOR ORFF

This short chapter deals with a generalization bound for the a regression problem with ORFF based on the results of Maurer [97] and Rahimi and Recht [122]. We also discuss the case of Ridge regression presented in ??.

Contents

7.1 GENERALIZATION BOUND

In this section, we are interested in finding a function $f_* : \mathcal{X} \to \mathcal{Y}$, where $\mathcal{X}$ is a Polish space and $\mathcal{Y}$ a separable Hilbert space such that for all x_i in $\mathcal{X}$ and all y_i in $\mathcal{Y}$ that minimizes a criterion. In statistical supervised learning, we consider a training set sequence $\mathbf{s} = (x_i, y_i)_{i=1}^N \in (\mathcal{X} \times \mathcal{Y})^N$, $N \in \mathbb{N}^*$ drawn i. i. d. from an unknown probability law $\mathbf{Pr}$. We suppose we are given a cost function $c : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, such that $c(f(x), y)$ returns the error of the prediction $f(x)$ w. r. t. the ground truth y. We define the true risk the sum of the cost over all possible training examples drawn from a latent probability law $\mathbf{Pr}$,

$$\mathfrak{R}\left(f\right) = \int_{\mathcal{X}\times\mathcal{Y}} L(x, f, y) d\mathbf{Pr}(x, y) = \int_{\mathcal{X}\times\mathcal{Y}} c(f(x), y) d\mathbf{Pr}(x, y)$$

Thus given a class of functions $\mathcal{F}$, the goal of a learning algorithm is to find an optimal model f_* that minimizes the true risk. Namely

$$f_* = \operatorname*{arg\,min}_{f\in\mathcal{F}} \mathfrak{R}\left(f\right) = \operatorname*{arg\,min}_{f\in\mathcal{F}} \int_{\mathcal{X}\times\mathcal{Y}} c(f(x), y) d\mathbf{Pr}(x, y).$$

Since in practice we do not have access to the joint probability law of (X,Y), we define its empirical counterpart as the empirical mean estimate, where the sequence $\mathbf{s} = (x_i, y_i)_{i=1}^N$ is made of $\mathcal{X}$-valued random vectors drawn i. i. d. from some law $\mathbf{Pr}$. The empirical risk then reads

$$\mathfrak{R}_{\text{emp}}\left(f, \mathbf{s}\right) = \frac{1}{N}\sum_{i=1}^N c(f(x_i), y_i), \quad (x_i, y_i) \sim \mathbf{Pr} \text{ i. i. d..}$$

As a result, in practice we seek a function $f_{\mathbf{s}}$ such that

$$f_{\mathbf{s}} = \operatorname*{arg\,min}_{f\in\mathcal{F}} \mathfrak{R}_{\text{emp}}\left(f, \mathbf{s}\right) = \operatorname*{arg\,min}_{f\in\mathcal{F}} \frac{1}{N}\sum_{i=1}^N c\left(f(x_i), y_i\right). \tag{7.1}$$

The basic requirement for any learning algorithm is the generalization property: the empir- ical error must be a good proxy of the expected error, that is the difference between the two must be "small" when N is large. A generalization bound allows to study, for any $f \in \mathcal{F}$ the difference between its true risk $\mathfrak{R}\left(f\right)$ and its empirical risk, $\mathfrak{R}\left(f, \mathbf{s}\right)$. This quantifies the impact of having a limited number of observations. Generalization (upper) bounds [158] involve two components: one being the empirical risk and the other depends on the dataset size as well as some capacity notion that reflects the richness of the family of functions $\mathcal{F}$ considered. First generalization bounds proved by Vapnik and Chervonenkis involve the dimension of Vapnik-Chervonenkis dimension of $\mathcal{F}$. In practice, generalization bounds are usually not enough tight to provide direct recommendations for choosing the appropriate family of functions to work with when N is fixed. However as the bias-variance dilemna, it suggests that when learning a function from a finite dataset, it is necessary to control the richness of the functions family $\mathcal{F}$. In practise, a regularizer is added to the data-fitting term in order to maintain the solution $f_{\mathbf{s}}$ of equation 7.1 in a ball of $\mathcal{F}$. As a result if $\mathcal{F}$ is a Banach space, it is common to find $f_{\mathbf{s}}$ such that

$$f_{\mathbf{s}} = \operatorname*{arg\,min}_{f \in \mathcal{F}} \mathfrak{R}_{\text{emp}}\left(f, \mathbf{s}\right) + \lambda \|f\|_{\mathcal{F}}^2.$$

(Tychonov regularization) or

$$f_{\mathbf{s}} = \begin{cases} \operatorname{arg\,min}_{f \in \mathcal{F}} & \mathfrak{R}_{\text{emp}}\left(f, \mathbf{s}\right) \\ \text{subject to} & \|f\|_{\mathcal{F}} < C \in \mathbb{R}_{>0} \end{cases}$$

(Ivanov regularization) or

$$f_{\mathbf{s}} = \begin{cases} \operatorname{arg\,min}_{f \in \mathcal{F}} & \mathfrak{R}_{\text{emp}}\left(f, \mathbf{s}\right) \\ \text{subject to} & \|f\|_{\infty} < C \in \mathbb{R}_{>0}. \end{cases}$$

7.1.1 *Generalization by bounding the function space complexity*

In the following we consider functions living in a Vector Valued Reproducing Kernel Hilbert Space, with kernel K (or $\widetilde{K}$).

Proposition 7.1 (Bartlett and Mendelson [14] and Maurer [97]). *Suppose that $f \in \mathcal{H}_K$ a VV-RKHS where*

$$\sup_{x \in \mathcal{X}} \mathbf{Tr}[K(x, x)] < T$$

and $\|f\|_{\mathcal{H}_K} < B$. Moreover let $c : \mathcal{Y} \to [0, C]$ be a L-Lipschitz cost function and $\mathcal{Y}$ a separable Hilbert space. Then if we are given N i. i. d. random variables with values in $\mathcal{X}$ (training samples, noted $\mathbf{s}$), then we have with at least probability $1 - \delta$, $\delta \in (0, 1)$ over the drawn training samples $\mathbf{s}$ that for any $f \in \mathcal{H}_K$,

$$\mathfrak{R}\left(f\right) \leq \mathfrak{R}_{emp}\left(f, \mathbf{s}\right) + 2\sqrt{\frac{2}{N}}\left(LBT^{1/2} + C\sqrt{\ln(2/\delta)}\right). \tag{7.2}$$

Proof *This proof is due to Maurer [97] generalizing the work of Bartlett and Mendelson [14, section 4.3]: we do not claim any originality for this proof. First let us introduce the notion of Rademacher complexity of a class of function $\mathcal{F}$. We recall that the probability mass function of a uniformly distributed Rademacher random variable is given for any $k \in \{-1, 1\}$ by*

$$f(k) = \begin{cases} 1/2 & \textit{if } k = -1 \\ 1/2 & \textit{otherwise.} \end{cases}$$

Definition 7.1 (Bartlett and Mendelson [14]). *Let $\mathcal{X}$ be any set. Let $\epsilon_1, \ldots, \epsilon_N$ be N independent Rademacher random variables, identically uniformly* *distributed on* $\{-1;1\}$. *For any class of functions* $F : \mathcal{X} \to \mathbb{R}$, *then for all* $x_1, \dots x_N \in \mathcal{X}$ *the quantity*

$$\mathcal{R}_N(F) := \mathbf{E}\left[\sup_{f\in\mathcal{F}} \sum_{i=1}^{N} \epsilon_i f(x_i) \,\middle|\, x_1, \dots, x_N\right]$$

is called Rademacher complexity of the class $\mathcal{F}$.

In a few words the Rademacher complexity measures the richness of a class a function by its capacity to be correlated to noise. In generalization bounds, the Rademacher complexity of a class of function often involves a composition between a target function to be learn and a cost function, part of the risk we want to minimize. The idea is to bound the Rademacher complexity with a term that does not depends on the cost function, but only on the target function.

Proposition 7.2 (Maurer [97]). *Let* $\mathcal{X}$ *be any set,* $x_1, \dots, x_N$ *in* $\mathcal{X}$, *let* $\mathcal{F}$ *be a class of function* $f : \mathcal{X} \to \mathcal{Y}$ *and for i=1, ..., N, each function* $h_i : \mathcal{Y} \to \mathbb{R}$ *be a L-Lipschitz function, where* $\mathcal{Y}$ *is a separable Hilbert space endowed with euclidean inner product. Then*

$$\begin{aligned}\mathbf{E}&\left[\sup_{f\in\mathcal{F}} \sum_{i=1}^{N} \epsilon_i h_i(f(x_i)) \,\middle|\, x_1, \dots, x_N\right] \\ &\le \sqrt{2}L\mathbf{E}\left[\sup_{f\in F} \sum_{i=1,k}^{i=N} \epsilon_{ik} f_k(x_i) \,\middle|\, x_1, \dots, x_N\right],\end{aligned}$$

where ϵ_{ik} *is a doubly indexed independent Rademacher sequence and* $f_k(x_i)$ *is the k-th component of* $f(x_i)$.

From now on, we consider functions $f \in \mathcal{H}_K$ *a Vector Valued Reproducing Kernel Hilbert Space. Then there exists an induced feature-map* $\Phi : \mathcal{X} \to \mathcal{L}(\mathcal{Y}, \mathcal{H})$ *such that for all* $y, y' \in \mathcal{Y}$ *the kernel is given by*

$$\langle y, K(x,z)y' \rangle = \langle \Phi_x y, \Phi_z y' \rangle.$$

We say that the feature space $\mathcal{H}$ *is embedded into the RKHS* $\mathcal{H}_K$ *by means of the feature operator* $(W\theta)(x) := (\Phi_x^* \theta)$. *Indeed* W *defines a partial isometry between* $\mathcal{H}$ *and* $\mathcal{H}_K$. *Suppose that* $\mathcal{Y}$ *a is separable Hilbert space and let the class of* $\mathcal{Y}$*-valued functions F be*

$$\mathcal{F} = \{ f \,|\, f : x \mapsto (W\theta)(x), \ \|\theta\|_{\mathcal{H}} < B \} \subset \mathcal{H}_K.$$

Let $c_{y_i} = c(\cdot - y_i)$, *for all in* $\in \mathbb{N}_N$. *Then from proposition 7.2 and if K is trace class, we have*

$$\begin{aligned}\mathbf{E} \sup_{\|\theta\|_{\mathcal{H}} < B} \sum_{i=1}^{N} \epsilon_i c_{y_i}(\Phi_{x_i}^* \theta) &\le \sqrt{2}L\mathbf{E} \sup_{\|\theta\|_{\mathcal{H}} < B} \sum_{i=1,k}^{i=N} \epsilon_{ik} \langle \Phi_{x_i}^* \theta, e_k \rangle \\ &= \sqrt{2}L\mathbf{E} \sup_{\|\theta\|_{\mathcal{H}} < B} \left\langle \theta, \sum_{i=1,k}^{i=N} \epsilon_{ik} \Phi_{x_i} e_k \right\rangle_{\mathcal{Y}}.\end{aligned}$$

Thus

$$\begin{aligned}\mathbf{E} \sup_{\|\theta\|_{\mathcal{H}} < B} \sum_{i=1}^{N} \epsilon_i c_{y_i}(\Phi_{x_i}^* \theta) &\leq \sqrt{2} L B \mathbf{E} \left\| \sum_{i=1,k}^{i=N} \epsilon_{ik} \Phi_{x_i} e_k \right\|_{\mathcal{Y}} \\ &\leq \sqrt{2} L B \sqrt{\sum_{i=1,k}^{i=N} \|\Phi_{x_i} e_k\|_{\mathcal{Y}}^2} \\ &\leq \sqrt{2} L B \sqrt{\sum_{i=1}^{N} \mathbf{Tr}\left[K(x_i, x_i)\right]} \\ &\leq \sqrt{2} L B \sqrt{N} \sqrt{\sup_{x \in \mathcal{X}} \mathbf{Tr}\left[K(x, x)\right]}. \end{aligned} \tag{7.3}$$

From Bartlett and Mendelson [14] and Maurer [97],

Theorem 7.1 *Let $\mathcal{X}$ be any set, $\mathcal{F}$ a class of functions $f: \mathcal{X} \to [0, C]$ and let $X_1, \ldots, X_N$ be a sequence of i. i. d. random variables with value in $\mathcal{X}$. Then for $\delta \in (0, 1)$, with probability at least $1 - \delta$, we have for all $f \in \mathcal{F}$ that*

$$\mathbf{E} f(X) \leq \frac{1}{N} \sum_{i=1}^{N} f(X_i) + \frac{2}{N} \mathcal{R}_N(\mathcal{F}) + C \sqrt{\frac{8 \ln(2/\delta)}{N}} \tag{7.4}$$

Conclude by pluging equation 7.3 in theorem 7.1. □

As an example, let us consider equation 6.1, which is a solution of the regularized empirical risk, and *algorithm* 4 when $D \to \infty$. We first list the following assumptions useful in the rest of the section. Let $\mathbf{s} = (x_i, y_i)_{i=1}^N \in \mathcal{X}^N \times \mathcal{Y}^N$ be the training samples.

Assumption 7.1 *There exists a positive constant $\kappa \in \mathbb{R}_{\geq 0}$ such that*

$$\max_{i \in \mathbb{N}_N^*} \|K(x_i, x_i)\|_{\mathcal{Y}, \mathcal{Y}} < \kappa.$$

Assumption 7.2 *There exists a positive constant $T \in \mathbb{R}_{\geq 0}$ such that*

$$\max_{i \in \mathbb{N}_N^*} \mathbf{Tr}\left[K(x_i, x_i)\right] < T.$$

Assumption 7.3 *There exists a positive constant $C \in \mathbb{R}_{\geq 0}$ such that*

$$\max_{i \in \mathbb{N}_N^*} \|y_i\|_{\mathcal{Y}} \leq C.$$

Assumption 7.4 *Given a Loss function L, there exists a positive constant $\xi \in \mathbb{R}_{\geq 0}$ such that for all $x \in \mathcal{X}$, for all $y \in \mathcal{Y}$ and for any $\mathbf{s} \in \mathcal{X}^N \times \mathcal{Y}^N$,*

$$L(x, f_{\mathbf{s}}, y) \leq \xi.$$

Under assumption 7.3, from remark 6.1, we know that $\|f\|_{\mathcal{H}_K} \leq \sqrt{\frac{2}{\lambda}} \sigma_y$, where

$$\frac{1}{N} \sum_{i=1}^{N} \|y_i\|_{\mathcal{Y}}^2 \leq \sigma_y^2 \leq C^2.$$

Thus we straight away see that it is possible to choose $B = \sqrt{\frac{2}{\lambda}}C$. Let $\kappa = \|K_e(e)\|_{\mathcal{Y},\mathcal{Y}}$ The Lipschitz constant of the least square loss $c(f_{\mathbf{s}}(x), y) = \frac{1}{2}\|f_{\mathbf{s}}(x) - y\|_{\mathcal{Y}}^2$ with respect to $f_{\mathbf{s}}(x)$ is $L = \max\left(\sqrt{\frac{2\kappa}{\lambda}}C, C\right)$ and the loss takes values in $\left[0, \frac{1}{2}L^2\right]$. Hence under assumption that $\lambda < 2\kappa$ and assumption 7.2, and assumption 7.3, equation 7.2 applies especially that for any $f_{\mathbf{s}} \in \mathcal{H}_K$, solution of algorithm 4,

$$\mathfrak{R}\left(f_{\mathbf{s}}\right) \leq \mathfrak{R}_{\text{emp}}\left(f_{\mathbf{s}}, \mathbf{s}\right) + 8\frac{C^2}{\lambda}\sqrt{\frac{\kappa}{N}}\left(T^{1/2} + \sqrt{\frac{\kappa \ln(1/\delta)}{2}}\right). \tag{7.5}$$

This bound is to be compared to the results of Kadri et al. [73] in the context of β-stability.

7.1.2 *Algorithm stability*

The approach to generalization bounds presented in proposition 7.1 is based on controlling the complexity of the hypothesis space[18] using Rademacher complexity. On the other hand, the idea of stability is that a reliable algorithm should not change its solution too much if we modify slightly the training data. Given a training sequence

[18] *Other methods using covering numbers [152, 174] or VC-dimension [157] have also been used as a proxy on the complexity of the hypothesis space.*

$$\mathbf{s} = \left((x_1, y_1), \dots (x_N, y_N)\right) \in (\mathcal{X} \times \mathcal{Y})^N,$$

we note $\mathbf{s}^{\setminus i}$ the training sequence

$$\mathbf{s}^{\setminus i} = \left((x_1, y_1), \dots, (x_{i-1}, y_{i-1}), \dots, (x_{i+1}, y_{i+1})(x_N, y_N)\right) \in (\mathcal{X} \times \mathcal{Y})^N,$$

the subsequence of $\mathbf{s}$ from which we removed the i-th element.

Definition 7.2 (Uniform stability Bousquet and Elisseeff [21, definition 6]). *A learning algorithm $\mathbf{s} \mapsto f_{\mathbf{s}}$ has uniform stability β with respect to the loss function L if the following holds*

$$\forall i \in \mathbb{N}_N^*, \forall \mathbf{s} \in (\mathcal{X} \times \mathcal{Y})^N \sup_{x \in \mathcal{X}, y \in \mathcal{Y}} |L(x, f_{\mathbf{s}}, y) - L(x, f_{\mathbf{s}^{\setminus i}}, y)| \leq \beta.$$

As shown by Bousquet and Elisseeff [21], algorithm stability has direct link with generalization. Indeed if an algorithm has β-stability, and a "bounded" loss for all $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ (assumption 7.4), it is possible to exhibit a generalization bound.

Theorem 7.2 (Bousquet and Elisseeff [21, theorem 12]). *Let* $\mathbf{s} \mapsto f_{\mathbf{s}}$ *be a learning algorithm with uniform stability β with respect to a loss L that satisfies assumption 7.4. Then $\forall N \in \mathbb{N}^*$, $\forall \delta \in (0,1)$, the following bound holds with probability at least $1-\delta$ over the i. i. d. drawn training smaples* $\mathbf{s}$.

$$\mathfrak{R}\left(f_{\mathbf{s}}\right) \leq \mathfrak{R}_{emp}\left(f_{\mathbf{s}}, \mathbf{s}\right) + 2\beta + (4N\beta + \xi)\sqrt{\frac{\ln(1/\delta)}{2N}}.$$

In their original paper on learning function-valued output data, Kadri et al. [73] showed that under assumption 7.1, assumption 7.3, and provided that K is weakly measurable, the algorithm is β-stable with β =. Moreover assumption 7.4 holds with ξ, where σ =. Thus another generalization bound for algorithm 4 is

$$\begin{aligned}\mathfrak{R}\left(f_{\mathbf{s}}\right) \leq \mathfrak{R}_{\text{emp}}\left(f_{\mathbf{s}}, \mathbf{s}\right) + \frac{\kappa^2 C^2 \left(1 + \frac{\kappa}{\sqrt{\lambda}}\right)^2}{\lambda N} \\ + C^2 \left(1 + \frac{\kappa}{\sqrt{\lambda}}\right)^2 \left(\frac{4\kappa^2}{\lambda} + 1\right) \sqrt{\frac{\ln(1/\delta)}{2N}}.\end{aligned} \tag{7.6}$$

7.2 CONSISTENCY OF LEARNING WITH ORFF

In this section we are interested by measuring how $\mathfrak{R}\left(\widetilde{f}_{\mathbf{s}}\right)$ is close to the smallest true risk achieved in the function class $\mathcal{F}$. The quantity of interest is:

$$\mathfrak{R}\left(\widetilde{f}_{\mathbf{s}}\right) - \min_{f \in \mathcal{F}} \mathfrak{R}\left(f\right).$$

In other words, we quantify the difference between risk of the optimal solution belonging to a given class of function $\mathcal{F}$, and the risk given a solution $f_{\mathbf{s}}$ returned by some learning algorithm. Here to derive a consistency result, we study an algorithm slightly different from algorithm 4. Given a loss function $L : \mathcal{X} \times \mathcal{F} \times \mathcal{Y} \to \mathbb{R}_+$ and its canonical cost function $c(f(x), y) := L(x, f, y)$ such that c is Lipschitz in its first argument. We consider learning with an ORFF $\widetilde{\Phi}(x) : \mathcal{Y} \to \bigoplus_{j=1}^{D} \mathcal{Y}'$ thanks to the algorithm

$$\theta_{\mathbf{s}} = \begin{cases} \arg\min_{\theta \in \bigoplus_{j=1}^{D} \mathcal{Y}'} & \frac{1}{N} \sum_{i=1}^{N} c(\widetilde{\Phi}(x)^* \theta) \\ \text{subject to} & \max_{j \in \mathbb{N}^*_D} \left\|\theta_j\right\|_{\mathcal{Y}} \leq \frac{B}{D}. \end{cases} \tag{7.7}$$

Then the associated output function return is $\widetilde{f}_{\mathbf{s}} = \widetilde{\Phi}(\cdot)^* \theta_{\mathbf{s}}$. We suppose that the operator $A(\omega)$ used in the construction of $\widetilde{\Phi}(x)$ has bounded trace $\mathbf{Pr}_{\rho, \widehat{\mathbf{Haar}}}$-almost everywhere.

Proposition 7.3 *Let $\Phi_x = (x, \cdot) B(\cdot)$ be a Fourier feature such that there exists a constant $T \in \mathbb{R}_+$ such that*

$$\operatorname*{ess\,sup}_{\omega \in \widehat{\mathcal{X}}} \mathbf{Tr}\left[A(\omega)\right] < T$$

and a constant $u \in \mathbb{R}_+$ such that

$$\operatorname*{ess\,sup}_{\omega \in \widehat{\mathcal{X}}} \|A(\omega)\|^2_{\mathcal{Y}, \mathcal{Y}} < u.$$

where $A(\omega) = B(\omega) B(\omega)^*$. Let ρ be the density of a probability distribution with respect to the Haar measure $\mathbf{Haar}$ *and define the set*

$$\mathcal{F} = \left\{ f \,\middle|\, f : x \mapsto \int_{\widehat{\mathcal{X}}} \Phi_x(\omega) \theta(\omega)^* d\widehat{\mathbf{Haar}}(\omega), \quad \|\theta(\omega)\|_{\mathcal{Y}} < B\rho(\omega) \right\} \subseteq \mathcal{H}_K.$$

Eventually let $c : \mathcal{Y}^2 \to [0, C]$ be a cost function L-Lipschitz in its first argument. Then for any $\delta \in (0,1)$, given a training sequence $\mathbf{s} = (x_i, y_i) \in (\mathcal{X} \times \mathcal{Y})^N$ drawn i. i. d., if $\widetilde{f}_{\mathbf{s}}$ is given by equation 7.7 then we have

$$\mathfrak{R}\left(\widetilde{f}_{\mathbf{s}}\right) - \min_{f \in \mathcal{F}} \mathfrak{R}\left(f\right) \leq \underbrace{4\sqrt{\frac{2}{N}}\left(LBT^{1/2} + C\sqrt{\ln(2/\delta)}\right)}_{\text{Estimation error.}} + \underbrace{\frac{uLB}{\sqrt{D}}\left(1 + \sqrt{2\ln(1/\delta)}\right)}_{\text{Approximation error.}}.$$

with probability $1 - 2\delta$ over the training sequence and the random vectors $(\omega_j)_{j=1}^D$.

Proof *We follow the proof idea of Rahimi and Recht [122] in the scalar case and adapt it to the vector-valued case in the light of the results of Maurer [97]. We first define the two following sets.*

$$\mathcal{F} = \left\{ f \;\middle|\; f : x \mapsto \int_{\widehat{\mathcal{X}}} \Phi_x(\omega)^* \theta(\omega) d\widehat{\mathbf{Haar}}(\omega), \quad \|\theta(\omega)\|_{\mathcal{Y}} < B\rho(\omega) \right\}$$

and

$$\widetilde{\mathcal{F}} = \left\{ f \;\middle|\; f : x \mapsto \sum_{j=1}^{D} \Phi_x(\omega_j)^* \theta_j, \quad \forall j \in \mathbb{N}_D^*, \left\|\theta_j\right\|_{\mathcal{Y}} < \frac{B}{D} \right\}.$$

Proposition 7.4 **(Existence of an approximate function).** *Let μ be a measure on $\mathcal{X}$, and f_* a function in $\mathcal{F}$. Moreover let $\operatorname{ess\,sup}_{\omega \in \widehat{\mathcal{X}}} \|B(\omega)\|_{\mathcal{Y},\mathcal{Y}}^2 \leq u$. If $(\omega_j)_{j=1}^D$ are drawn i. i. d. from a probability distribution of density ρ w. r. t. $\widehat{\mathbf{Haar}}$, then for any $\delta \in (0,1)$, with probability at least $1 - \delta$ over $(\omega_j)_{j=1}^D$, there exists a function $\widetilde{f}$ in $\widetilde{\mathcal{F}}$ such that*

$$\sqrt{\int_{\mathcal{X}} \left\|\widetilde{f}(x) - f_*(x)\right\|_{\mathcal{Y}}^2 d\mu(x)} \leq \frac{uB}{\sqrt{D}}\left(1 + \sqrt{2\ln(1/\delta)}\right).$$

Proof *Since $f_* \in \mathcal{F}$, we can write $f_*(x) = \int_{\mathcal{X}} \overline{(x,\omega)} B(\omega)\theta(\omega) d\widehat{\mathbf{Haar}}(\omega)$. Construct the functions $f_j = (\cdot, \omega_j) B(\omega_j) \beta_j$ with $\beta_k := \frac{\theta(\omega_j)}{\rho(\omega)}$, so that $\mathbf{E}_{\rho, \widehat{\mathbf{Haar}}} f_j = f_*$ pointwise. Let*

$$\widetilde{f}(x) = \sum_{j=1}^{D} \Phi_x(\omega_j)^* \frac{\beta_j}{D}$$

be the sample average of these functions. Then, $\widetilde{f} \in \widetilde{\mathcal{F}}$ because $\left\|\beta_j\right\|_{\mathcal{Y}}/D < B/D$. Also, under the inner product $\int_{\mathcal{X}} \langle f(x), g(x) \rangle_{\mathcal{Y}} d\mu(x)$, we have that

$$\left\|\overline{(\cdot, \omega_j)} B(\omega_j) \beta_j\right\|_{L^2(\mathcal{X},\mu;\mathcal{Y})} \leq \operatorname*{ess\,sup}_{\omega \in \widehat{\mathcal{X}}} \|B(\omega)B(\omega)^*\|_{\mathcal{Y},\mathcal{Y}}^2 B.$$

We introduce the following technical lemma of Rahimi and Recht [122] for concentration of random variable in Hilbert spaces (similar to Pinelis [118]).

Lemma 7.1 *Let $X_1, \dots, X_D$ be i. i. d. random variables with values in a ball of radius M centered at the origin in a Hilbert space $\mathcal{H}$. Denote the sample average $\bar{X} = \frac{1}{D}\sum_{j=1}^D X_j$. Then for any $\delta \in (0,1)$ with probability $1-\delta$,*

$$\|\mathbf{E}\bar{X} - \bar{X}\|_{\mathcal{Y}} \le \frac{M}{\sqrt{D}}\left(1 + \sqrt{2\ln(1/\delta)}\right).$$

Eventually apply lemma 7.1 to $f_1, \dots, f_D$ under the canonical inner product of the vector valued function space $L^2(\mathcal{X}, \mu; \mathcal{Y})$ to conclude the proof. □

Proposition 7.5 (Bound on the approximation error). *Let $L(x, f, y)$ be a loss function and $c_y(f(x)) = L(x, f, y)$ be a L-Lipschitz cost function for all $y \in \mathcal{Y}$. Let f_* be a function in $\mathcal{F}$. Suppose there exists a constant $u \in \mathbb{R}_+$ such that*

$$\operatorname*{ess\,sup}_{\omega \in \widehat{\mathcal{X}}} \|A(\omega)\|^2_{\mathcal{Y},\mathcal{Y}} \le u.$$

If $(\omega_j)_{j=1}^D$ are i. i. d. random variables drawn from a probability distribution of density ρ, then for any $\delta \in (0,1)$ there exits, with probability $1-\delta$ over $(\omega_j)_{j=1}^D$, a function $\widetilde{f} \in \widetilde{\mathcal{F}}$ such that

$$\mathfrak{R}\left(\widetilde{f}\right) \le \mathfrak{R}\left(f_*\right) + \frac{uLB}{\sqrt{D}}\left(1 + \sqrt{2\ln(1/\delta)}\right).$$

Proof *Given any functions f and g in $\mathcal{F}$, the Lipschitz hypothesis on c_{y_i} followed by the concavity of the square root (Jensen's inequality) gives*

$$\begin{aligned}
\mathfrak{R}\left(f\right) - \mathfrak{R}\left(g\right) &= \mathbf{E}_\mu c_y(f(x)) - c_y(g(x)) \\
&\le \mathbf{E}_\mu \left|c_y(f(x)) - c_y(g(x))\right| \\
&\le L\mathbf{E}_\mu \|f(x) - g(x)\|_{\mathcal{Y}} \\
&\le L\sqrt{\mathbf{E}_\mu \|f(x) - g(x)\|^2_{\mathcal{Y}}}.
\end{aligned}$$

apply proposition 7.4 to conclude.

Proposition 7.6 (Bound on the estimation error). *Let $c_y : \mathcal{Y} \to [0; C]$ be a L-Lipschitz cost function for all $y \in \mathcal{Y}$. Let $(\omega_j)_{j=1}^D$ be D fixed vectors in $\widehat{\mathcal{X}}$. If $\mathbf{s} = (x_i, y_i)_{i=1}^N \in (\mathcal{X} \times \mathcal{Y})^N$ are i. i. d. random variables then for all $\delta \in (0,1)$, then it holds with probability $1-\delta$ for all $\widetilde{f} \in \widetilde{\mathcal{F}}$ that*

$$\mathfrak{R}\left(\widetilde{f}\right) \le \mathfrak{R}_{emp}\left(\widetilde{f}, \mathbf{s}\right) + 2\sqrt{\frac{2}{N}}\left(LBT^{1/2} + C\sqrt{\ln(2/\delta)}\right).$$

where $\mathbf{Tr}\left[\widetilde{K}_e(e)\right] < T \in \mathbb{R}_+$.

Proof *Since $\widetilde{f} \in \widetilde{\mathcal{F}}$ and forall $j \in \mathbb{N}^*_D$, $\|\theta_j\|_{\mathcal{Y}} < B/D$, Thus $\|\theta\|_{\bigoplus_{j=1}^D \mathcal{Y}} < B/\sqrt{D}$. Moreover, if we define $\widetilde{\Phi}(x) = \bigoplus_{j=1}^D \Phi_x(\omega_j)$, it gives birth to a RKHS with kernel $D\Phi_x^*\Phi_z$ for all $x, z \in \mathcal{X}$. Thus with arguments similar to equation 7.3, noticing that the terms in $\sqrt{D}$ cancels out, we obtain a bound on the Rademacher complexity*

$$\mathcal{R}_N\left(\widetilde{\mathcal{F}}\right) \le \sqrt{2}BL\sqrt{N\,\mathbf{Tr}\left[\widetilde{K}_e(e)\right]}.$$

Eventually apply theorem 7.1. □

We are now ready to prove the main claim. Let f_* be a minimizer of $\mathcal{R}$ over $\mathcal{F}$, $\widetilde{f}$ a minimizer of $\mathcal{R}_{emp}$ over $\widetilde{\mathcal{F}}$ and $\widetilde{f}_*$ a minimizer of $\mathcal{R}$ over $\widetilde{\mathcal{F}}$. Then

$$\mathfrak{R}\left(\widetilde{f}\right) - \mathfrak{R}\left(f_*\right) = \mathfrak{R}\left(\widetilde{f}\right) - \mathfrak{R}\left(\widetilde{f}_*\right) + \mathfrak{R}\left(\widetilde{f}_*\right) - \mathfrak{R}\left(f_*\right). \tag{7.8}$$

The first difference in the right hand side of the equation is The estimation error. By proposition 7.6, with probability $1-\delta$, $\mathfrak{R}\left(\widetilde{f}_*\right) - \mathfrak{R}_{emp}\left(\widetilde{f}_*\right) \leq \epsilon_{est}$ and simultaneously, $\mathfrak{R}\left(\widetilde{f}\right) - \mathfrak{R}_{emp}\left(\widetilde{f}\right) \leq \epsilon_{est}$. By optimality of $\widetilde{f}$, $\mathfrak{R}_{emp}\left(\widetilde{f}\right) \leq \mathfrak{R}\left(\widetilde{f}_*\right)$. Combining these facts, with probability $1-\delta$,

$$\mathfrak{R}\left(\widetilde{f}\right) - \mathfrak{R}\left(\widetilde{f}_*\right) \leq 4\sqrt{\frac{2}{N}}\left(LBT^{1/2} + C\sqrt{\ln(2/\delta)}\right) = 2\epsilon_{est}.$$

Applying proposition 7.5 yields

$$\mathfrak{R}\left(\widetilde{f}_*\right) - \mathfrak{R}\left(f_*\right) \leq \frac{uLB}{\sqrt{D}}\left(1 + \sqrt{2\ln(1/\delta)}\right) = \epsilon_{app}.$$

Conclude by the union bound with probability $1-2\delta$ equation 7.8 is bounded by above by $2\epsilon_{est} + \epsilon_{app}$. Notice that $\mathbf{Tr}\left[\widetilde{K}_e(e)\right] = \frac{1}{D}\sum_{j=1}^D A(\omega_j)$. Thus if we have $\operatorname{ess\,sup}_{x\in\widehat{\mathcal{X}}} \mathbf{Tr}\left[A(\omega)\right] < \infty$, $\mathbf{Tr}\left[\widetilde{K}_e(e)\right] \leq \operatorname{ess\,sup}_{x\in\widehat{\mathcal{X}}} \mathbf{Tr}\left[A(\omega)\right]$. □

7.3 DISCUSSION

In this chapter we reviewed two ways of obtaining generalization bounds (see section 7.1 and subsection 7.1.2) for OVKs by bounding the function class complexity (Maurer [97]) or using algorithm stability arguments (Kadri et al. [73]). Then we used the results of Maurer [97] to prove the consistency of the algorithm obtained by minimizing equation 7.7, which is a variant of algorithm 4, where we replace the Tychonov regularizer by a projection in a $\|\cdot\|_\infty$ ball. This bound generalizes the work of Rahimi and Recht [122] to vector-valued learning.

Notice that we cannot directly derive a consistency bound from proposition 7.3 to algorithm 4. Indeed with arguments similar to remark 6.1, we can show that $\widetilde{f}_{\mathbf{s}} = \widetilde{\Phi}(x)^*\theta$ has a parameter vector θ such that $\left\|\theta_j\right\|_{\mathcal{Y}} < \sqrt{\frac{2}{\lambda D}}\sigma_{\mathcal{Y}}$, where $\sigma_y^2 = \frac{1}{N}\sum_{i=1}^N \|y_i\|_{\mathcal{Y}}^2$. Thus if $\widetilde{f}_{\mathbf{s}}$ is a solution of algorithm 4, we do not have $\widetilde{f}_{\mathbf{s}} \in \widetilde{\mathcal{F}}$. i. e. the Tichonov regularization is not "powerfull" enough to guarantee that $\widetilde{f}_{\mathbf{s}}$ belongs to $\widetilde{\mathcal{F}}$. One could argue that we could choose $\lambda = O(\sqrt{D})$ to obtain consistency with Tychonov regularization, however this makes little sense since in this case if $D \to \infty$ then $\lambda \to \infty$ the algorithm 4 will always return $\widetilde{f}_{\mathbf{s}} = 0$.

While the bound in proposition 7.3 shows the consistency of learning with ORFF it still has low and possibly suboptimal rate. Moreover it does

not allow to derive a number of features D smaller than the number of data since both of them decrease the error in $O(D^{-1/2})$ (respectively $O(N^{-1/2})$) as in the reference bound for scalar-valued random features by Rahimi and Recht [122]. In the scalar-valued kernel litterature, recent work of Bach [10] with much more involved analysis, gives simlar results to Rahimi and Recht [122] in the case of Tichonov regularization. Moreover it suggests that the number of features D to guarantee an error below some constant is linked to the decrease rate of the eigenvalues of the Mercer decomposition of scalar-valued kernel k. If the eigenvalues decrease in $O(m^{-2s})$ then the error is in $O\left(\log(D)^s D^{-s}\right)$. Lastly the new results of Rudi, Camoriano, and Rosasco [126] shows that for scalar-valued kernels, the kernel ridge regression algorithm (which is algorithm 4 with $A = 1$) generalizes optimality with a number of features $D = O(\sqrt{N})$. Thus the time complexity required for optimal generalization with RFFs in the case of kernel ridge regression is $O(ND^2) = O(N^2)$ and the space complexity is in $O(N^{1.5})$, if the random features are all stored and not computed in an online fashion[19]
.

[19] See subsection 6.2.2.

8

APPLICATION TO TIME SERIES MODELLING

This chapter shows how to use the ORFF methodology to non-linear vector autoregression. It is an instanciation of the ORFF framework to $\mathcal{X} = \mathcal{Y} = (\mathbb{R}^d, +)$. We also give a generalization of a stochastic gradient descent [43] to ORFF. This is a joint work with Néhémy Lim and Florence d'Alché-Buc and has been published at a workshop of ECML. It is based on the previous work Lim et al. [85] for time series vector autoregression with operator-valued kernels [25].

Contents

8.1 INTRODUCTION

Time series are ubiquitous in various fields such as climate, biomedical signal processing, videos understanding to name but a few. When linear models are not appropriate, a generic nonparametric approach to modeling is relevant. In this work we build on a recent work about Vector Autoregressive models using Operator-Valued Kernels [85, 86]. Vector autoregression is addressed in a Vector Valued Reproducing Kernel Hilbert Space with the important property to allow for couplings between outputs. Given a d-dimensional time series of N data points $\{ x_1, \dots, x_N \}$, autoregressive models based on operator-valued kernels have the form $\hat{x}_{t+1} = h(x_t) = \sum_{\ell=1}^{N-1} K(x_t, x_\ell)c_\ell$ where coefficients $c_\ell \in \mathbb{R}^d, \ell = 1, \dots, N-1$ are the model parameters. A naive approach for training such a model requires a memory complexity $O(N^2 d^2)$, which makes the method prohibitive for large-scale problems.

To scale up standard algorithms, we define an approximated operator-valued feature map $\tilde{\Phi} : \mathbb{R}^d \to \mathbb{R}^D$ that allows to approximate the aforementioned model h in the RKHS by the following function

$$\tilde{h}(x_t) = \tilde{\Phi}(x_t)^* \theta \approx h(x_t).$$

The features maps are matrices of size $D \times d$ where D controls the quality of the approximation, d is the dimension of the inputs and θ is here the parameter vector to learn. This formulation allows to reduce the memory complexity to $O((N-1)D + (N-1)d)$ which is now linear w. r. t. the number of data points (see section 6.1). The principle used for building the feature map extends the idea of scalar Random Fourier Features to the operator-valued case [121, 148].

8.2 OPERATOR-VALUED KERNELS FOR VECTOR AUTOREGRESSION

Assume that we observe a dynamical system composed of $d \in \mathbb{N}^*$ state variables at $N \in \mathbb{N}^*$ evenly-spaced time points. The resulting discrete multivariate time series is denoted by $x_{1:N} = (x_\ell)_{i=1}^N$ where $x_\ell \in \mathbb{R}^d$ denotes the state of the system at time t_ℓ, $\ell \in \mathbb{N}_N^*$. It is generally assumed that the evolution of the state of the system is governed by a function h, such that $x_t = h(x_{t-p}, \dots, x_{t-1}) + u_t$ where t is a discrete measure of time and u_t is a zero-mean noise random vector. Then h is usually referred to as a vector autoregressive model of order p. In the remainder of the paper, we consider first-order vector autoregressive models, that is $p = 1$. In a supervised learning setting, the vector autoregression problem consists in learning a model $\hat{h} : \mathbb{R}^d \to \mathbb{R}^d$ from a given training set

$$\mathbf{s} = ((x_1, x_2), \dots, (x_{N-1}, x_N)) \in \left(\mathbb{R}^d \times \mathbb{R}^d\right)^N.$$

In the literature, a standard approach to vector autoregressive modeling is to fit a VAR model. The VAR(1) model reads $h(x_t) = Ax_t$ where A is an

$d \times d$ matrix whose structure encodes the temporal relationships among the d state variables.

However, due to their intrinsically linear nature, VAR models fail to capture the nonlinearities underlying realistic dynamical systems. This paper builds upon the work of Lim et al. [85] where the authors introduced a family of nonparametric nonlinear autoregressive models called OKVAR. OKVAR models rely on the theory of operator-valued kernels [115, 134], which provides a versatile framework for learning vector-valued functions [4, 34, 98]. Those models can be regarded as natural extensions of VAR models to the nonlinear case.

Next, we recall key elements of the theory of VV-RKHS of functions from $\mathbb{R}^d$ to $\mathbb{R}^d$ (see section 4.2 for the detailed construction). We first introduced the matrix-valued kernel which is an instance of OVKss.

Definition 8.1 (Matrix-valued kernels). *A function $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^{d\times d}$ is said to be a positive $\mathbb{R}^{d\times d}$-valued kernel if:*

1. $\forall x, \ z \in \mathbb{R}^d, \ K(x, z) = K(z, x)^*$,
2. $\forall m \in \mathbb{N}, \ \forall ((x_i, y_i))_{i=1}^m \in \left(\mathbb{R}^d \times \mathbb{R}^d\right)^m, \ \sum_{i,j=1}^m y_i^* K(x_i, x_j) y_j \geq 0.$

Furthermore, for a given $\mathbb{R}^{d\times d}$-valued kernel K, we associate K with a unique VV-RKHS $(\mathcal{H}_K, \langle \cdot, \cdot \rangle_{\mathcal{H}_K})$ of functions from $\mathbb{R}^d$ to $\mathbb{R}^d$. The precise construction of $\mathcal{H}_k$ can be found in section 3.3. In this section, we assume that all functions $h \in \mathcal{H}_K$ are continuous. Then K is called an $\mathbb{R}^d$-Mercer kernel (see definition 3.8).

Similarly to the case of scalar-valued kernels, working within the framework of VV-RKHS allows to take advantage of representer theorems (theorem 6.2) for a class of regularized loss functions such as ridge regression. More precisely, we consider h, a nonparametric vector autoregressive model of the following form assuming we have observed N data points. Given x_t the state vector at time t, we have $\hat{x}_{t+1} = \sum_{\ell=1}^{N-1} K(x_t, x_\ell) c_\ell$ where $x_{1:N} = (x_i)_{i=1}^N \in \left(\mathbb{R}^d\right)^N$ is the observed time series, $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^{d\times d}$ is a matrix-valued kernel and $(c_1)_{i=1}^{N-1} \in \left(\mathbb{R}^d\right)^{N-1}$ are the model parameters. We call OKVAR any model of the above form. In Lim et al. [85], the authors developed a family of OKVAR models based on appropriate choices of kernels to address the problem of network inference where both the parameters c_ℓ, $\ell \in \mathbb{N}^*_{N-1}$ and the OVK itself are learned using a proximal block coordinate descent algorithm [20] under sparsity constraints. In the following, we will not consider the kernel learning problem and will use a simple ridge loss. We will also illustrate our approach to a well known class of OVK, called *decomposable* or *separable* matrix-valued kernels [32, 98], and intance of Decomposable OVK that were originally introduced to solve multi-task learning problems [51]. Other kernels may also be considered as developed in subsection 3.3.3.

[20] *See for instance Parikh, Boyd, et al. [112] about proximal algorithms and Fercoq and Peter [54], Fercoq and Richtárik [55], and Richtárik and Takáč [123] for proximal block coordinate descent.*

Proposition 8.1 (Decomposable matrix-valued kernels). *Let the function $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ be a scalar-valued kernel and $\Gamma \in \mathbb{R}^{d\times d}$ a positive semidefinite matrix of size $d \times d$. Then function $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^{d\times d}$ defined for all $(x, z) \in \mathbb{R}^d \times \mathbb{R}^d$ as $K(x, z) = k(x, z)\Gamma$ is a decomposable matrix-valued kernel.*

A common choice for the scalar-valued kernel is the Gaussian kernel

$$k_{\text{Gauss}}(x, z) = \exp(-\|x - z\|_2^2/(2\sigma^2))$$

for any $x, z \in \mathbb{R}^d$ and $\sigma \in \mathbb{R}_+$. The corresponding decomposable kernel is referred to as K_{dec} and is as $K_{\text{dec}}(x, z) = k_{\text{Gauss}}(x, z)\Gamma$ with Γ a positive semidefinite matrix.

While the model parameters c_ℓ's are estimated under sparsity constraints in Lim et al. [85], here we consider the classic kernel ridge regression setting where the loss function to minimize is

$$\mathcal{J}_\lambda(h) = \frac{1}{N-1}\sum_{\ell=2}^{N}\|h(x_{\ell-1}) - x_\ell\|_2^2 + \lambda\|h\|_{\mathcal{H}_K}^2 \tag{8.1}$$

with $\lambda \geq 0$ and $\|h\|_{\mathcal{H}_K}^2 = \sum_{t,\ell=1}^{N-1} c_t^* K(x_t, x_\ell)c_\ell$. The optimization problem is solved using a L-BFGS-B [31] which is well suited for optimization problems with a large number of parameters, and is wildly used as a training algorithm on small/medium-scale problems. However, like standard kernel methods, OKVAR suffers from unfavorable computational complexity both in time and memory since it needs to store the full Gram matrix, preventing its ability to scale to large data sets and making it really slow on medium scale problem. We argue that this obstacle can be effectively overcome: in the following we develop a method to scale up OKVAR to successfully tackle medium/large scale autoregression problems.

8.3 OPERATOR-VALUED RANDOM FOURIER FEATURES

We now introduce our methodology to approximate OVKs. Given a translation-invariant kernel $K(x, z) = K_0(x - z)$, we approximate K by finding an explicit feature map such that $\tilde{\Phi}(x)^*\tilde{\Phi}(z) \approx K_0(x - z)$. The idea is to use a generalization of Bochner's theorem for the OVK family that states that any translation-invariant OVK can be written as the Fourier transform of a positive operator-valued measure. More precisely, we build on the following proposition first proved in [34]. More details can be found in section 4.2.

In the following, suppose that $K_0 = k_0(\cdot)A$ is a decomposable kernel. Decomposable kernels belong to the family of translation-invariant OVKs. From proposition 4.3 we see that $C(\omega)_{ij} = \mathcal{F}^{-1}[k_0(\cdot)](\omega)A_{ij}$. We decompose A as $A = BB^*$, note that A does not depend on ω, and we denote $\bigoplus_{j=1}^{D} z_j$ the Dm-long column vector obtained by stacking vectors

$z_j \in \mathbb{R}^m$. Then we define an approximate feature map for K_0, called Operator-valued Random Fourier Feature (ORFF) map [25] as follows (see subsection 4.2.2 and section 4.3). For all $x \in \mathbb{R}^d$,

$$\tilde{\Phi}^{dec}(x) = \frac{1}{\sqrt{D}} \bigoplus_{j=1}^{D} \begin{pmatrix} \cos \langle x, \omega_j \rangle B^* \\ \sin \langle x, \omega_j \rangle B^* \end{pmatrix}, \quad \omega_j \sim \mathcal{F}^{-1}\left[k_0\right],$$

which can also be expressed as a Kronecker product $\otimes$ of a scalar feature map with a matrix (see subsection 6.2.4): $\tilde{\Phi}^{dec}(x) = \tilde{\varphi}(x) \otimes B^*$ where

$$\tilde{\varphi}(x) = \frac{1}{\sqrt{D}} \bigoplus_{j=1}^{D} \begin{pmatrix} \cos \langle x, \omega_j \rangle \\ \sin \langle x, \omega_j \rangle \end{pmatrix}, \quad \omega_j \sim \mathcal{F}^{-1}\left[k_0\right]$$

is a scalar-valued feature map. In particular, if k_0 is a Gaussian kernel with bandwidth σ^2, then $\mathcal{F}^{-1}\left[k_0\right] = \mathcal{N}(0, 1/\sigma^2)$ as proven in Rahimi and Recht [121]. More examples on different OVK can be found in subsection 4.2.2 as well as a proof of the uniform convergence of the kernel approximation in section 5.1 defined by $\tilde{K}(x, z) = \tilde{\Phi}(x)^* \tilde{\Phi}(z)$ towards the true kernel. In the case of vector autoregression, we consider a model $\tilde{h}$ of the form: $\hat{x}_{t+1} = \tilde{\Phi}(x_t)^* \theta$. That model is referred to as ORFFVAR in the remainder of the section. Now, given the operator-valued feature map, we get a linear model, and we want to minimize the regularized risk

$$\mathcal{J}_\lambda(\theta) = \frac{1}{N-1} \sum_{\ell=2}^{N} \|(\tilde{\varphi}(x_{\ell-1})^* \otimes B)\theta - x_\ell\|_2^2 + \lambda \|\theta\|_2^2$$

with $\lambda > 0$ instead of equation 8.1 (see theorem 6.3). In their paper Brault, Lim, and d'Alché-Buc [25] proposed to formulate the learning problem as a Stein equation when dealing with decomposable kernels, and then used an appropriate solver [144]. We opted here for a more general algorithm, which is a variant of the doubly stochastic gradient descent [43]. In a few words, this algorithm is a stochastic gradient descent that takes advantage of the feature representation of the kernel allowing the number of features to grow along with the number of points. Dai et al. [43] show that the number of iterations needed for achieving a desired accuracy ϵ using a stochastic approximation is $\Omega(1/\epsilon)$, making it competitive compared to other stochastic methods for kernels such as NORMA [76] and its OVK adaptation ONORMA [7]. We propose here in algorithm 5, an extension of the doubly stochastic gradient descent of Dai et al. [43] to OVKs. Additionally we consider a batch approach w. r. t. the data and the features, and make it possible to "cap" the maximum number of features. The inputs of the algorithm are: $\mathcal{X}$ the input data, $\mathcal{Y}$ the targets, K_e the OVK used for learning, γ_t the learning rate (see Dai et al. [43] for a discussion on the selection of a proper learning rate), T the number of iterations, n the size of data batch, b the size of the feature batch, and D the maximum number of features. Note that if K_0 is a scalar kernel, $D = T$, $b = 1$ and $n = 1$, we retrieve the algorithm formulated in Dai et al. [43].

Algorithm 5: Block-coordinate mini-batch doubly SGD.

Data: $\mathcal{X}$, $\mathcal{Y}$, K_e, γ_t, λ, T, n, D, b
Result: Find θ
Let $D_b = D/b$ and find (ω, x), $B(\omega)$ and $\mu(\omega)$ from K_e;
for $i = 1$ **to** D_b **do**
  $\theta^1_{i,\cdot} = 0$;
end
for $t = 1$ **to** T **do**
  $\mathcal{A}_t = \mathcal{X}_t \times \mathcal{Y}_t$, a random subsample of n data from $\mathcal{X} \times \mathcal{Y}$;
  $h\left(\mathcal{X}_t\right) = \text{predict}\left(\mathcal{X}_t, \theta^t, K_e\right)$; `// Make a prediction.`
  $\Omega_i \sim \mu$ with seed i, where $i = ((t-1) \mod D_b) + 1$; `// Sample b features from μ.`
  for $\omega \in \Omega_i$ `// Update the parameters from the gradient.`
  do
    $$\theta^{t+1}_{i,\omega} = \theta^t_{i,\omega} - \gamma_t \left(\frac{1}{|\mathcal{A}_t|} \sum_{(x,y)\in\mathcal{A}_t} \frac{B(\omega)^*(\omega, x)(h(x) - y)}{\sqrt{D}} + \lambda \theta^t_{i,\omega} \right);$$
  end
end
return θ^{t+1}

In addition, the convergence of the algorithm can be speeded-up by preconditioning by the Hessian of the system. An experimental C++ code is available at `https://github.com/RomainBrault/OV2SGD`.

Algorithm 6: $h\left(\mathcal{X}\right)$ =predict$\left(\mathcal{X}, \theta, K_e\right)$

Data: $\mathcal{X}$, θ, K_0
Find (ω, x), $B(\omega)$ and $\mu(\omega)$ from K_0;
$f(\mathcal{X}) = 0$;
for $x \in \mathcal{X}$ **do**
  for $i = 1$ **to** D **do**
    $\Omega_i \sim \mu(\omega)$ with seed i;
    for $\omega \in \Omega_i$ **do**
      $h(x) = h(x) + \overline{(\omega, x)} B(\omega) \theta_{i,\omega}$;
    end
  end
end
return $h(\mathcal{X})$

8.3.1 *Numerical Performance*

We now apply algorithm 5 to toy and real datasets.

8.3.2 *Simulated data*

To assess the performance of our models, we start our investigation by generating discrete d-dimensional time series $(x_t)_{t\geq 1}$ as follows

$$\begin{cases} x_1 \sim \mathcal{N}(0, \Sigma_x) \\ x_{t+1} = h(x_t) + u_{t+1}, \quad \forall t > 0. \end{cases} \tag{8.2}$$

where the residuals are homoscedastic and distributed according to $u_t \sim \mathcal{N}(0, \Sigma_u)$. We study two different kinds of noise: an isotropic noise with covariance $\Sigma_u = \sigma_u^2 I_d$ and an anisotropic noise with Toeplitz structure $\Sigma_{u,ij} = \nu^{|i-j|}$, where ν lives in $(0, 1)$. We generated $N = 1000$ data points and used a Sequential cross-validation (SCV) with time windows $N_t = N/2$ to measure the Mean Squared Error SCV-MSE of the different models. Next, we compare the performances of VAR(1), OKVAR and ORFFVAR through three scenarios. Across the simulations, the topological structures of the underlying dynamical systems are encoded by a matrix A of size 5×5. All entries of A are set to zero except for the diagonal where all coefficients are equal to 0.9 for Settings 1 and 3 and 0.5 for Setting 2. Then five off-diagonal coefficients are drawn randomly from $\mathcal{N}(0, 0.3)$ for Settings 1 and 3 and $\mathcal{N}(0, 0.5)$ for Setting 2. We check that all the eigenvalues of A are less than one to ensure the stability of the system. More specifically, we picked the following values of parameters for each scenario.

- **Setting 1: Linear model.**: $h(x_t) = Ax_t$, $\nu = 0.9$ and $\sigma_u = 0.9$,
- **Setting 2: Exponential model.**: $h(x_t) = A\exp(x_t)$ where exp is the element-wise exponential function, $\nu = 0.09$ and $\sigma_u = 0.09$,
- **Setting 3: Sine model.**: $h(x_t) = A\sin(x_t)$ where sin is the element-wise sine function, $\nu = 0.9$ and $\sigma_u = 0.009$.

ORFFVAR is instantiated with $D = 25$ random features in presence of a white noise while we set $D = 50$ in case of a Toeplitz noise. We summarize the computational efficiency and the statistical accuracy of the models in table 8.1. Throughout all the experiments, we set B as the identity matrix of size $d \times d$. This reflects the absence of a prior on the structure of the data. A further study on the influence of the choice of B can be found in Álvarez, Rosasco, and Lawrence [4] and propositions 3.8 and 4.12.

In Setting 1, we observe that OKVAR does not provide any advantage over VAR(1) as expected since the data were generated according to a linear VAR(1) model. Note that OKVAR takes orders of magnitude more time to achieve the same performance as VAR(1) while ORFFVAR performs equally well with a competitive timing. In nonlinear scenarios (Settings 2 and 3), OKVAR and ORFFVAR consistently outperform VAR(1). Noticeably, ORFFVAR reaches the accuracy of OKVAR with the computation time of VAR(1).

Setting		1			2			3		
model	noise	SVC-MSE	variance	time	SVC-MSE	variance	time	SVC-MSE	variance	time
VAR(1)	White	**0.914979**	**0.572485**	0.002467(s)	0.001275	0.000994	0.002346(s)	0.009534	0.006003	**0.001697(s)**
	Toeplitz	**1.091096**	**1.267880**	0.004822(s)	0.017014	0.013498	**0.002050(s)**	**0.116901**	0.127396	0.001702(s)
ORFFVAR	White	0.919663	0.572936	**0.000994(s)**	**0.001003**	**0.000647**	**0.001284(s)**	0.009536	0.005998	0.002377(s)
	Toeplitz	1.097183	1.268978	**0.001022(s)**	**0.012635**	**0.008837**	0.012144(s)	0.116964	**0.127395**	**0.000934(s)**
OKVAR	White	0.958790	0.591934	0.104706(s)	0.001100	0.000731	0.027099(s)	**0.009227**	**0.005717**	0.014458(s)
	Toeplitz	1.410969	1.312243	0.289046(s)	0.013854	0.010977	1.856988(s)	0.160133	0.136570	0.019170(s)

Table 8.1: Sequential SCV-MSE and computation times for VAR(1), ORFFVAR and OKVAR on synthetic data (Settings 1, 2 and 3).

8.3.3 *Influence of the number of random features*

Here, we investigate the impact of D, the number of random features for ORFFVAR. To this end, we generated $N = 10000$ data points following equation 8.2, with exponential nonlinearities and white noise as in Setting 2. We performed a sequential cross-validation on a window of $N/2$ data. As expected the error decreases with the number of random features D (table 8.2). For the same computation time ($D = 25$) as VAR(1), ORFFVAR achieves an SCV-MSE that is twice as small.

model	$D = 1$	$D = 5$	$D = 10$	$D = 25$	$D = 50$	$D = 100$	VAR(1)
SVC-MSE	0.005342	0.001111	0.000991	0.000962	0.000949	0.000944	0.001660
variance	0.008639	0.000793	0.000660	0.000618	0.000608	0.000605	0.001363
time	0.001191(s)	0.002384(s)	0.003614(s)	0.018469(s)	0.038229(s)	0.069294(s)	0.019634(s)

Table 8.2: SVC-MSE with respect to D the number of random features for ORFFVAR.

8.3.4 *Real datasets*

We now investigate three real datasets. The performances of the models on those datasets are recorded in table 8.3. Throughout the experiments, the hyperparameters are set as follows: the bandwidth of the Gaussian kernel σ is chosen as the median of the Euclidean pairwise distances and the regularization parameter λ was tuned on a grid. The number of random features D and the parameters in algorithm 5 were picked so as to reach the level of accuracy of OKVAR/VAR.

MACRODATA This dataset is part of the Python library `Statmodels`[1]. It contains 204 US macroeconomic data points collected on the period 1959–2009. Each data point represents 12 economic features. No pre-processing is applied before learning. We measure SCV-MSE using a window of 25 years (50 points). We instantiated algorithm 5 as follows: $\gamma_t = 1$, $\lambda = 10^{-3}$, $D = 100$, $T = 2$ and $b = 50$ for ORFF and $\lambda = 0.00025$ and $\sigma = 11.18$ for OKVAR.

GESTURE PHASE. This dataset[2] is constructed using features extracted from seven videos with people gesticulating. We present the results for videos 1 and 4, consisting in 1069 data points and 31 features. Data are normalized prior to learning. We measure SCV-MSE using a time window of 200 points. We implemented ORFFVAR with $\gamma_t = 1$, $\lambda = 10^{-3}$, $D = 100$, $T = 2$ and $b = 50$.

CLIMATE. This dataset [91] contains monthly meteorological measurements of 18 variables (temperature, CO2 concentration, ...) collected at 135 different locations throughout the USA and recorded over 13 years, thus resulting in 135 time series of dimension 18 and length 156. Data are standardized at each station. A unique model is learned for all stations. SCV-MSE is measured on a window of 1872 points, corresponding to the data of all the 135 stations over one year. Specifically, we set the parameters of ORFFVAR as follows: $\gamma_t = 1$, $\lambda = 10^{-6}$, $D = 100$, $T = 1$ and $b = 100$.

HEART. The dataset is a multivariate time-serie recorded from a patient in the sleep laboratory of the Beth Israel Hospital in Boston, Massachusetts[3]. The attributes are the heart rate, the chest volume (respiration force) and the blood oxygen concentration. The time-serie contains 17000 points recorded at 2Hz during roughly 4 hours 30 minutes. We used a window of 240 points for the sequential cross-validation (corresponding to 2 minutes of observations).

			ORFFVAR			VAR(1)			OKVAR		
Dataset	N	d	SCV-MSE	variance	time	SCV-MSE	variance	time	SCV-MSE	variance	time
Macrodata	#203	#12	**445.9**	84.5	0.014(s)	449.1	1021	0.0005(s)	499.8	793.0	0.641(s)
Gesture phase 1	#1743	#31	**0.741**	2.999	0.009(s)	0.980	3.370	0.0014(s)	N. A.	N. A.	N. A.
Gesture phase 4	#1069	#31	**0.473**	2.406	0.061(s)	0.768	6.49	0.0075(s)	N. A.	N. A.	N. A.
Climate	#19375	#18	**0.237**	0.2128	0.396(s)	0.266	0.218	0.0124(s)	N. A.	N. A.	N. A.
Heart	#16999	#3	0.262	1.020	0.011(s)	**0.259**	1.040	0.0010(s)	N. A.	N. A.	N. A.

Table 8.3: SCV-MSE and computation times for ORFFVAR, VAR(1) and OKVAR on real datasets.

1 https://github.com/statsmodels/statsmodels
2 https://archive.ics.uci.edu/ml/datasets/Gesture+Phase+Segmentation
3 http://www-psych.stanford.edu/~andreas/Time-Series/SantaFe.html

8.4 DISCUSSION

Operator-Valued Random Fourier Feature provides a way to approximate OVK and in the context of time series, allows for nonlinear Vector Autoregressive models that can be efficiently learned both in terms of computing time and memory. We illustrate the approach with a simple family of Operator-valued kernels, the so-called decomposable kernels but other kernels may be used. While we focused on first-order autoregressive models, we will consider extensions of our models for higher orders. In this work, the kernel hyperparameter B is given prior to learning, however it would be interesting to learn B as in OKVAR. Thus, a promising perspective is to use these models in tasks such as network inference and search for causality graphs among the state variables for large-scale time series [85, 86].

Part III

WORK IN PROGRESS AND FINAL WORDS

9

WORK IN PROGRESS

9.1 LEARNING FUNCTION-VALUED FUNCTIONS

9.1.0.1 *One class SVM revisited*

9.1.0.2 *Many quantile regression*

9.2 NEURAL NETWORKS, DEEP LEARNING

9.3 OPERALIB

https://github.com/operalib/operalib

10

CONLUSIONS

Part IV

APPENDIX

A

PROOFS OF THEOREMS

In this appendix we detail the proofs of corollary 5.2 and corollary 5.3. These two corollaries applying on compact subsets of Banach spaces are the consequences of more generic propositions (proposition A.1 and proposition A.2) working on any compact subsets of Polish spaces. Eventually we give a proof on the variance bound given in proposition A.3.

Contents

A.1 PROOF OF THE ERROR BOUND WITH HIGH PROBABILITY OF THE ORFF ESTIMATOR

We recall the notations $\delta = x \star z^{-1}$, for all $x, z \in \mathcal{X}$, $\tilde{K}(x,z) = \widetilde{\Phi}(x)^* \widetilde{\Phi}(z)$, $\tilde{K}^j(x,z) = \Phi_x(\omega_j)^* \Phi_z(\omega_j)$, where $\omega_j \sim \mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho}$ and $K_e(\delta) = K(x,z)$ and $\tilde{K}_e(\delta) = \tilde{K}(x,z)$. For the sake of readabilty, we use throughout the proof the quantities

$$F(\delta) := \tilde{K}(x,z) - K(x,z)$$
$$F^j(\delta) := \frac{1}{D}\left(\tilde{K}^j(x,z) - K(x,z)\right)$$

We also view $\mathcal{X}$ as a metric space endowed with the distance $d_{\mathcal{X}} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$. Compared to the scalar case, the proof follows the same scheme as the one described in [121, 148], but we consider an operator norm as measure of the error and therefore concentration inequality dealing with these operator norm. The main feature of proposition A.1 is that it covers the case of bounded ORFF as well as unbounded ORFF. In the case of bounded ORFF, a Bernstein inequality for matrix concentration such that the one proved in Mackey et al. [95, Corollary 5.2] or the formulation of Tropp [154] recalled in Koltchinskii et al. [77] is suitable. However some kernels like the curl and the divergence-free kernels do not have obvious bounded $\|F^j\|_{\mathcal{Y},\mathcal{Y}}$ but exhibit F^j with subexponential tails. Therefore, we use an operator Bernstein concentration inequality adapted for random matrices with subexponential norms.

A.1.1 *Epsilon-net*

Let $\mathcal{C} \subseteq \mathcal{X}$ be a compact subset of $\mathcal{X}$. Let $\mathcal{D}_{\mathcal{C}} = \left\{ x \star z^{-1} \;\middle|\; x, z \in \mathcal{C} \right\}$ with diameter at most $2|\mathcal{C}|$ where $|\mathcal{C}|$ is the diameter of $\mathcal{C}$. Since $\mathcal{C}$ is supposed compact, so is $\mathcal{D}_{\mathcal{C}}$. Since $\mathcal{D}_{\mathcal{C}}$ is also a metric space it is well known that a compact metric space is totally bounded. Thus it is possible to find a finite ϵ-net covering $\mathcal{D}_{\mathcal{C}}$. We call $T = \mathcal{N}(\mathcal{D}_{\mathcal{C}}, r)$ the number of closed balls of radius r required to cover $\mathcal{D}_{\mathcal{C}}$. For instance if $\mathcal{D}_{\mathcal{C}}$ is a subspace finite dimensional Banach space with diameter at most $2|\mathcal{C}|$ it is possible to cover the space with at most $T = (4|\mathcal{C}|/r)^d$ balls of radius r (see Cucker and Smale [42, proposition 5]).

Let us call $\delta_i, i = 1, \dots, T$ the center of the i-th ball, also called anchor of the ϵ-net. Denote L_F the Lipschitz constant of F. Let $\|\cdot\|_{\mathcal{Y},\mathcal{Y}}$ be the operator norm on $\mathcal{L}(\mathcal{Y})$ (largest eigenvalue). We introduce the following technical lemma.

Lemma A.1 *$\forall \delta \in \mathcal{D}_{\mathcal{C}}$, if*

$$L_F \leq \frac{\epsilon}{2r} \tag{A.1}$$

and

$$\|F(\delta_i)\|_{\mathcal{Y},\mathcal{Y}} \leq \frac{\epsilon}{2}, \quad \textit{for all } i \in \mathbb{N}^*_T \tag{A.2}$$

then $\|F(\delta)\|_{\mathcal{Y},\mathcal{Y}} \le \epsilon$.

Proof

$$\begin{aligned}\|F(\delta)\|_{\mathcal{Y},\mathcal{Y}} &= \|F(\delta) - F(\delta_i) + F(\delta_i)\|_{\mathcal{Y},\mathcal{Y}} \\ &\le \|F(\delta) - F(\delta_i)\|_{\mathcal{Y},\mathcal{Y}} + \|F(\delta_i)\|_{\mathcal{Y},\mathcal{Y}}\end{aligned}$$

for all $0 < i < T$. *Using the Lipschitz continuity of* F *we have*

$$\|F(\delta) - F(\delta_i)\|_{\mathcal{Y},\mathcal{Y}} \le d_{\mathcal{X}}(\delta, \delta_i) L_F \le r L_F$$

hence

$$\|F(\delta)\|_{\mathcal{Y},\mathcal{Y}} \le r L_F + \|F(\delta_i)\|_{\mathcal{Y},\mathcal{Y}} = \frac{r\epsilon}{2r} + \frac{\epsilon}{2} = \epsilon.$$

To apply the lemma, we must bound the Lipschitz constant of the operator-valued function F (equation A.1) and $\|F(\delta_i)\|_{\mathcal{Y},\mathcal{Y}}$, for all $i = 1, \dots, T$ as well (equation A.2).

A.1.2 *Bounding the Lipschitz constant*

This proof is a slight generalization of Minh [101] to arbitrary metric spaces. It differ from our first approach [24], based on the proof of Sutherland and Schneider [148] which was only valid for a finite dimensional input space $\mathcal{X}$ and imposed a twice differentiability condition on the considered kernel.

Lemma A.2 *Let* $H_\omega \in \mathbb{R}_+$ *be the Lipschitz constant of* $h_\omega(\cdot)$ *and assume that*

$$\int_{\widehat{\mathcal{X}}} H_\omega \|A(\omega)\|_{\mathcal{Y},\mathcal{Y}} d\mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho}(\omega) < \infty.$$

Then the operator-valued function $K_e : \mathcal{X} \to \mathcal{L}(\mathcal{Y})$ *is Lipschitz with*

$$\|K_e(x) - K_e(z)\|_{\mathcal{Y},\mathcal{Y}} \le d_{\mathcal{X}}(x, z) \int_{\widehat{\mathcal{X}}} H_\omega \|A(\omega)\|_{\mathcal{Y},\mathcal{Y}} d\mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho}(\omega). \quad (A.3)$$

Proof *We use the fact that the cosine function is Lipschitz with constant* 1 *and* h_ω *Lipschitz with constant* H_ω. *For all* $x, z \in \mathcal{X}$ *we have*

$$\begin{aligned}\left\|\tilde{K}_e(x) - K_e(z)\right\|_{\mathcal{Y},\mathcal{Y}} &= \left\|\int_{\widehat{\mathcal{X}}} (\cos h_\omega(x) - \cos h_\omega(z)) A(\omega) d\mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho}\right\|_{\mathcal{Y},\mathcal{Y}} \\ &\le \int_{\widehat{\mathcal{X}}} |\cos h_\omega(x) - \cos h_\omega(z)| \|A(\omega)\|_{\mathcal{Y},\mathcal{Y}} d\mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho} \\ &\le \int_{\widehat{\mathcal{X}}} |h_\omega(x) - h_\omega(z)| \|A(\omega)\|_{\mathcal{Y},\mathcal{Y}} d\mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho} \\ &\le d_{\mathcal{X}}(x, z) \int_{\widehat{\mathcal{X}}} H_\omega \|A(\omega)\|_{\mathcal{Y},\mathcal{Y}} d\mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho}\end{aligned}$$

In the same way, considering $\tilde{K}_e(\delta) = \frac{1}{D}\sum_{j=1}^{D} \cos h_{\omega_j}(\delta) A(\omega_j)$, where $\omega_j \sim \mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho}$, we can show that $\tilde{K}_e$ is Lipschitz with

$$\left\|\tilde{K}_e(x) - \tilde{K}_e(z)\right\|_{\mathcal{Y},\mathcal{Y}} \le d_{\mathcal{X}}(x, z) \frac{1}{D} \sum_{j=1}^{D} H_{\omega_j} \left\|A(\omega_j)\right\|_{\mathcal{Y},\mathcal{Y}}.$$

Combining the Lipschitz continuity of $\tilde{K}_e$ and $\tilde{K}$ (lemma A.2) we obtain

$$\begin{aligned}\|F(x)-F(z)\|_{\mathcal{Y},\mathcal{Y}} &= \left\|\tilde{K}_e(x)-\tilde{K}_e(x)-\tilde{K}_e(z)+K_e(z)\right\|_{\mathcal{Y},\mathcal{Y}} \\ &\leq \left\|\tilde{K}_e(x)-\tilde{K}_e(z)\right\|_{\mathcal{Y},\mathcal{Y}}+\|K_e(x)-K_e(z)\|_{\mathcal{Y},\mathcal{Y}} \\ &\leq d_{\mathcal{X}}(x,z)\Bigg(\int_{\widehat{\mathcal{X}}} H_\omega \|A(\omega)\|_{\mathcal{Y},\mathcal{Y}} d\mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho} \\ &\qquad + \frac{1}{D}\sum_{j=1}^{D} H_{\omega_j}\left\|A(\omega_j)\right\|_{\mathcal{Y},\mathcal{Y}}\Bigg)\end{aligned}$$

Taking the expectation yields

$$\mathbf{E}_{\widehat{\mathbf{Haar}},\rho}\left[L_F\right]=2\int_{\widehat{\mathcal{X}}} H_\omega\|A(\omega)\|_{\mathcal{Y},\mathcal{Y}} d\mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho}$$

Thus by Markov's inequality,

$$\begin{aligned}\mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho}\left\{(\omega_j)_{j=1}^D \mid L_F\geq\epsilon\right\} &\leq \frac{\mathbf{E}_{\widehat{\mathbf{Haar}},\rho}\left[L_F\right]}{\epsilon} \\ &\leq \frac{2}{\epsilon}\int_{\widehat{\mathcal{X}}} H_\omega\|A(\omega)\|_{\mathcal{Y},\mathcal{Y}} d\mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho}.\end{aligned} \tag{A.4}$$

A.1.3 *Bounding F on a given anchor point δ_i*

To bound $\|F(\delta_i)\|_{\mathcal{Y},\mathcal{Y}}$, Hoeffding inequality devoted to matrix concentration [95] can be applied. We prefer here to turn to tighter and refined inequalities such as Matrix Bernstein inequalities (Sutherland and Schneider [148] also pointed that for the scalar case). The first non-commutative (matrix) concetration inequalities are due to the pioneer work of Ahlswede and Winter [2], using bound on the moment generating function. This gave rise to many applications Koltchinskii et al. [77], Oliveira [110], and Tropp [154] ranging from analysis of randomized optimization algorithm to analysis of random graphs and generalization bounds usefull in machine learning. The following inequality has been proposed in [77].

Theorem A.1 (Bounded non-commutative Bernstein). *From Theorem 3 of Koltchinskii et al. [77], consider a sequence $(X_j)_{j=1}^D$ of D independent Hermitian $p\times p$ random matrices acting on a finite dimensional Hilbert space $\mathcal{Y}$ that satisfy* $\mathbf{E}X_j=0$, *and suppose that there exist some constant $U\geq\left\|X_j\right\|_{\mathcal{Y},\mathcal{Y}}$ for each index j. Denote the proxy bound on the matrix variance*

$$V\succcurlyeq\sum_{j=1}^{D}\mathbf{E}X_j^2.$$

Then, for all $\epsilon\geq 0$,

$$\mathbf{Pr}\left\{\left\|\sum_{j=1}^{D}X_j\right\|_{\mathcal{Y},\mathcal{Y}}\geq\epsilon\right\}\leq p\exp\left(-\frac{\epsilon^2}{2\|V\|_{\mathcal{Y},\mathcal{Y}}+2U\epsilon/3}\right)$$

This bound we used in our original paper [24] has the default to grow lineraly with the dimension p of the output space $\mathcal{Y}$. However if the evaluation of the operator-valued kernel at two points yields a low-rank matrix, this bound could be improved since only a few principal dimensions are relevent. Moreover this bound cannot be used when dealing with operator-valued kernel acting on infinite dimensional Hilbert spaces. Yet when the evaluation of the kernel at two points is a compact operator, we could expect the concentration inequality to be valid since it has finite trace. Recent results of Minsker [104] consider the notion of intrinsic dimension to avoid this “curse of dimensionality”.

Definition A.1 *Let A be a trace class operator acting on a Hilbert space $\mathcal{Y}$. We call intrinsic dimension the quantity*

$$IntDim(A) = \frac{\mathbf{Tr}\left[A\right]}{\|A\|_{\mathcal{Y},\mathcal{Y}}}.$$

When A is approximately low-rank (i. e. many eigenvalues are small), or go quickly to zero, the intrinsic dimension can be much lower than the dimensionality. Indeed,

$$1 \le \operatorname{IntDim}(A) \le \operatorname{Rank}(A) \le \dim(A).$$

Theorem A.2 (Bounded non-commutative Bernstein with intrinsic dimension [104, 155]). *Consider a sequence $(X_j)_{j=1}^D$ of D independent Hilbert-Schmidt self-adjoint random operators acting on a separable Hilbert $\mathcal{Y}$ space that satisfy $\mathbf{E}X_j = 0$ for all $j \in \mathbb{N}^*_D$. Suppose that there exist some constant $U \ge 2\|X_j\|_{\mathcal{Y},\mathcal{Y}}$ almost surely for all $j \in \mathbb{N}^*_D$. Define a semi-definite upper bound for the the operator-valued variance*

$$V \succcurlyeq \sum_{j=1}^D \mathbf{E}X_j^2.$$

Then for all $\epsilon \ge \sqrt{\|V\|_{\mathcal{Y},\mathcal{Y}}} + U/3$,

$$\mathbf{Pr}\left\{\left\|\sum_{j=1}^D X_j\right\|_{\mathcal{Y},\mathcal{Y}} \ge \epsilon\right\} \le 4 IntDim(V) \exp\left(-\psi_{V,U}(\epsilon)\right)$$

where $\psi_{V,U}(\epsilon) = \frac{\epsilon^2}{2\|V\|_{\mathcal{Y},\mathcal{Y}} + 2U\epsilon/3}$

Essentially, compared to theorem A.1, theorem A.2 replace the dimension of $\mathcal{Y}$ by four times the intrinsic dimension of the variance of the matrix valued random variable. The concentration inequality is restricted to the case where $\epsilon \ge \sqrt{\|V\|_{\mathcal{Y},\mathcal{Y}}} + U/3$ since the probability is vacuous on the contrary. The assumption that X_j’s are Hilbert-Schmidt operators comes from the fact that the product of two such operator yields a trace-class operator, for which the intrinsic dimension is well defined.

However, to cover the general case including unbounded ORFFs like curl and divergence-free ORFFs, we choose a version of Bernstein matrix concentration inequality proposed in [77] that allows to consider matrices that are not uniformly bounded but have subexponential tails. In the following we use the notion of Orlicz norm to bound random variable by their tail behavior rather than their value.

Definition A.2 (Orlicz norm). *We follow the definition given by Koltchinskii et al. [77]. Let $\psi : \mathbb{R}_+ \to \mathbb{R}_+$ be a non-decreasing convex function with $\psi(0) = 0$. For a random variable X on a measured space $(\Omega, \mathcal{T}(\Omega), \mu)$*

$$\|X\|_{\psi} := \inf \left\{ C > 0 \;\middle|\; \mathbf{E}[\psi\left(|X|/C\right)] \leq 1 \right\}.$$

For the sake of simplicity, we now fix $\psi(t) = \psi_1(t) = \exp(t) - 1$. Although the Orlicz norm should be adapted to the tail of the distribution of the random operator we want to quantify to obtain the sharpest bounds. We also introduce two technical lemmas related to Orlicz norm. The first one relates the ψ_1-Orlicz norm to the moment generating function (MGF).

Lemma A.3 *Let X be a random variable with a strictly monotonic moment-generating function. We have $\|X\|_{\psi_1}^{-1} = MGF_{|X|}^{-1}(2)$.*

Proof *We have*

$$\begin{aligned}\|X\|_{\psi_1} &= \inf \left\{ C > 0 \;\middle|\; \mathbf{E}[\exp\left(|X|/C\right)] \leq 2 \right\} \\ &= \frac{1}{\sup \left\{ C > 0 \;\middle|\; MGF_{|X|}(C) \leq 2 \right\}}\end{aligned}$$

X has strictly monotonic moment-generating thus $C^{-1} = MGF_{|X|}^{-1}(2)$. Hence

$$\|X\|_{\psi_1}^{-1} = MGF_{|X|}^{-1}(2).$$

The second lemma gives the Orlicz norm of a positive constant.

Lemma A.4 *If $a \in \mathbb{R}_+$ then $\|a\|_{\psi_1} = \frac{a}{\ln(2)} < 2a$.*

Proof *We consider a as a positive constant random variable, whose Moment Generating Function (MGF) is*

$$MGF_a(t) = \exp(at).$$

From lemma A.3, $\|a\|_{\psi_1} = \frac{1}{MGF_X^{-1}(2)}$. Then $MGF_{|a|}^{-1}(2) = \frac{\ln(2)}{|a|}$, $a \neq 0$. If $a = 0$ then $\|a\|_{\psi_1} = 0$ by definition of a norm. Thus $\|a\|_{\psi_1} = \frac{a}{\ln(2)}$. □

We now turn our attention to Minsker [104]'s theorem to for unbounded random variables.

Theorem A.3 (Unbounded non-commutative Bernstein with intrinsic dimension). *Consider a sequence $(X_j)_{j=1}^D$ of D independent self-adjoint random operators acting on a finite dimensional Hilbert space $\mathcal{Y}$ of dimension p that satisfy $\mathbf{E}X_j = 0$ for all $j \in \mathbb{N}_D^*$. Suppose that there exist some constant*

$U \geq \left\| \|X_j\|_{\mathcal{Y},\mathcal{Y}} \right\|_{\psi}$ *for all* $j \in \mathbb{N}^*_D$. *Define a semi-definite upper bound for the the operator-valued variance*

$$V \succcurlyeq \sum_{j=1}^{D} \mathbf{E} X_j^2.$$

Then for all $\epsilon > 0$,

$$\begin{aligned}
&\mathbf{Pr}\left\{ \left\| \sum_{j=1}^{D} X_j \right\|_{\mathcal{Y},\mathcal{Y}} \geq \epsilon \right\} \\
&\leq \begin{cases} 2\mathit{IntDim}(V) \exp\left(-\frac{\epsilon^2}{2\|V\|_{\mathcal{Y},\mathcal{Y}}\left(1+\frac{1}{p}\right)} \right) r_V(\epsilon), & \epsilon \leq \frac{\|V\|_{\mathcal{Y},\mathcal{Y}}}{2U} \frac{1+1/p}{K(V,p)} \\ 2\mathit{IntDim}(V) \exp\left(-\frac{\epsilon}{4UK(V,p)} \right) r_V(\epsilon), & \textit{otherwise.} \end{cases}
\end{aligned}$$

where $K(V,p) = \log\left(16\sqrt{2}p\right) + \log\left(\frac{DU^2}{\|V\|_{\mathcal{Y},\mathcal{Y}}}\right)$ *and* $r_V(\epsilon) = 1 + \frac{3}{\epsilon^2 \log^2(1+\epsilon/\|V\|_{\mathcal{Y},\mathcal{Y}})}$

Let $\psi = \psi_1$. To use theorem A.3, we set $X_j = F^j(\delta_i)$. We have indeed $\mathbf{E}_{\widehat{\mathbf{Haar}},\rho}[F^j(\delta_i)] = 0$ since $\tilde{K}(\delta_i)$ is the Monte-Carlo approximation of $K_e(\delta_i)$ and the matrices $F^j(\delta_i)$ are self-adjoint. We assume we can bound all the Orlicz norms of the $F^j(\delta_i) = \frac{1}{D}(\tilde{K}^j(\delta_i) - K_e(\delta_i))$. In the following we use constants u_i such that $u_i = DU$. Using lemma A.4 and the sub-additivity of the $\|\cdot\|_{\mathcal{Y},\mathcal{Y}}$ and $\|\cdot\|_{\psi_1}$ norm,

$$\begin{aligned}
u_i &= 2D \max_{1\leq j\leq D} \left\| \|F^j(\delta_i)\|_{\mathcal{Y},\mathcal{Y}} \right\|_{\psi_1} \\
&\leq 2 \max_{1\leq j\leq D} \left\| \|\tilde{K}^j(\delta_i)\|_{\mathcal{Y},\mathcal{Y}} \right\|_{\psi_1} + 2\left\| \|K_e(\delta_i)\|_{\mathcal{Y},\mathcal{Y}} \right\|_{\psi_1} \\
&< 4 \max_{1\leq j\leq D} \left\| \|A(\omega_j)\|_{\mathcal{Y},\mathcal{Y}} \right\|_{\psi_1} + 4\|K_e(\delta_i)\|_{\mathcal{Y},\mathcal{Y}} \\
&= 4\left(\left\| \|A(\omega)\|_{\mathcal{Y},\mathcal{Y}} \right\|_{\psi_1} + \|K_e(\delta_i)\|_{\mathcal{Y},\mathcal{Y}} \right)
\end{aligned}$$

In the same way we defined the constants $v_i = DV$,

$$\begin{aligned}
v_i &= D \sum_{j=1}^{D} \mathbf{E}_{\widehat{\mathbf{Haar}},\rho} F^j(\delta_i)^2 \\
&= D\mathbf{Var}_{\widehat{\mathbf{Haar}},\rho}\left[\tilde{K}(\delta_i)\right]
\end{aligned}$$

Then applying theorem A.3, we get for all $i \in \mathbb{N}^*_{\mathcal{N}(\mathcal{D}_C,r)}$ (i is the index of each anchor)

$$\begin{aligned}
&\mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho}\left\{ (\omega_j)_{j=1}^D \;\middle|\; \|F(\delta_i)\|_{\mathcal{Y},\mathcal{Y}} \geq \epsilon \right\} \\
&\leq \begin{cases} 4\mathrm{IntDim}(v_i) \exp\left(-D\frac{\epsilon^2}{2\|v_i\|_{\mathcal{Y},\mathcal{Y}}\left(1+\frac{1}{p}\right)} \right) r_{v_i/D}(\epsilon), & \epsilon \leq \frac{\|v_i\|_{\mathcal{Y},\mathcal{Y}}}{2u_i} \frac{1+1/p}{K(v_i,p)} \\ 4\mathrm{IntDim}(v_i) \exp\left(-D\frac{\epsilon}{4u_iK(v_i,p)} \right) r_{v_i/D}(\epsilon), & \text{otherwise.} \end{cases}
\end{aligned}$$

with

$$K(v_i, p) = \log\left(16\sqrt{2}p\right) + \log\left(\frac{u_i^2}{\|v_i\|_{\mathcal{Y},\mathcal{Y}}}\right)$$

and

$$r_{v_i/D} = 1 + \frac{3}{\epsilon^2 \log^2(1 + D\epsilon/\|v_i\|_{\mathcal{Y},\mathcal{Y}})}.$$

To unify the bound on each anchor we define two constant

$$u = 4\left(\left\|\|A(\omega)\|_{\mathcal{Y},\mathcal{Y}}\right\|_{\psi_1} + \sup_{\delta\in\mathcal{D}_{\mathcal{C}}} \|K_e(\delta)\|_{\mathcal{Y},\mathcal{Y}}\right) \geq \max_{i=1,\dots T} u_i$$

and

$$v = \sup_{\delta\in\mathcal{D}_{\mathcal{C}}} D\mathbf{Var}_{\widehat{\mathbf{Haar}},\rho}\left[\tilde{K}_e(\delta)\right] \geq \max_{i=1,\dots T} v_i.$$

A.1.4 *Union Bound and examples*

Taking the union bound over the anchors yields

$$\begin{aligned}
&\mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho}\left\{ (\omega_j)_{j=1}^D \,\middle|\, \bigcup_{i=1}^{\mathcal{N}(\mathcal{D}_{\mathcal{C}},r)} \|F(\delta_i)\|_{\mathcal{Y},\mathcal{Y}} \geq \epsilon \right\} \\
&\quad\leq 4\mathcal{N}(\mathcal{D}_{\mathcal{C}}, r) r_{v/D}(\epsilon)\operatorname{IntDim}(v) \\
&\quad\begin{cases} \exp\left(-D\frac{\epsilon^2}{2\|v\|_{\mathcal{Y},\mathcal{Y}}\left(1+\frac{1}{p}\right)}\right), & \epsilon \leq \frac{\|v\|_{\mathcal{Y},\mathcal{Y}}}{2u}\frac{1+1/p}{K(v,p)} \\ \exp\left(-D\frac{\epsilon}{4uK(v,p)}\right), & \text{otherwise.} \end{cases}
\end{aligned} \tag{A.5}$$

Hence combining equation A.4 and equation A.5 gives and summing up the hypothesis yields the following proposition

Proposition A.1 *Let $K : \mathcal{X} \times \mathcal{X} \to \mathcal{L}(\mathcal{Y})$ be a shift-invariant $\mathcal{Y}$-Mercer kernel, where $\mathcal{Y}$ is a finite dimensional Hilbert space of dimension p and $\mathcal{X}$ a metric space. Moreover, let $\mathcal{C}$ be a compact subset of $\mathcal{X}$, $A : \widehat{\mathcal{X}} \to \mathcal{L}(\mathcal{Y})$ and $\mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho}$ a pair such that*

$$\tilde{K}_e = \sum_{j=1}^D \cos(\cdot, \omega_j)A(\omega_j) \approx K_e, \quad \omega_j \sim \mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho} \textit{ i. i. d..}$$

Let

$$V(\delta) \succcurlyeq \mathbf{Var}_{\widehat{\mathbf{Haar}},\rho}\tilde{K}_e(\delta), \quad \textit{for all } \delta \in \mathcal{D}_{\mathcal{C}}$$

and H_ω be the Lipschitz constant of the function $h : x \mapsto (x, \omega)$. If the three following constant exists

$$m \geq \int_{\widehat{\mathcal{X}}} H_\omega \|A(\omega)\|_{\mathcal{Y},\mathcal{Y}} d\mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho} < \infty$$

and

$$u \geq 4\left(\left\|\|A(\omega)\|_{\mathcal{Y},\mathcal{Y}}\right\|_{\psi_1} + \sup_{\delta\in\mathcal{D}_{\mathcal{C}}}\|K_e(\delta)\|_{\mathcal{Y},\mathcal{Y}}\right) < \infty$$

and

$$v \geq \sup_{\delta\in\mathcal{D}_{\mathcal{C}}} D\|V(\delta)\|_{\mathcal{Y},\mathcal{Y}} < \infty.$$

Define $p_{int} \geq \sup_{\delta\in\mathcal{D}_{\mathcal{C}}} IntDim(V(\delta))$ *then for all* $r \in \mathbb{R}_+^*$ *and all* $\epsilon \in \mathbb{R}_+^*$,

$$\begin{aligned}
&\mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho}\left\{ (\omega_j)_{j=1}^D \,\middle|\, \|\tilde{K} - K\|_{\mathcal{C}\times\mathcal{C}} \geq \epsilon \right\} \\
&\leq 4\left(\frac{rm}{\epsilon} + p_{int}\mathcal{N}(\mathcal{D}_{\mathcal{C}}, r) r_{v/D}(\epsilon)\right. \\
&\left.\begin{cases} \exp\left(-D\frac{\epsilon^2}{8v\left(1+\frac{1}{p}\right)}\right), & \epsilon \leq \frac{v}{u}\frac{1+1/p}{K(v,p)} \\ \exp\left(-D\frac{\epsilon}{8uK(v,p)}\right), & \text{otherwise.}\end{cases}\right)
\end{aligned}$$

where

$$K(v,p) = \log\left(16\sqrt{2}p\right) + \log\left(\frac{u^2}{\|v\|_{\mathcal{Y},\mathcal{Y}}}\right)$$

and

$$r_{v/D}(\epsilon) = 1 + \frac{3}{\epsilon^2\log^2(1 + D\epsilon/\|v\|_{\mathcal{Y},\mathcal{Y}})}.$$

Proof *Let* $m = \int_{\widehat{\mathcal{X}}} H_\omega \|A(\omega)\|_{\mathcal{Y},\mathcal{Y}} d\mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho}$. *From lemma A.2,*

$$\mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho}\left\{ (\omega_j)_{j=1}^D \,\middle|\, L_F \geq \frac{\epsilon}{2r} \right\} \leq \frac{4rm}{\epsilon}.$$

Thus from lemma A.1, for all $r \in \mathbb{R}_+^*$,

$$\begin{aligned}
&\mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho}\left\{ (\omega_j)_{j=1}^D \,\middle|\, \sup_{\delta\in\mathcal{D}_{\mathcal{C}}}\|F(\delta)\|_{\mathcal{Y},\mathcal{Y}} \geq \epsilon \right\} \\
&\leq \mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho}\left\{ (\omega_j)_{j=1}^D \,\middle|\, L_F \geq \frac{\epsilon}{2r} \right\} \\
&\quad + \mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho}\left\{ (\omega_j)_{j=1}^D \,\middle|\, \bigcup_{i=1}^{\mathcal{N}(\mathcal{D}_{\mathcal{C}},r)} \|F(\delta_i)\|_{\mathcal{Y},\mathcal{Y}} \geq \epsilon \right\} \\
&= 4\frac{rm}{\epsilon} + 4\mathcal{N}(\mathcal{D}_{\mathcal{C}}, r) r_{v/D}(\epsilon) IntDim(v) \\
&\quad \begin{cases} \exp\left(-D\frac{\epsilon^2}{8\|v\|_{\mathcal{Y},\mathcal{Y}}\left(1+\frac{1}{p}\right)}\right), & \epsilon \leq \frac{\|v\|_{\mathcal{Y},\mathcal{Y}}}{u}\frac{1+1/p}{K(v,p)} \\ \exp\left(-D\frac{\epsilon}{8uK(v,p)}\right), & \text{otherwise.}\end{cases}
\end{aligned}$$

With slight modification we can obtain a second inequality for the case where the random operators $A(\omega_j)$ are bounded almost surely. This second bound with more restrictions on A has the advantage of working in infinite dimension as long as $A(\omega_j)$ is a Hilbert-Schmidt operator.

Proposition A.2 *Let $K : \mathcal{X} \times \mathcal{X} \to \mathcal{L}(\mathcal{Y})$ be a shift-invariant $\mathcal{Y}$-Mercer kernel, where $\mathcal{Y}$ is a Hilbert space and $\mathcal{X}$ a metric space. Moreover, let $\mathcal{C}$ be a compact subset of $\mathcal{X}$, $A : \widehat{\mathcal{X}} \to \mathcal{L}(\mathcal{Y})$ and $\mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho}$ a pair such that*

$$\tilde{K}_e = \sum_{j=1}^{D} \cos(\cdot, \omega_j) A(\omega_j) \approx K_e, \quad \omega_j \sim \mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho} \text{ i. i. d..}$$

where $A(\omega_j)$ is a Hilbert-Schmidt operator for all $j \in \mathbb{N}^*_D$. Let $\mathcal{D}_{\mathcal{C}} = \mathcal{C} \star \mathcal{C}^{-1}$ and

$$V(\delta) \succcurlyeq \mathbf{Var}_{\widehat{\mathbf{Haar}},\rho} \tilde{K}_e(\delta), \quad \text{for all } \delta \in \mathcal{D}_{\mathcal{C}}$$

and H_ω be the Lipschitz constant of the function $h : x \mapsto (x, \omega)$. If the three following constant exists

$$m \geq \int_{\widehat{\mathcal{X}}} H_\omega \|A(\omega)\|_{\mathcal{Y},\mathcal{Y}} d\mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho} < \infty$$

and

$$u \geq \operatorname*{ess\,sup}_{\omega \in \widehat{\mathcal{X}}} \|A(\omega)\|_{\mathcal{Y},\mathcal{Y}} + \sup_{\delta \in \mathcal{D}_{\mathcal{C}}} \|K_e(\delta)\|_{\mathcal{Y},\mathcal{Y}} < \infty$$

and

$$v \geq \sup_{\delta \in \mathcal{D}_{\mathcal{C}}} D \|V(\delta)\|_{\mathcal{Y},\mathcal{Y}} < \infty.$$

define $p_{int} \geq \sup_{\delta \in \mathcal{D}_{\mathcal{C}}} \mathit{IntDim}(V(\delta))$ then for all $r \in \mathbb{R}^*_+$ and all $\epsilon > \sqrt{\frac{v}{D}} + \frac{1}{3D}u$,

$$\begin{aligned} &\mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho} \left\{ (\omega_j)_{j=1}^D \;\middle|\; \sup_{\delta \in \mathcal{D}_{\mathcal{C}}} \|F(\delta)\|_{\mathcal{Y},\mathcal{Y}} \geq \epsilon \right\} \\ &\quad \leq 4 \left(\frac{rm}{\epsilon} + p_{int} \mathcal{N}(\mathcal{D}_{\mathcal{C}}, r) \exp\left(-D\psi_{v,u}(\epsilon)\right) \right) \end{aligned}$$

where $\psi_{v,u}(\epsilon) = \frac{\epsilon^2}{2(v + u\epsilon/3)}$.

When the covering number $\mathcal{N}(\mathcal{D}_{\mathcal{C}}, r)$ of the metric space $\mathcal{D}_{\mathcal{C}}$ has an analytical form, it is possible to optimize the bound over the radius r of the covering balls. As an example, we refine proposition A.1 and proposition A.2 in the case where $\mathcal{C}$ is a finite dimensional Banach space.

Corollary A.1 *Let $K : \mathcal{X} \times \mathcal{X} \to \mathcal{L}(\mathcal{Y})$ be a shift-invariant $\mathcal{Y}$-Mercer kernel, where $\mathcal{Y}$ is a finite dimensional Hilbert space of dimension p and $\mathcal{X}$ a finite dimensional Banach space of dimension d. Moreover, let $\mathcal{C}$ be a closed ball of $\mathcal{X}$*

centered at the origin of diameter $|\mathcal{C}|$, $A : \widehat{\mathcal{X}} \to \mathcal{L}(\mathcal{Y})$ *and* $\mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho}$ *a pair such that*

$$\tilde{K}_e = \sum_{j=1}^{D} \cos(\cdot, \omega_j) A(\omega_j) \approx K_e, \quad \omega_j \sim \mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho} \text{ i. i. d..}$$

Let $\mathcal{D}_{\mathcal{C}} = \mathcal{C} \star \mathcal{C}^{-1}$ *and*

$$V(\delta) \succcurlyeq \mathbf{Var}_{\widehat{\mathbf{Haar}},\rho} \tilde{K}_e(\delta), \quad \text{for all } \delta \in \mathcal{D}_{\mathcal{C}}$$

Let H_ω *be the Lipschitz constant of* $h_\omega : x \mapsto (x, \omega)$*. If the three following constant exists*

$$m \geq \int_{\widehat{\mathcal{X}}} H_\omega \|A(\omega)\|_{\mathcal{Y},\mathcal{Y}} d\mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho} < \infty$$

and

$$u \geq 4 \left(\left\| \|A(\omega)\|_{\mathcal{Y},\mathcal{Y}} \right\|_{\psi_1} + \sup_{\delta \in \mathcal{D}_{\mathcal{C}}} \|K_e(\delta)\|_{\mathcal{Y},\mathcal{Y}} \right) < \infty$$

and

$$v \geq \sup_{\delta \in \mathcal{D}_{\mathcal{C}}} D \|V(\delta)\|_{\mathcal{Y},\mathcal{Y}} < \infty.$$

Define $p_{int} \geq \sup_{\delta \in \mathcal{D}_{\mathcal{C}}} \mathit{IntDim}(V(\delta))$*, then for all* $0 < \epsilon \leq m|C|$,

$$\mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho} \left\{ (\omega_j)_{j=1}^D \;\middle|\; \|\tilde{K} - K\|_{\mathcal{C} \times \mathcal{C}} \geq \epsilon \right\}$$

$$\leq 8\sqrt{2} \left(\frac{m|\mathcal{C}|}{\epsilon} \right) \left(p_{int} r_{v/D}(\epsilon) \right)^{\frac{1}{d+1}} \begin{cases} \exp\left(-D \frac{\epsilon^2}{8v(d+1)\left(1+\frac{1}{p}\right)} \right), & \epsilon \leq \frac{v}{u} \frac{1+1/p}{K(v,p)} \\ \exp\left(-D \frac{\epsilon}{8u(d+1)K(v,p)} \right), & \text{otherwise,} \end{cases}$$

where $K(v,p) = \log\left(16\sqrt{2}p\right) + \log\left(\frac{u^2}{v}\right)$ *and* $r_{v/D}(\epsilon) = 1 + \frac{3}{\epsilon^2 \log^2(1 + D\epsilon/v)}$.

Proof *As we have seen in appendix A.1.1, suppose that* $\mathcal{X}$ *is a finite dimensional Banach space. Let* $\mathcal{C} \subset \mathcal{X}$ *be a closed ball centered at the origin of diameter* $|\mathcal{C}| = C$ *then the difference ball centered at the origin*

$$\begin{aligned} \mathcal{D}_{\mathcal{C}} &= \mathcal{C} \star \mathcal{C}^{-1} \\ &= \left\{ x \star z^{-1} \;\middle|\; \|x\|_{\mathcal{X}} \leq C/2, \|z\|_{\mathcal{X}} \leq C/2, (x, z) \in \mathcal{X}^2 \right\} \subset \mathcal{X} \end{aligned}$$

is closed and bounded, so compact and has diameter $|\mathcal{C}| = 2C$*. It is possible to cover it with*

$$\mathcal{N}(\mathcal{D}_{\mathcal{C}}, r) = \left(\frac{2|\mathcal{C}|}{r} \right)^d$$

closed balls of radius r. Pluging back into equation A.5 yields

$$\mathbf{Pr}_{\widehat{\mathbf{Haar},\rho}}\left\{ (\omega_j)_{j=1}^D \;\middle|\; \|\tilde{K}-K\|_{\mathcal{C}\times\mathcal{C}} \geq \epsilon \right\}$$
$$\leq 4\left(\frac{rm}{\epsilon} + p_{int}\left(\frac{2|C|}{r}\right)^d r_{v/D}(\epsilon) \begin{cases} \exp\left(-D\frac{\epsilon^2}{8v\left(1+\frac{1}{p}\right)}\right), & \epsilon \leq \frac{v}{u}\frac{1+1/p}{K(v,p)} \\ \exp\left(-D\frac{\epsilon}{8uK(v,p)}\right), & \textit{otherwise.} \end{cases}\right)$$

The right hand side of the equation has the form $ar + br^{-d}$ *with*

$$a = \frac{m}{\epsilon}$$

and

$$b = p_{int}(2|\mathcal{C}|)^d r_{v/D}(\epsilon) \begin{cases} \exp\left(-D\frac{\epsilon^2}{8v\left(1+\frac{1}{p}\right)}\right), & \epsilon \leq \frac{v}{u}\frac{1+1/p}{K(v,p)} \\ \exp\left(-D\frac{\epsilon}{8uK(v,p)}\right), & \textit{otherwise.} \end{cases}$$

Following [101, 121, 148], we optimize over r. It is a convex continuous function on $\mathbb{R}_+$ *and achieve minimum at*

$$r = \left(\frac{bd}{a}\right)^{\frac{1}{d+1}}$$

and the minimum value is

$$r_* = a^{\frac{d}{d+1}} b^{\frac{1}{d+1}} \left(d^{\frac{1}{d+1}} + d^{-\frac{d}{d+1}}\right),$$

hence

$$\mathbf{Pr}_{\widehat{\mathbf{Haar},\rho}}\left\{ (\omega_j)_{j=1}^D \;\middle|\; \|\tilde{K}-K\|_{\mathcal{C}\times\mathcal{C}} \geq \epsilon \right\}$$
$$\leq C_d\left(\frac{2m|\mathcal{C}|}{\epsilon}\right)^{\frac{d}{d+1}} \left(p_{int} r_{v/D}(\epsilon)\right)^{\frac{1}{d+1}} \begin{cases} \exp\left(-D\frac{\epsilon^2}{8v(d+1)\left(1+\frac{1}{p}\right)}\right), & \epsilon \leq \frac{v}{u}\frac{1+1/p}{K(v,p)} \\ \exp\left(-D\frac{\epsilon}{8u(d+1)K(v,p)}\right), & \textit{otherwise,} \end{cases}$$
$$\leq 8\sqrt{2}\left(\frac{m|\mathcal{C}|}{\epsilon}\right) \left(p_{int} r_{v/D}(\epsilon)\right)^{\frac{1}{d+1}} \begin{cases} \exp\left(-D\frac{\epsilon^2}{8v(d+1)\left(1+\frac{1}{p}\right)}\right), & \epsilon \leq \frac{v}{u}\frac{1+1/p}{K(v,p)} \\ \exp\left(-D\frac{\epsilon}{8u(d+1)K(v,p)}\right), & \textit{otherwise,} \end{cases}$$

where $C_d = 4\left(d^{\frac{1}{d+1}} + d^{-\frac{d}{d+1}}\right)$. *Eventually when* $\mathcal{X}$ *is a Banach space, the Lipschitz constant of* h_ω *is the supremum of the gradient*

$$H_\omega = \sup_{\delta\in\mathcal{D}_\mathcal{C}} \|(\nabla h_\omega)(\delta)\|_{\widehat{\mathcal{X}}}.$$

Following the same proof technique we obtain the second bound for bounded ORFF.

Corollary A.2 *Let $K : \mathcal{X} \times \mathcal{X} \to \mathcal{L}(\mathcal{Y})$ be a shift-invariant $\mathcal{Y}$-Mercer kernel, where $\mathcal{Y}$ is a Hilbert space and $\mathcal{X}$ a finite dimensional Banach space of dimension D. Moreover, let $\mathcal{C}$ be a closed ball of $\mathcal{X}$ centered at the origin of diameter $|\mathcal{C}|$, subset of $\mathcal{X}$, $A : \widehat{\mathcal{X}} \to \mathcal{L}(\mathcal{Y})$ and $\mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho}$ a pair such that*

$$\tilde{K}_e = \sum_{j=1}^{D} \cos\left(\cdot, \omega_j\right) A(\omega_j) \approx K_e, \quad \omega_j \sim \mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho} \text{ i. i. d..}$$

where $A(\omega_j)$ is a Hilbert-Schmidt operator for all $j \in \mathbb{N}^*_D$. Let $\mathcal{D}_{\mathcal{C}} = \mathcal{C} \star \mathcal{C}^{-1}$ and

$$V(\delta) \succcurlyeq \mathbf{Var}_{\widehat{\mathbf{Haar}},\rho} \tilde{K}_e(\delta), \quad \text{for all } \delta \in \mathcal{D}_{\mathcal{C}}$$

and H_ω be the Lipschitz constant of the function $h : x \mapsto (x, \omega)$. If the three following constant exists

$$m \geq \int_{\widehat{\mathcal{X}}} H_\omega \|A(\omega)\|_{\mathcal{Y},\mathcal{Y}} d\mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho} < \infty$$

and

$$u \geq \operatorname*{ess\,sup}_{\omega \in \widehat{\mathcal{X}}} \|A(\omega)\|_{\mathcal{Y},\mathcal{Y}} + \sup_{\delta \in \mathcal{D}_{\mathcal{C}}} \|K_e(\delta)\|_{\mathcal{Y},\mathcal{Y}} < \infty$$

and

$$v \geq \sup_{\delta \in \mathcal{D}_{\mathcal{C}}} D \|V(\delta)\|_{\mathcal{Y},\mathcal{Y}} < \infty.$$

define $p_{int} \geq \sup_{\delta \in \mathcal{D}_{\mathcal{C}}} \mathit{IntDim}\left(V(\delta)\right)$ then for all $\sqrt{\frac{v}{D}} + \frac{u}{3D} < \epsilon < m|\mathcal{C}|$,

$$\begin{aligned} &\mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho} \left\{ (\omega_j)_{j=1}^{D} \;\middle|\; \sup_{\delta \in \mathcal{D}_{\mathcal{C}}} \|F(\delta)\|_{\mathcal{Y},\mathcal{Y}} \geq \epsilon \right\} \\ &\leq 8\sqrt{2} \left(\frac{m|\mathcal{C}|}{\epsilon} \right) p_{int}^{\frac{1}{d+1}} \exp\left(-D\psi_{v,d,u}(\epsilon)\right) \end{aligned}$$

where $\psi_{v,d,u}(\epsilon) = \frac{\epsilon^2}{2(d+1)(v+u\epsilon/3)}$.

A.2 PROOF OF THE ORFF ESTIMATOR VARIANCE BOUND

We use the notations $\delta = x \star z^{-1}$ for all $x, z \in \mathcal{X}$, $\tilde{K}(x,z) = \tilde{\Phi}(x)^* \tilde{\Phi}(z)$, $\tilde{K}^j(x,z) = \Phi_x(\omega_j)^* \Phi_z(\omega_j)$ and $K_e(\delta) = K_e(x,z)$.

Proposition A.3 (Bounding the *variance* of $\tilde{K}$). *Let K be a shift invariant $\mathcal{Y}$-Mercer kernel on a second countable LCA topological space $\mathcal{X}$. Let $A : \widehat{\mathcal{X}} \to \mathcal{L}(\mathcal{Y})$ and $\mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho}$ a pair such that*

$$\tilde{K}_e = \sum_{j=1}^{D} \cos\left(\cdot, \omega_j\right) A(\omega_j) \approx K_e, \quad \omega_j \sim \mathbf{Pr}_{\widehat{\mathbf{Haar}},\rho} \text{ i. i. d..}$$

Then,

$$\mathbf{Var}_{\widehat{\mathbf{Haar}},\rho}\left[\tilde{K}_e(\delta)\right] \preccurlyeq \frac{1}{2D}\Big((K_e(2\delta)+K_e(e))\,\mathbf{E}_{\widehat{\mathbf{Haar}},\rho}[A(\omega)] - 2K_e(\delta)^2 + \mathbf{Var}_{\widehat{\mathbf{Haar}},\rho}[A(\omega)]\Big)$$

Proof *Let $\delta \in \mathcal{D}_{\mathcal{C}}$ be a constant. From the definition of the variance of a random variable and using the fact that the $(\omega_j)_{j=1}^D$ are i. i. d. random variables,*

$$\begin{aligned}\mathbf{Var}_{\widehat{\mathbf{Haar}},\rho}\left[\tilde{K}_e(\delta)\right] &= \mathbf{E}_{\widehat{\mathbf{Haar}},\rho}\left[\frac{1}{D}\sum_{j=1}^{D}\tilde{K}_e^j(\delta)-K_e(\delta)\right]^2\\ &= \frac{1}{D^2}\mathbf{E}_{\widehat{\mathbf{Haar}},\rho}\left[\sum_{j=1}^{D}\tilde{K}_e^j(\delta)-K_e(\delta)\right]^2\\ &= \frac{1}{D}\mathbf{E}_{\widehat{\mathbf{Haar}},\rho}\left[\tilde{K}_e^j(\delta)^2-\tilde{K}_e^j(\delta)K_e(\delta)-K_e(\delta)\tilde{K}_e^j(\delta)+K_e(\delta)^2\right]\end{aligned}$$

From the definition of $\tilde{K}_e^j$, $\mathbf{E}_{\widehat{\mathbf{Haar}},\rho}\tilde{K}_e^j(\delta) = K_e(\delta)$, which leads to

$$\mathbf{Var}_{\widehat{\mathbf{Haar}},\rho}\left[\tilde{K}_e(\delta)\right] = \frac{1}{D}\mathbf{E}_{\widehat{\mathbf{Haar}},\rho}\left[\tilde{K}_e^j(\delta)^2-K_e(\delta)^2\right]$$

A trigonometric identity gives us $(\cos(\delta,\omega))^2 = \frac{1}{2}\left(\cos(2\delta,\omega)+\cos(e,\omega)\right)$. *Thus*

$$\mathbf{Var}_{\widehat{\mathbf{Haar}},\rho}\left[\tilde{K}_e(\delta)\right] = \frac{1}{2D}\mathbf{E}_{\widehat{\mathbf{Haar}},\rho}\left[(\cos(2\delta,\omega)+\cos(e,\omega))\,A(\omega)^2-2K_e(\delta)^2\right].$$

Also,

$$\begin{aligned}\mathbf{E}_{\widehat{\mathbf{Haar}},\rho}\left[\cos(2\delta,\omega)A(\omega)^2\right] &= \mathbf{E}_{\widehat{\mathbf{Haar}},\rho}[\cos(2\delta,\omega)A(\omega)]\,\mathbf{E}_{\widehat{\mathbf{Haar}},\rho}[A(\omega)]\\ &\quad+\mathbf{Cov}_{\widehat{\mathbf{Haar}},\rho}[\cos(2\delta,\omega)A(\omega),A(\omega)]\\ &= K_e(2\delta)\mathbf{E}_{\widehat{\mathbf{Haar}},\rho}[A(\omega)]\\ &\quad+\mathbf{Cov}_{\widehat{\mathbf{Haar}},\rho}[\cos(2\delta,\omega)A(\omega),A(\omega)]\end{aligned}$$

Similarly we obtain

$$\begin{aligned}\mathbf{E}_{\widehat{\mathbf{Haar}},\rho}\left[\cos(e,\omega)A(\omega)^2\right] &= K_e(e)\mathbf{E}_{\widehat{\mathbf{Haar}},\rho}[A(\omega)]\\ &\quad+\mathbf{Cov}_{\widehat{\mathbf{Haar}},\rho}[\cos(e,\omega)A(\omega),A(\omega)]\end{aligned}$$

Therefore

$$\begin{aligned}\mathbf{Var}_{\widehat{\mathbf{Haar}},\rho}\left[\tilde{K}_e(\delta)\right] &= \frac{1}{2D}\Big((K_e(2\delta)+K_e(e))\,\mathbf{E}_{\widehat{\mathbf{Haar}},\rho}[A(\omega)]-2K_e(\delta)^2\\ &\quad+\mathbf{Cov}_{\widehat{\mathbf{Haar}},\rho}[(\cos(2\delta,\omega)+\cos(e,\omega))\,A(\omega),A(\omega)]\Big)\\ &= \frac{1}{2D}\Big((K_e(2\delta)+K_e(e))\,\mathbf{E}_{\widehat{\mathbf{Haar}},\rho}[A(\omega)]-2K_e(\delta)^2\\ &\quad+\mathbf{Cov}_{\widehat{\mathbf{Haar}},\rho}\left[(\cos(\delta,\omega))^2A(\omega),A(\omega)\right]\Big)\\ &\preccurlyeq \frac{1}{2D}\Big((K_e(2\delta)+K_e(e))\,\mathbf{E}_{\widehat{\mathbf{Haar}},\rho}[A(\omega)]-2K_e(\delta)^2\\ &\quad+\mathbf{Var}_{\widehat{\mathbf{Haar}},\rho}[A(\omega)]\Big)\end{aligned}$$

B

MISCELLANEOUS

Contents

B.1 OPERATOR-VALUED FUNCTIONS AND INTEGRATION

B.2 ABOUT CONCENTRATION INEQUALITIES

B.3 ORFF ENGINEERING

B.4 LEARNING WITH SEMI-SUPERVISION

We present here an extension of chapter 6 to semi-supervised learning with ORFF in the framework of Minh, Bazzani, and Murino [103]. This framework includes Vector-valued Manifold Regularization [16, 29, 102] and Co-regularized Multi-view Learning [26, 125, 140, 147].

B.4.1 *Representer theorem and feature equivalence*

We suppose that we are given a training sample $\mathbf{u} = (x_i)_{i=N}^{N+U} \in \mathcal{X}^U$ of unlabeled exemples. We note $\mathbf{z} \in (\mathcal{X} \times \mathcal{Y})^N \times \mathcal{X}^U$ the sequence $\mathbf{z} = \mathbf{su}$ concatenating both labeled ($\mathbf{s}$) and unlabeled ($\mathbf{u}$) training examples.

Theorem B.1 (Representer theorem, Minh, Bazzani, and Murino [103]). *Let K be a $\mathcal{U}$-Mercer Operator-Valued Kernel and $\mathcal{H}_K$ its corresponding $\mathcal{U}$-Reproducing Kernel Hilbert space.*

Let $V : \mathcal{U} \to \mathcal{Y}$ be a bounded linear operator and let $c : \mathcal{Y} \times \mathcal{Y} \to \overline{\mathbb{R}}$ be a cost function such that $L(x, f, y) = c(Vf(x), y)$ is a proper convex lower semi-continuous function in f for all $x \in \mathcal{X}$ and all $y \in \mathcal{Y}$.

Eventually let $\lambda_K \in \mathbb{R}_{>0}$ and $\lambda_M \in \mathbb{R}_+$ be two regularization hyperparameters and $(M_{ik})_{i,k=1}^{N+U}$ be a sequence of data dependent bounded linear operators in $\mathcal{L}(\mathcal{U})$, such that

$$\sum_{i,j=1}^{N+U} \langle u_i, M_{ik} u_k \rangle \geq 0, \quad \forall (u_i)_{i=1}^{N+U} \in \mathcal{U}^{N+U} \text{ and } M_{ik} = M_{ki}^*.$$

The solution $f_{\mathbf{z}} \in \mathcal{H}_K$ of the regularized optimization problem

$$\begin{aligned} f_{\mathbf{z}} = \operatorname*{arg\,min}_{f \in \mathcal{H}_K} & \frac{1}{N} \sum_{i=1}^{N} c(Vf(x_i), y_i) + \frac{\lambda_K}{2} \|f\|_K^2 \\ & + \frac{\lambda_M}{2} \sum_{i,k=1}^{N+U} \langle f(x_i), M_{ik} f(x_k) \rangle_{\mathcal{U}} \end{aligned} \tag{B.1}$$

has the form $f_{\mathbf{z}} = \sum_{j=1}^{N+U} K(\cdot, x_j) u_{\mathbf{z},j}$ where $u_{\mathbf{z},j} \in \mathcal{U}$ and

$$\begin{aligned} u_{\mathbf{z}} = \operatorname*{arg\,min}_{u \in \bigoplus_{i=1}^{N+U} \mathcal{U}} & \frac{1}{N} \sum_{i=1}^{N} c\left(V \sum_{k=1}^{N+U} K(x_i, x_j) u_j, y_i \right) + \frac{\lambda_K}{2} \sum_{k=1}^{N+U} u_i^* K(x_i, x_k) u_k \\ & + \frac{\lambda_M}{2} \sum_{i,k=1}^{N+U} \left\langle \sum_{j=1}^{N+U} K(x_i, x_j) u_j, M_{ik} \sum_{j=1}^{N+U} K(x_k, x_j) u_j \right\rangle_{\mathcal{U}} . \end{aligned} \tag{B.2}$$

We present here the proof of the formulation proposed by Minh, Bazzani, and Murino [103]. In the mean time we clarify some elements of the proof. Indeed the existence of a global minimizer is not trivial and we must invoke the Mazur-Schauder theorem. Moreover the coercivity of the objective function required by the Mazur-Schauder theorem is not obvious when we do not require the cost function to take only positive values. However a corollary of Hahn-Banach theorem linking strong convexity to coercivity gives the solution.

Proof *Since $f(x) = K_x^* f$ (see equation 3.14), the optimization problem reads*

$$f_{\mathbf{z}} = \arg\min_{f \in \mathcal{H}_K} \frac{1}{N} \sum_{i=1}^{N} c(VK_{x_i}^* f, y_i) + \frac{\lambda_K}{2} \|f\|_K^2 + \frac{\lambda_M}{2} \sum_{i,k=1}^{N+U} \langle K_{x_i}^* f, M_{ik} K_{x_k}^* f \rangle_{\mathcal{U}}$$

Let $W_{V,\mathbf{s}} : \mathcal{H}_K \to \bigoplus_{i=1}^N \mathcal{Y}$ be the restriction linear operator defined as

$$W_{V,\mathbf{s}} f = \bigoplus_{i=1}^{N} VK_{x_i}^* f,$$

with $VK_{x_i}^* : \mathcal{H}_K \to \mathcal{Y}$ and $K_{x_i} V^* : \mathcal{Y} \to \mathcal{H}_K$. Let $\Upsilon = \bigoplus_{i=1}^N y_i \in \mathcal{Y}^N$. We have

$$\langle \Upsilon, W_{V,\mathbf{s}} f \rangle_{\bigoplus_{i=1}^N \mathcal{Y}} = \sum_{i=1}^{N} \langle y_i, VK_{x_i}^* f \rangle_{\mathcal{Y}} = \sum_{i=1}^{N} \langle K_{x_i} V^* y_i, f \rangle_{\mathcal{H}_K}.$$

Thus the adjoint operator $W_{V,\mathbf{s}}^* : \bigoplus_{i=1}^N \mathcal{Y} \to \mathcal{H}_K$ is

$$W_{V,\mathbf{s}}^* \Upsilon = \sum_{i=1}^{N} K_{x_i} V^* y_i,$$

and the operator $W_{V,\mathbf{s}}^* W_{V,\mathbf{s}} : \mathcal{H}_K \to \mathcal{H}_K$ is

$$W_{V,\mathbf{s}}^* W_{V,\mathbf{s}} f = \sum_{i=1}^{N} K_{x_i} V^* V K_{x_i}^* f$$

where $V^* V \in \mathcal{L}(\mathcal{U})$. Let

$$\mathcal{J}_{\lambda_K}(f) = \underbrace{\frac{1}{N} \sum_{i=1}^{N} c(Vf(x_i), y_i)}_{=\mathcal{J}_c} + \frac{\lambda_K}{2} \|f\|_K^2 + \underbrace{\frac{\lambda_M}{2} \sum_{i,k=1}^{N+U} \langle f(x_i), M_{ik} f(x_k) \rangle_{\mathcal{U}}}_{=\mathcal{J}_M}$$

Since c is proper, lower semi-continuous and convex by assumption, thus the term $\mathcal{J}_c$ is also proper, lower semi-continuous and convex. Moreover the term $\mathcal{J}_M$ is always *positive for any $f \in \mathcal{H}_K$ and $\frac{\lambda_K}{2}\|f\|_K^2$ is strongly convex. Thus $\mathcal{J}_{\lambda_K}$ is strongly convex. Apply lemma 6.1 to obtain the coercivity of $\mathcal{J}_{\lambda_K}$, and then theorem 6.1 to show that $\mathcal{J}_{\lambda_K}$ has a unique minimizer and is attained. Then let*

$$\mathcal{H}_{K,\mathbf{z}} = \left\{ \sum_{j=1}^{N+U} K_{x_j} u_j \;\middle|\; \forall (u_i)_{i=1}^{N+U} \in \mathcal{U}^{N+U} \right\}.$$

For $f \in \mathcal{H}_{K,\mathbf{z}}^{\perp}$, the operator $W_{V,\mathbf{s}}$ satisfies

$$\langle Y, W_{V,\mathbf{s}} f \rangle_{\bigoplus_{i=1}^{N} \mathcal{Y}} = \langle \underbrace{f}_{\in \mathcal{H}_{K,\mathbf{z}}^{\perp}}, \underbrace{\sum_{i=1}^{N+U} K_{x_i} V^* y_i}_{\in \mathcal{H}_{K,\mathbf{z}}} \rangle_{\mathcal{H}_K} = 0$$

for all sequences $(y_i)_{i=1}^N$, since $V^* y_i \in \mathcal{U}$. Hence,

$$(Vf(x_i))_{i=1}^N = 0 \tag{B.3}$$

In the same way,

$$\sum_{i=1}^{N+U} \langle K_{x_i}^* f, u_i \rangle_{\mathcal{U}} = \langle \underbrace{f}_{\in \mathcal{H}_{K,\mathbf{z}}^{\perp}}, \underbrace{\sum_{j=1}^{N+U} K_{x_j} u_j}_{\in \mathcal{H}_{K,\mathbf{z}}} \rangle_{\mathcal{H}_K} = 0.$$

for all sequences $(u_i)_{i=1}^{N+U} \in \mathcal{U}^{N+U}$. As a result,

$$(f(x_i))_{i=1}^{U+N} = 0. \tag{B.4}$$

Now for an arbitrary $f \in \mathcal{H}_{\mathcal{K}}$, consider the orthogonal decomposition $f = f^{\perp} + f^{\parallel}$, where $f^{\perp} \in \mathcal{H}_{K,\mathbf{z}}^{\perp}$ and $f^{\parallel} \in \mathcal{H}_{K,\mathbf{z}}$. Then since $\left\|f^{\perp} + f^{\parallel}\right\|_{\mathcal{H}_K}^2 = \|f^{\perp}\|_{\mathcal{H}_K}^2 + \left\|f^{\parallel}\right\|_{\mathcal{H}_K}^2$, equation B.3 and equation B.4 shows that if $\lambda_K > 0$, clearly then

$$\mathcal{J}_{\lambda_K}(f) = \mathcal{J}_{\lambda_K}\left(f^{\perp} + f^{\parallel}\right) \geq \mathcal{J}_{\lambda_K}\left(f^{\parallel}\right)$$

The last inequality holds only when $\|f^{\perp}\|_{\mathcal{H}_K} = 0$, that is when $f^{\perp} = 0$. As a result since the minimizer of $\mathcal{J}_{\lambda_K}$ is unique and attained, it must lies in $\mathcal{H}_{K,\mathbf{z}}$. □

Theorem B.2 (Feature equivalence). *Let $\widetilde{K}$ be an Operator-Valued Kernel such that for all $x, z \in \mathcal{X}$, $\widetilde{\Phi}(x)^* \widetilde{\Phi}(z) = \widetilde{K}(x, z)$ where $\widetilde{K}$ is a $\mathcal{U}$-Mercer OVK and $\mathcal{H}_{\widetilde{K}}$ its corresponding $\mathcal{U}$-Reproducing kernel Hilbert space.*

Let $V : \mathcal{U} \to \mathcal{Y}$ be a bounded linear operator and let $c : \mathcal{Y} \times \mathcal{Y} \to \overline{\mathbb{R}}$ be a cost function such that $L(x, \widetilde{f}, y) = c(V\widetilde{f}(x), y)$ is a proper convex lower semi-continuous function in $\widetilde{f} \in \mathcal{H}_{\widetilde{K}}$ for all $x \in \mathcal{X}$ and all $y \in \mathcal{Y}$.

Eventually let $\lambda_K \in \mathbb{R}_{>0}$ and $\lambda_M \in \mathbb{R}_+$ be two regularization hyperparameters and $(M_{ik})_{i,k=1}^{N+U}$ be a sequence of data dependent bounded linear operators in $\mathcal{L}(\mathcal{U})$, such that

$$\sum_{i,j=1}^{N+U} \langle u_i, M_{ik} u_k \rangle \geq 0, \quad \forall (u_i)_{i=1}^{N+U} \in \mathcal{U}^{N+U} \text{ and } M_{ik} = M_{ki}^*.$$

The solution $f_{\mathbf{z}} \in \mathcal{H}_{\widetilde{K}}$ of the regularized optimization problem

$$\begin{aligned} \widetilde{f}_{\mathbf{z}} = \operatorname*{arg\,min}_{\widetilde{f} \in \mathcal{H}_{\widetilde{K}}} \frac{1}{N} \sum_{i=1}^{N} c\left(V\widetilde{f}(x_i), y_i\right) + \frac{\lambda_K}{2} \left\|\widetilde{f}\right\|_{\widetilde{K}}^2 \\ + \frac{\lambda_M}{2} \sum_{i,k=1}^{N+U} \langle \widetilde{f}(x_i), M_{ik} \widetilde{f}(x_k) \rangle_{\mathcal{U}} \end{aligned} \tag{B.5}$$

has the form $\widetilde{f}_{\mathbf{z}} = \widetilde{\Phi}(\cdot)^* \theta_{\mathbf{z}}$, where $\theta_{\mathbf{z}} \in (Ker\ \widetilde{W})^\perp$ and

$$\begin{aligned} \theta_{\mathbf{z}} = \operatorname*{arg\,min}_{\theta \in \widetilde{\mathcal{H}}} \frac{1}{N} \sum_{i=1}^{N} c\left(V\widetilde{\Phi}(x_i)^* \theta, y_i\right) + \frac{\lambda_K}{2} \|\theta\|_{\widetilde{\mathcal{H}}}^2 \\ + \frac{\lambda_M}{2} \sum_{i,k=1}^{N+U} \langle \theta, \widetilde{\Phi}(x_i) M_{ik} \widetilde{\Phi}(x_k)^* \theta \rangle_{\widetilde{\mathcal{H}}}. \end{aligned} \tag{B.6}$$

Proof *Since $\widetilde{K}$ is an operator-valued kernel, from theorem B.1, equation B.5 has a solution of the form*

$$\begin{aligned} \widetilde{f}_{\mathbf{z}} &= \sum_{i=1}^{N+U} \widetilde{K}(\cdot, x_i) u_i, \quad u_i \in \mathcal{U}, x_i \in \mathcal{X} \\ &= \sum_{i=1}^{N} \widetilde{\Phi}(\cdot)^* \widetilde{\Phi}(x_i) u_i = \widetilde{\Phi}(\cdot)^* \underbrace{\left(\sum_{i=1}^{N+U} \widetilde{\Phi}(x_i) u_i \right)}_{=\theta \in \left(Ker\ \widetilde{W}\right)^\perp \subset \widetilde{\mathcal{H}}}. \end{aligned}$$

Let

$$\begin{aligned} \theta_{\mathbf{z}} = \operatorname*{arg\,min}_{\theta \in \left(Ker\ \widetilde{W}\right)^\perp} \frac{1}{N} \sum_{i=1}^{N} c\left(V\widetilde{\Phi}(x_i)^* \theta, y_i\right) + \frac{\lambda_K}{2} \left\|\widetilde{\Phi}(\cdot)^* \theta\right\|_{\widetilde{K}}^2 \\ + \frac{\lambda_M}{2} \sum_{i,k=1}^{N+U} \left\langle \widetilde{\Phi}(x_i)^* \theta, M_{ik} \widetilde{\Phi}(x_k)^* \theta \right\rangle_{\mathcal{U}}. \end{aligned}$$

Since $\theta \in (Ker\ \widetilde{W})^\perp$ and W is an isometry from $(Ker\ \widetilde{W})^\perp \subset \widetilde{\mathcal{H}}$ onto $\mathcal{H}_{\widetilde{K}}$, we have $\left\|\widetilde{\Phi}(\cdot)^* \theta\right\|_{\widetilde{K}}^2 = \|\theta\|_{\widetilde{\mathcal{H}}}^2$. Hence

$$\begin{aligned} \theta_{\mathbf{z}} = \operatorname*{arg\,min}_{\theta \in \left(Ker\ \widetilde{W}\right)^\perp} \frac{1}{N} \sum_{i=1}^{N} c\left(V\widetilde{\Phi}(x_i)^* \theta, y_i\right) + \frac{\lambda_K}{2} \|\theta\|_{\widetilde{\mathcal{H}}}^2 \\ + \frac{\lambda_M}{2} \sum_{i,k=1}^{N+U} \langle \widetilde{\Phi}(x_i)^* \theta, M_{ik} \widetilde{\Phi}(x_k)^* \theta \rangle_{\mathcal{U}}. \end{aligned}$$

Finding a minimizer $\theta_{\mathbf{z}}$ over $\left(Ker\ \widetilde{W}\right)^{\perp}$ is not the same as finding a minimizer over $\widetilde{\mathcal{H}}$. Although in both cases Mazur-Schauder's theorem guarantees that the respective minimizers are unique, they might not be the same. Since $\widetilde{W}$ is bounded, Ker $\widetilde{W}$ is closed, so that we can perform the decomposition $\widetilde{\mathcal{H}} = \left(Ker\ \widetilde{W}\right)^{\perp} \oplus \left(Ker\ \widetilde{W}\right)$. Then clearly by linearity of W and the fact that for all $\theta^{\parallel} \in Ker\ \widetilde{W}$, $\widetilde{W}\theta^{\parallel} = 0$, if $\lambda > 0$ we have

$$\theta_{\mathbf{z}} = \underset{\theta\in\widetilde{\mathcal{H}}}{\arg\min} \frac{1}{N}\sum_{i=1}^{N} c\left(V\widetilde{\Phi}(x_i)^*\theta, y_i\right) + \frac{\lambda_K}{2}\|\theta\|^2_{\widetilde{\mathcal{H}}} + \frac{\lambda_M}{2}\sum_{i,k=1}^{N+U}\left\langle \widetilde{\Phi}(x_i)^*\theta, M_{ik}\widetilde{\Phi}(x_k)^*\theta\right\rangle_{\mathcal{U}}$$

Thus

$$\begin{aligned}\theta_{\mathbf{z}} = &\underset{\substack{\theta^{\perp}\in\left(Ker\ \widetilde{W}\right)^{\perp},\\ \theta^{\parallel}\in Ker\ \widetilde{W}}}{\arg\min} \frac{1}{N}\sum_{i=1}^{N} c\left(V\left(\widetilde{W}\theta^{\perp}\right)(x) + \underbrace{V\left(\widetilde{W}\theta^{\parallel}\right)(x)}_{=0\ \ for\ all\ \ \theta^{\parallel}}, y_i\right)\\
&+ \frac{\lambda_K}{2}\left\|\theta^{\perp}\right\|^2_{\widetilde{\mathcal{H}}} + \underbrace{\frac{\lambda_K}{2}\left\|\theta^{\parallel}\right\|^2_{\widetilde{\mathcal{H}}}}_{=0\ \ only\ if\ \ \theta^{\parallel}=0}\\
&+\frac{\lambda_M}{2}\sum_{i,k=1}^{N+U}\left\langle \widetilde{\Phi}(x_i)^*\theta^{\perp}, M_{ik}\left(\widetilde{W}\theta^{\perp}\right)(x_k)\right\rangle_{\mathcal{U}}\\
&+\frac{\lambda_M}{2}\sum_{i,k=1}^{N+U}\left\langle \underbrace{\left(\widetilde{W}\theta^{\parallel}\right)(x_i)}_{=0\ \ for\ all\ \ \theta^{\parallel}}, M_{ik}\left(\widetilde{W}\theta^{\perp}\right)(x_k)\right\rangle_{\mathcal{U}}\\
&+\frac{\lambda_M}{2}\sum_{i,k=1}^{N+U}\left\langle \left(\widetilde{W}\theta^{\perp}\right)(x_i), M_{ik}\underbrace{\left(\widetilde{W}\theta^{\parallel}\right)(x_k)}_{=0\ \ for\ all\ \ \theta^{\parallel}}\right\rangle_{\mathcal{U}}\\
&+\frac{\lambda_M}{2}\sum_{i,k=1}^{N+U}\left\langle \underbrace{\left(\widetilde{W}\theta^{\parallel}\right)(x_i)}_{=0\ \ for\ all\ \ \theta^{\parallel}}, M_{ik}\underbrace{\left(\widetilde{W}\theta^{\parallel}\right)(x_k)}_{=0\ \ for\ all\ \ \theta^{\parallel}}\right\rangle_{\mathcal{U}}.\end{aligned}$$

Thus

$$\begin{aligned}\theta_{\mathbf{z}} = &\underset{\theta^{\perp}\in\left(Ker\ \widetilde{W}\right)^{\perp}}{\arg\min} \frac{1}{N}\sum_{i=1}^{N} c\left(V\left(\widetilde{W}\theta^{\perp}\right)(x), y_i\right) + \frac{\lambda_K}{2}\left\|\theta^{\perp}\right\|^2_{\widetilde{\mathcal{H}}}\\
&+\frac{\lambda_M}{2}\sum_{i,k=1}^{N+U}\left\langle \widetilde{\Phi}(x_i)^*\theta^{\perp}, M_{ik}\left(\widetilde{W}\theta^{\perp}\right)(x_k)\right\rangle_{\mathcal{U}}.\end{aligned}$$

Hence minimizing over $\left(Ker\ \widetilde{W}\right)^{\perp}$ *or* $\widetilde{\mathcal{H}}$ *is the same when* $\lambda_K > 0$*. Eventually,*

$$\begin{aligned}
\theta_{\mathbf{z}} &= \underset{\theta\in\widetilde{\mathcal{H}}}{\arg\min}\ \frac{1}{N}\sum_{i=1}^{N} c\left(V\widetilde{\Phi}(x_i)^*\theta, y_i\right) + \frac{\lambda_K}{2}\|\theta\|^2_{\widetilde{\mathcal{H}}} \\
&\quad + \frac{\lambda_M}{2}\sum_{i,k=1}^{N+U}\left\langle \widetilde{\Phi}(x_i)^*\theta, M_{ik}\widetilde{\Phi}(x_k)^*\theta\right\rangle_{\mathcal{U}} \\
&= \underset{\theta\in\widetilde{\mathcal{H}}}{\arg\min}\ \frac{1}{N}\sum_{i=1}^{N} c\left(V\widetilde{\Phi}(x_i)^*\theta, y_i\right) + \frac{\lambda_K}{2}\|\theta\|^2_{\widetilde{\mathcal{H}}} \\
&\quad + \frac{\lambda_M}{2}\sum_{i,k=1}^{N+U}\left\langle \theta, \widetilde{\Phi}(x_i)M_{ik}\widetilde{\Phi}(x_k)^*\theta\right\rangle_{\widetilde{\mathcal{H}}}.
\end{aligned}$$

This theorem is illustrated by figure B.1. We use the classic two moons dataset[1]. We first perform an unsupervised spectral clustering step [161] and construct the matrix where C_{ik} is 1 if x_i and x_k are in the same cluster, 0 otherwise. Then we take the inverse Laplacian of this matrix and use it as the data dependent operator M. E

B.4.2 *Gradients*

By linearity and applying the chaine rule to equation B.6 and since $M_{ik}^* = M_{ki}$ for all $i, k \in \mathbb{N}^*_{N+U}$, we have

$$\begin{aligned}
\nabla_\theta c\left(V\widetilde{\Phi}(x_i)^*\theta, y_i\right) &= \widetilde{\Phi}(x_i)V^*\left(\frac{\partial}{\partial y}c\left(y, y_i\right)\bigg|_{y=V\widetilde{\Phi}(x_i)^*\theta}\right)^*, \\
\nabla_\theta\left\langle \widetilde{\Phi}(x_i)^*\theta, M_{ik}\widetilde{\Phi}(x_k)^*\theta\right\rangle_{\mathcal{U}} &= \widetilde{\Phi}(x_i)\left(M_{ik} + M_{ki}^*\right)\widetilde{\Phi}(x_k)^*\theta, \\
\nabla_\theta\|\theta\|^2_{\widetilde{\mathcal{H}}} &= 2\theta.
\end{aligned}$$

Provided that $c(y, y_i)$ is Frechet differentiable w. r. t. y, for all y and $y_i \in \mathcal{Y}$ we have $\nabla_\theta \mathcal{J}_{\lambda_K}(\theta) \in \widetilde{\mathcal{H}}$ and

$$\begin{aligned}
\nabla_\theta \mathcal{J}_{\lambda_K}(\theta) &= \frac{1}{N}\sum_{i=1}^{N}\widetilde{\Phi}(x_i)V^*\left(\frac{\partial}{\partial y}c\left(y, y_i\right)\bigg|_{y=V\widetilde{\Phi}(x_i)^*\theta}\right)^* \\
&\quad + \lambda_K\theta + \lambda_M\sum_{i,k=1}^{N+U}\widetilde{\Phi}(x_i)M_{ik}\widetilde{\Phi}(x_k)^*\theta
\end{aligned}$$

Therefore after factorization, considering $\lambda_K > 0$,

$$\begin{aligned}
\nabla_\theta \mathcal{J}_{\lambda_K}(\theta) &= \frac{1}{N}\sum_{i=1}^{N}\widetilde{\Phi}(x_i)V^*\left(\frac{\partial}{\partial y}c\left(y, y_i\right)\bigg|_{y=V\widetilde{\Phi}(x_i)^*\theta}\right)^* \\
&\quad + \lambda_K\left(I_{\widetilde{\mathcal{H}}} + \frac{\lambda_M}{\lambda_K}\sum_{i,k=1}^{N+U}\widetilde{\Phi}(x_i)M_{ik}\widetilde{\Phi}(x_k)^*\right)\theta
\end{aligned}$$

1 Available at http://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_moons.html.

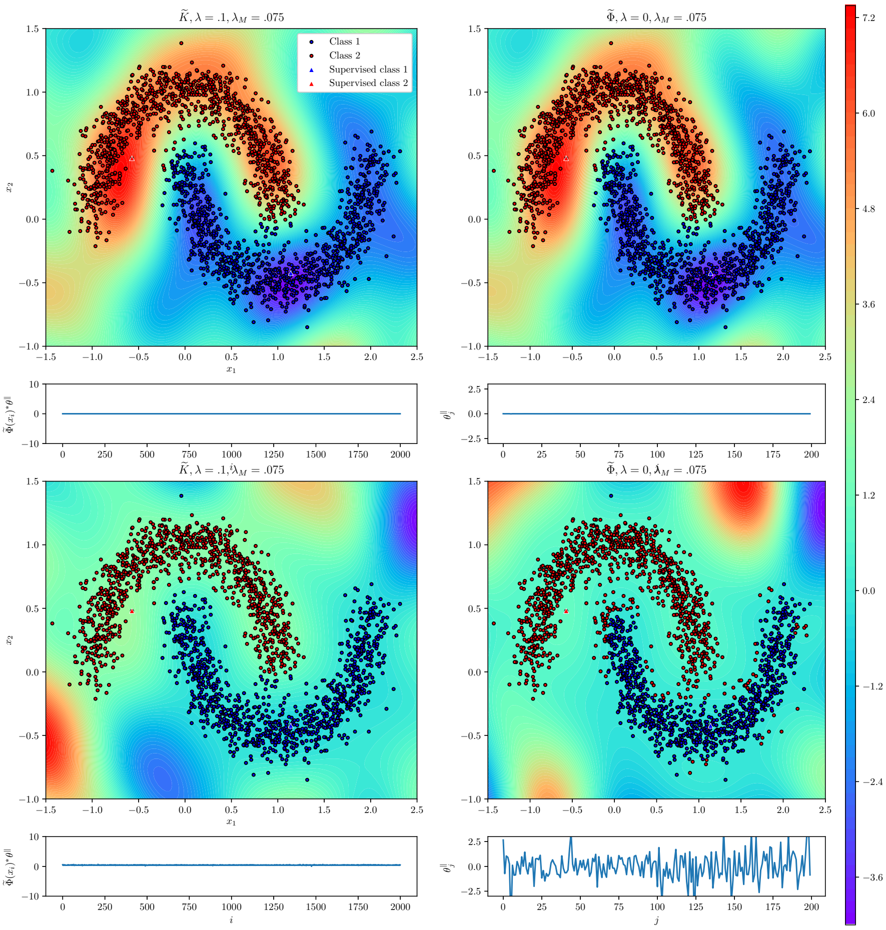

Figure B.1: ORFF equivalence theorem (semi-supervised). Each row compares the scalar ORFF $\widetilde{\Phi}$ method constructed from a Gaussian with the kernel method where $\widetilde{K} = \widetilde{\Phi}^\top \widetilde{\Phi}$. The top row corresponds to the case $\lambda_K = 0.1$ and $\lambda_M = 0.075$. Since $\lambda_K > 0$, the solution with $\widetilde{K}$ and $\widetilde{\Phi}$ are exactly the same (theorem B.2 applies) and we see that $\theta^{\parallel} = 0$. The bottom row corresponds to the case $\lambda_K = 0$ and $\lambda_M = 0.075$. Here the solution with $\widetilde{K}$ and $\widetilde{\Phi}$ doesn't match (theorem B.2 fails to apply since $\lambda_K = 0$). Moreover we can see that $\theta^{\parallel} \neq 0$ and $\widetilde{\Phi}(x)^* \theta \neq 0$, thus θ is not in $(\operatorname{Ker} W)^{\perp}$.

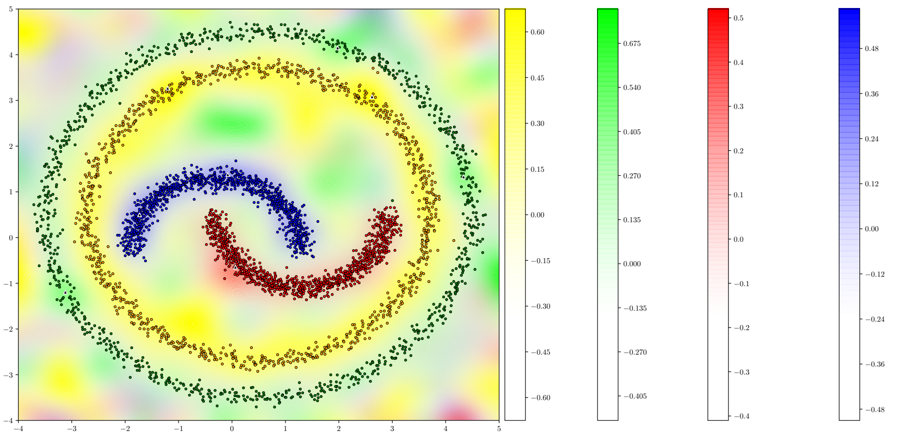

Figure B.2: Semi-supervised learning on a multiclass nested circle and two moons dataset. We performed an unsupervised spectral clustering [161] step to construc the matrices M_{ik}'s and, choose the matrix B of the ORFF using the simplex coding [105].

We note the quantity

$$\widetilde{\mathbf{M}}_{(\lambda_K,\lambda_M)} = I_{\widetilde{\mathcal{H}}} + \frac{\lambda_M}{\lambda_K}\sum_{i,k=1}^{N+U}\widetilde{\Phi}(x_i)M_{ik}\widetilde{\Phi}(x_k)^* \in \mathcal{L}(\widetilde{\mathcal{H}}) \tag{B.7}$$

so that

$$\nabla_\theta \mathfrak{J}_{\lambda_K}(\theta) = \frac{1}{N}\sum_{i=1}^{N}\widetilde{\Phi}(x_i)V^*\left(\frac{\partial}{\partial y}c\left(y,y_i\right)\Big|_{y=V\widetilde{\Phi}(x_i)^*\theta}\right)^* + \lambda_K\widetilde{\mathbf{M}}_{(\lambda_K,\lambda_M)}\theta \tag{B.8}$$

Example B.1 (Naive closed form for the squared error cost). *Consider the cost function defined for all $y, y' \in \mathcal{Y}$ by $c(y,y') = \frac{1}{2}\|y-y\|_{\mathcal{Y}}^2$. Then*

$$\left(\frac{\partial}{\partial y}c\left(y,y_i\right)\Big|_{y=V\widetilde{\Phi}(x_i)^*\theta}\right)^* = \left(V\widetilde{\Phi}(x_i)^*\theta - y_i\right).$$

Thus, since the optimal solution $\theta_{\mathbf{z}}$ verifies $\nabla_{\theta_{\mathbf{z}}}\mathfrak{J}_{\lambda_K}(\theta_{\mathbf{z}}) = 0$ we have

$$\frac{1}{N}\sum_{i=1}^{N}\widetilde{\Phi}(x_i)V^*\left(V\widetilde{\Phi}(x_i)^*\theta_{\mathbf{z}} - y_i\right) + \lambda_K\widetilde{\mathbf{M}}_{(\lambda_K,\lambda_M)}\theta_{\mathbf{z}} = 0.$$

Therefore,

$$\left(\frac{1}{N}\sum_{i=1}^{N}\widetilde{\Phi}(x_i)V^*V\widetilde{\Phi}(x_i)^* + \lambda_K\widetilde{\mathbf{M}}_{(\lambda_K,\lambda_M)}\right)\theta_{\mathbf{z}} = \frac{1}{N}\sum_{i=1}^{N}\widetilde{\Phi}(x_i)V^*y_i. \tag{B.9}$$

Suppose that $\mathcal{Y} \subseteq \mathbb{R}^p$, $V : \mathcal{U} \to \mathcal{Y}$ where $\mathcal{U} \subseteq \mathbb{R}^u$ and for all $x \in \mathcal{X}$, $\widetilde{\Phi}(x) : \mathbb{R}^r \to \mathbb{R}^u$ where all spaces are endowed with the euclidean inner product. From this we can derive algorithm 7 which returns the closed form solution of equation 6.11 for $c(y,y') = \frac{1}{2}\|y-y'\|_2^2$.

Algorithm 7: Naive closed form for the squared error cost.

Input :

- $\mathbf{s} = (x_i, y_i)_{i=1}^N \in (\mathcal{X} \times \mathbb{R}^p)^N$ a sequence of supervised training points,
- $\mathbf{u} = (x_i)_{i=N+1}^{N+U} \in \mathcal{X}^U$ a sequence of unsupervised training points,
- $\widetilde{\Phi}(x_i) \in \mathcal{L}(\mathbb{R}^u, \mathbb{R}^r)$ a feature map defined for all $x_i \in \mathcal{X}$,
- $(M_{ik})_{i,k=1}^{N+U}$ a sequence of data dependent operators (see theorem B.2),
- $V \in \mathcal{L}(\mathbb{R}^u, \mathbb{R}^p)$ a combination operator,
- $\lambda_K \in \mathbb{R}_{>0}$ the Tychonov regularization term,
- $\lambda_M \in \mathbb{R}_+$ the manifold regularization term.

Output : A model

$$h : \begin{cases} \mathcal{X} \to \mathbb{R}^p \\ x \mapsto \widetilde{\Phi}(x)^\top \theta_{\mathbf{z}}, \end{cases}$$

such that $\theta_{\mathbf{z}}$ minimize equation 6.11, where $c(y, y') = \|y - y'\|_2^2$ and $\mathbb{R}^u$, $\mathbb{R}^r$ and $\mathbb{R}^p$ are Hilbert spaces endowed with the euclidean inner product.

1 $\mathbf{P} \leftarrow \frac{1}{N}\sum_{i=1}^N \widetilde{\Phi}(x_i) V^\top V \widetilde{\Phi}(x_i)^\top \in \mathcal{L}(\mathbb{R}^r, \mathbb{R}^r)$;
2 **if** $\lambda_M = 0$ **then**
3 | $\widetilde{\mathbf{M}}_{(\lambda_K, \lambda_M)} \leftarrow I_r \in \mathcal{L}(\mathbb{R}^r, \mathbb{R}^r)$;
4 **else**
5 | $\widetilde{\mathbf{M}}_{(\lambda_K, \lambda_M)} \leftarrow \left(I_r + \frac{\lambda_M}{\lambda_K}\sum_{i,k=1}^{N+U} \widetilde{\Phi}(x_i) M_{ik} \widetilde{\Phi}(x_k)^\top\right) \in \mathcal{L}(\mathbb{R}^r, \mathbb{R}^r)$;
6 **end**
7 $\mathbf{Y} \leftarrow \frac{1}{N}\sum_{i=1}^N \widetilde{\Phi}(x_i) V^\top y_i \in \mathbb{R}^r$;
8 $\theta_{\mathbf{z}} \leftarrow \text{solve}_\theta\left(\left(\mathbf{P} + \lambda_K \widetilde{\mathbf{M}}_{(\lambda_K, \lambda_M)}\right)\theta = \mathbf{Y}\right) \in \mathbb{R}^r$;
9 **return** $h : x \mapsto \widetilde{\Phi}(x)^\top \theta_{\mathbf{z}}$;

B.4.3 *Complexity*

Suppose that $u = \dim(\mathcal{U}) < +\infty$ and $u' = \dim(\mathcal{U}') < \infty$ and for all $x \in \mathcal{X}$, $\widetilde{\Phi}(x) : \mathcal{U}' \to \widetilde{\mathcal{H}}$ where $r = \dim(\widetilde{\mathcal{H}}) < \infty$ is the dimension of the redescription space $\widetilde{\mathcal{H}} = \mathbb{R}^r$. Since u, u', and $r < \infty$, we view the operators $\widetilde{\Phi}(x)$, V and $\widetilde{\mathbf{M}}_{(\lambda_K, \lambda_M)}$ as matrices. Computing $V^* V$ cost $O_t(u^2 p)$. Step 1 costs $O_t(r^2 u + r u^2)$. Steps 5 (optional) has the same cost except that the sum is done over all pair of $N + U$ points thus it costs $O_t((N + U)^2(r^2 u + r u^2))$. Steps 7 costs $O_t(N(ru + up))$. For step 8, the naive inversion of the operator costs $O_t(r^3)$. Eventually the overall complexity of algorithm 7 is

$$O_t\left(ru(r+u)\begin{cases}(N+U)^2 & \text{if } \lambda_M>0\\ N & \text{if } \lambda_M=0\end{cases}+r^3+Nu(r+p)\right),$$

while the space complexity is $O_s(r^2)$. This complexity is to compare with the kernelized solution proposed by Minh, Bazzani, and Murino [103]. Let

$$\mathbf{K}:\begin{cases}\mathcal{U}^{N+U}\to\mathcal{U}^{N+U}\\ u\mapsto\bigoplus_{i=1}^{N+U}\sum_{j=1}^{N+U}K(x_i,x_j)u_j\end{cases}$$

and

$$\mathbf{M}:\begin{cases}\mathcal{U}^{N+U}\to\mathcal{U}^{N+U}\\ u\mapsto\bigoplus_{i=1}^{N+U}\sum_{k=1}^{N+U}M_{ik}u_k.\end{cases}$$

When $\mathcal{U}=\mathbb{R}$,

$$\mathbf{K}=\begin{pmatrix}K(x_1,x_1) & \dots & K(x_1,x_{N+U})\\ \vdots & \ddots & \vdots\\ K(x_{N+U},x_1) & \dots & K(x_{N+U},x_{N+U})\end{pmatrix}$$

is called the Gram matrix of K. When $\mathcal{U}=\mathbb{R}^p$, $\mathbf{K}$ is a matrix-valued Gram matrix of size $u(N+U)\times u(N+U)$ where each entry $\mathbf{K}_{ij}\in\mathcal{M}_{u,u}(\mathbb{R})$. When $\mathcal{U}=\mathbb{R}^u$, $\mathbf{M}$ can also be seen as a matrix-valued matrix where each entry is $M_{ik}\in\mathcal{M}_{u,u}(\mathbb{R})$. We also introduce the matrices $\mathbf{V}^\top\mathbf{V}:=I_{N+U}\otimes(V^\top V)$ and

$$\mathbf{P}:\begin{cases}\mathcal{U}^{N+U}\to\mathcal{U}^{N+U}\\ u\mapsto\left(\bigoplus_{j=1}^{N}u_j\right)\oplus\left(\bigoplus_{j=N+1}^{N+U}0\right)\end{cases}$$

The operator $\mathbf{P}$ is a projection that sets all the terms u_j, $N<j\leq N+U$ of u to zero. When $\mathcal{U}=\mathbb{R}^u$ it can also be seen as the block matrix of size $u(N+U)\times u(N+U)$ and

$$\mathbf{P}=\begin{pmatrix} & & & 0 & \dots & 0\\ & I_u\otimes I_N & & \vdots & \ddots & \vdots\\ & & & 0 & \dots & 0\\ 0 & \dots & 0 & 0 & \dots & 0\\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots\\ 0 & \dots & 0 & 0 & \dots & 0\end{pmatrix}$$

Then the equivalent kernelized solution $u_{\mathbf{z}}$ of theorem B.1 is given by Minh, Bazzani, and Murino [103]

$$\left(\frac{1}{N}\mathbf{V}^\top\mathbf{VPK}+\lambda_M\mathbf{MK}+\lambda_K I_{\bigoplus_{i=1}^{N+U}\mathcal{U}}\right)u_{\mathbf{z}}=\left(\bigoplus_{i=1}^{N}V^\top y_i\right)\oplus\left(\bigoplus_{i=N+1}^{N+U}0\right).$$

which has time complexity $O_t(((N+U)u)^3 + Nup)$ and space complexity $O_s(((N+U)u)^2)$. Notice that computing the data dependent norm (manifold regularization) is expensive. Indeed when $\lambda_M = 0$, algorithm 7 has a linear complexity with respect to the number of supervised training points N while the complexity becomes quadratic in the number of supervised and unsupervised training points $N+U$ when $\lambda_M > 0$.

C

RELEVANT PIECE OF CODE

In the following, we give minimal short samples of Python code showing how to implement efficient ORFF. Each section represent an independent snippet of code, and a simple copy-paste in a python editor should generate the corresponding figure presented in this manuscipt.

Contents

C.1 PYTHON CODE FOR FIGURE 3.1

```python
r"""Plot figure: Different outcomes of a Gaussian kernel approximation."""

import matplotlib
import numpy as np
import matplotlib.pyplot as plt

from sklearn.metrics import pairwise_kernels


def phi(x, w, D):
    r"""RFF map."""
    Z = np.dot(x, w)
    return np.hstack((np.cos(Z), np.sin(Z))) / np.sqrt(D)


def createColorbar(lwr, upr, fig, axes):
    r"""Create colorbar for multiple Pyplot plot."""
    cax = fig.add_axes([.92, 0.1, 0.01, 0.8])
    norm = matplotlib.colors.LogNorm(vmin=lwr, vmax=upr, clip=False)
    c = matplotlib.colorbar.ColorbarBase(cax, cmap=plt.get_cmap('rainbow'),
                                         norm=norm, label='D=')
    plt.title(r'$\widetilde{K}$')
    return c


def main():
    r"""Plot figure: Different outcomes of a Gaussian kernel approximation."""
    T = 25  # Number of curves

    cm_subsection = np.linspace(0, 1, T + 1)
    colors = [matplotlib.cm.rainbow(x) for x in cm_subsection]

    d = 1  # Dimension of the input
    N = 250  # Number of points per curves

    # Generate N data in (-1, 1) and exact Gram matrix
    np.random.seed(0)
    X = np.linspace(-1, 1, N).reshape((N, d))
    K = pairwise_kernels(X, metric='rbf', gamma=1. / (2. * .1 ** 2))

    # A Matrix for the decomposable kernel. Link the outputs to some mean value
    c = np.random.randn(N, 2)
    A = .5 * np.eye(2) + .5 * np.ones((2, 2))

    plt.close()
    plt.rc('text', usetex=True)
    plt.rc('font', family='serif')
    f, axes = plt.subplots(2, 2, figsize=(12, 8), sharex=True, sharey=True)

    # For each curve with different D
    for k, D in enumerate(np.logspace(0, 4, T)):
        D = int(D)
        np.random.seed(0)

        w = np.random.randn(d, D) / .1
        phiX = phi(X, w, D)
        Kt = np.dot(phiX, phiX.T)

        # Generate outputs with the exact Gram matrix
        pred = np.dot(np.dot(Kt, c), A)
        axes[0, 0].plot(X, pred[:, 0], c=colors[k], lw=.5, linestyle='-')
        axes[0, 0].set_ylabel(r'$y_1$')
        axes[0, 1].plot(X, pred[:, 1], c=colors[k], lw=.5, linestyle='-')
        axes[0, 1].set_ylabel(r'$y_2$')
```

```
        # Generate outputs with the a realization of the random Gram matrix
        w = np.random.randn(d, D) / .1
        phiX = phi(X, w, D)
        Kt = np.dot(phiX, phiX.T)

        pred = np.dot(np.dot(Kt, c), A)
        axes[1, 0].plot(X, pred[:, 0], c=colors[k], lw=.5, linestyle='-')
        axes[1, 0].set_xlabel(r'$x$')
        axes[1, 0].set_ylabel(r'$y_1$')
        axes[1, 1].plot(X, pred[:, 1], c=colors[k], lw=.5, linestyle='-')
        axes[1, 1].set_xlabel(r'$x$')
        axes[1, 1].set_ylabel(r'$y_2$')

    axes[0, 0].plot(X, np.dot(np.dot(K, c), A)[:, 0], c='k', lw=.5, label='K')
    axes[0, 1].plot(X, np.dot(np.dot(K, c), A)[:, 1], c='k', lw=.5, label='K')
    axes[1, 0].plot(X, np.dot(np.dot(K, c), A)[:, 0], c='k', lw=.5, label='K')
    axes[1, 1].plot(X, np.dot(np.dot(K, c), A)[:, 1], c='k', lw=.5, label='K')

    axes[0, 0].set_title(r'$\widetilde{K}u \approx Ku$, realization 1', x=1.1)
    axes[1, 0].set_title(r'$\widetilde{K}u \approx Ku$, realization 2', x=1.1)

    for xx in axes.ravel():
        xx.legend(loc=4)

    createColorbar(1, D, f, axes)
    plt.savefig('not_Mercer.pgf', bbox_inches='tight')

if __name__ == "__main__":
    main()
```

C.2 PYTHON CODE FOR FIGURE 5.1

```
r"""Plot figure: ORFF Representer theorem pt.1."""

import numpy as np
import matplotlib.pyplot as plt
import matplotlib


def phi(x, w, D):
    r"""RFF map."""
    Z = np.dot(x, w)
    return np.hstack((np.cos(Z), np.sin(Z))) / np.sqrt(D)


def main():
    r"""Plot figure: ORFF Representer theorem pt.1."""
    d = 1  # dimensionality of the inputs
    D = 50  # number of random features

    N = 50
    Nt = 200

    # N training points in (0,1)
    np.random.seed(0)
    x = 2 * np.random.rand(N, d) - 1
    y = np.sin(10 * x)
    y += .5 * np.random.randn(y.shape[0], y.shape[1]) + 2. * x ** 2

    # Nt testing points in (0,1)
    xt = np.linspace(-1, 1, Nt).reshape((-1, 1))
    yt = np.sin(10 * xt) + 2. * xt ** 2
    yt += .5 * np.random.randn(yt.shape[0], yt.shape[1])
```

```
sigma = .3
w = np.random.randn(d, D) / sigma  # Realization of (\omega_j)_{j=1}^D

phiX = phi(x, w, D)  # Train RFF
phiXt = phi(xt, w, D)  # Test RFF

# Create plot
plt.close()
plt.rc('text', usetex=True)
plt.rc('font', family='serif')
f, axis = plt.subplots(4, 3, gridspec_kw={'width_ratios': [3, 3, 1.5]},
                       figsize=(16, 6), sharex='col', sharey='col')
f.subplots_adjust(hspace=.25)
formatter = matplotlib.ticker.ScalarFormatter()
formatter.set_powerlimits((-3, 4))

# For different hyperparameters \lambda
for k, lbda in enumerate([1e-2, 5e-6, 1e-10, 0]):
    # Train with ORFF with kernel approximation (dual)
    ck = np.linalg.lstsq(np.dot(phiX, phiX.T) + lbda * np.eye(N),
                         y, rcond=-1)[0]
    # Train with ORFF without kernel approximation (primal)
    c = np.linalg.lstsq(np.dot(phiX.T, phiX) + lbda * np.eye(2 * D),
                        np.dot(phiX.T, y), rcond=-1)[0]
    cc = np.sum((phi(x, w, D) * ck), axis=0)
    # Link dual coefficient with primal coefficients
    cr = (cc - c.ravel()) / np.linalg.norm(c) * 100
    err = np.array([np.linalg.norm(np.dot(phiXt, c) - yt) ** 2 / Nt,
                    np.linalg.norm(np.dot(np.dot(phiXt,
                                                 phiX.T),
                                          ck) - yt) ** 2 / Nt,
                    np.linalg.norm(np.dot(phiXt, cr)) ** 2 / Nt,
                    np.linalg.norm(cr)])

    # Plot
    lmin = -1.8
    lmax = 3.
    axis[k, 0].set_xlim([-1.5, 1])
    axis[k, 0].set_ylim([lmin, lmax])
    axis[k, 0].plot(xt, np.dot(phiXt, c),
                    label=r'$\widetilde{\Phi}^* \theta$')
    axis[k, 0].plot(xt, np.dot(np.dot(phiXt, phiX.T), ck),
                    label=r'$\widetilde{K}u$', linestyle='-.')
    axis[k, 0].scatter(x, y, c='r', marker='+', label='train', lw=2)
    axis[k, 0].scatter(xt, yt, c='k', marker='.', label='test')
    axis[k, 0].legend(loc=3)
    axis[k, 0].set_ylabel('y')
    if k == 3:
        axis[k, 0].set_xlabel('x')

    lmin = -1.8
    lmax = 3.
    pred = np.dot(phi(xt, w, D), cr)
    axis[k, 1].set_xlim([-1.5, 1])
    axis[k, 1].set_ylim([lmin, lmax])
    axis[k, 1].plot(xt, pred,
                    label=r'$\widetilde{\Phi}^* \theta^{\parallel}$')
    axis[k, 1].scatter(x, y, c='r', marker='+', label='train', lw=2)
    axis[k, 1].scatter(xt, yt, c='k', marker='.', label='test')
    axis[k, 1].legend(loc=3)
    if k == 3:
        axis[k, 1].set_xlabel('x')

    xs = np.arange(cr.size)
    axis[k, 2].barh(xs, np.abs(cr), edgecolor="none", log=True)
    axis[k, 2].set_ylabel(r'$j$')
```

```
        if k == 3:
            axis[k, 2].set_xlabel(
                r'$|\theta^{\parallel}_j|$, \% of relative error')
    plt.savefig('representer.pgf', bbox_inches='tight')

    return err

if __name__ == "__main__":
    main()
```

C.3 PYTHON CODE FOR FIGURE 5.3

```
r"""Efficient implementation of the Gaussian ORFF decomposable kernel."""

from time import time

from pympler.asizeof import asizeof

from numpy.linalg import svd
from numpy.random import rand, seed
from numpy import (dot, diag, sqrt, kron, zeros,
                   logspace, log10, matrix, eye, int, float)
from scipy.sparse.linalg import LinearOperator
from sklearn.kernel_approximation import RBFSampler
from matplotlib.pyplot import savefig, subplots, tight_layout


def NaiveDecomposableGaussianORFF(X, A, gamma=1.,
                                  D=100, eps=1e-5, random_state=0):
    r"""Return the Naive ORFF map associated with the data X.

    Parameters
    ----------
    X : {array-like}, shape = [n_samples, n_features]
        Samples.
    A : {array-like}, shape = [n_targets, n_targets]
        Operator of the Decomposable kernel (positive semi-definite)
    gamma : {float},
        Gamma parameter of the RBF kernel.
    D : {integer},
        Number of random features.
    eps : {float},
        Cutoff threshold for the singular values of A.
    random_state : {integer},
        Seed of the generator.

    Returns
    -------
    \tilde{\Phi}(X) : array
    """
    # Decompose A=BB^T
    u, s, v = svd(A, full_matrices=False, compute_uv=True)
    B = dot(diag(sqrt(s[s > eps])), v[s > eps, :])

    # Sample a RFF from the scalar Gaussian kernel
    phi_s = RBFSampler(gamma=gamma, n_components=D, random_state=random_state)
    phiX = phi_s.fit_transform(X)

    # Create the ORFF linear operator
    return matrix(kron(phiX, B))


def EfficientDecomposableGaussianORFF(X, A, gamma=1.,
                                      D=100, eps=1e-5, random_state=0):
    r"""Return the Efficient ORFF map associated with the data X.
```

```
    Parameters
    ----------
    X : {array-like}, shape = [n_samples, n_features]
        Samples.
    A : {array-like}, shape = [n_targets, n_targets]
        Operator of the Decomposable kernel (positive semi-definite)
    gamma : {float},
        Gamma parameter of the RBF kernel.
    D : {integer},
        Number of random features.
    eps : {float},
        Cutoff threshold for the singular values of A.
    random_state : {integer},
        Seed of the generator.

    Returns
    -------
    \tilde{\Phi}(X) : Linear Operator, callable
    """
    # Decompose A=BB^T
    u, s, v = svd(A, full_matrices=False, compute_uv=True)
    B = dot(diag(sqrt(s[s > eps])), v[s > eps, :])

    # Sample a RFF from the scalar Gaussian kernel
    phi_s = RBFSampler(gamma=gamma, n_components=D, random_state=random_state)
    phiX = phi_s.fit_transform(X)

    # Create the ORFF linear operator
    cshape = (D, B.shape[0])
    rshape = (X.shape[0], B.shape[1])
    return LinearOperator((phiX.shape[0] * B.shape[1], D * B.shape[0]),
                          matvec=lambda b: dot(phiX, dot(b.reshape(cshape),
                                                         B)),
                          rmatvec=lambda r: dot(phiX.T, dot(r.reshape(rshape),
                                                          B.T)),
                          dtype=float)


def main():
    r"""Plot figure: Efficient decomposable gaussian ORFF."""
    N = 100  # Number of points
    pmax = 100  # Maximum output dimension
    d = 20  # Input dimension
    D = 100  # Number of random features

    seed(0)
    X = rand(N, d)

    R, T = 10, 10
    time_Efficient, mem_Efficient = zeros((R, T, 2)), zeros((R, T))
    time_naive, mem_naive = zeros((R, T, 2)), zeros((R, T))

    for i, p in enumerate(logspace(0, log10(pmax), T)):
        A = rand(int(p), int(p))
        A = dot(A.T, A) + eye(int(p))

        # Perform \Phi(X)^T \theta with Efficient implementation
        for j in range(R):
            start = time()
            phiX1 = EfficientDecomposableGaussianORFF(X, A, D)
            time_Efficient[j, i, 0] = time() - start
            theta = rand(phiX1.shape[1], 1)
            start = time()
            phiX1 * theta
            time_Efficient[j, i, 1] = time() - start
```

```
            mem_Efficient[j, i] = asizeof(phiX1, code=True)

        # Perform \Phi(X)^T \theta with naive implementation
        for j in range(R):
            start = time()
            phiX2 = NaiveDecomposableGaussianORFF(X, A, D)
            time_naive[j, i, 0] = time() - start
            theta = rand(phiX2.shape[1], 1)
            start = time()
            phiX2 * theta
            time_naive[j, i, 1] = time() - start
            mem_naive[j, i] = asizeof(phiX2, code=True)

    # Plot
    f, axes = subplots(1, 3, figsize=(10, 4), sharex=True, sharey=False)
    axes[0].errorbar(logspace(0, log10(pmax), T).astype(int),
                     time_Efficient[:, :, 0].mean(axis=0),
                     time_Efficient[:, :, 0].std(axis=0),
                     label='Efficient decomposable ORFF')
    axes[0].errorbar(logspace(0, log10(pmax), T).astype(int),
                     time_naive[:, :, 0].mean(axis=0),
                     time_naive[:, :, 0].std(axis=0),
                     label='Naive decomposable ORFF')
    axes[1].errorbar(logspace(0, log10(pmax), T).astype(int),
                     time_Efficient[:, :, 1].mean(axis=0),
                     time_Efficient[:, :, 1].std(axis=0),
                     label='Efficient decomposable ORFF')
    axes[1].errorbar(logspace(0, log10(pmax), T).astype(int),
                     time_naive[:, :, 1].mean(axis=0),
                     time_naive[:, :, 1].std(axis=0),
                     label='Naive decomposable ORFF')
    axes[2].errorbar(logspace(0, log10(pmax), T).astype(int),
                     mem_Efficient[:, :].mean(axis=0),
                     mem_Efficient[:, :].std(axis=0),
                     label='Efficient decomposable ORFF')
    axes[2].errorbar(logspace(0, log10(pmax), T).astype(int),
                     mem_naive[:, :].mean(axis=0),
                     mem_naive[:, :].std(axis=0),
                     label='Naive decomposable ORFF')
    axes[0].set_xscale('log')
    axes[0].set_yscale('log')
    axes[1].set_xscale('log')
    axes[1].set_yscale('log')
    axes[2].set_xscale('log')
    axes[2].set_yscale('log')
    axes[0].set_xlabel(r'$p=\dim(\mathcal{Y})$')
    axes[1].set_xlabel(r'$p=\dim(\mathcal{Y})$')
    axes[2].set_xlabel(r'$p=\dim(\mathcal{Y})$')
    axes[0].set_ylabel(r'time (s)')
    axes[2].set_ylabel(r'memory (bytes)')
    axes[0].set_title(r'Preprocessing time')
    axes[1].set_title(r'$\widetilde{\Phi}(X)^T \theta$ computation time')
    axes[2].set_title(r'$\widetilde{\Phi}(X)^T$ required memory')
    axes[0].legend(loc=2)
    tight_layout()
    savefig('efficient_decomposable_gaussian.pgf', bbox_inches='tight')

if __name__ == "__main__":
    main()
```

C.4 PYTHON CODE FOR FIGURE 5.4

```
r"""Efficient implementation of the Gaussian curl-free kernel."""

from time import time
```

```python
from pympler.asizeof import asizeof

from numpy.random import rand, seed
from numpy import dot, zeros, logspace, log10, matrix, int, float
from scipy.sparse.linalg import LinearOperator
from sklearn.kernel_approximation import RBFSampler
from matplotlib.pyplot import savefig, subplots, tight_layout


def NaiveCurlFreeGaussianORFF(X, gamma=1.,
                              D=100, eps=1e-5, random_state=0):
    r"""Return the Naive ORFF map associated with the data X.

    Parameters
    ----------
    X : {array-like}, shape = [n_samples, n_features]
        Samples.
    gamma : {float},
        Gamma parameter of the RBF kernel.
    D : {integer},
        Number of random features.
    eps : {float},
        Cutoff threshold for the singular values of A.
    random_state : {integer},
        Seed of the generator.

    Returns
    -------
    \tilde{\Phi}(X) : array
    """
    phi_s = RBFSampler(gamma=gamma, n_components=D,
                       random_state=random_state)
    phiX = phi_s.fit_transform(X)
    phiX = (phiX.reshape((phiX.shape[0], 1, phiX.shape[1])) *
            phi_s.random_weights_.reshape((1, -1, phiX.shape[1])))

    return matrix(phiX.reshape((-1, phiX.shape[2])))


def EfficientCurlFreeGaussianORFF(X, gamma=1.,
                                  D=100, eps=1e-5, random_state=0):
    r"""Return the Efficient ORFF map associated with the data X.

    Parameters
    ----------
    X : {array-like}, shape = [n_samples, n_features]
        Samples.
    gamma : {float},
        Gamma parameter of the RBF kernel.
    D : {integer},
        Number of random features.
    eps : {float},
        Cutoff threshold for the singular values of A.
    random_state : {integer},
        Seed of the generator.

    Returns
    -------
    \tilde{\Phi}(X) : array
    """
    phi_s = RBFSampler(gamma=gamma, n_components=D,
                       random_state=random_state)
    phiX = phi_s.fit_transform(X)

    return LinearOperator((phiX.shape[0] * X.shape[1], phiX.shape[1]),
```

```
                      matvec=lambda b:
                      dot(phiX.reshape((phiX.shape[0], 1, phiX.shape[1])) *
                      phi_s.random_weights_.reshape((1, -1,
                                                     phiX.shape[1])), b),
                      rmatvec=lambda r:
                      dot((phiX.reshape((phiX.shape[0], 1,
                                         phiX.shape[1])) *
                           phi_s.random_weights_.reshape((1, -1,
                                                          phiX.shape
                                                          [1]))).reshape
                      (phiX.shape[0] * X.shape[1], phiX.shape[1]).T, r),
                      dtype=float)


def main():
    r"""Plot figure: Efficient decomposable gaussian ORFF."""
    N = 1000  # Number of points
    dmax = 100  # Input dimension
    D = 500  # Number of random features

    seed(0)

    R, T = 50, 10
    time_Efficient, mem_Efficient = zeros((R, T, 2)), zeros((R, T))
    time_naive, mem_naive = zeros((R, T, 2)), zeros((R, T))

    for i, d in enumerate(logspace(0, log10(dmax), T)):
        X = rand(N, int(d))

        # Perform \Phi(X)^T \theta with Efficient implementation
        for j in range(R):
            start = time()
            phiX1 = EfficientCurlFreeGaussianORFF(X, D)
            time_Efficient[j, i, 0] = time() - start
            start = time()
            phiX1 * rand(phiX1.shape[1], 1)
            time_Efficient[j, i, 1] = time() - start
            mem_Efficient[j, i] = asizeof(phiX1, code=True)

        # Perform \Phi(X)^T \theta with naive implementation
        for j in range(R):
            start = time()
            phiX2 = NaiveCurlFreeGaussianORFF(X, D)
            time_naive[j, i, 0] = time() - start
            start = time()
            phiX2 * rand(phiX2.shape[1], 1)
            time_naive[j, i, 1] = time() - start
            mem_naive[j, i] = asizeof(phiX2, code=True)

    # Plot
    f, axes = subplots(1, 3, figsize=(10, 4), sharex=True, sharey=False)
    axes[0].errorbar(logspace(0, log10(dmax), T).astype(int),
                     time_Efficient[:, :, 0].mean(axis=0),
                     time_Efficient[:, :, 0].std(axis=0),
                     label='Efficient decomposable ORFF')
    axes[0].errorbar(logspace(0, log10(dmax), T).astype(int),
                     time_naive[:, :, 0].mean(axis=0),
                     time_naive[:, :, 0].std(axis=0),
                     label='Naive decomposable ORFF')
    axes[1].errorbar(logspace(0, log10(dmax), T).astype(int),
                     time_Efficient[:, :, 1].mean(axis=0),
                     time_Efficient[:, :, 1].std(axis=0),
                     label='Efficient decomposable ORFF')
    axes[1].errorbar(logspace(0, log10(dmax), T).astype(int),
                     time_naive[:, :, 1].mean(axis=0),
                     time_naive[:, :, 1].std(axis=0),
```

```
                     label='Naive decomposable ORFF')
    axes[2].errorbar(logspace(0, log10(dmax), T).astype(int),
                     mem_Efficient[:, :].mean(axis=0),
                     mem_Efficient[:, :].std(axis=0),
                     label='Efficient decomposable ORFF')
    axes[2].errorbar(logspace(0, log10(dmax), T).astype(int),
                     mem_naive[:, :].mean(axis=0),
                     mem_naive[:, :].std(axis=0),
                     label='Naive decomposable ORFF')
    axes[0].set_xscale('log')
    axes[0].set_yscale('log')
    axes[1].set_xscale('log')
    axes[1].set_yscale('log')
    axes[2].set_xscale('log')
    axes[2].set_yscale('log')
    axes[0].set_xlabel(r'$p=\dim(\mathcal{Y})$')
    axes[1].set_xlabel(r'$p=\dim(\mathcal{Y})$')
    axes[2].set_xlabel(r'$p=\dim(\mathcal{Y})$')
    axes[0].set_ylabel(r'time (s)')
    axes[2].set_ylabel(r'memory (bytes)')
    axes[0].set_title(r'Preprocessing time')
    axes[1].set_title(r'$\widetilde{\Phi}(X)^T \theta$ computation time')
    axes[2].set_title(r'$\widetilde{\Phi}(X)^T$ required memory')
    axes[0].legend(loc=2)
    tight_layout()
    savefig('efficient_curlfree_gaussian.pgf', bbox_inches='tight')

if __name__ == "__main__":
    main()
```

C.5 PYTHON CODE FOR FIGURE 5.5

```
r"""Efficient implementation of the Gaussian divergence-free kernel."""

from time import time

from pympler.asizeof import asizeof

from numpy.random import rand, seed
from numpy.linalg import norm
from numpy import dot, zeros, logspace, log10, matrix, int, eye, float
from scipy.sparse.linalg import LinearOperator
from sklearn.kernel_approximation import RBFSampler
from matplotlib.pyplot import savefig, subplots, tight_layout


def _rebase(phiX, W, Wn):
    return (phiX.reshape((phiX.shape[0], 1, 1, phiX.shape[1])) *
            (eye(W.shape[1]).reshape(1, W.shape[1], W.shape[1], 1) * Wn -
            W * W.reshape(1, 1, W.shape[1], phiX.shape[1]) / Wn)).reshape(
        (-1, W.shape[1] * Wn.shape[3]))


def NaiveDivergenceFreeGaussianORFF(X, gamma=1.,
                                    D=100, eps=1e-5, random_state=0):
    r"""Return the Naive ORFF map associated with the data X.

    Parameters
    ----------
    X : {array-like}, shape = [n_samples, n_features]
        Samples.
    gamma : {float},
        Gamma parameter of the RBF kernel.
    D : {integer},
        Number of random features.
```

```
    eps : {float},
        Cutoff threshold for the singular values of A.
    random_state : {integer},
        Seed of the generator.

    Returns
    -------
    \tilde{\Phi}(X) : array
    """
    phi_s = RBFSampler(gamma=gamma, n_components=D,
                       random_state=random_state)

    phiX = _rebase(phi_s.fit_transform(X),
                   phi_s.random_weights_.reshape((1, -1, 1, D)),
                   norm(phi_s.random_weights_, axis=0).reshape((1, 1, 1, -1)))

    return matrix(phiX)


def EfficientDivergenceFreeGaussianORFF(X, gamma=1.,
                                        D=100, eps=1e-5, random_state=0):
    r"""Return the Efficient ORFF map associated with the data X.

    Parameters
    ----------
    X : {array-like}, shape = [n_samples, n_features]
        Samples.
    gamma : {float},
        Gamma parameter of the RBF kernel.
    D : {integer},
        Number of random features.
    eps : {float},
        Cutoff threshold for the singular values of A.
    random_state : {integer},
        Seed of the generator.

    Returns
    -------
    \tilde{\Phi}(X) : array
    """
    phi_s = RBFSampler(gamma=gamma, n_components=D,
                       random_state=random_state)
    phiX = phi_s.fit_transform(X)
    W = phi_s.random_weights_.reshape((1, -1, 1, phiX.shape[1]))
    Wn = norm(phi_s.random_weights_, axis=0).reshape((1, 1, 1, -1))
    return LinearOperator((phiX.shape[0] * X.shape[1],
                           phiX.shape[1] * X.shape[1]),
                          matvec=lambda b: dot(_rebase(phiX, W, Wn), b),
                          rmatvec=lambda r: dot(_rebase(phiX, W, Wn).T, r),
                          dtype=float)


def main():
    r"""Plot figure: Efficient decomposable gaussian ORFF."""
    N = 100  # Number of points
    dmax = 100  # Input dimension
    D = 100  # Number of random features

    seed(0)

    R, T = 10, 10
    time_Efficient, mem_Efficient = zeros((R, T, 2)), zeros((R, T))
    time_naive, mem_naive = zeros((R, T, 2)), zeros((R, T))

    for i, d in enumerate(logspace(0, log10(dmax), T)):
        X = rand(N, int(d))
```

```
        # Perform \Phi(X)^T \theta with Efficient implementation
        for j in range(R):
            start = time()
            phiX1 = EfficientDivergenceFreeGaussianORFF(X, D)
            time_Efficient[j, i, 0] = time() - start
            theta = rand(phiX1.shape[1], 1)
            start = time()
            phiX1 * theta
            time_Efficient[j, i, 1] = time() - start
            mem_Efficient[j, i] = asizeof(phiX1, code=True)

        # Perform \Phi(X)^T \theta with naive implementation
        for j in range(R):
            start = time()
            phiX2 = NaiveDivergenceFreeGaussianORFF(X, D)
            time_naive[j, i, 0] = time() - start
            theta = rand(phiX2.shape[1], 1)
            start = time()
            phiX2 * theta
            time_naive[j, i, 1] = time() - start
            mem_naive[j, i] = asizeof(phiX2, code=True)

    # Plot
    f, axes = subplots(1, 3, figsize=(10, 4), sharex=True, sharey=False)
    axes[0].errorbar(logspace(0, log10(dmax), T).astype(int),
                     time_Efficient[:, :, 0].mean(axis=0),
                     time_Efficient[:, :, 0].std(axis=0),
                     label='Efficient decomposable ORFF')
    axes[0].errorbar(logspace(0, log10(dmax), T).astype(int),
                     time_naive[:, :, 0].mean(axis=0),
                     time_naive[:, :, 0].std(axis=0),
                     label='Naive decomposable ORFF')
    axes[1].errorbar(logspace(0, log10(dmax), T).astype(int),
                     time_Efficient[:, :, 1].mean(axis=0),
                     time_Efficient[:, :, 1].std(axis=0),
                     label='Efficient decomposable ORFF')
    axes[1].errorbar(logspace(0, log10(dmax), T).astype(int),
                     time_naive[:, :, 1].mean(axis=0),
                     time_naive[:, :, 1].std(axis=0),
                     label='Naive decomposable ORFF')
    axes[2].errorbar(logspace(0, log10(dmax), T).astype(int),
                     mem_Efficient[:, :].mean(axis=0),
                     mem_Efficient[:, :].std(axis=0),
                     label='Efficient decomposable ORFF')
    axes[2].errorbar(logspace(0, log10(dmax), T).astype(int),
                     mem_naive[:, :].mean(axis=0),
                     mem_naive[:, :].std(axis=0),
                     label='Naive decomposable ORFF')
    axes[0].set_xscale('log')
    axes[0].set_yscale('log')
    axes[1].set_xscale('log')
    axes[1].set_yscale('log')
    axes[2].set_xscale('log')
    axes[2].set_yscale('log')
    axes[0].set_xlabel(r'$p=\dim(\mathcal{Y})$')
    axes[1].set_xlabel(r'$p=\dim(\mathcal{Y})$')
    axes[2].set_xlabel(r'$p=\dim(\mathcal{Y})$')
    axes[0].set_ylabel(r'time (s)')
    axes[2].set_ylabel(r'memory (bytes)')
    axes[0].set_title(r'Preprocessing time')
    axes[1].set_title(r'$\widetilde{\Phi}(X)^T \theta$ computation time')
    axes[2].set_title(r'$\widetilde{\Phi}(X)^T$ required memory')
    axes[0].legend(loc=2)
    tight_layout()
    savefig('efficient_divfree_gaussian.pgf', bbox_inches='tight')
```

```
if __name__ == "__main__":
    main()
```

D

ONE CLASS SPLITTING CRITERIA FOR RANDOM FORESTS

This thesis has also been the opportunity to be part of a collaborative work on anomaly detection with random forests, with other fellow Ph. D. students of Télécom ParisTech. The following paper Goix et al. [64] is based on a original idea of Nicolas Goix and a joint work with Nicolas Drougard and Maël Chiapino. It is currently under review at ECML. Our original paper can be found at https://arxiv.org/pdf/1611.01971.pdf.

Contents

Anomalies, novelties or outliers are usually assumed to lie in low probability regions of the data generating process. This assumption drives many statistical anomaly detection methods. Parametric techniques [13, 50] suppose that the inliers are generated by a distribution belonging to some specific parametric model a priori known. Here and hereafter, we denote by inliers the "not abnormal" data, and by outliers/anomalies/novelties the data from the abnormal class. Classical non-parametric approaches are based on density (level set) estimation [28, 120, 129, 133], on dimensionality reduction [1, 138] or on decision trees [89, 137]. Relevant overviews of current research on anomaly detection can be found in Chandola, Banerjee, and Kumar [35], Hodge and Austin [69], Markou and Singh [96], and Patcha and Park [113].

The algorithm proposed in this paper lies in the *novelty detection* setting, also called *one-class classification*. In this framework, we assume that we only observe examples of one class (referred to as the normal class, or inlier class). The second –hidden– class is called the abnormal class, or outlier class. The goal is to identify characteristics of the inlier class, such as its support or some density level sets with levels close to zero. This setup is for instance used in some –non-parametric– kernel methods such as OCSM [129], which extends the Support Vector Machine (SVM) methodology [40, 136] to handle training using only inliers. Recently, LSAD [120], a kernel method similarly extends a multi-class probabilistic classifier [146] to the one-class setting.

Random Forests (RFs) are strong machine learning tools [27], comparing well with state-of-the-art methods such as SVM or boosting algorithms [58], and used in a wide range of domains [46, 61, 149]. These estimators fit a number of decision tree classifiers on different random sub-samples of the dataset. Each tree is built recursively, according to a splitting criterion based on some impurity measure of a node. The prediction is done by an average over each tree prediction. In classification the averaging is based on a majority vote. Practical and theoretical insights on RFs are given in Biau, Devroye, and Lugosi [17], Biau and Scornet [18], Genuer, Poggi, and Tuleau [60], and Louppe [92].

Yet few attempts have been made to transfer the idea of RFs to one-class classification [44, 89, 137]. In Liu, Ting, and Zhou [89], the novel concept of *isolation* is introduced. The Isolation Forest algorithm isolates anomalies, instead of profiling the inlier behavior which is the usual approach. It avoids adapting splitting rules to the one-class setting by using extremely randomized trees, also named extra trees [62]: isolation trees are built completely randomly, without any splitting rule. Therefore, Isolation Forest is not really based on RFs, the base estimators being extra trees instead of classical decision trees. Isolation Forest performs very well in practice with low memory and time complexities. In Désir et al. [44]

and Shi and Horvath [137], outliers are generated to artificially form a second class. In Désir et al. [44] the authors propose a technique to reduce the number of outliers needed by shrinking the dimension of the input space. The outliers are then generated from the reduced space using a distribution complementary to the inlier distribution. Thus their algorithm artificially generates a second class, to use classical RFs. In Shi and Horvath [137], two different outliers generating processes are compared. In the first one, an artificial second class is created by randomly sampling from the product of empirical marginal –inlier– distributions. In the second one outliers are uniformly generated from the hyper-rectangle that contains the observed data. The first option is claimed to work best in practice, which can be understood from the curse of dimensionality argument: in large dimension [151], when the outliers distribution is not tightly defined around the target set, the chance for an outlier to be in the target set becomes very small, so that a huge number of outliers is needed.

Looking beyond the RF literature, Scott and Nowak [133] proposes a methodology to build dyadic decision trees to estimate minimum-volume sets [49, 119]. This is done by reformulating their structural risk minimization problem to be able to use the algorithm in Blanchard, Schäfer, and Rozenholc [19]. While this methodology can also be used for non-dyadic trees pruning (assuming such a tree has been previously constructed, e. g. using some greedy heuristic), it does not allow to grow such trees. Also, the theoretical guaranties derived there relies on the dyadic structure assumption. In the same spirit, Clémençon and Robbiano [36] proposes to use the two-class splitting criterion defined in Clémençon and Vayatis [37]. This two-class splitting rule aims at producing oriented decision trees with a "left-to-right" structure to address the bipartite ranking task. Extension to the one-class setting is done by assuming a uniform distribution for the outlier class. Consistency and rate bounds relies also on this left-to-right structure. building process to a recursive optimization procedure, thus allowing Thus, these two references [36, 133] impose constraints on the tree structure (designed to allow a statistical study) which differs then significantly from the general structure of the base estimators in RF. The price to pay is the flexibility of the model, and its ability to capture complex broader patterns or structural characteristics from the data.

In this paper, we make the choice to stick to the RF framework. We do not assume any structure for the binary decision trees. The price to pay is the lack of statistical guaranties –the consistency of RFs has only been proved recently [131] and in the context of regression additive models. The gain is that we preserve the flexibility and strength of RFs, the algorithm presented here being able to compete well with state-of-the-art anomaly detection algorithms. Besides, we do not assume any –fixed in advance– outlier distribution as in Clémençon and Robbiano [36], but define it in an adaptive way during the tree building process.

To the best of our knowledge, no algorithm structurally extends (without second class sampling and without alternative base estimators) RFs to one-class classification. The main purpose of this work is to introduce such a methodology. It builds on a natural adaptation of two-class Gini-based criterion specially designed for splitting criteria to the one-class setting, as well as an adaptation of the two-class majority vote.

The basic underlying idea is the following. To split a node without second class examples (outliers), we proceed as follows. Each time we look for the best split for a node t, we simply replace (in the two-class *impurity decrease* to be maximized going to the left child node t_L by the proportion expectation $\mathbf{Leb}(t_L)/\mathbf{Leb}(t)$ (idem for the right node), $\mathbf{Leb}(t)$ being the Lebesgue measure, i. e. the volume of the rectangular cell corresponding to node t. It ensures that one child node manages to capture the maximum number of observations with a minimal volume, while the other child looks for the opposite.

This simple idea corresponds to an adaptive modeling of the outlier distribution. The proportion expectation mentioned above is weighted proportionally to the number of inliers in node t. Thus, the resulting outlier distribution is tightly concentrated around the inliers. the latter concentrates outside, closely around but also inside the support of the normal distribution. Besides, and this attests the consistency of our approach with the two-class framework, it turns out that the one-class model promoted here corresponds to the asymptotic behavior of an adaptive outliers generating methodology.

This paper is structured as follows. Appendix D.1 provides the reader with necessary background, to address appendix D.2 which proposes an adaptation of RFs to the one-class setting and describes a generic one-class random forest algorithm. The latter is compared empirically with state-of-the-art anomaly detection methods in appendix D.3. Finally a theoretical justification of the one-class criterion is given in appendix D.4.

D.1 BACKGROUND ON DECISION TREES

Let us denote by $\mathcal{X} \subset \mathbb{R}^d$ the d-dimensional hyper-rectangle containing all the observations. Consider a binary tree on $\mathcal{X}$ whose node values are subsets of $\mathcal{X}$, iteratively produced by splitting $\mathcal{X}$ into two disjoint subsets. Each internal node t with value $\mathcal{X}_t$ is labeled with a split feature m_t and split value c_t (along that feature), in such a way that it divides $\mathcal{X}_t$ into two disjoint spaces $\mathcal{X}_{t_L} := \{ x \in \mathcal{X}_t \mid x_{m_t} < c_t \}$ and $\mathcal{X}_{t_R} := \{ x \in \mathcal{X}_t \mid x_{m_t} \geq c_t \}$, where t_L (respectively t_R) denotes the left (respectively right) children of node t, and x_j denotes the jth coordinate of vector x. Such a binary tree is grown from a sample $X_1, \dots, X_n$ ($\forall i \in \mathbb{N}_n^*, X_i \in \mathcal{X}$) and its finite depth is determined either by a fixed maximum depth value or by a stopping criterion evaluated on the nodes (e. g. based on an impurity measure). The external nodes (the *leaves*) form a partition of $\mathcal{X}$.

In a supervised classification setting, these binary trees are called *classification trees* and prediction is made by assigning to each sample $x \in \mathcal{X}$ the majority class of the leaves containing x. This is called the *majority vote*. Classification trees are usually built using an impurity measure $i(t)$ whose decrease is maximized at each split of a node t, yielding an optimal split (m_t^*, c_t^*). The decrease of impurity (also called *goodness of split*) $\Delta i(t, t_L, t_R)$ w. r. t. the split (m_t, c_t) and corresponding to the partition $\mathcal{X}_t = \mathcal{X}_{t_L} \sqcup \mathcal{X}_{t_R}$ of the node t is defined as

$$\Delta i(t, t_L, t_R) = i(t) - p_L i(t_L) - p_R i(t_R), \tag{D.1}$$

where $p_L = p_L(t)$ (respectively $p_R = p_R(t)$) is the proportion of samples from $\mathcal{X}_t$ going to $\mathcal{X}_{t_L}$ (respectively to $\mathcal{X}_{t_R}$). The impurity measure $i(t)$ reflects the goodness of node t: the smaller $i(t)$, the purer the node t and the better the prediction by majority vote on this node. Usual choices for $i(t)$ are the Gini index [63] or the Shannon entropy [135]. To produce a randomized tree, these optimization steps are usually partially randomized (conditionally on the data, splits (m_t^*, c_t^*)'s become random variables). A classification tree can even be grown totally randomly [62]. In a two-class classification setup, the Gini index is

$$i_G(t) = 2\left(\frac{n_t}{n_t + n'_t}\right)\left(\frac{n'_t}{n_t + n'_t}\right) \tag{D.2}$$

where n_t (respectively n'_t) stands for the number of observations with label 0 (respectively 1) in node t. The Gini index is maximal when $n_t/(n_t + n'_t) = n'_t/(n_t + n'_t) = 0.5$, namely when the conditional probability to have label 0 given that we are in node t is the same as to have label 0 unconditionally: the node t does not discriminate at all between the two classes. For a node t, maximizing the impurity decrease equation D.1 is equivalent to minimizing $p_L i(t_L) + p_R i(t_R)$. Since $p_L = (n_{t_L} + n'_{t_L})/(n_t + n'_t)$ and $p_R = (n_{t_R} + n'_{t_R})/(n_t + n'_t)$, and the quantity $(n_t + n'_t)$ being constant in the optimization problem, this is equivalent to minimizing the following proxy of the impurity decrease,

$$I(t_L, t_R) = (n_{t_L} + n'_{t_L})i(t_L) + (n_{t_R} + n'_{t_R})i(t_R). \tag{D.3}$$

Note that with the Gini index $i_G(t)$ given in equation D.2, the corresponding proxy of the impurity decrease is

$$I_G(t_L, t_R) = \frac{n_{t_L} n'_{t_L}}{n_{t_L} + n'_{t_L}} + \frac{n_{t_R} n'_{t_R}}{n_{t_R} + n'_{t_R}}. \tag{D.4}$$

In the one-class setting, no label is available, hence the impurity measure $i(t)$ does not apply to this setup. The standard splitting criterion which consists in minimizing the latter cannot be used anymore.

D.2 ADAPTATION TO THE ONE-CLASS SETTING

The two reasons why RFs do not apply to one-class classification are that the standard splitting criterion does not apply to this setup, as well as the majority vote. In this section, we propose a one-class splitting criterion and a one-class version of the majority vote.

D.2.1 *One-class splitting criterion*

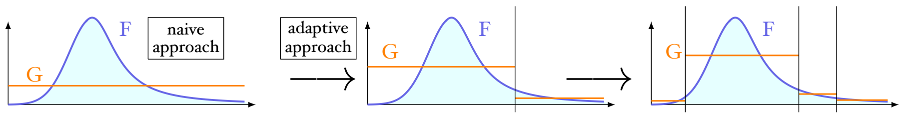

Figure D.1: Outliers distribution G in the naive and adaptive approach. In the naive approach, G does not depends on the tree and is constant on the input space. In the adaptive approach the distribution depends on the inlier distribution F through the tree. The outliers density is constant and equal to the average of F on each node before splitting it.

As one does not observe the second-class (outliers), n'_t needs to be defined. In the naive approach below, it is defined as $n'_t := n'\mathbf{Leb}(\mathcal{X}_t)/\mathbf{Leb}(\mathcal{X})$, where n' is the assumed total number of –hidden– outliers. In the adaptive approach hereafter, it is defined as $n'_t := \gamma n_t$, with typically $\gamma = 1$. Thus, the class ratio $\gamma_t := n'_t/n_t$ is well defined in both approaches and in the naive approach, goes to 0 when $\mathbf{Leb}(\mathcal{X}_t) \to 0$ while it is maintained constant to γ in the adaptive one.

D.2.1.1 *Naive approach*

A naive approach to extend the Gini splitting criterion to the one-class setting is to assume a uniform distribution for the second class (outliers), and to replace their number n'_t in node t by the expectation $n'\mathbf{Leb}(\mathcal{X}_t)/\mathbf{Leb}(\mathcal{X})$, where n' denotes the total number of outliers (for instance, it can be chosen as a proportion of the number of inliers). The problem with this approach appears when the dimension is *not small*. As mentioned in the introduction (curse of dimensionality), when actually generating n' uniform outliers on $\mathcal{X}$, the probability that a node (sufficiently small to yield a good precision) contains at least one of them is very close to zero. That is why data-dependent distributions for the outlier class are often considered [44, 137]. Taking the expectation $n'\mathbf{Leb}(\mathcal{X}_t)/\mathbf{Leb}(\mathcal{X})$ to replace the number of points in node t does not solve the curse of dimensionality mentioned in the introduction: the volume proportion $L_t := \mathbf{Leb}(\mathcal{X}_t)/\mathbf{Leb}(\mathcal{X})$ is very close to 0 for nodes t deep in the tree, especially in large dimension. In addition, we typically grow trees on sub-samples of the input data, meaning that even the root node of the trees may be very small compared to the hyper-rectangle containing all the input data. An other problem is that the Gini splitting criterion is skew-sensitive [56], and has here to be apply on nodes t with $0 \simeq n'_t \ll n_t$. When trying empirically this approach, we observe that splitting such nodes produces a child containing (almost) all the data (see appendix D.4).

Example D.1 *To illustrate the fact that the volume proportion*

$$L_t := \frac{\mathbf{Leb}(\mathcal{X}_t)}{\mathbf{Leb}(\mathcal{X})}$$

becomes very close to zero in large dimension for lots of nodes t (in particular the leaves), suppose for the sake of simplicity that the input space is $\mathcal{X} = [0,1]^d$. *Suppose that we are looking for a rough precision of* $1/2^3 = 0.125$ *in each dimension, i. e. a unit cube precision of* 2^{-3d}. *To achieve such a precision, the splitting criterion has to be used on nodes/cells t of volume of order* 2^{-3d}, *namely with* $L_t = 1/2^{3d}$. *Note that if we decide to choose* n' *to be* 2^{3d} *times larger than the number of inliers in order that* $n'L_t$ *is not negligible w. r. t. the number of inliers, the same –reversed– problem of unbalanced classes appears on nodes with small depth.*

D.2.1.2 *Adaptive approach*

Our solution is to remove the uniform assumption on the outliers, and to choose their distribution adaptively in such a way it is tightly concentrated around the inlier distribution. Formally, the idea is to maintain constant the class ratio $\gamma_t := n'_t/n_t$ on each node t: before looking for the best split, we update the number of outliers to be equal (up to a scaling constant γ) to the number of inliers, $n'_t = \gamma n_t$, i. e. $\gamma_t \equiv \gamma$. These –hidden– outliers are uniformly distributed on node t. The parameter γ is typically set to $\gamma = 1$, see suppl. Appendix D.6.1 for a discussion on the relevance of this choice (in a nutshell, γ has an influence on optimal splits).

With this methodology, one cannot derive a one-class version of the Gini index equation D.2, but we can define a one-class version of the proxy of the impurity decrease equation D.4, by simply replacing n'_{t_L} (respectively n'_{t_R}) by $n'_t\lambda_L$ (resp. $n'_t\lambda_R$), where $\lambda_L := \mathbf{Leb}(\mathcal{X}_{t_L})/\mathbf{Leb}(\mathcal{X}_t)$ and $\lambda_R := \mathbf{Leb}(\mathcal{X}_{t_R})/\mathbf{Leb}(\mathcal{X}_t)$ are the volume proportion of the two child nodes

$$I_G^{OC-ad}(t_L, t_R) = \frac{n_{t_L}\gamma n_t \lambda_L}{n_{t_L} + \gamma n_t \lambda_L} + \frac{n_{t_R}\gamma n_t \lambda_R}{n_{t_R} + \gamma n_t \lambda_R}. \tag{D.5}$$

Minimization of the one-class Gini improvement proxy equation D.5 is illustrated in figure D.2. Note that $n'_t\lambda_L$ (resp. $n'_t\lambda_R$) is the expectation of the number of uniform observations (on $\mathcal{X}_t$) among n'_t (fixed to $n'_t = \gamma n_t$) falling into the left (respectively right) node.

Choosing the split minimizing $I_G^{OC-ad}(t_L, t_R)$ at each step of the tree building process, corresponds to generating $n'_t = \gamma n_t$ outliers each time the best split has to be chosen for node t, and then using the classical two-class Gini proxy equation D.4. The only difference is that n'_{t_L} and n'_{t_R} are replaced by their expectations $n'_t\lambda_{t_L}$ and $n'_t\lambda_{t_R}$ in our method.

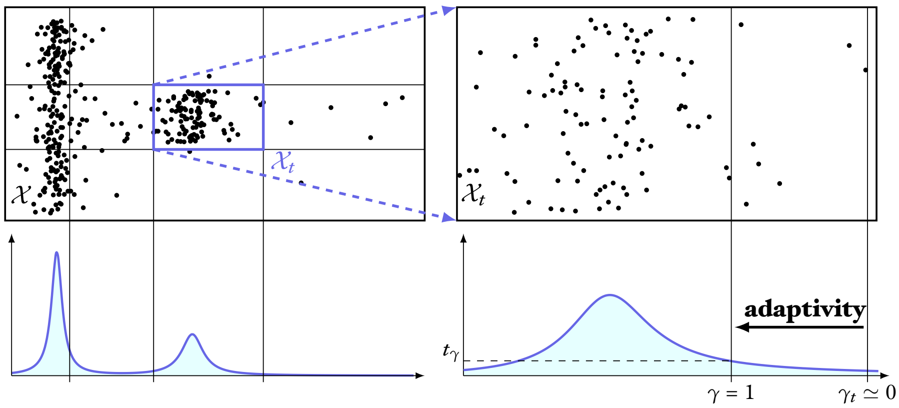

Figure D.2: The left part represents the dataset under study and the underlying density. The node $\mathcal{X}_t$ obtained after some splits is illustrated in the right part of this figure: without the proposed adaptive approach, the class ratio γ_t becomes too small and yields poor splits (all the data are in the "inlier side" of the split, which thus does not discriminate at all). Contrariwise, setting γ to one, i. e. using the adaptive approach, is far preferable.

D.2.1.3 *Resulting outlier distribution*

Figure D.1 shows the corresponding outlier density G (we drop the dependence in the number of splits to keep the notations uncluttered). Note that G is a piece-wise constant approximation of the inlier distribution F. Considering the Neyman-Pearson test $X \sim F$ versus $X \sim G$ instead of $X \sim F$ versus $X \sim \mathcal{U}$ may seem surprising at first sight. Let us try to give some intuition on why this works in practice. First, there exists (at each step) $\epsilon > 0$ such that $G > \epsilon$ on the entire input space, since the density G is constant on each node and equal to the average of F on this node *before splitting it*. If the average of F was estimated to be zero (no inlier in the node), the node would obviously not have been split, from where the existence of ϵ. Thus, at each step, one can also view G as a piece-wise approximation of $F_\epsilon := (1-\epsilon)F + \epsilon\mathcal{U}$, which is a mixture of F and the uniform distribution. (ϵ depending on the step/number of splits) Yet, one can easily show that optimal tests for the Neyman-Pearson problem $H_0 : X \sim F$ vs. $H_1 : X \sim F_\epsilon$ are identical to the optimal tests for $H_0 : X \sim F$ vs. $H_1 : X \sim \mathcal{U}$, since the corresponding likelihood ratios are related by a monotone transformation, see Scott and Blanchard [132] for instance (in fact, this reference shows that these two problems are even equivalent in terms of consistency and rates of convergence of the learning rules). An other intuitive justification is as follows. In the first step, the algorithm tries to discriminate F from $\mathcal{U}$. When going deeper in the tree, splits manage to discriminate F from a (more and more accurate) approximation of F. Asymptotically, splits become irrelevant since they are trying to discriminate F from itself (a perfect approximation, $\epsilon \to 0$).

Remark D.1 *(Consistency with the two-class framework) Consider the following method to generate outliers –tightly concentrated around the support of the inlier* *distribution. Sample uniformly* $n' = \gamma n$ *outliers on the rectangular cell containing all the inliers. Split this root node using classical two-class impurity criterion (e. g. minimizing equation D.4). Apply recursively the three following steps: for each node t, remove the potential outliers inside* $\mathcal{X}_t$*, re-sample* $n'_t = \gamma n_t$ *uniform outliers on* $\mathcal{X}_t$*, and use the latter to find the best split using Equation D.4. Then, each optimization problem equation D.4 we have solved is equivalent (in expectation) to its one-class version equation D.5. In other words, by generating outliers adaptively, we can recover (in average) a tree grown using the one-class impurity, from a tree grown using the two-class impurity.*

Remark D.2 *(Extension to other impurity criteria) Our extension to the one-class setting also applies to other impurity criteria. For instance, in the case of the Shannon entropy defined in the two-class setup by*

$$i_S(t) = \frac{n_t}{n_t + n'_t} \log_2 \frac{n_t + n'_t}{n_t} + \frac{n'_t}{n_t + n'_t} \log_2 \frac{n_t + n'_t}{n'_t},$$

the one-class impurity improvement proxy becomes

$$I_S^{OC-ad}(t_L, t_R) = n_{t_L} \log_2 \frac{n_{t_L} + \gamma n_t \lambda_L}{n_{t_L}} + n_{t_R} \log_2 \frac{n_{t_R} + \gamma n_t \lambda_R}{n_{t_R}}.$$

D.2.2 *Prediction: scoring function of the forest*

Now that RFs can be grown in the one-class setting using the one-class splitting criterion, the forest has to return a prediction adapted to this framework. In other words we also need to extend the concept of majority vote.

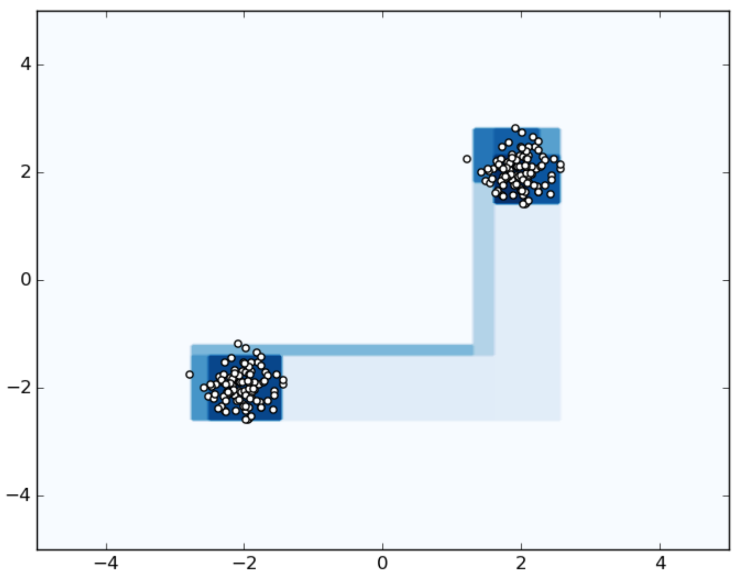

Figure D.3: OneClassRF with one tree, level-sets of the scoring function.

Most usual one-class (or more generally anomaly detection) algorithms actually provide more than just a level-set estimate or a predicted label for any new observation, abnormal versus normal. Instead, they return a real valued function, termed *scoring function*, defining a preorder/ranking on the input space. Such a function $s : \mathbb{R}^d \to \mathbb{R}$ allows to rank any observations according to their supposed "degree of abnormality". Thresholding it provides level-set estimates, as well as a decision rule that splits the input space into inlier/normal and outlier/abnormal regions. The scoring function $s(x)$ we use is the one defined in Liu, Ting, and Zhou [89] in view of its established high performance. It is a decreasing function of the average depth of the leaves containing x in the forest. An average term is added to each node containing more than one sample, say containing N samples. This term $c(N)$ is the average depth of an extremely randomized tree [62] (i. e. built without minimizing any criterion, by randomly choosing one feature and one uniform value over this feature to split on) on N samples. Formally,

$$\log_2 s(x) = -\left(\sum_{t \text{ leaves}} \mathbb{1}_{\{x \in t\}} d_t + c(n_t) \right) / c(n), \tag{D.6}$$

where d_t is the depth of node t, and $c(n) = 2H(n-1) - 2(n-1)/n$, $H(i)$ being the harmonic number. Alternative scoring functions can be defined for this one-class setting (see appendix D.6.2).

D.2.3 *OneClassRF: a Generic One-Class Random Forest algorithm*

Let us summarize the One Class Random Forest algorithm, based on generic RFs [27]. It has 6 parameters, namely `max_samples`, `max_features_tree`, `max_features_node`, `gamma`, `max_depth`, `n_trees`.

Each tree is classically grown on a random subset of both the input samples and the input features [68, 111]. This random subset is a sub-sample of size `max_samples`, with `max_features_tree` variables chosen at random without replacement (replacement is only done after the tree is grown). The tree is built by minimizing equation D.5 for each split, using parameter γ (recall that $n'_t := \gamma n_t$), until either the maximal depth `max_depth` is achieved or the node contains only one point. Minimizing equation D.5 is done as introduced in Amit and Geman [5]: at each node, we search the best split over a random selection of features with fixed size *maxfeaturesnode*. The forest is composed of a number *ntrees* of trees. The predicted score of a point x is given by $s(x)$, with s defined by Equation D.6. Remarks on alternative stopping criteria and variable importances are available in appendix D.6.3.

Figure D.3 represents the level sets of the scoring function produced by OneClassRF, with only one tree of maximal depth 4, without sub-

Table D.1: Original datasets characteristics

Datasets	nb of samples	nb of features	anomaly class	
adult	48842	6	class '> $50K$'	(23.9%)
annthyroid	7200	6	classes $\neq$ 3	(7.42%)
arrhythmia	452	164	classes $\neq$ 1 (features 10-14 removed)	(45.8%)
forestcover	286048	10	class 4 (versus class 2)	(0.96%)
http	567498	3	attack	(0.39%)
ionosphere	351	32	bad	(35.9%)
pendigits	10992	16	class 4	(10.4%)
pima	768	8	pos (class 1)	(34.9%)
shuttle	85849	9	classes $\neq$ 1 (class 4 removed)	(7.17%)
smtp	95156	3	attack	(0.03%)
spambase	4601	57	spam	(39.4%)
wilt	4839	5	class 'w' (diseased trees)	(5.39%)

sampling, and using the Gini-based one-class splitting criterion with $\gamma = 1$.

D.3 BENCHMARKS

In this section, we compare the OneClassRF algorithm described above to seven state-of-art anomaly detection algorithms: the IForest algorithm [89], a one-class RFs algorithm based on sampling a second class OCRF-sampling [44], One-Class Support Vector Machine (OCSM) [129], Local Outlier Factor (LOF) [28], Orca [15], Least Squares Anomaly Detection (LSAD) [120], Random Forest Clustering (RFC) [137].

D.3.1 *Default parameters of OneClassRF*

The default parameters taken for our algorithm are the followings.

- `max_samples` is fixed to 20% of the training sample size (with a minimum of 100);
- `max_features_tree` is fixed to 50% of the total number of features with a minimum of 5 (i. e. each tree is built on 50% of the total number of features);
- `max_features_node` is fixed to 5;
- γ is fixed to 1;

- `max_depth` is fixed to $\log_2$ (logarithm in base 2) of the training sample size as in Liu, Ting, and Zhou [89];
- `n_trees` is fixed to 100 as in the previous reference.

The other algorithms in the benchmark are trained with their recommended (default) hyper-parameters as seen in their respective paper or author's implementation. See appendix D.7 for details. The characteristics of the twelve reference datasets considered here are summarized in table D.1. They are all available on the UCI repository [84] and the preprocessing is done as usually in the literature (see appendix D.8).

D.3.2 *Results*

All the code is available at `https://github.com/ngoix/OCRF`. The experiments are performed in the novelty detection framework, where the training set consists of inliers only. No significance level test are given, but experiements or each algorithm are repeated 10 times on random training and testing datasets are performed, yielding averaged ROC and PR curves whose AUCs are summarized in table D.3 (higher is better). The training time of each algorithm has been limited (for each experiment among the 10 performed for each dataset) to 30 minutes, where N. A. indicates that the algorithm could not finish training within the allowed time limit. In average on all the datasets, our proposed algorithm OneClassRF achieves both best AUC ROC and AUC PR scores (with LSAD for AUC ROC). It also achieves the lowest cumulative training time. For further insights on the benchmarks c. f. appendix D.6. It appears that OneClassRF has the best performance on five datasets in terms of ROC AUCs, and is also the best in average. Computation times (training plus testing) of OneClassRF are also very competitive. Experiments in an outlier detection framework (the training set is polluted by outliers) have also been made (see appendix D.9). The anomaly rate is arbitrarily bounded to 10% max (before splitting data into training and testing sets).

D.4 THEORETICAL ANALYSIS

This section aims at recovering equation D.5 from a natural modeling of the one-class framework, along with a theoretical study of the problem raised by the naive approach.

D.4.1 *Underlying model*

In order to generalize the two-class framework to the one-class one, we need to consider the population versions associated to empirical quantities equation D.1, equation D.2 and equation D.3, as well as the underlying model assumption. The latter can be described as follows.

Table D.3: Results for the novelty detection setting.

Datasets	OneClassRF		IForest		OCRFsampling		OCSM		LOF		Orca		LSAD		RFC	
AUC	ROC	PR	ROC	PR	ROC	PR	ROC	PR	ROC	PR	ROC	PR	ROC	PR	ROC	PR
adult	**0.665**	**0.278**	0.661	0.227	N. A.	N. A.	0.638	0.201	0.615	0.188	0.606	0.218	0.647	0.258	N. A.	N. A.
annthyroid	**0.936**	0.468	0.913	0.456	0.918	**0.532**	0.706	0.242	0.832	0.446	0.587	0.181	0.810	0.327	N. A.	N. A.
arrhythmia	0.684	0.510	0.763	0.492	0.639	0.249	**0.922**	**0.639**	0.761	0.473	0.720	0.466	0.778	0.514	0.716	0.299
forestcover	0.968	0.457	0.863	0.046	N. A.	N. A.	N. A.	N. A.	**0.990**	**0.795**	0.946	0.558	0.952	0.166	N. A.	N. A.
http	**0.999**	**0.838**	0.994	0.197	N. A.	N. A.	N. A.	N. A.	N. A.	N. A.	**0.999**	0.812	0.981	0.537	N. A.	N. A.
ionosphere	0.909	0.643	0.902	0.535	0.859	0.609	0.973	0.849	0.959	0.807	0.928	**0.910**	**0.978**	0.893	0.950	0.754
pendigits	0.960	0.559	0.810	0.197	0.968	0.694	0.603	0.110	0.983	0.827	**0.993**	**0.925**	0.983	0.752	N. A.	N. A.
pima	0.719	0.247	0.726	0.183	**0.759**	**0.266**	0.716	0.237	0.700	0.152	0.588	0.175	0.713	0.216	0.506	0.090
shuttle	**0.999**	**0.998**	0.996	0.973	N. A.	N. A.	0.992	0.924	**0.999**	0.995	0.890	0.782	0.996	0.956	N. A.	N. A.
smtp	0.922	0.499	0.907	0.005	N. A.	N. A.	0.881	**0.656**	**0.924**	0.149	0.782	0.142	0.877	0.381	N. A.	N. A.
spambase	**0.850**	0.373	0.824	0.372	0.797	**0.485**	0.737	0.208	0.746	0.160	0.631	0.252	0.806	0.330	0.723	0.151
wilt	0.593	0.070	0.491	0.045	0.442	0.038	0.323	0.036	0.697	0.092	0.441	0.030	0.677	0.074	**0.896**	**0.631**
average	**0.850**	**0.495**	0.821	0.311	0.769	0.410	0.749	0.410	0.837	0.462	0.759	0.454	**0.850**	0.450	0.758	0.385
cum. train time	**61s**		68s		N. A.		N. A.		N. A.		2232s		73s		N. A.	

D.4.1.1 *Existing Two-Class Model (n, α).*

We consider a R. V. $X : \Omega \to \mathbb{R}^d$ w. r. t. a probability space $(\Omega, \mathcal{F}, \mathbf{Pr})$. The law of X depends on another R. V. $y \in \{0, 1\}$, verifying $\mathbf{Pr}\{\, y = 1 \,\} = 1 - \mathbf{Pr}\{\, y = 0 \,\} = \alpha$. We assume that conditionally on $y = 0$, X follows a law F, and conditionally on $y = 1$ a law G;

$$\begin{array}{llll} X \mid y = 0 & \sim & F, & \mathbf{Pr}\{\, y = 0 \,\} = 1 - \alpha, \\ X \mid y = 1 & \sim & G, & \mathbf{Pr}\{\, y = 1 \,\} = \alpha. \end{array}$$

Then, considering

$$p(t_L|t) = \mathbf{Pr}\{\, X \in \mathcal{X}_{t_L} \mid X \in \mathcal{X}_t \,\},$$

and

$$p(t_R|t) = \mathbf{Pr}\{\, X \in \mathcal{X}_{t_R} \mid X \in \mathcal{X}_t \,\},$$

the population version (probabilistic version) of Equation D.1 is

$$\Delta i^{theo}(t, t_L, t_R) = i^{theo}(t) - p(t_L|t) i^{theo}(t_L) - p(t_R|t) i^{theo}(t_R). \tag{D.7}$$

It can be used with the Gini index i_G^{theo},

$$i_G^{theo}(t) = 2\mathbf{Pr}\{\, y = 0 \mid X \in \mathcal{X}_t \,\}\, \mathbf{Pr}\{\, y = 1 \mid X \in \mathcal{X}_t \,\} \tag{D.8}$$

which is the population version of equation D.2.

D.4.1.2 *One-Class-Model (n, α).*

We model the one-class framework as follows. Among the n i. i. d. observations, we only observe those with $y = 0$ (the inliers), namely N realizations of $(X \mid y = 0)$, where N is itself a realization of a R. V. $\mathbf{N}$ of law $\mathbf{N} \sim \text{Bin}(n, (1 - \alpha))$. Here and hereafter, $\text{Bin}(n, p)$ denotes the binomial distribution with parameters (n, p). As outliers are not observed, it is natural to assume that G follows a uniform distribution on the hyper-rectangle $\mathcal{X}$ containing all the observations, so that G has a constant density $g(\cdot) = 1/\mathbf{Leb}(\mathcal{X})$ on $\mathcal{X}$. Note that this assumption *will be removed* in the adaptive approach described below – which aims at maintaining a non-negligible proportion of (hidden) outliers in every nodes.

Let us define $L_t = \mathbf{Leb}(\mathcal{X}_t)/\mathbf{Leb}(\mathcal{X})$. Then, $\mathbf{Pr}\{\, X \in \mathcal{X}_t \mid y = 1 \,\} = \mathbf{Pr}\{\, y = 1 \,\}\, \mathbf{Pr}\{\, X \in \mathcal{X}_t \mid y = 1 \,\} = \alpha L_t$. Replacing $\mathbf{Pr}\{\, X \in \mathcal{X}_t \mid y = 0 \,\}$ by its empirical version n_t/n in equation D.8, we obtain the one-class empirical Gini index

$$i_G^{OC}(t) = \frac{n_t \alpha n L_t}{(n_t + \alpha n L_t)^2}. \tag{D.9}$$

This one-class index can be seen as a *semi-empirical* version of equation D.8, in the sense that it is obtained by considering empirical quantities for the (observed) inlier behavior and population quantities for the (non-observed) outlier behavior. Now, maximizing the population version of the impurity decrease $\Delta i_G^{theo}(t, t_L, t_R)$ as defined in equation D.7 is equivalent to minimizing

$$p(t_L|t) i_G^{theo}(t_L) + p(t_R|t) i_G^{theo}(t_R). \tag{D.10}$$

Considering semi-empirical versions of $p(t_L|t)$ and $p(t_R|t)$, as for Equation D.9, gives $p_n(t_L|t) = (n_{t_L} + \alpha n L_{t_L})/(n_t + \alpha n L_t)$ and $p_n(t_R|t) = (n_{t_R} + \alpha n L_{t_R})/(n_t + \alpha n L_t)$. Then, the semi-empirical version of Equation D.10 is

$$p_n(t_L|t) i_G^{OC}(t_L) + p_n(t_R|t) i_G^{OC}(t_R) = \frac{1}{(n_t + \alpha n L_t)} \left(\frac{n_{t_L} \alpha n L_{t_L}}{n_{t_L} + \alpha n L_{t_L}} + \frac{n_{t_R} \alpha n L_{t_R}}{n_{t_R} + \alpha n L_{t_R}} \right) \tag{D.11}$$

where $1/(n_t + \alpha n L_t)$ is constant when the split varies. This means that finding the split minimizing equation D.11 is equivalent to finding the split minimizing

$$I_G^{OC}(t_L, t_R) = \frac{n_{t_L} \alpha n L_{t_L}}{n_{t_L} + \alpha n L_{t_L}} + \frac{n_{t_R} \alpha n L_{t_R}}{n_{t_R} + \alpha n L_{t_R}}. \tag{D.12}$$

Note that equation D.12 can be obtained from the two-class impurity decrease equation D.4 as described in the naive approach paragraph in appendix D.2. In other words, it is the naive one-class version of equation D.4.

Remark D.3 (Direct link with the two-class framework). *The two-class proxy of the Gini impurity decrease equation D.4 is recovered from equation D.12 by replacing $\alpha n L_{t_L}$ (resp. $\alpha n L_{t_R}$) by n'_{t_L} (respectively n'_{t_R}), the number of second class instances in t_L (respectively in t_R). When generating αn of them uniformly on $\mathcal{X}$, $\alpha n L_t$ is the expectation of n'_t.*

As detailed in appendix D.2.1, this approach suffers from the curse of dimensionality. We can summarize the problem as follows. Note that when setting $n'_t := \alpha n L_t$, the class ratio $\gamma_t = n'_t / n_t$ is then equal to

$$\gamma_t = \alpha n L_t / n_t. \tag{D.13}$$

This class ratio is close to 0 for lots of nodes t, which makes the Gini criterion unable to discriminate accurately between the –hidden– outliers and the inliers. Minimizing this criterion produces splits corresponding to $\gamma_t \simeq 0$ in figure D.2: one of the two child nodes, say t_L contains almost all the data.

D.4.2 *Adaptive approach*

The solution presented appendix D.2 is to remove the uniform assumption for the outlier class. From the theoretical point of view, the idea is to choose in an adaptive way (w. r. t. the volume of $\mathcal{X}_t$) the number αn, which can be interpreted as the number of (hidden) outliers. α). Doing so, we aim at avoiding $\alpha n L_t \ll n_t$ when L_t is too small. Namely, with γ_t defined in equation D.13, we aim at avoiding $\gamma_t \simeq 0$ when $L_t \simeq 0$. The idea is to consider $\alpha(L_t)$ and $n(L_t)$ such that $\alpha(L_t) \to 1$, $n(L_t) \to \infty$ when $L_t \to 0$. We then define the one-class adaptive proxy of the impurity decrease by

$$\begin{aligned} I_G^{OC-ad}(t_L, t_R) = & \frac{n_{t_L} \alpha(L_t) n(L_t) L_{t_L}}{n_{t_L} + \alpha(L_t) n(L_t) L_{t_L}} \\ & + \frac{n_{t_R} \alpha(L_t) n(L_t) L_{t_R}}{n_{t_R} + \alpha(L_t) \cdot n(L_t) \cdot L_{t_R}}. \end{aligned} \tag{D.14}$$

In other words, instead of considering one general model One-Class-Model(n, α) defined in appendix D.4.1, we adapt it to each node t, considering One-Class-Model($n(L_t)$, $\alpha(L_t)$) *before searching the best split*. We still consider the N inliers as a realization of this model. When growing the tree, using One-Class-Model($n(L_t)$, $\alpha(L_t)$) allows to maintain a non-negligible expected proportion of outliers in the node to be split, despite L_t becomes close to zero. Of course, constraints have to be imposed to ensure consistency between these models. Recalling that the number N of inliers is a realization of $\mathbf{N}$ following a Binomial distribution with parameters $(n, 1-\alpha)$, a first natural constraint on $(n(L_t), \alpha(L_t))$ is

$$(1-\alpha)n = (1-\alpha(L_t)) \cdot n(L_t), \quad \text{for all } t, \tag{D.15}$$

so that the expectation of $\mathbf{N}$ remains unchanged.

Remark D.4 *In our adaptive model One-Class-Model($n(L_t)$, $\alpha(L_t)$) which varies when we grow the tree, let us denote by $\mathbf{N}(L_t) \sim Bin\,(n(L_t), 1-\alpha(L_t))$ the R. V. ruling the number of inliers. The number of inliers N is still viewed as a realization of it. Note that the distribution of $\mathbf{N}(L_t)$ converges in distribution to $\mathcal{P}\,((1-\alpha)n)$ a Poisson distribution with parameter $(1-\alpha)n$ when $L_t \to 0$, while the distribution $Bin\,(n(L_t), \alpha(L_t))$ of the R. V. $n(L_t) - \mathbf{N}(L_t)$ ruling the number of (hidden) outliers goes to infinity almost surely. In other words, the asymptotic model (when $L_t \to 0$) consists in assuming that the number of inliers N we observed is a realization of $\mathbf{N}_\infty \sim \mathcal{P}\,((1-\alpha)n)$, and that an infinite number of outliers have been hidden.*

A second natural constraint on $(\alpha(L_t), n(L_t))$ is related to the class ratio γ_t. As explained in appendix D.2.1, we do not want γ_t to go to zero when L_t does. Let us say we want γ_t to be constant for all node t, equal to $\gamma > 0$. From the constraint $\gamma_t = \gamma$ and equation D.13, we get

$$\alpha(L_t) n(L_t) L_t = \gamma n_t := n_t'. \tag{D.16}$$

The constant γ is a parameter ruling the expected proportion of outliers in each node. Typically, $\gamma = 1$ so that there is as much expected uniform (hidden) outliers than inliers at each time we want to find the best split minimizing equation D.14. Equation D.15 and equation D.16 allow to explicitly determine $\alpha(L_t)$ and $n(L_t)$: $\alpha(L_t) = n_t'/((1-\alpha)nL_t + n_t')$ and $n(L_t) = ((1-\alpha)nL_t + n_t')/L_t$. Regarding equation D.14, $\alpha(L_t)n(L_t)L_{t_L} = \frac{n_t'}{L_t} L_{t_L} = n_t' \frac{\mathrm{Leb}(\mathcal{X}_{t_L})}{\mathrm{Leb}(\mathcal{X}_t)}$ by equation D.16 and $\alpha(L_t)n(L_t)L_{t_R} = n_t' \frac{\mathrm{Leb}(\mathcal{X}_{t_R})}{\mathrm{Leb}(\mathcal{X}_t)}$, so that we recover equation D.5.

D.5 CONCLUSION

Through a natural adaptation of both (two-class) splitting criteria and majority vote, this paper introduces a methodology to structurally extend RFs to the one-class setting. Our one-class splitting criteria correspond to the asymptotic behavior of an adaptive outliers generating methodology, so that consistency with two-class RFs seems respected. While no statistical guaranties have been derived in this paper, a strong empirical performance attests the relevance of this methodology.

D.6 FURTHER INSIGHTS ON THE ALGORITHM

D.6.1 *Interpretation of parameter gamma*

In order for the splitting criterion equation D.5 to perform well, n_t' is expected to be of the same order of magnitude as the number of inliers n_t. If $\gamma = n_t'/n_t \ll 1$, the split puts every inliers on the same side, even the ones which are far in the tail of the distribution, thus widely over-estimating the support of inliers. If $\gamma \gg 1$, the opposite effect happens, yielding an estimate of a t-level set with t not close enough to 0. Figure D.2 illustrates the splitting criterion when γ varies. It clearly shows that there is a link between parameter γ and the level t_γ of the induced level-set estimate. But from the theory, an explicit relation between γ and t_γ is hard to derive. By default we set γ to 1. One could object that in some situations, it is useful to randomize this parameter. For instance, in the case of a bi-modal distribution for the inlier/normal behavior, one split of the tree needs to separate two clusters, in order for the level set estimate to distinguish between the two modes. As illustrated in figure D.4, it can only occur if n_t' is large with respect to n_t ($\gamma \gg 1$). However, the randomization of γ is somehow included in the randomization of each tree, thanks to the sub-sampling inherent to RFs. Moreover, small clusters tend to vanish when the sub-sample size is sufficiently small: a small sub-sampling size is used by Liu, Ting, and Zhou [89] to isolate outliers even when they form clusters.

D.6.2 *Alternative scoring functions*

Although we use the scoring function defined in equation D.6 because of its established high performance [89], other scoring functions can be defined. A natural idea to adapt the majority vote to the one-class setting is to change the single vote of a leaf node t into the fraction $\frac{n_t}{\mathbf{Leb}(\mathcal{X}_t)}$, the forest output being the average of the latter quantity over the forest, $s(x) = \sum_{t \text{ leaves}} \mathbb{1}_{\{x \in t\}} \frac{n_t}{\mathbf{Leb}(\mathcal{X}_t)}$. In such a case, each tree of the forest yields a piece-wise density estimate on its induced partition. The output produced by the forest is then a *step-wise density estimate*. We could also think about the *local density of a typical cell*. For each point x of the input space, it returns the average number of observations in the leaves containing x, divided by the average volume of such leaves. The output of OneClassRF is then the scoring function $s(x) = \left(\sum_{t \text{ leaves}} \mathbb{1}_{\{x \in t\}} n_t\right) \left(\sum_{t \text{ leaves}} \mathbb{1}_{\{x \in t\}} \mathbf{Leb}(\mathcal{X}_t)\right)^{-1}$, where the sums are over each leave of each tree in the forest. This score can be interpreted as the local density of a "typical" cell (typical among those usually containing x).

D.6.3 *Alternative stopping criteria*

Other stopping criteria than a maximal depth may be considered. We could stop splitting a node t when it contains less than `n_min` observations, or when the quantity $n_t/\mathbf{Leb}(\mathcal{X}_t)$ is large enough (all the points in the cell $\mathcal{X}_t$ are likely to be inliers) or close enough to 0 (all the points in the cell $\mathcal{X}_t$ are likely to be outliers). These options are not discussed in this work.

D.6.4 *Variable importance*

In the multiclass setting, Breiman [27] proposed to evaluate the importance of a feature $j \in \{1, \ldots d\}$ for prediction by adding up the weighted impurity decreases for all nodes t where X_j is used, averaged over all the trees. The analogue quantity can be computed with respect to the one-class impurity decrease proxy. In our one-class setting, this quantity represents the size of the tail of X_j, and can be interpreted as the capacity of feature j to discriminate between inliers/outliers.

D.7 HYPER-PARAMETERS OF TESTED ALGORITHMS

Overall we chose to train the different algorithms with their (default) hyper-parameters as seen in their respective paper or author's implementation. Indeed, since we are in an unsupervised setting, there is no trivial way to select/learn the hyperparameters of the different algorithm in the training phase – the labels are not supposed to be available. Hence the more realistic way to test the algorithms is to use their recommended/default hyperparameters.

The OCSM algorithm uses default parameters: `kernel='rbf'` with `tol=1e-3`, `nu=0.5`, `shrinking=True` and `gamma=1/n_features`, where tol is the tolerance for stopping criterion.

The LOF algorithm uses default parameters: `n_neighbors=5` with the `leaf_size=30` and `metric='minkowski'` and `contamination=0.1` and `algorithm='auto'`, where the algorithm parameters stipulates how to compute the nearest neighbors (either ball-tree, kd-tree or brute-force).

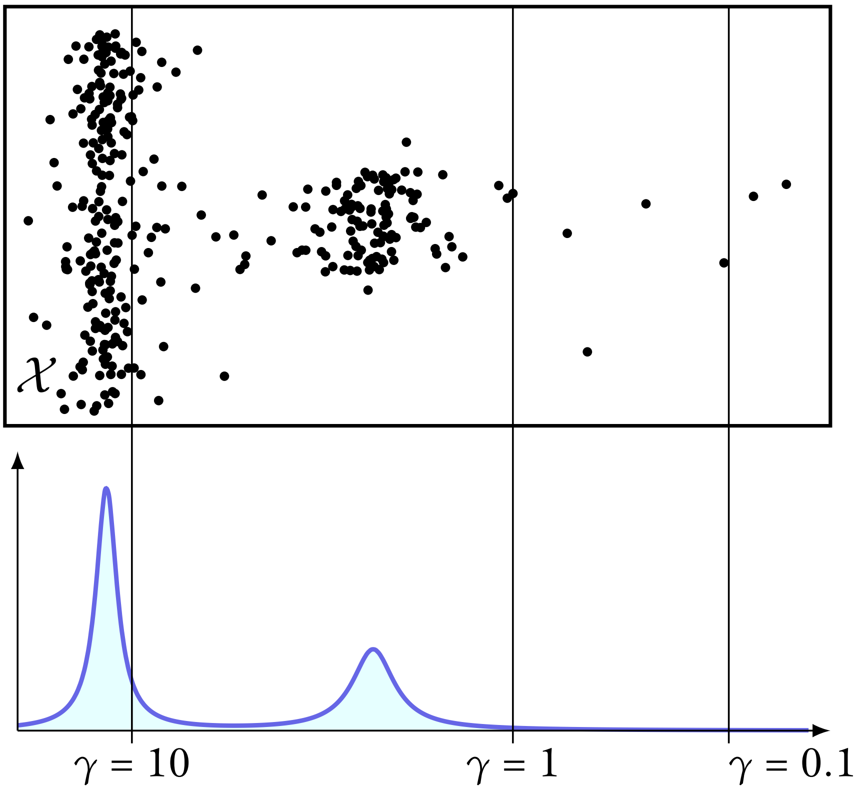

Figure D.4: Illustration of the standard splitting criterion on two modes when the proportion γ varies.

The IFOREST algorithm uses default parameters: `n_estimators=100` and `max_samples=min(256, n_samples)` and `max_features=1` and setting `bootstrap=false`, where bootstrap states whether samples are drawn with replacement.

The OCRFSAMPLING algorithm uses default parameters: the number of dimensions for the Random Subspace Method `krsm=-1`, the number of features randomly selected at each node during the induction of the tree `krfs=-1`, `n_tree=100`, the factor controlling the extension of the outlier domain used to sample outliers according to the volume of the hyper-box surrounding the target data `alpha=1.2`, the factor controlling the number of outlier data generated according to the number of target data `beta=10`, whether outliers are generated from uniform distribution `optimize=0` and eventually whether data outside target bounds are considered as outlier data `rejectOutOfBounds=0`.

The *Orca* algorithm uses default parameter `k=5` (number of nearest neighbors) as well as `N=n/8` (how many anomalies are to be reported). The last setting, set up in the empirical evaluation of iForest in Liu, Ting, and Zhou [90], allows a better computation time without impacting Orca's performance.

The RFC algorithm uses default parameters: `no.forests=25` with the number of trees `no.trees=3000`, the Addcl1 Random Forest dissimilarity `addcl1=T`, `addcl2=F` use the importance measure `imp=T`, the data generating process `oob.prox1=T`, the number of features sampled at each split `mtry1=3`.

The LSAD algorithm uses default parameters: the maximum number of samples per kernel `n_kernels_max=500`, the center of each kernel (the center of the random sample subset by default) `kernel_pos='None'`, the kernel scale parameter (using the pairwise median trick by default [71]) `gamma='None'`, the regularization parameter `rho=0.1`.

D.8 DESCRIPTION OF THE DATASETS

The characteristics of the twelve reference datasets considered here are summarized in table D.1. They are all available on the UCI repository [84] and the preprocessing is done in a classical way. In anomaly detection, we typically have data from two class (inliers/outliers) – in novelty detection, the second class is unavailable in training in outlier detection, training data are polluted by second class (anonymous) examples. The classical approach to adapt multi-class data to this framework is to set classes forming the outlier class, while the other classes form the inlier class.

We removed all categorial attributes. Indeed, our method is designed to handle data whose distribution is absolutely continuous w. r. t. the Lebesgue measure. The *http* and *smtp* datasets belong to the KDD Cup '99 dataset [75, 150], which consist of a wide variety of hand-injected attacks (anomalies) in a closed network (normal/inlier background). They are classically obtained as described in Yamanishi et al. [167]. This two datasets are available on the *scikit-learn* library [114]. The *shuttle* dataset is the fusion of the training and testing datasets available in the UCI repository. As in Liu, Ting, and Zhou [89], we use instances from all different classes but class 4. In the *forestcover* data, the inliers are the instances from class 2 while instances from class 4 are anomalies (as in Liu, Ting, and Zhou [89]). The *ionosphere* dataset differentiates "good" from "bad" radars, considered here as abnormal. A "good" radar shows evidence of some type of structure in the ionosphere. A "bad" radar does not, its signal passing through the ionosphere. The *spambase* dataset consists of spam or non-spam emails. The former constitute our anomaly class. The *annthyroid* medical dataset on hypothyroidism contains one normal class and two abnormal ones, which form our outliers. The *arrhythmia* dataset reflects the presence and absence (class 1) of cardiac arrhythmia. The number of attributes being large considering the sample size, we removed attributes containing missing data. Besides, we removed attributes taking less than 10 different values, the latter breaking too strongly our absolutely continuous assumption (w. r. t. to **Leb**). The *pendigits* dataset contains 10 classes corresponding to the digits from 0 to 9, examples being handwriting samples. As in Schubert et al. [130], the outliers are chosen to be those from class 4. The *pima* dataset consists of medical data on diabetes. Patients suffering from diabetes (inlier class) were considered outliers. The *wild* dataset involves detecting diseased trees in Quickbird imagery. Diseased trees (class 'w') is our outlier class. In the *adult* dataset, the goal is to predict whether income exceeds $ 50K/year based on census data. We only keep the 6 continuous attributes.

D.9 FURTHER DETAILS ON BENCHMARKS AND OUTLIER DETECTION RESULTS

Figure D.5 shows that the amount of time to train[1] and test any dataset takes less than one minute with OneClassRF, whereas some algorithms have far higher computation times (OCRFsampling, OCSM, LOF and Orca have computation times higher than 30 minutes in some datasets). Our approach yields results similar to quite new algorithms such as IForest and LSAD.

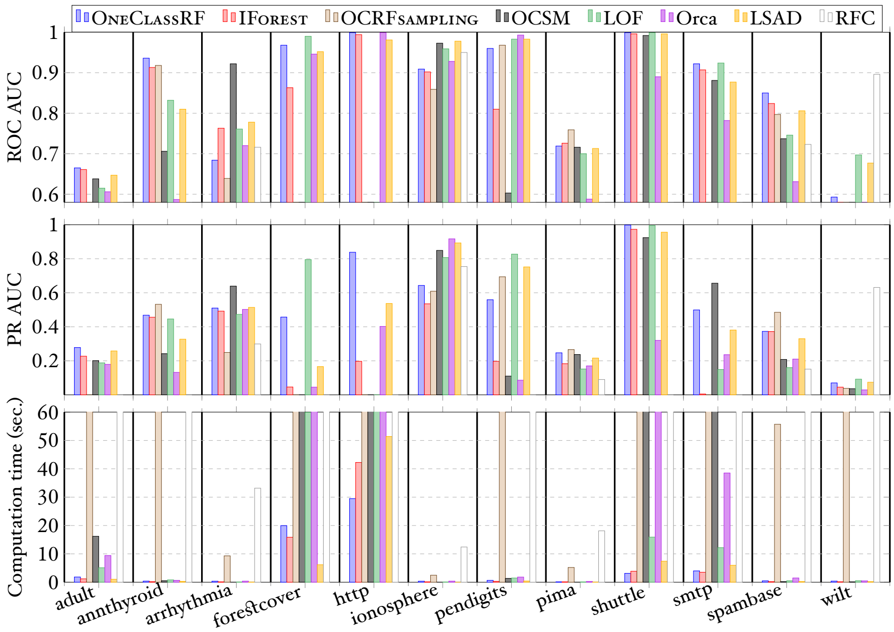

Figure D.5: Performances of the algorithms on each dataset in the novelty detection framework: ROC AUCs are displayed on the top, PR AUCs in the middle and training times on the bottom, for each dataset and algorithm. The x-axis represents the datasets.

In this section present experiments in the outlier detections setting. For each algorithm, 10 experiments on random training and testing datasets are performed. Averaged ROC and PR curves AUC are summarized in ta-

1 For OneClassRF, Orca and RFC, testing and training time cannot be isolated because of algorithms implementation: for these algorithms, the sum of the training and testing times are displayed in Figure D.5 and Figure D.6.

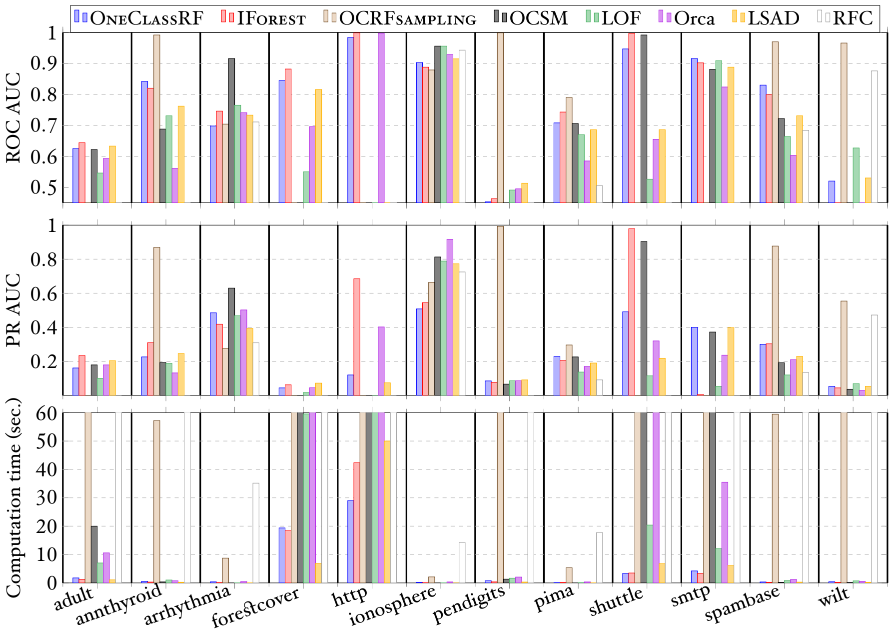

Figure D.6: Performances of the algorithms on each dataset in the outlier detection framework: ROC AUCs are on the top, PR AUCs in the middle and processing times are displayed below (for each dataset and algorithm). The x-axis represents the datasets.

ble D.4. For the experiments made in an unsupervised framework (meaning that the training set is polluted by outliers), the anomaly rate is arbitrarily bounded to 10% max (before splitting data into training and testing sets).

Figure D.7: ROC and PR curves for OneClassRF (novelty detection framework)

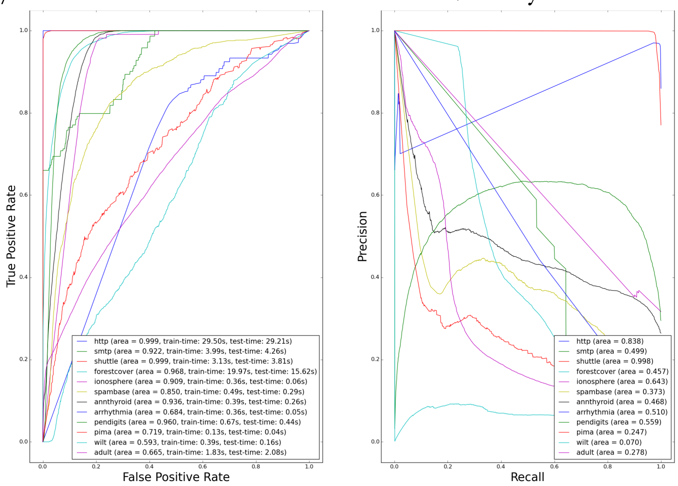

Figure D.8: ROC and PR curves for OneClassRF (outlier detection framework)

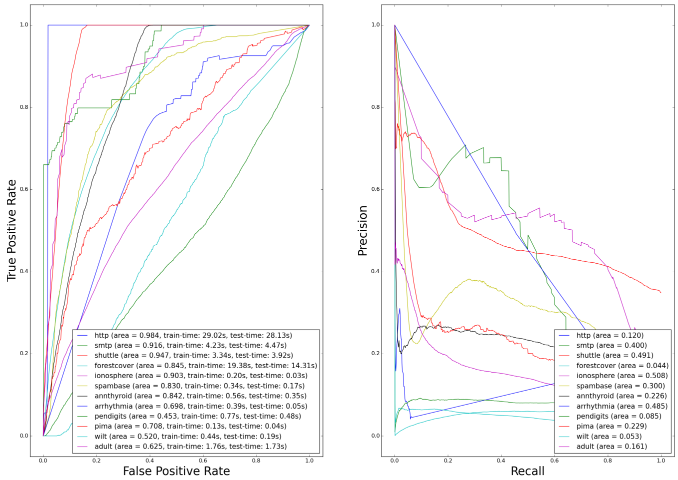

Figure D.9: ROC and PR curves for IForest (novelty detection framework)

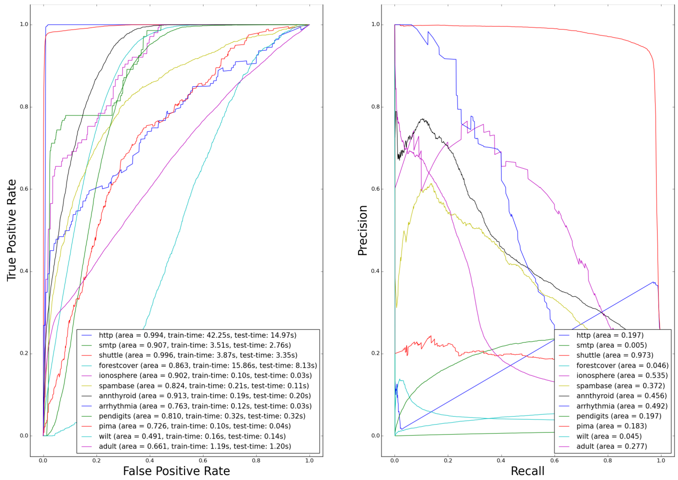

Figure D.10: ROC and PR curves for IForest (outlier detection framework)

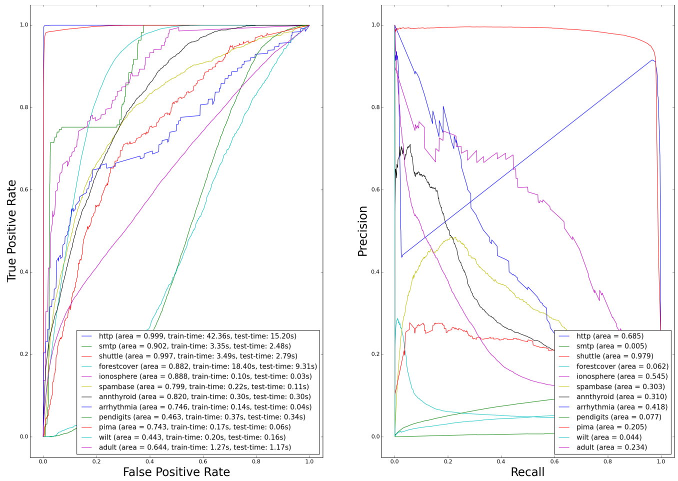

Figure D.11: ROC and PR curves for OCRFsampling (novelty detection framework)

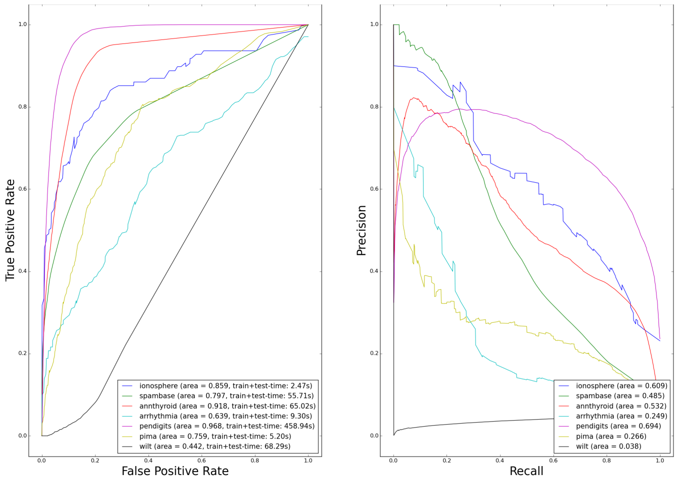

Figure D.12: ROC and PR curves for OCRFsampling (outlier detection framework)

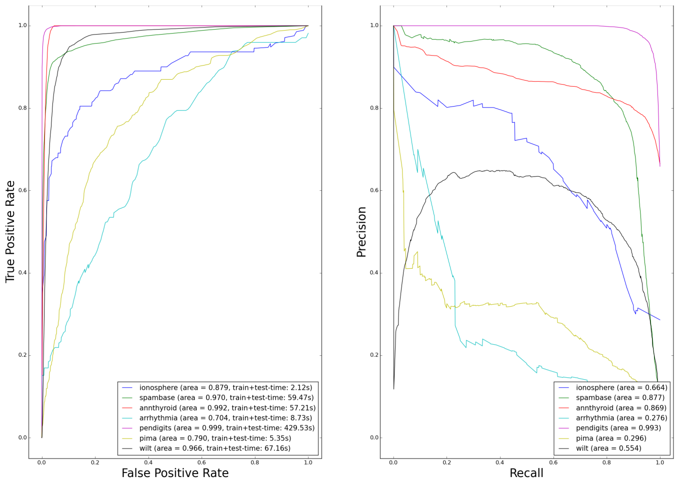

Figure D.13: ROC and PR curves for OCSM (novelty detection framework)

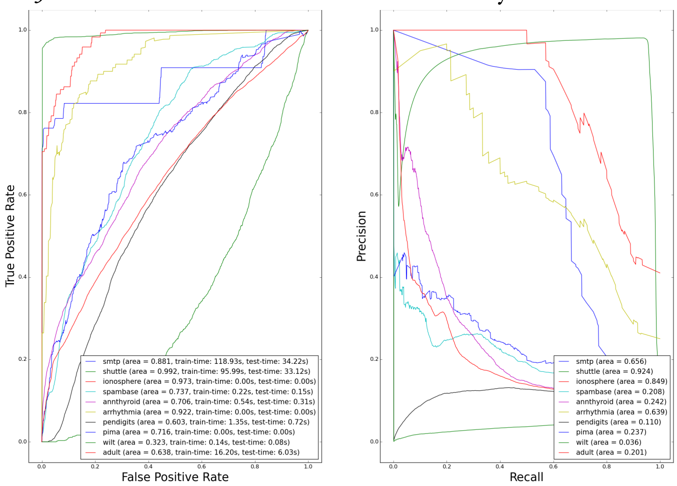

Figure D.14: ROC and PR curves for OCSM (outlier detection framework)

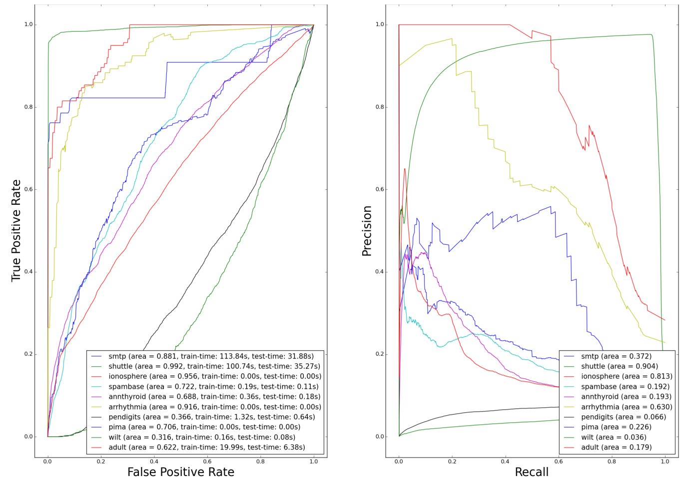

Figure D.15: ROC and PR curves for LOF (novelty detection framework)

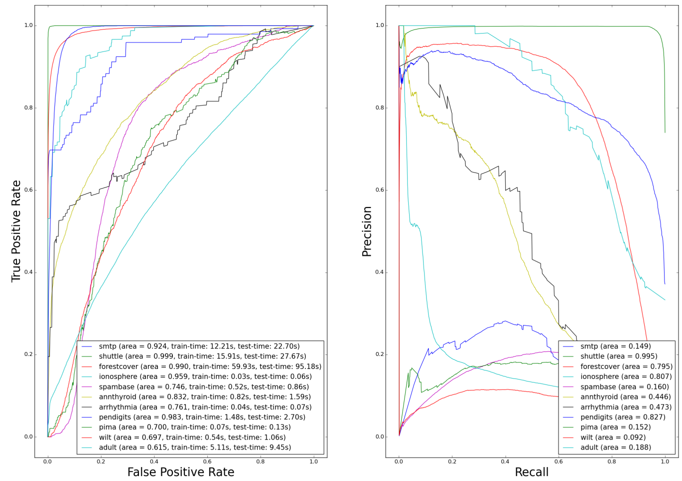

Figure D.16: ROC and PR curves for LOF (outlier detection framework)

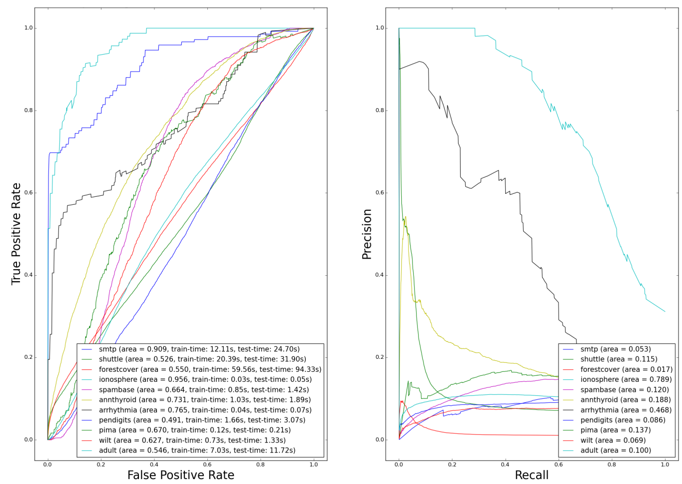

Figure D.17: ROC and PR curves for Orca (novelty detection framework)

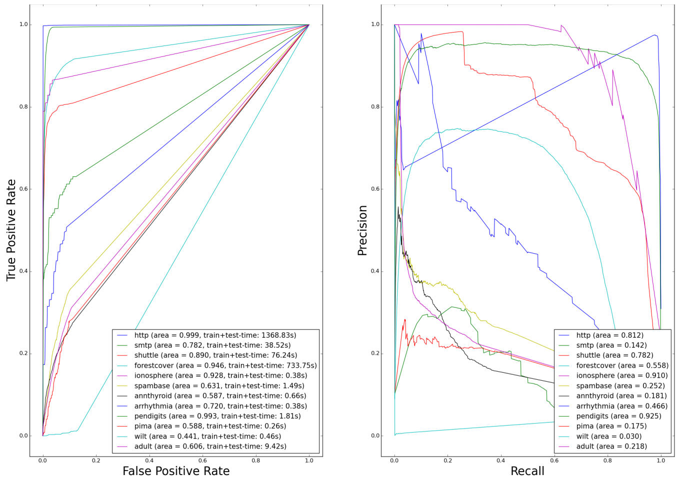

Figure D.18: ROC and PR curves for Orca (outlier detection framework)

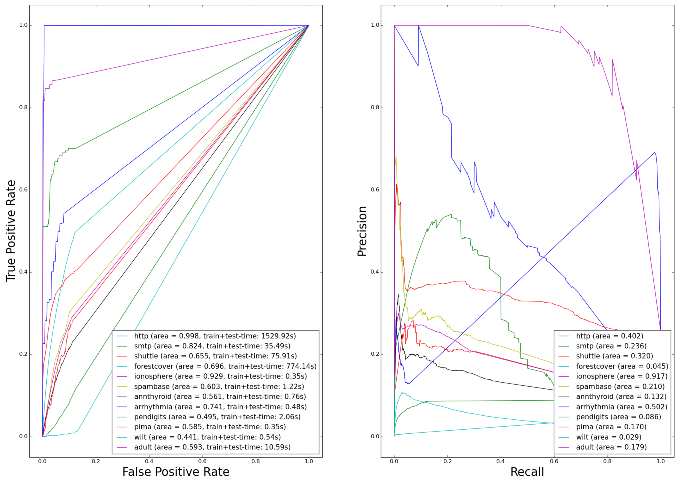

Figure D.19: ROC and PR curves for LSAD (novelty detection framework)

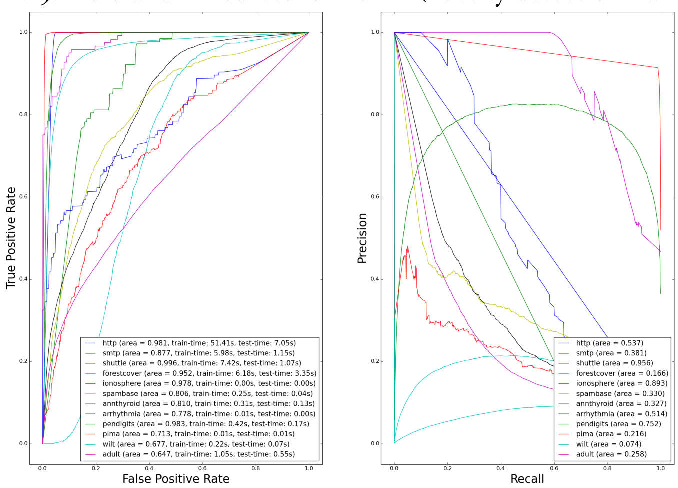

Figure D.20: ROC and PR curves for LSAD (outlier detection framework)

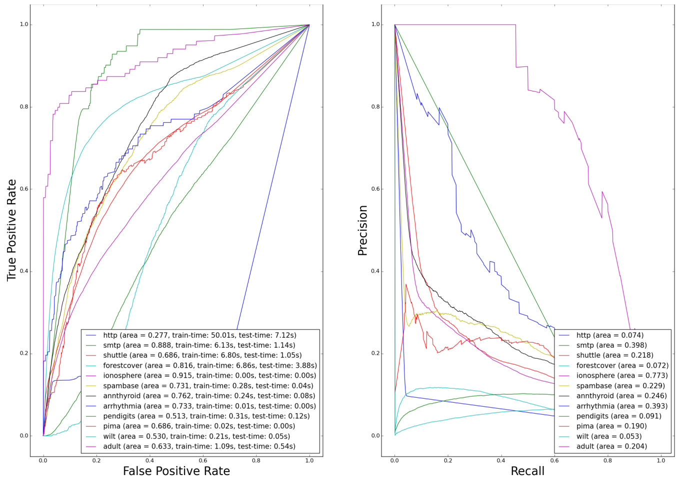

Figure D.21: ROC and PR curves for RFC (novelty detection framework)

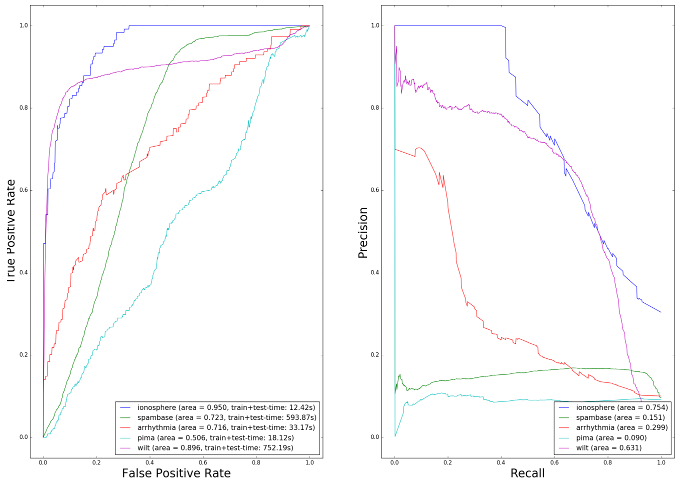

Figure D.22: ROC and PR curves for RFC (outlier detection framework)

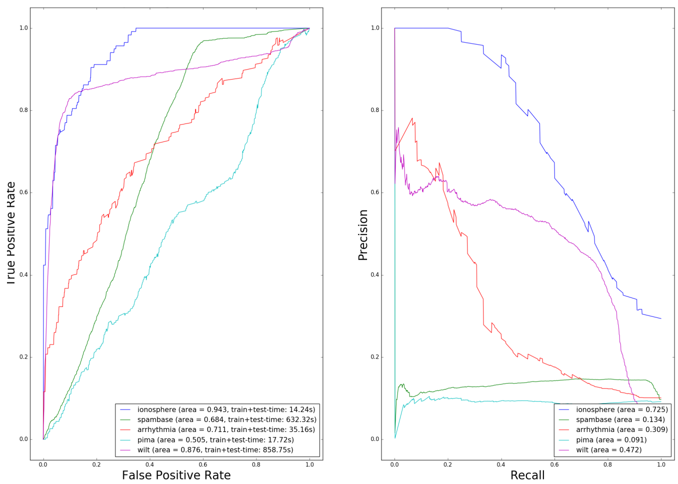

Table D.4: Results for the outlier detection setting

Dataset	OneClassRF		IForest		OCRFsampling		OCSM		LOF		Orca		LSAD		RFC	
AUC	ROC	PR	ROC	PR	ROC	PR	ROC	PR	ROC	PR	ROC	PR	ROC	PR	ROC	PR
adult	0.625	0.161	**0.644**	0.234	N. A.	N. A.	0.622	0.179	0.546	0.100	0.593	0.179	0.633	0.204	N. A.	N. A.
annthyroid	0.842	0.226	0.820	0.310	**0.992**	0.869	0.688	0.193	0.731	0.188	0.561	0.132	0.762	0.246	N. A.	N. A.
arrhythmia	0.698	0.485	0.746	0.418	0.704	0.276	**0.916**	0.630	0.765	0.468	0.741	0.502	0.733	0.393	0.711	0.309
forestcover	0.845	0.044	**0.882**	0.062	N. A.	N. A.	N. A.	N. A.	0.550	0.017	0.696	0.045	0.816	0.072	N. A.	N. A.
http	0.984	0.120	**0.999**	0.685	N. A.	N. A.	N. A.	N. A.	N. A.	N. A.	0.998	0.402	0.277	0.074	N. A.	N. A.
ionosphere	0.903	0.508	0.888	0.545	0.879	0.664	**0.956**	0.813	**0.956**	0.789	0.929	0.917	0.915	0.773	0.943	0.725
pendigits	0.453	0.085	0.463	0.077	**0.999**	0.993	0.366	0.066	0.491	0.086	0.495	0.086	0.513	0.091	N. A.	N. A.
pima	0.708	0.229	0.743	0.205	**0.790**	0.296	0.706	0.226	0.670	0.137	0.585	0.170	0.686	0.190	0.505	0.091
shuttle	0.947	0.491	**0.997**	0.979	N. A.	N. A.	0.992	0.904	0.526	0.115	0.655	0.320	0.686	0.218	N. A.	N. A.
smtp	**0.916**	0.400	0.902	0.005	N. A.	N. A.	0.881	0.372	0.909	0.053	0.824	0.236	0.888	0.398	N. A.	N. A.
spambase	0.830	0.300	0.799	0.303	**0.970**	0.877	0.722	0.192	0.664	0.120	0.603	0.210	0.731	0.229	0.684	0.134
wilt	0.520	0.053	0.443	0.044	**0.966**	0.554	0.316	0.036	0.627	0.069	0.441	0.029	0.530	0.053	0.876	0.472
average	0.773	0.259	0.777	0.322	**0.900**	0.647	0.717	0.361	0.676	0.195	0.677	0.269	0.681	0.245	0.744	0.346
cum. train time	**61s**		70s		N. A.		N. A.		N. A.		2432s		72s		N. A.	

PUBLICATIONS

[23] R. Brault and F. d'Alché–Buc. "Borne sur l'approximation de noyaux à valeurs opérateurs à l'aide de transformées de Fourier." In: *Société Francaise des Statistiques*. SFDS. 2016 (cit. on pp. 4, 79).

[24] R. Brault, M. Heinonen, and F. d'Alché-Buc. "Random Fourier Features For Operator-Valued Kernels." In: *Proceedings of The 8th Asian Conference on Machine Learning*. 2016, pp. 110–125 (cit. on pp. 4, 52, 79, 84, 155, 157).

[25] R. Brault, N. Lim, and F. d'Alché-Buc. "Scaling up Vector Autoregressive Models With Operator-Valued Random Fourier Features." In: *Proceedings of AALTD 2016: Second ECML/PKDD International Workshop on Advanced Analytics and Learning on Temporal Data*. 2016, p. 3 (cit. on pp. 5, 109, 116, 135, 139).

[64] N. Goix, R. Brault, N. Drougard, and M. Chiapino. "One Class Splitting Criteria for Random Forests." In: *arXiv preprint arXiv:1611.01971* (2016) (cit. on p. 193).

[2] R. Ahlswede and A. Winter. "Strong converse for identification via quantum channels." In: *IEEE Trans. Inform. Theory 48(3)* (2002), pp. 569–679 (cit. on p. 156).

[3] E. M. Alfsen. "A simplified constructive proof of the existence and uniqueness of Haar measure." In: *Mathematica Scandinavica* 12.1 (1964), pp. 106–116 (cit. on p. 29).

[4] M. A. Álvarez, L. Rosasco, and N. D. Lawrence. "Kernels for vector-valued functions: a review." In: *Foundations and Trends in Machine Learning* 4.3 (2012), pp. 195–266 (cit. on pp. 4, 44, 95, 137, 141).

[6] N. Aronszajn. "Theory of reproducing kernels." In: *Transactions of the American mathematical society* (1950), pp. 337–404 (cit. on pp. 11–13, 35).

[7] J. Audiffren and H. Kadri. "Online learning with operator-valued kernels." In: *European symposium on artificial neural networks (ESANN)*. 2015 (cit. on pp. 116, 139).

[8] I. Avramidi. "Notes on Hilbert Spaces." In: (2000) (cit. on p. 28).

[9] F. Bach. *On the Equivalence between Quadrature Rules and Random Features*. HAl-report-/hal-01118276. 2015 (cit. on pp. 4, 17).

[10] F. Bach. "On the equivalence between quadrature rules and random features." In: *arXiv preprint arXiv:1502.06800* (2015) (cit. on p. 133).

[11] L. Baldassarre, L. Rosasco, A. Barla, and A. Verri. "Vector Field Learning via Spectral Filtering." In: *ECML/PKDD*. Ed. by J. Balcazar, F. Bonchi, A. Gionis, and M. Sebag. Vol. 6321. LNCS. Springer Berlin / Heidelberg, 2010, pp. 56–71 (cit. on p. 44).

[12] L. Baldassarre, L. Rosasco, A. Barla, and A. Verri. "Multi-output learning via spectral filtering." In: *Machine Learning* 87.3 (2012), pp. 259–301 (cit. on p. 45).

[14] P. L. Bartlett and S. Mendelson. "Rademacher and Gaussian complexities: Risk bounds and structural results." In: *Journal of Machine Learning Research* 3.Nov (2002), pp. 463–482 (cit. on pp. 17, 125, 127).

[16] M. Belkin, P. Niyogi, and V. Sindhwani. "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples." In: *Journal of machine learning research* 7.Nov (2006), pp. 2399–2434 (cit. on p. 168).

[20] S. Boucheron, G. Lugosi, and P. Massart. *Concentration Inequalities*. Oxford Press, 2013 (cit. on pp. 81, 83).

[21] O. Bousquet and A. Elisseeff. "Stability and generalization." In: *Journal of Machine Learning Research* 2.Mar (2002), pp. 499–526 (cit. on p. 128).

[22] G. E. Box. "Robustness in the strategy of scientific model building." In: *Robustness in statistics* 1 (1979), pp. 201–236 (cit. on p. 7).

[26] U. Brefeld, T. Gärtner, T. Scheffer, and S. Wrobel. "Efficient co-regularised least squares regression." In: *Proceedings of the 23rd international conference on Machine learning*. ACM. 2006, pp. 137–144 (cit. on p. 168).

[29] C. Brouard, F. d'Alché-Buc, and M. Szafranski. "Semi-supervised Penalized Output Kernel Regression for Link Prediction." In: *Proc. of the 28th Int. Conf. on Machine Learning*. 2011 (cit. on pp. 96, 168).

[30] C. Brouard, F. d'Alché-Buc, and M. Szafranski. "Input Output Kernel Regression." In: *to appear in JMLR* (2016) (cit. on p. 95).

[31] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu. "A limited memory algorithm for bound constrained optimization." In: *SIAM Journal on Scientific Computing* 16.5 (1995), pp. 1190–1208 (cit. on pp. 116, 138).

[32] A. Caponnetto, C. A. Micchelli, and Y. Ying. "Universal MultiTask Kernels." In: *Journal of Machine Learning Research* 9 (2008), pp. 1615–1646 (cit. on pp. 44, 46, 137).

[33] C. Carmeli, E. De Vito, and A. Toigo. "Vector valued reproducing kernel Hilbert spaces of integrable functions and Mercer theorem." In: *Analysis and Applications* 4.04 (2006), pp. 377–408 (cit. on pp. 36–38, 42, 54).

[34] C. Carmeli, E. De Vito, A. Toigo, and V. Umanità. "Vector valued reproducing kernel Hilbert spaces and universality." In: *Analysis and Applications* 8 (2010), pp. 19–61 (cit. on pp. 4, 36, 40, 41, 43, 44, 52–54, 65, 137, 138).

[38] J. B. Conway. *A course in functional analysis*. Vol. 96. Springer Science & Business Media, 2013 (cit. on pp. 29, 101).

[39] A. Cornuéjols and L. Miclet. *Apprentissage artificiel: concepts et algorithmes*. Editions Eyrolles, 2011 (cit. on p. 9).

[41] I. I. Cotaescu. "Elements of Linear Algebra. Lecture Notes." In: *arXiv preprint arXiv:1602.03006* (2016) (cit. on p. 22).

[42] F. Cucker and S. Smale. "On the mathematical foundation of learning." In: *AMERICAN MATHEMATICAL SOCIETY* 39.1 (2001), pp. 1–49 (cit. on pp. 85, 154).

[43] B. Dai, B. Xie, N. He, Y. Liang, A. Raj, M.-F. F. Balcan, and L. Song. "Scalable kernel methods via doubly stochastic gradients." In: *Advances in Neural Information Processing Systems*. 2014, pp. 3041–3049 (cit. on pp. 4, 5, 19, 135, 139).

[45] F. Dinuzzo, C. Ong, P. Gehler, and G. Pillonetto. "Learning Output Kernels with Block Coordinate Descent." In: *Proc. of the 28th Int. Conf. on Machine Learning*. 2011 (cit. on p. 44).

[47] P. Drineas and M. W. Mahoney. "On the Nyström method for approximating a Gram matrix for improved kernel-based learning." In: *journal of machine learning research* 6.Dec (2005), pp. 2153–2175 (cit. on pp. 4, 17).

[48] R. M. Dudley. "The sizes of compact subsets of Hilbert space and continuity of Gaussian processes." In: *Journal of Functional Analysis* 1.3 (1967), pp. 290–330 (cit. on p. 83).

[51] T. Evgeniou, C. A. Micchelli, and M. Pontil. "Learning Multiple Tasks with kernel methods." In: *JMLR* 6 (2005), pp. 615–637 (cit. on pp. 44, 137).

[52] P. L. Falb. "On a theorem of Bochner." In: *Publications Mathématiques de l'IHÉS* 36 (1969), pp. 59–67 (cit. on p. 26).

[53] X. Y. Felix, A. T. Suresh, K. M. Choromanski, D. N. Holtmann-Rice, and S. Kumar. "Orthogonal random features." In: *Advances in Neural Information Processing Systems*. 2016, pp. 1975–1983 (cit. on pp. 4, 18).

[54] O. Fercoq and R. Peter. "Accelerated, parallel, and proximal coordinate descent." In: *SIAM Journal on Optimization* 25.4 (2015), pp. 1997–2023 (cit. on p. 137).

[55] O. Fercoq and P. Richtárik. "Smooth minimization of nonsmooth functions with parallel coordinate descent methods." In: *arXiv preprint arXiv:1309.5885* (2013) (cit. on p. 137).

[57] G. B. Folland. *A course in abstract harmonic analysis*. CRC press, 1994 (cit. on pp. 15, 29, 30, 32–34).

[59] E. Fuselier. "Refined Error Estimates for Matrix-Valued Radial Basis Functions." PhD thesis. Texas A&M University, 2006 (cit. on p. 45).

[65] L. Górniewicz. *Topological fixed point theory of multivalued mappings*. Vol. 495. Springer, 1999 (cit. on p. 94).

[66] G. Guennebaud, B. Jacob, et al. *Eigen v3*. 2010. URL: `http://eigen.tuxfamily.org` (cit. on p. 111).

[67] R. Hamid, Y. Xiao, A. Gittens, and D. DeCoste. "Compact Random Feature Maps." In: *ICML*. 2014, pp. 19–27 (cit. on p. 18).

[70] K.-U. Hoffgen, H.-U. Simon, and K. S. Vanhorn. "Robust trainability of single neurons." In: *Journal of Computer and System Sciences* 50.1 (1995), pp. 114–125 (cit. on p. 10).

[71] T. Jaakkola, M. Diekhans, and D. Haussler. "Using the Fisher kernel method to detect remote protein homologies." In: *ISMB*. Vol. 99. 1999, pp. 149–158 (cit. on p. 212).

[72] E. Jones, T. Oliphant, and P. Peterson. “{SciPy}: open source scientific tools for {Python}.” In: (2014) (cit. on p. 107).

[73] H. Kadri, E. Duflos, P. Preux, S. Canu, A. Rakotomamonjy, and J. Audiffren. “Operator-valued kernels for learning from functional response data.” In: *Journal of Machine Learning Research* 16 (2015), pp. 1–54 (cit. on pp. 46, 47, 95, 96, 128, 129, 132).

[74] P. Kar and H. Karnick. “Random Feature Maps for Dot Product Kernels.” In: *AISTATS*. Vol. 22. 2012, pp. 583–591 (cit. on p. 18).

[76] J. Kivinen, A. J. Smola, and R. C. Williamson. “Online learning with kernels.” In: *IEEE transactions on signal processing* 52.8 (2004), pp. 2165–2176 (cit. on p. 139).

[77] V. Koltchinskii et al. “A remark on low rank matrix recovery and noncommutative Bernstein type inequalities.” In: *From Probability to Statistics and Back: High-Dimensional Models and Processes–A Festschrift in Honor of Jon A. Wellner*. Institute of Mathematical Statistics, 2013, pp. 213–226 (cit. on pp. 4, 79, 84, 86, 154, 156, 158).

[78] A. J. Kurdila and M. Zabarankin. *Convex functional analysis*. 2006 (cit. on pp. 22, 94, 97, 101).

[79] Q. V. Le, T. Sarlós, and A. J. Smola. “Fastfood - Computing Hilbert Space Expansions in loglinear time.” In: *Proc. of ICML 2013, Atlanta, USA, 16-21 June 2013*. 2013, pp. 244–252 (cit. on pp. 4, 18).

[80] Y. LeCun, Y. Bengio, et al. “Convolutional networks for images, speech, and time series.” In: *The handbook of brain theory and neural networks* 3361.10 (1995), p. 1995 (cit. on p. 19).

[81] M. Ledoux. *The concentration of measure phenomenon*. 89. American Mathematical Soc., 2005 (cit. on p. 81).

[82] M. Ledoux and M. Talagrand. *Probability in Banach Spaces: isoperimetry and processes*. Springer Science & Business Media, 2013 (cit. on p. 84).

[83] F. Li, C. Ionescu, and C. Sminchisescu. “Pattern Recognition: 32nd DAGM Symposium, Darmstadt, Germany, September 22-24, 2010. Proc.” In: ed. by M. Goesele, S. Roth, A. Kuijper, B. Schiele, and K. Schindler. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. Chap. Random Fourier Approximations for Skewed Multiplicative Histogram Kernels, pp. 262–271. ISBN: 978-3-642-15986-2. DOI: 10.1007/978-3-642-15986-2_27. URL: http://dx.doi.org/10.1007/978-3-642-15986-2_27 (cit. on pp. 15, 18, 32, 61, 75).

[85] N. Lim, F. d’Alché-Buc, C. Auliac, and G. Michailidis. “Operator-valued kernel-based vector autoregressive models for network inference.” In: *Machine Learning* 99.3 (2015), pp. 489–513 (cit. on pp. 5, 44, 135–138, 144).

[86] N. Lim, Y. Senbabaoglu, G. Michailidis, and F. d'Alché-Buc. "OKVAR-Boost: a novel boosting algorithm to infer nonlinear dynamics and interactions in gene regulatory networks." In: *Bioinformatics* 29.11 (2013), pp. 1416–1423 (cit. on pp. 136, 144).

[87] N. Lim, F. d'Alché-Buc, C. Auliac, and G. Michailidis. "Operator-valued kernel-based vector autoregressive models for network inference." In: *Machine learning* 99.3 (2015), pp. 489–513 (cit. on p. 46).

[88] N. Lim, Y. Şenbabaoğlu, G. Michailidis, and F. d'Alché-Buc. "OKVAR-Boost: a novel boosting algorithm to infer nonlinear dynamics and interactions in gene regulatory networks." In: *Bioinformatics* 29.11 (2013), pp. 1416–1423 (cit. on p. 46).

[91] Y. Liu, A. Niculescu-Mizil, A. C. Lozano, and Y. Lu. "Learning temporal causal graphs for relational time-series analysis." In: *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*. 2010, pp. 687–694 (cit. on p. 143).

[93] Z. Lu, A. May, K. Liu, A. B. Garakani, D. Guo, A. Bellet, L. Fan, M. Collins, B. Kingsbury, M. Picheny, et al. "How to scale up kernel methods to be as good as deep neural nets." In: *arXiv preprint arXiv:1411.4000* (2014) (cit. on pp. 4, 19).

[94] Y. Macedo and R. Castro. *Learning Div-Free and Curl-Free Vector Fields by Matrix-Valued Kernels*. Tech. rep. Preprint A 679/2010 IMPA, 2008 (cit. on p. 45).

[95] L. Mackey, M. I. Jordan, R. Chen, B. Farrel, and J. Tropp. "Matrix Concentration Inequalities via the method of exchangeable pairs." In: *The Annals of Probability* 42:3 (2014), pp. 906–945 (cit. on pp. 154, 156).

[97] A. Maurer. "A vector-contraction inequality for Rademacher complexities." In: *International Conference on Algorithmic Learning Theory*. Springer. 2016, pp. 3–17 (cit. on pp. 5, 123, 125–127, 130, 132).

[98] C. A. Micchelli and M. A. Pontil. "On Learning Vector-Valued Functions." In: *Neural Computation* 17 (2005), pp. 177–204 (cit. on pp. 4, 36, 95, 96, 137).

[99] M. Micheli and J. Glaunes. *Matrix-valued kernels for shape deformation analysis*. Tech. rep. Arxiv report, 2013 (cit. on pp. 44, 45).

[100] H. Q. Minh, L. Bazzani, and V. Murino. "A unifying framework for vector-valued manifold regularization and multi-view learning." In: *Proc. of the 30th International Conference on Machine Learning*. 2013 (cit. on p. 95).

[101] H. Q. Minh. "Operator-Valued Bochner Theorem, Fourier Feature Maps for Operator-Valued Kernels, and Vector-Valued Learning." In: *arXiv preprint arXiv:1608.05639* (2016) (cit. on pp. 57, 58, 87, 155, 164).

[102] H. Q. Minh, L. Bazzani, and V. Murino. "A unifying framework for vector-valued manifold regularization and multi-view learning." In: *ICML (2)*. 2013, pp. 100–108 (cit. on p. 168).

[103] H. Q. Minh, L. Bazzani, and V. Murino. "A unifying framework in vector-valued reproducing kernel Hilbert spaces for manifold regularization and co-regularized multi-view learning." In: *Journal of Machine Learning Research* 17.25 (2016), pp. 1–72 (cit. on pp. 95, 96, 168, 169, 177).

[104] S. Minsker. "On some extensions of Bernstein's inequality for self-adjoint operators." In: *arXiv preprint arXiv:1112.5448* (2011) (cit. on pp. 4, 5, 79, 84, 87, 157, 158).

[105] Y. Mroueh, T. Poggio, L. Rosasco, and J.-j. Slotine. "Multiclass learning with simplex coding." In: *Advances in NIPS*. 2012, pp. 2789–2797 (cit. on pp. 115, 175).

[106] Y. Mukuta and T. Harada. "Kernel Approximation via Empirical Orthogonal Decomposition for Unsupervised Feature Learning." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 5222–5230 (cit. on p. 18).

[107] A. Naor. "On the Banach-space-valued Azuma inequality and small-set isoperimetry of Alon–Roichman graphs." In: *Combinatorics, Probability and Computing* 21.04 (2012), pp. 623–634 (cit. on p. 81).

[108] K.-H. Neeb. "Operator-valued positive definite kernels on tubes." In: *Monatshefte für Mathematik* 126.2 (1998), pp. 125–160 (cit. on p. 52).

[109] T. E. Oliphant. *A guide to NumPy*. Vol. 1. Trelgol Publishing USA, 2006 (cit. on p. 107).

[110] R. I. Oliveira. "Concentration of the adjacency matrix and of the Laplacian in random graphs with independent edges." In: *arXiv preprint arXiv:0911.0600* (2009) (cit. on p. 156).

[112] N. Parikh, S. Boyd, et al. "Proximal algorithms." In: *Foundations and Trends® in Optimization* 1.3 (2014), pp. 127–239 (cit. on p. 137).

[114] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al. "Scikit-learn: Machine learning in Python." In: *Journal of Machine Learning Research* 12.Oct (2011), pp. 2825–2830 (cit. on pp. 107, 212).

[115] G. Pedrick. *Theory of reproducing kernels for Hilbert spaces of vector-valued functions*. Tech. rep. University of Kansas, Department of Mathematics, 1957 (cit. on pp. 36, 137).

[116] T. Penzl. "Numerical solution of generalized Lyapunov equations." In: *Advances in Computational Mathematics* 8.1 (1998), pp. 33–48 (cit. on p. 109).

[117] N. Pham and R. Pagh. "Fast and scalable polynomial kernels via explicit feature maps." In: *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM. 2013, pp. 239–247 (cit. on p. 18).

[118] I. Pinelis. "Optimum bounds for the distributions of martingales in Banach spaces." In: *The Annals of Probability* (1994), pp. 1679–1706 (cit. on pp. 81, 84, 130).

[121] A. Rahimi and B. Recht. "Random Features for Large-Scale Kernel Machines." In: *NIPS 2007*. 2007, pp. 1177–1184 (cit. on pp. 4, 15–18, 73, 79, 82, 83, 85, 87, 136, 139, 154, 164).

[122] A. Rahimi and B. Recht. "Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning." In: *Advances in neural information processing systems*. 2009, pp. 1313–1320 (cit. on pp. 5, 18, 123, 130, 132, 133).

[123] P. Richtárik and M. Takáč. "Parallel coordinate descent methods for big data optimization." In: *Mathematical Programming* 156.1-2 (2016), pp. 433–484 (cit. on p. 137).

[124] L. Rosasco, M. Belkin, and E. D. Vito. "On learning with integral operators." In: *Journal of Machine Learning Research* 11.Feb (2010), pp. 905–934 (cit. on pp. 17, 96, 109).

[125] D. Rosenberg, V. Sindhwani, P. Bartlett, and P. Niyogi. "A kernel for semi-supervised learning with multi-view point cloud regularization." In: *IEEE Signal Processing Magazine* 26.5 (2009), pp. 145–150 (cit. on p. 168).

[126] A. Rudi, R. Camoriano, and L. Rosasco. "Generalization properties of learning with random features." In: *arXiv preprint arXiv:1602.04474* (2016) (cit. on pp. 4, 17, 18, 133).

[127] A. L. Samuel. "Some studies in machine learning using the game of checkers." In: *IBM Journal of research and development* 3.3 (1959), pp. 210–229 (cit. on p. ix).

[128] M. Sangnier, O. Fercoq, and F. D'Alché-Buc. "Joint quantile regression in vector-valued RKHSs." In: *Neural Information Processing Systems* (2016) (cit. on p. 95).

[134] E. Senkene and A. Tempel'man. "Hilbert Spaces of operator-valued functions." In: *Lithuanian Mathematical Journal* 13.4 (1973), pp. 665–670 (cit. on pp. 36, 137).

[139] V. Sindhwani, H. Q. Minh, and A. Lozano. "Scalable Matrix-valued Kernel Learning for High-dimensional Nonlinear Multivariate Regression and Granger Causality." In: *Proc. of UAI'13, Bellevue, WA, USA, August 11-15, 2013*. AUAI Press, Corvallis, Oregon, 2013 (cit. on pp. 44, 109).

[140] V. Sindhwani and D. S. Rosenberg. "An RKHS for multi-view learning and manifold co-regularization." In: *Proceedings of the 25th international conference on Machine learning*. ACM. 2008, pp. 976–983 (cit. on p. 168).

[141] G. L. Sleijpen, P. Sonneveld, and M. B. Van Gijzen. "Bi-CGSTAB as an induced dimension reduction method." In: *Applied Numerical Mathematics* 60.11 (2010), pp. 1100–1114 (cit. on p. 109).

[142] S. Smale and D.-X. Zhou. "Learning theory estimates via integral operators and their approximations." In: *Constructive approximation* 26.2 (2007), pp. 153–172 (cit. on p. 81).

[143] A. J. Smola, B. Schölkopf, and K.-R. Müller. "The connection between regularization operators and support vector kernels." In: *Neural networks* 11.4 (1998), pp. 637–649 (cit. on p. 76).

[144] P. Sonneveld and M. B. van Gijzen. "IDR (s): A family of simple and fast algorithms for solving large nonsymmetric systems of linear equations." In: *SIAM Journal on Scientific Computing* 31.2 (2008), pp. 1035–1062 (cit. on p. 139).

[145] B. Sriperumbudur and Z. Szabo. "Optimal Rates for Random Fourier Features." In: *Advances in NIPS 28*. Ed. by C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett. 2015, pp. 1144–1152 (cit. on pp. 4, 16, 82, 83).

[147] S. Sun. "Multi-view Laplacian support vector machines." In: *International Conference on Advanced Data Mining and Applications*. Springer. 2011, pp. 209–222 (cit. on p. 168).

[148] D. J. Sutherland and J. G. Schneider. "On the Error of Random Fourier Features." In: *Proc. of UAI 2015, July 12-16, 2015, Amsterdam, The Netherlands*. 2015, pp. 862–871 (cit. on pp. 4, 5, 16, 17, 79, 82, 83, 87, 89, 136, 154–156, 164).

[152] A. Tewari and P. L. Bartlett. "Learning theory." In: *Academic Press Library in Signal Processing* 1 (), pp. 775–816 (cit. on p. 128).

[153] S. Tornier. *Haar measures*. `https://people.math.ethz.ch/~torniers/download/2014/haar_measures.pdf`. 2014 (cit. on p. 29).

[154] J. A. Tropp. "User-friendly tail bounds for sums of random matrices." In: *Foundations of computational mathematics* 12.4 (2012), pp. 389–434 (cit. on pp. 154, 156).

[155] J. A. Tropp et al. "An introduction to matrix concentration inequalities." In: *Foundations and Trends® in Machine Learning* 8.1-2 (2015), pp. 1–230 (cit. on pp. 4, 5, 79, 84, 157).

[156] A. W. Van Der Vaart and J. A. Wellner. "Weak Convergence." In: *Weak Convergence and Empirical Processes*. Springer, 1996, pp. 16–28 (cit. on p. 84).

[157] V. Vapnik. “Principles of risk minimization for learning theory.” In: *Advances in neural information processing systems*. 1992, pp. 831–838 (cit. on p. 128).

[158] V. N. Vapnik and V. Vapnik. *Statistical learning theory*. Vol. 1. Wiley New York, 1998 (cit. on p. 124).

[159] E. Vazquez and E. Walter. “Multi-output support vector regression.” In: *13th IFAC Symposium on System Identification*. 2003, pp. 1820–1825 (cit. on p. 46).

[160] J.-P. Vert. “Regularization of Kernel Methods by Decreasing the Bandwidth of the Gaussian Kernel.” In: () (cit. on p. 76).

[161] U. Von Luxburg. “A tutorial on spectral clustering.” In: *Statistics and computing* 17.4 (2007), pp. 395–416 (cit. on pp. 173, 175).

[162] G. Wahba. *Spline model for observational data*. Philadelphia, Society for Industrial and Applied Mathematics, 1990 (cit. on pp. 12, 96).

[163] N. Wahlström, M. Kok, T. Schön, and F. Gustafsson. “Modeling magnetic fields using Gaussian processes.” In: *in Proc. of the 38th ICASSP*. 2013 (cit. on p. 45).

[164] S. v. d. Walt, S. C. Colbert, and G. Varoquaux. “The NumPy array: a structure for efficient numerical computation.” In: *Computing in Science & Engineering* 13.2 (2011), pp. 22–30 (cit. on p. 109).

[165] C. K. I. Williams and M. Seeger. “Using the Nyström Method to Speed Up Kernel Machines.” In: *Advances in Neural Information Processing Systems 13*. Ed. by T. K. Leen, T. G. Dietterich, and V. Tresp. MIT Press, 2001, pp. 682–688 (cit. on p. 16).

[166] B. Xie, Y. Liang, and L. Song. “Scale up nonlinear component analysis with doubly stochastic gradients.” In: *Advances in Neural Information Processing Systems*. 2015, pp. 2341–2349 (cit. on p. 19).

[168] J. Yang, V. Sindhwani, H. Avron, and M. Mahoney. “Quasi-Monte Carlo feature maps for shift-invariant kernels.” In: *Proceedings of The 31st International Conference on Machine Learning (ICML-14)*. 2014, pp. 485–493 (cit. on p. 18).

[169] J. Yang, V. Sindhwani, Q. Fan, H. Avron, and M. W. Mahoney. “Random laplace feature maps for semigroup kernels on histograms.” In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 971–978 (cit. on p. 19).

[170] T. Yang, Y.-F. Li, M. Mahdavi, R. Jin, and Z. Zhou. “Nyström Method vs Random Fourier Features: A Theoretical and Empirical Comparison.” In: *NIPS 25*. Ed. by F. Pereira, C. Burges, L. Bottou, and K. Weinberger. 2012, pp. 476–484 (cit. on pp. 17, 18, 76).

[171] Z. Yang, M. Moczulski, M. Denil, N. de Freitas, A. Smola, L. Song, and Z. Wang. "Deep fried convnets." In: *Proceedings of the IEEE International Conference on Computer Vision*. 2015, pp. 1476–1483 (cit. on pp. 4, 19).

[172] Z. Yang, A. G. Wilson, A. J. Smola, and L. Song. "A la Carte - Learning Fast Kernels." In: *Proc. of AISTATS 2015, San Diego, California, USA, 2015*. 2015 (cit. on p. 19).

[173] H. Zhang, Y. Xu, and Q. Zhang. "Refinement of Operator-valued Reproducing Kernels." In: *Journal of Machine Learning Research* 13 (2012), pp. 91–136 (cit. on p. 52).

[174] D.-X. Zhou. "The covering number in learning theory." In: *Journal of Complexity* 18.3 (2002), pp. 739–767 (cit. on p. 128).

REFERENCES ON EXTREMS

[1] C. Aggarwal and P. Yu. "Outlier detection for high dimensional data." In: *ACM Sigmod Record*. 2001 (cit. on p. 194).

[5] Y. Amit and D. Geman. "Shape quantization and recognition with randomized trees." In: *Neural comp.* (1997) (cit. on p. 202).

[13] V. Barnett and T. Lewis. *Outliers in statistical data*. Wiley New York, 1994 (cit. on p. 194).

[15] S. Bay and M. Schwabacher. "Mining distance-based outliers in near linear time with randomization and a simple pruning rule." In: *KDD*. 2003 (cit. on p. 203).

[17] G. Biau, L. Devroye, and G. Lugosi. "Consistency of random forests and other averaging classifiers." In: *Journal of Machine Learning Research* 9.Sep (2008), pp. 2015–2033 (cit. on p. 194).

[18] G. Biau and E. Scornet. "A random forest guided tour." In: *Test* 25.2 (2016), pp. 197–227 (cit. on p. 194).

[19] G. Blanchard, C. Schäfer, and Y. Rozenholc. "Oracle bounds and exact algorithm for dyadic classification trees." In: *International Conference on Computational Learning Theory*. Springer. 2004, pp. 378–392 (cit. on p. 195).

[27] L. Breiman. "Random Forests." English. In: *Machine Learning* (2001). ISSN: 0885-6125 (cit. on pp. 194, 202, 210).

[28] M. Breunig, H. Kriegel, R. Ng, and J. Sander. "LOF: identifying density-based local outliers." In: *ACM Sigmod Rec*. 2000 (cit. on pp. 194, 203).

[35] V. Chandola, A. Banerjee, and V. Kumar. "Anomaly detection: A survey." In: *ACM Comput. Surv.* (2009) (cit. on p. 194).

[36] S. Clémençon and S. Robbiano. "Anomaly Ranking as Supervised Bipartite Ranking." In: *ICML*. 2014 (cit. on p. 195).

[37] S. Clémençon and N. Vayatis. "Tree-based ranking methods." In: *IEEE Transactions on Information Theory* 55.9 (2009), pp. 4316–4336 (cit. on p. 195).

[40] C. Cortes and V. Vapnik. "Support-Vector Networks." In: *Machine Learning* (1995) (cit. on p. 194).

[44] C. Désir, S. Bernard, C. Petitjean, and L. Heutte. "One Class Random Forests." In: *Pattern Recogn.* (2013) (cit. on pp. 194, 195, 198, 203).

[46] R. Díaz-Uriarte and S. De Andres. "Gene selection and classification of microarray data using random forest." In: *BMC bioinformatics* (2006) (cit. on p. 194).

[49] J. H. Einmahl and D. M. Mason. "Generalized quantile processes." In: *The Annals of Statistics* (1992), pp. 1062–1078 (cit. on p. 195).

[50] E. Eskin. "Anomaly Detection over Noisy Data using Learned Probability Distributions." In: *ICML*. 2000 (cit. on p. 194).

[56] P. Flach. "The geometry of ROC space: understanding ml metrics through ROC isometrics." In: *ICML*. 2003 (cit. on p. 199).

[58] Y. Freund, R. Schapire, et al. "Experiments with a new boosting algorithm." In: *ICML*. 1996 (cit. on p. 194).

[60] R. Genuer, J.-M. Poggi, and C. Tuleau. "Random Forests: some methodological insights." In: *arXiv:0811.3619* (2008) (cit. on p. 194).

[61] R. Genuer, J.-M. Poggi, and C. Tuleau-Malot. "Variable selection using random forests." In: *Pattern Recog. Letters* (2010) (cit. on p. 194).

[62] P. Geurts, D. Ernst, and L. Wehenkel. "Extremely randomized trees." In: *Machine learning* (2006) (cit. on pp. 194, 197, 202).

[63] C. Gini. "Variabilita e mutabilita." In: *Memorie di metodologia statistica* (1912) (cit. on p. 197).

[68] T. Ho. "The random subspace method for constructing decision forests." In: *TPAMI* (1998) (cit. on p. 202).

[69] V. Hodge and J. Austin. "A survey of outlier detection methodologies." In: *Artif. Intel. Review* (2004) (cit. on p. 194).

[75] KDDCup. "The third international knowledge discovery and data mining tools competition dataset." In: (1999) (cit. on p. 212).

[84] M. Lichman. *UCI Machine Learning Repository*. 2013. URL: http://archive.ics.uci.edu/ml (cit. on pp. 204, 212).

[89] F. Liu, K. Ting, and Z. Zhou. "Isolation Forest." In: *ICDM*. 2008 (cit. on pp. 194, 202–204, 209, 212).

[90] F. Liu, K. Ting, and Z.-H. Zhou. "Isolation-Based Anomaly Detection." In: *ACM Trans. Knowl. Discov. Data* (Mar. 2012). ISSN: 1556-4681. DOI: 10.1145/2133360.2133363. URL: http://doi.acm.org/10.1145/2133360.2133363 (cit. on p. 211).

[92] G. Louppe. "Understanding random forests: From theory to practice." In: *arXiv:1407.7502* (2014) (cit. on p. 194).

[96] M. Markou and S. Singh. "Novelty detection: a review part 1: statistical approaches." In: *Signal proc.* (2003) (cit. on p. 194).

[111] P. Panov and S. Džeroski. *Combining bagging and random subspaces to create better ensembles*. Springer, 2007 (cit. on p. 202).

[113] A. Patcha and J. Park. "An overview of anomaly detection techniques: Existing solutions and latest technological trends." In: *Computer Networks* (2007) (cit. on p. 194).

[119] W. Polonik. "Minimum volume sets and generalized quantile processes." In: *Stochastic Processes and their Applications* (1997) (cit. on p. 195).

[120] J. Quinn and M. Sugiyama. "A least-squares approach to anomaly detection in static and sequential data." In: *Pattern Recognition Letters* (2014) (cit. on pp. 194, 203).

[129] B. Schölkopf, J. Platt, J. Shawe-Taylor, A. Smola, and R. Williamson. "Estimating the support of a high-dimensional distribution." In: *Neural computation* (2001) (cit. on pp. 194, 203).

[130] E. Schubert, R. Wojdanowski, A. Zimek, and H.-P. Kriegel. "On Evaluation of Outlier Rankings and Outlier Scores." In: *SDM*. 2012 (cit. on p. 213).

[131] E. Scornet, G. Biau, J.-P. Vert, et al. "Consistency of random forests." In: *The Annals of Statistics* 43.4 (2015), pp. 1716–1741 (cit. on p. 195).

[132] C. Scott and G. Blanchard. "Novelty detection: Unlabeled data definitely help." In: *AISTATS*. 2009, pp. 464–471 (cit. on p. 200).

[133] C. Scott and R. Nowak. "Learning minimum volume sets." In: *JMLR* (2006) (cit. on pp. 194, 195).

[135] C. E. Shannon. "A mathematical theory of communication." In: *ACM SIGMOBILE MC2R* (2001) (cit. on p. 197).

[136] J. Shawe-Taylor and N. Cristianini. *Kernel methods for pattern analysis*. Cambridge univ. press, 2004 (cit. on p. 194).

[137] T. Shi and S. Horvath. "Unsupervised learning with random forest predictors." In: *J. Comp. Graph. Stat.* (2012) (cit. on pp. 194, 195, 198, 203).

[138] M. Shyu, S. Chen, K. Sarinnapakorn, and L. Chang. *A novel anomaly detection scheme based on principal component classifier*. Tech. rep. DTIC Document, 2003 (cit. on p. 194).

[146] M. Sugiyama. "Superfast-trainable multi-class probabilistic classifier by least-squares posterior fitting." In: *IEICE Transactions on Information and Systems* (2010) (cit. on p. 194).

[149] V. Svetnik, A. Liaw, C. Tong, J. C. Culberson, R. Sheridan, and B. Feuston. "Random forest: a classification and regression tool for compound classification and QSAR modeling." In: *J. Chem. Inf. Model.* (2003) (cit. on p. 194).

[150] M. Tavallaee, E. Bagheri, W. Lu, and A. Ghorbani. "A detailed analysis of the KDD CUP 99 data set." In: *IEEE CISDA*. 2009 (cit. on p. 212).

[151] D. Tax and R. Duin. "Uniform object generation for optimizing one-class classifiers." In: *JMLR* (2002) (cit. on p. 195).

[167] K. Yamanishi, J. Takeuchi, G. Williams, and P. Milne. "On-line unsupervised outlier detection using finite mixtures with discounting learning algorithms." In: *KDD*. 2000 (cit. on p. 212).

DECLARATION

I hereby declare that this manuscript entitled "*Data Are Not Reals! Large-Scale Learning On Structured Input-Output Data With Operator-Valued Kernels*" is a presentation of my original research work done during my thesis, carried out at Université d'Évry-Val-d'Essonne, Évry, Télécom ParisTech and Université Paris-Saclay, Paris, for the degree of Doctor of Philosophy in Computer Science of Université Paris-Saclay.

The interpretations put forth are based on my reading and understanding of the original texts and they are not published anywhere in the form of books, monographs or articles. Wherever contributions of others are involved, every effort is made to indicate this clearly, with due reference to the literature, and acknowledgement of collaborative research and discussions. For the present thesis, which I am submitting to the University, no degree or diploma or distinction has been conferred on me before, either in this or in any other University. The present work was done under the guidance of Professor Florence d'Alché-Buc, at Télécom ParisTech, Université Paris-Saclay, Paris.

Place:	(Mr. Romain Raymond Brault)
Date:	Research student

This is to certify that the work incorporated in the manuscript "*Data Are Not Reals! Large-Scale Learning On Structured Input-Output Data With Operator-Valued Kernels*" submitted by Romain Raymond Brault was carried out by the candidate under my guidance. Such materials as has been obtained from other sources have been duly acknowledged in the thesis. In my capacity as supervisor of the candidate's thesis, I certify that the above statements of this page are true to the best of my knowledge.

Place:	(Prof. Florence d'Alché-Buc)
Date:	Research supervisor

46, Rue Barrault, 75013 — Paris, France, May 20, 2017.

Romain Brault

COLOPHON

This document was typeset using the typographical look-and-feel `classicthesis` developed by André Miede. The style was inspired by Robert Bringhurst's seminal book on typography "*The Elements of Typographic Style*". `classicthesis` is available for both LaTeX and LyX at

https://bitbucket.org/amiede/classicthesis/.

Université Paris-Saclay
ED STIC — 580
Université Paris Sud, Bâtiment 650 Ada Lovelace, 91405 Orsay Cedex, France.

ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Large-Scale Learning On Structured Input-Output Data With Operator-Valued Kernels

Keywords: Operator-Valued Kernels, Large Scale Learning, Random Fourier Features

Abstract: Many problems in Machine Learning can be cast into vector-valued functions approximation. Operator-Valued Kernels *Operator-Valued Kernels* and vector-valued Reproducing Kernel Hilbert Spaces provide a theoretical and practical framework to address that issue, extending nicely the well-known setting of scalar-valued kernels. However large scale applications are usually not affordable with these tools that require an important computational power along with a large memory capacity. In this thesis, we propose and study scalable methods to perform regression with *Operator-Valued Kernels*. To achieve this goal, we extend Random Fourier Features, an approximation technique originally introduced for scalar-valued kernels, to *Operator-Valued Kernels*. The idea is to take advantage of an approximated operator-valued feature map in order to come up with a linear model in a finite-dimensional space.

This thesis is structured as follows. First we develop a general framework devoted to the approximation of shift-invariant Mercer kernels on Locally Compact Abelian groups and study their properties along with the complexity of the algorithms based on them. Second we show theoretical guarantees by bounding the error due to the approximation, with high probability. Third, we study various applications of Operator Random Fourier Features (ORFF) to different tasks of Machine learning such as multiclass classification, multi-task learning, time serie modeling, functional regression and anomaly detection. We also compare the proposed framework with other state of the art methods. Fourth, we conclude by drawing short-term and mid-term perspectives of this work.

Régression À Noyaux À Valeurs Opérateurs Pour Grands Ensembles De Données

Mots clefs: Noyaux à Valeurs Opérateurs, Passage à l'échelle, Random Fourier Features

Résumé: De nombreuses problématiques d'apprentissage artificiel peuvent êtres modélisés grâce à des fonctions à valeurs vectorielles. Les *noyaux à valeurs opérateurs* et leur espace de Hilbert à noyaux reproduisant à valeurs vectorielles associés donnent un cadre théorique et pratique pour apprendre de telles fonctions, étendant la littérature existante des noyaux scalaires. Cependant, lorsque les données sont nombreuses, ces méthodes sont peu utilisables, ne passant pas à l'échelle, car elle nécéssite une quantité de mémoire évoluant quadratiquement et un temps de calcul évoluant cubiquement vis à vis du nombre de données, dans leur implémentation la plus naïve. Afin de faire passer les *noyaux à valeurs opérateurs* à l'échelle, nous étendons une technique d'approximation stochastique introduite dans le cadres des noyaux scalaires. L'idée est de tirer parti d'une fonction de redescription charactétisant le *noyau à valeurs opérateurs*, dont les fonctions associées vivent dans un espace de dimension infinie, afin d'obtenir un problème d'optimisation linéaire de dimension finie.

Dans cette thèse nous développons dans un premier temps un cadre général afin de permettre l'approximation de noyaux de Mercer définits sur des groupes commutatifs localement compacts et étudions leurs propriétés ainsi que la complexité des algorithmes en découlant. Dans un second temps nous montrons des garanties théoriques en bornant l'erreur comise par l'approximation, avec grande probabilité. Enfin, nous mettons en évidence plusieurs applications des Représentations Opérateurs Aléatoires de Fourier (ORFF) telles que la classification multiple, l'apprentissage multi-tâche, la modélisation de séries temporelles, la régression fonctionelle et la détection d'anomalies. Nous comparons également ce cadre avec d'autres méthodes de la littérature et concluons par des perspectives à moyen et long terme.