Distribution-Dissimilarities in Machine Learning

Carl-Johann Simon-Gabriel

2018

Distribution-Dissimilarities in Machine Learning

Dissertation

der Mathematisch-Naturwissenschaftlichen Fakultät
der Eberhard Karls Universität Tübingen
zur Erlangung des Grades eines

Doktor der Naturwissenschaften

(Dr. rer. nat.)

vorgelegt von

Carl-Johann P.-M. Simon-Gabriel

aus Neuilly-sur-Seine/Frankreich

Tübingen
2018

Gedruckt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der
Eberhard Karls Universität Tübingen

Tag der mündlichen Qualifikation:	17.12.2018
Dekan:	Prof. Dr. Wolfgang Rosenstiel
1. Berichterstatter:	Prof. Dr. Bernhard Schölkopf
2. Berichterstatter:	Prof. Dr. Ulrike von Luxburg

To my parents, who gave me everything.
To my wife, who became my everything.

Abstract

While point-dissimilarities and classifiers are the core of machine learning since its beginnings, distribution-dissimilarities have long seemed a mere theoretical tool for statistical proofs. But both are closely connected: training a binary classifier – more precisely, a score-function – indeed amounts to computing a distribution-dissimilarity. The smaller the classifier's capacity, the weaker the resulting dissimilarity. Almost all usual dissimilarities are classifier-based. But they happen to be extremely strong: total variation, Hellinger distance, KL-divergence, etc. Weakening them has many advantages: they can get easier to compute and stop saturating on samples. But only up to a certain point, after which they also stop providing enough discrimination. So, what is the right capacity?

We study this question first on maximum mean discrepancies (MMD), a class of weakened total variation dissimilarities. We show that they stay perfectly discriminative if and only if the classifier has enough capacity to approximate a unit ball of continuous functions. Surprisingly, in that case, and only that case, the dissimilarity also stays strong enough to metrize the weak convergence of probability measures. Similar results are provided for what we call *targeted* convergence, as opposed to *global* convergence. We then provide straight-forward applications in the context of probabilistic programming and the estimation of functions of random variables. We also show that MMDs can be extended from probability measures to generalized measures, called Schwartz-distributions. They enable us to work with derivatives of probability measures, even when those measures are discrete, like any empirical measure. This leads to new results on kernel Stein discrepancies, an MMD specifically designed for sample-quality tests.

We then turn towards generative models. Contrary to their usual presentation, we introduce several popular models – GANs, VAEs, etc. – as a dissimilarity minimization task. All of them, we will see, minimize approximate f-divergences. So we complement them with another family of distribution-dissimilarities: optimal transport metrics. Our approach leads to *Wasserstein auto-encoders* and unveils new links between VAEs and GANs.

Finally, we focus on two specific deficiencies of the weak distribution-dissimilarities used in generative modeling: mode-collapse and adversarial examples. Mode-collapse is a state where the generator produces only samples of a specific type, ignoring the overall data-diversity. We propose an AdaBoost-like solution called *AdaGAN*. It trains several generators sequentially. Each new generator views an automatically reweighted dataset that focuses only on regions not covered by the previous generator. After training, the generators get combined into a single, diversified generative mixture.

As for adversarial examples, they are targeted, but typically imperceptible sample perturbations, that can break even the most accurate classifiers. Because most distribution-dissimilarities used in generative modeling are classifier-based, adversarial vulnerability shows that they are far from a human-like dissimilarity: two almost identical samples can look completely different to them. We explain why. Contrary to human perception, where higher resolution helps, we show, both empirically and theoretically, that the adversarial vulnerability of feed-forward networks increases as the square-root of the input-dimension, almost independently of the architecture. Our findings strongly suggest that, to build robust classifiers and dissimilarities with human-like perception, we need to significantly rethink our network architectures.

Zusammenfassung

Punktdissimilaritäten und Klassifikatoren sind seit Anfang Kernbestand des maschinellen Lernens. Verteilungsdissimilaritäten dagegen blieben lange nur ein theoretisches Werkzeug für statistische Beweise. Dabei sind beide eigentlich kaum trennbar: wer einen binären Klassifikator trainiert, genauer gesagt, dessen Score-Funktion, der berechnet eigentlich eine Verteilungsdissimilarität. Je kleiner die Kapazität des Klassifikators, desto schwächer die daraus resultierende Dissimilarität. Nahezu alle üblichen Dissimilaritäten sind klassifikatorbasiert: Gesamtvariation, Hellingerdistanz, KL-Divergenz, etc. Aber sie sind auch außerordentlich stark. Sie zu schwächen, hat viele Vorteile: sie werden oft einfacher zu berechnen und schwieriger zu sättigen. Doch schwächt man sie zu sehr, so sorgen sie nicht mehr für genügend Diskriminierung. Was, also, ist die richtige Schwächung, die richtige Kapazität des zugrundeliegenden Klassifikators?

Wir untersuchen diese Frage zunächst anhand von MMDs (*engl.* maximum mean discrepancies), einer Klasse von geschwächten totalen Variations Dissimilaritäten. Wir beweisen, dass MMDs genau dann perfekt diskriminierend sind, wenn der Klassifikator über genügend Kapazität verfügt, um eine Einheitskugel kontinuierlicher Funktionen zu approximieren. Überraschenderweise bleibt auch genau dann die Dissimilarität stark genug, um die schwache Konvergenz über Wahrscheinlichkeitsmaße zu dominieren. Ähnliche Ergebnisse liefern wir auch für das neu eingeführte Konzept der *gezielten* Konvergenz (statt der üblichen, *globalen* Konvergenz). Anschließend werden Anwendungen im Rahmen der probabilistischen Programmierung und Schätzung von Funktionen von Zufallsvariablen geboten. Nebenbei stellen wir fest, dass MMDs sich leicht von Wahrscheinlichkeitsmessungen auf Schwartz-Verteilungen verallgemeinern lassen. Letztere ermöglichen es, mit Ableitungen von Wahrscheinlichkeitsmaßen zu arbeiten, und zwar auch von diskreten Maßen. Dies führt zu neuen Ergebnissen über Kernel-Stein-Diskrepanzen, also bestimmte MMDs, die speziell zur Anpassungsgütetests gedacht sind.

Danach wenden wir uns generativen Modellen zu. Im Gegensatz zu ihrer üblichen Präsentation führen wir gängige Modelle wie GANs und VAEs als Dissimilaritätminimierungsmodelle ein. Sie alle minimieren nämlich approximative f-Divergenzen. Wir ergänzen diese Modelle mit einer weiteren Klasse von Dissimilaritätminimierungsmodellen: solche, die optimale Transportmetriken minimieren. Unser Ansatz führt zu *Wasserstein Auto-Encodern* und enthüllt neue Verbindungen zwischen VAEs und GANs.

Schließlich konzentrieren wir uns auf zwei spezifische Defizite solcher Verteilungsdissimilaritäten, die bei generativen Modellen verwendet werden: Modus-Kollaps (*engl.* mode-collapse) und gegnerische Beispiele (*engl.* adversarial examples). Modus-Kollaps ist ein Zustand, in dem der Generator nur noch Proben eines bestimmten Typs erzeugt und dabei die gesamte Datendiversität ignoriert. Wir schlagen eine AdaBoost-ähnliche Lösung vor: *AdaGAN*. Dabei werden nacheinander mehrere Generatoren trainiert. Für jeden Generator wird der Datensatz automatisch so neu gewichtet, dass er sich nur auf jene Bereiche konzentriert, die nicht durch die vorherigen Generatoren abgedeckt wurden. Nach dem Training werden die Generatoren zu einem einzigen, diversifiziertem, generativen Modell zusammengefügt.

Bei gegnerischen Beispielen geht es um gezielte, meist aber nicht wahrnehmbare Veränderungen der Eingangsdaten, die selbst die besten neuronalen Klassifikatoren verwirren können. Da die meisten Verteilungsdissimilaritäten generativer Modelle auf solchen Klassifikatoren beruhen, beweisen gegnerische Beispiele, dass solche Dissimilaritäten kaum der menschlichen Wahrnehmung entsprechen. Wir erklären, warum. Beim Menschen hilft es, Bilder in höherer Auflösung zu sehen. Bei feed-forward Netzwerken dagegen zeigen wir sowohl empirisch als auch theoretisch, dass deren gegnerische Verletzlichkeit mit der Quadratwurzel der Datendimension zunimmt, ziemlich unabhängig von der Netzwerkarchitektur. Unsere Ergebnisse deuten nachdrücklich darauf hin, dass auf Netzwerk beruhende Klassifikatoren und Dissimilaritäten eine menschenähnliche Wahrnehmung nur bekommen könnten, indem wir deren Netzwerkarchitekturen neu überdenken.

Thanks!

Writing a thesis is an incredible opportunity. I could dedicate four entire years to what I probably like most: learning and thinking about fascinating new problems. But it would not have been possible and certainly less fun without so many fantastic people around, which I would like to thank here. In particular:

Bernhard Schölkopf, for his constant trust and support, without which this thesis would never have been possible; *Ulrike von Luxburg*, who was always available when I needed council; *Google*, and the *French and German tax payers*, for their generous financial support during my studies and my PhD; *Ilya Tolstikhin*, with whom I shared not only a room, but also many stimulating discussions and beautiful moments; *David Lopez-Paz*, who always seemed to believe in me, and convinced me to apply to Facebook AI Research for an amazing internship; *Yann Ollivier*, who, despite his tight schedule and seniority, always found time to fruitfully help me right down to the slightest details of our common research; *Léon Bottou*, a man with visions and the patience to share them; *Olivier Bousquet* and *Sylvain Gelly*, for being so reliable, available, and involved in our common projects; *Lester Mackey*, for his precious and kind guidance on kernel Stein discrepancies; *Jonas Peters*, always ready to proof-read my papers *and* their bibliography, and constantly pushing me both in music and teaching; *Edgar Klenske*, for kindly providing his thesis' source file as template for this work; and so many other fabulous collegues, collaborators and friends such as Mateo Rojas-Carulla, Tatiana Fomina, Sebastian Gomez-Gonzalez, Martin Arjovsky, Motonobu Kanagawa, Paul Rubinstein, Krikamol Muandet, Diego Fiovaranti, Arthur Gretton, Manuel Gomez-Rodriguez, etc.

Of course, it is one thing to acknowledge the support I got during my thesis, but it would be ungrateful not to mention those that prepared its foundations; in particular my wonderful "prépa" teachers, *Frédéric Cuvellier*, *Thierry Meyer*, *Frédéric Paviet-Salomon* and *Patrick Génaux*, all extraordinary in their way; and my primary school teachers, *Frau Christea* and *Madame Garry*, who taught me the joys of study and work.

And last but not least, there are no words to thank my family, my parents and my wife. Their love rendered all the rest... Merci.

Contents

Abbreviations & Acronyms

∫spd	integrally strictly positive definite
AAE	adversarial auto-encoder
AVB	adversarial variational Bayes
CelebA	dataset of celebrity faces with some attributes
CIFAR-10	dataset with 10 classes from the Canadian Institute For Advanced Research
CNN	convolutional neural network
cpd	conditionally positive definite
GAN	generative adversarial network
i.e.	id est
iff	if and only if
iid	independent and identically distributed
ImageNet	dataset with 1000 classes
IPM	integral probability metric
JS	Jensen-Shannon (divergence)
KL	Kullback-Leibler (divergence)
KME	kernel mean embedding
KPP	kernel probabilistic programming
loc. cv.	locally convex
ML	machine learning
MLP	multi-layer perceptron
MMD	maximum mean discrepancy
MNIST	digit dataset from the Mixed National Institute of Standards and Technology
OT	optimal transport/trasnportation
R.V.	random variable
resp.	respectively
RGB	Red-Green-Blue (for colored images)
s.t.	such that
SGD	stochastic gradient descent
SNR	signal-to-noise ratio
spd	strictly positive definite
TV	total variation (metric/divergence)
TVS	topological vector space
UGAN	unrolled generative neural network
VAE	variational auto-encoder
WAE	Wasserstein auto-encoder
WGAN	Wasserstein generative adversarial network
wrt	with respect to

Symbols & Notations

Points, Vectors, Sets and Vector Spaces

$\mathbb{C}$	Complex numbers
E	Locally convex, Hausdorff, topological vector space
$\mathbb{N}$	Non-negative integers
$\mathbf{p}$	Element of $\mathbb{N}^d$ and/or path in a (network-) graph
$\mathbb{R}$	Real numbers
$\mathcal{X}$, $\mathcal{Z}$, $\mathcal{Y}$	Arbitrary sets, equipped with a Hausdorff topology and the Borel sigma-algebra. Usually: input, latent and output/label space resp.
x, y, z, ...	Scalar or arbitrary points
$\mathbf{x}$, $\mathbf{y}$, $\mathbf{z}$, ...	Typically Vectors
X, Y, Z, ...	Random variables

Functions and Function Spaces

$\mathbb{1}$, $\mathbb{1}_S$	Constant function 1, and indicator function of set S resp.
f	Real-valued convex function s.t. $f(1) = 0$, used to define an f-divergence
G	Generator function (usually defined by a network, for GAN-, VAE-like algorithms)
k, k_x, k_z, k_{xy}	Mercer kernels, defined over an arbitrary input space, $\mathcal{X}$, $\mathcal{Z}$, and $\mathcal{X} \times \mathcal{Y}$ resp.
κ	Stein kernel derived from kernel k
$\mathcal{L}$	Loss function
μ_X^k, $\hat{\mu}_X^k$	Kernel mean embedding of random variable X w.r.t. kernel k, and its estimator
$\Phi_k(P)$	Kernel mean embedding of distribution P w.r.t. kernel k
φ	Scalar-valued test function
$\mathscr{C}$, $\mathscr{C}_b$	Continuous (resp. and bounded) scalar-valued functions
$\mathscr{C}^m$, $\mathscr{C}_b^m$	Scalar-valued, m-times continuously differentiable functions (resp. with all their partial-derivatives bounded up to order m)
$\mathbb{C}^{\mathcal{X}}$	All functions from $\mathcal{X}$ to $\mathbb{C}$
$\mathcal{F}$	Topological vector space of scalar-valued functions.
$\mathcal{F}_{\text{div}}$	Set of (convex) f-divergence functions $f : (0, \infty) \to \mathbb{R}$ s.t. $f(1) = 0$
$\mathcal{G}$	Set of attainable generator functions
$\mathcal{H}$, $\mathcal{H}_k$	Reproducing kernel Hilbert space (RKHS) with kernel k
L^m	Lebesgue-space of m-integrable functions

Distributions and Distribution Spaces

$\mathcal{D}$	Arbitrary (sub)set of Schwartz-distributions
$\mathcal{D}^m$	Schwartz-distributions of order m
$\mathcal{D}_{L^1}^m$	Integrable Schwartz-distributions of order m: dual of $\mathscr{C}_b^m$

$\mathscr{E}^m$	Schwartz-distributions of order m with compact support: dual of $\mathscr{C}^m$
$\Gamma(P_X, P_Y)$	Couplings (i.e. joint probability distributions) whose marginal laws are P_X and P_Y
$\mathcal{M}_c$	Signed measures with compact support
$\mathcal{M}_\delta$	Signed measures with *finite* support
$\mathcal{M}_f$	Signed finite measures
$\mathcal{M}_r$	Signed regular measures
$\mathcal{N}(X; \alpha, \sigma^2)$	Random variable X following a Gaussian distribution with mean α and variance σ^2
$\mathcal{P}$	Probability measures
P, Q	Arbitrary probability distribution
p, q, p_X	Density of distributions P, Q & P_X resp. w.r.t. the Lebesgue measure
dP, dQ	Density of distributions P & Q w.r.t. an arbitrary reference measure μ
P_X, $P_{X,Y}$	Probability (resp. joint probability) distribution of random variable X (resp. (X, Y))
$P_{X\|Y}$, $Q_{X\|Y}$	Conditional probability distribution of X given Y
$X \sim P$	Random variable X has distribution P
$X \perp\!\!\!\perp Y$	Random variables X and Y are indendent
$\mathbb{E}_P[\varphi(X)]$	Expectation of random variable $\varphi(X)$ when $X \sim P$
$Var X$	Variance of random variable X
$\mathbf{s}_Q$	KSE score function with target-density q

Norms, Distances, Dissimilarities

$\|x\|$, $\|\mathbf{p}\|$	Module of scalar x, & $\sum_{i=1}^d p_i$ if $\mathbf{p} = (p_1, \dots, p_d) \in \mathbb{N}^d$
$\|\cdot\|$, $\|\cdot\|_p$, $\|\cdot\|_k$	Arbitrary-, ℓ_p-, and RKHS- (with kernel k) norms
$\|\!\|\cdot\|\!\|$	Dual norm of $\|\cdot\|$
$\langle\cdot\,,\cdot\rangle$, $\langle\cdot\,,\cdot\rangle_k$	Arbitrary and RKHS inner products
$\|P\|_k$, $\langle P\,,\,Q\rangle_k$	RKHS norm and inner products of the kernel mean embeddings of P and Q
$D(P\|Q)$	Arbitrary dissimilarity between probability distributions P and Q
$\mathrm{MMD}_k(P, Q)$	Maximum mean discrepancy between P and Q w.r.t. kernel k, i.e. $\|P - Q\|_k$
$\mathrm{KSD}_{k,P}(Q)$	Kernel Stein discrepancy between Q and target P, i.e. $\mathrm{MMD}_\kappa(P, Q)$.
$P_n \to_b P$, $P_n \to_\sigma P$, $P_n \to_w P$, $P_n \to_{\|\cdot\|_k} P$, $P_n \to_\alpha P$	Convergence of $(P_n)_n$ to P in bounded, weak, weak-*, RKHS and Wasserstein-α topologies resp. See Table 1.2 and Prop. 3.1.1

Miscalleneous

$\partial_x \varphi$, $\partial^{\mathbf{p}} \varphi$	partial derivative of φ w.r.t. x & $\frac{\partial^{\|\mathbf{p}\|} f}{\partial^{p_1} x_1 \partial^{p_2} x_2 \cdots \partial^{p_d} x_d}$, where $\mathbf{p} = (p_1, \dots, p_d) \in \mathbb{N}^d$
$\boldsymbol{\partial}\varphi$, $\boldsymbol{\partial}_{\mathbf{x}}\varphi$	gradient & gradient w.r.t. vector $\mathbf{x}$ of function φ
$\operatorname{dom}\varphi$	Input domain of function φ
$\mathbf{d}_{\mathbf{p}}$	Path-degree vector of path $\mathbf{p}$ in a network
$\mathcal{P}(x, o)$	Set of all paths $\mathbf{p}$ from node x to node o
$D(x, o)$	set of path-degree vectors for all paths going from node x to node o.
$\mathscr{F}\varphi$	Fourier transform of φ
$\operatorname{span} S$	Linear span of set S
$E_1 \hookrightarrow E_2$	Continuous inclusion, i.e. $E_1 \subset E_2$ and the canonical embedding is continuous
$\operatorname{supp} P$	Support of distribution P

Introduction

FOUR YEARS AGO, Goodfellow et al. [37] proposed a new algorithm that gained instant popularity: Generative Adversarial Nets (GANs). To generate fake but realistically looking data, they proposed to use two networks – a generator and a discriminator – with opposite goals: the discriminator tries to distinguish true from fake data, while the generator produces fake data that tries to fool the discriminator. Both networks permanently evolve while adapting to the strengths and weaknesses of each other until the generator starts producing realistic data.

[37] Goodfellow et al., *Generative Adversarial Nets*, 2014

Remarkably, GANs do not need any predefined measure to quantify what it means to "look realistic" – or so it seems. Where previous algorithms used refined, closed-form dissimilarity measures such as the KL-divergence or the total variation between fake and true probability distributions P and Q, GANs judge the quality of P by their discriminator's ability to distinguish it from Q. They optimize P to minimize the average reward of a (logit-) discriminator φ trained over a set of functions $\mathcal{F}$ to distinguish P from Q. It turns out that the maximal average reward of φ is a distribution-dissimilarity too: we call it a *classifier-based* dissimilarity. For a loss function $\mathcal{L}$ (the negative reward), it can formally be written as

GANs Minimize a Classifier-Based Dissimilarity

$$D_{\mathcal{L},\mathcal{F},\pi}(P \,\|\, Q) := \sup_{\varphi \in \mathcal{F}} \mathop{\mathbb{E}}_{X,C} -\mathcal{L}(\varphi(X), C)\,, \tag{0.1}$$

where (X, C) denotes a sample X coming from P if $C = 1$ and from Q if $C = 0$, and where π is the (prior) probability that $C = 1$.[1]

[1] The dependence on π becomes clear when noting that $\mathbb{E}_{X,C}\,\mathcal{L}(\varphi(X), C) = \begin{cases} \pi\,\mathbb{E}_X\,\mathcal{L}(\varphi(X), 1) + \\ (1-\pi)\,\mathbb{E}_X\,\mathcal{L}(\varphi(X), 0). \end{cases}$

Classifier-based dissimilarities are extremely general: by varying $\mathcal{L}$, $\mathcal{F}$ and π, one covers all integral probability metrics (IPMs) and f-divergences. That includes the KL-, reverse KL-, Jensen-Shannon- and squared-Hellinger divergences, the total variation-, Wasserstein-1-, Dudley- and maximum mean discrepancy (MMD) distances, etc. The original GAN algorithm used a logistic loss $\mathcal{L}$ and $\pi = 1/2$. In that case $D_{\mathcal{L},\mathcal{F},\pi}$ happens to approximate the Jensen-Shannon divergence more and more accurately with growing capacity of $\mathcal{F}$. But could we use other f-divergences or IPMs as proposed by [82]? What would be the pros and cons? More generally:

[82] Nowozin et al., *f-GAN*, 2016

What are the properties of such distribution-dissimilarities?
What are they good for?

This is the core question which serves as leitmotiv to the present thesis. Rather than trying to answer it exhaustively, we will focus on the properties of some *selected* distribution-dissimilarities and present *selected* applications, GANs being only one of them. Now, we propose to give a brief overview of the major distribution-dissimilarities that we will encounter throughout this thesis and then present the contents of the chapters to come.

Name	Abbrev.	$\mathcal{F}$
Total Variation	TV	$\{\varphi \in \mathscr{C} : \|\varphi\|_\infty \leqslant 1\}$
Wasserstein-1	W_1	$\{\varphi \in \mathscr{C} : \|\varphi\|_L \leqslant 1\}$
Bounded Lipschitz (Dudley)	BL	$\{\varphi \in \mathscr{C} : \|\varphi\|_\infty + \|\varphi\|_L \leqslant 1\}$
Max. Mean Discrepancy	MMD	$\{\varphi \in \mathcal{H}_k : \|\varphi\|_k \leqslant 1\}$

Table 0.1: IPM examples. $\|\cdot\|_\infty$, $\|\cdot\|_\infty$ and $\|\cdot\|_k$ denote the supremum, the Lipschitz and an RKHS norm (of RKHS $\mathcal{H}_k$ with kernel k) respectively.

0.1 Distribution Dissimilarities

Let $\mathcal{X}$ be the input space, typically a Polish space (separable, complete and metrizable) equipped with its Borel sigma-algebra. We call *distribution-dissimilarity* or simply *dissimilarity* any real function D of two probability distributions P, Q defined on $\mathcal{X}$ that gets minimized when $P = Q$. $D(P\|Q)$ is typically non-negative and equal to 0 when $P = Q$, but not necessarily. We will encounter three big families of distribution-dissimilarities: Integral Probability Metrics (IPMs), f-divergences, and Optimal Transport (OT) dissimilarities. Let us present them here, and then see how they relate to the classifier-based dissimilarity (0.1).

Definition of Distribution-Dissimilarity

0.1.1 Textbook Dissimilarities

There are several ways to present IPMs, f-divergences and OT dissimilarities. They can usually be defined using either their primal or their dual formulation. Both are equivalent. Here, we opt for the more unusual *dual* definitions, so as to make the link between all three dissimilarity families appear more explicitly. Indeed, all of them can be written as

$$D(P\|Q) = \sup_{(\varphi,\psi)\in(\mathcal{F},\mathcal{G})} P(\varphi) - Q(\psi)\,, \tag{0.2}$$

IPMs, f-Divergences and OT Dissimilarities All Verify (0.2).

where $P(\varphi)$ is a short-hand for $\mathbb{E}_{X\sim P}\,\varphi(X)$ and $(\mathcal{F},\mathcal{G})$ denotes a set of measurable function pairs (φ,ψ) such that φ and ψ are P- and Q-integrable respectively.

Integral Probability Metrics (IPM) correspond to $(\mathcal{F},\mathcal{G}) = \{(\varphi,\varphi) : \varphi \in \mathcal{F}\}$, where $\mathcal{F}$ is some fixed (usually balanced) set of functions.

$$D_{\mathcal{F}}(P\|Q) = \sup_{\varphi\in\mathcal{F}} P(\varphi) - Q(\varphi)\,. \tag{0.3}$$

IPMs satisfy the axioms of a metric (or distance), except that they may take infinite values ($D_{\mathcal{F}}(P\|Q) = \infty$) and need not be definite, id est (i.e.) perfectly discriminative.[2] Various IPM examples are given in Table 0.1.

[2] *Definite* (or *perfectly discriminative*) dissimilarity: $D_{\mathcal{F}}(P\|Q) = 0 \Rightarrow P = Q$.

Name	Abbrev.	$f(s)$	$f^*(t)$	$\operatorname{dom} f^*$
Total Variation	TV	$\lvert s-1\rvert$	t	$-1 \leqslant t \leqslant 1$
Kullback-Leibler	KL	$s \log s$	$\exp(t-1)$	$\mathbb{R}$
Reverse-KL	/	$-\log s$	$-1-\log(-t)$	$t < 0$
Squared Hellinger	/	$(\sqrt{s}-1)^2$	$\frac{t}{1-t}$	$t < 1$
Pearson χ^2	χ^2	$(s-1)^2$	$t^2/4+t$	$\mathbb{R}$
Jensen-Shannon	JS	$-(s+1)\log\frac{1+s}{2}+s\log s$	$-\log(2-\exp t)$	$t < \log 2$

Table 0.2: Examples of f-divergences, with f, its Fenchel conjugate f^* and the latter's input domain $\operatorname{dom} f^*$. Taken from [82]

f-Divergences correspond to $(\mathcal{F}, \mathcal{G}) = \{(\varphi, f^*(\varphi))\}$ where $f : (0,\infty) \to \mathbb{R}$ is any fixed convex function satisfying $f(1) = 0$, f^* is its Fenchel conjugate,[3] and where φ spans the set of all measurable functions from $\mathcal{X}$ to the input domain $\operatorname{dom} f^*$ of f^*. See [82] & [85].

$$D_f(P \,\|\, Q) = \sup_{\varphi\,:\,\mathcal{X} \to \operatorname{dom} f^*} P(\varphi) - Q(f^*(\varphi)) \tag{0.4}$$

[3] Fenchel conjugate definition: $f^*(t) := \sup_{s \in \operatorname{dom} f} st - f(s)$

[82] Nowozin et al., *f-GAN*, 2016
[85] Peyré and Cuturi, *Computational Optimal Transport*, 2018, Rmk 8.1

f-divergences may not satisfy the triangular inequality, but they are always non-negative and definite. Examples of f-divergences include the Kullback-Leibler (KL), reverse-KL, Jensen-Shannon (JS), squared Hellinger and χ^2-divergences, and the TV distance. The latter corresponds to $f^*(x) = x$ for $-1 \leqslant x \leqslant 1$. It is the only f-divergence that is also an IPM. There are many connections between the different f-divergences, and between f-divergences and statistical information [67].

[67] Liese and Vajda, *On Divergences and Informations in Statistics and Information Theory*, 2006

Optimal transport (OT) dissimilarities correspond to $(\mathcal{F}, \mathcal{G}) = \{(\varphi, \psi) : \forall x, y \in \mathcal{X},\ \varphi(x) + \psi(y) \leqslant c(x,y)\}$, where $c(x,y)$ is any predefined, non-negative cost function. It is usually interpreted as the cost of transporting a unit mass from point x to point y.

$$D_c(P \,\|\, Q) = \sup_{\substack{(\varphi,\psi):\ \forall x,y \in \mathcal{X} \\ \varphi(x)+\psi(y) \leqslant c(x,y)}} P(\varphi) - Q(\psi)$$

In general, optimal transportation measures satisfy neither definiteness, nor the triangular inequality, but they are always non-negative and finite if c is. When c is a distance, then D_c (and more generally $D_{c^p}^{1/p}$) also satisfies the axioms of a distance, except that it may take infinite values. It is called the c-Wasserstein-1 (resp. c-Wasserstein-p) distance, or simply the Wasserstein-1 distance when c coincides with the underlying metric of $\mathcal{X}$. The c-Wasserstein-1 is an IPM, co-

inciding with the total variation metric when c is the discrete distance [121].

[121] Villani, *Optimal Transport: Old and New*, 2009, Chap. 6

0.1.2 Classifier-Based Dissimilarities

How do the previous three dissimilarity families relate to GANs and classifier-based dissimilarities, as discussed in (0.1)? Let us just illustrate here the main ideas of the answer on f-divergences. More details and links are given in Chapter 4.

The trick is to see every test function φ in the definition (0.4) of f-divergences as a score-function. For brevity, we may refer to φ as *the classifier*, but it is more accurate to think of it as the logit function of a binary classifier. It takes an input $x \in \mathcal{X}$ and tries to determine if x was drawn from distribution P or from Q. The higher the score $\varphi(x)$, the more it attributes x to P. Now, define the reward for score $\varphi(x)$ as $\varphi(x)$ if x comes from P, and $-f^*(\varphi(x))$ otherwise. Then $P(\varphi) - Q(f^*(\varphi))$, i.e. $\mathbb{E}_{X\sim P}[\varphi(X)] - \mathbb{E}_{X\sim Q}[f^*(\varphi(X))]$, is the average reward of φ when x comes with equal probability from P or Q. And $D_f(P\,\|\,Q)$ is the maximal average reward that we can get when φ is optimized over all functions from $\mathcal{X}$ to $\operatorname{dom} f^*$. f-divergences are hence, by definition, *classifier-based* dissimilarities. They satisfy (0.1) for the particular choices $\pi = 1/2$ (equal prior probability to come from P or Q), $\mathcal{F} = [-1,1]^{\mathcal{X}} := \{\varphi : \mathcal{X} \to \operatorname{dom} f^*\}$, and

f-Divergences are Classifier-Based Dissimilarities

$$\mathcal{L}(\varphi(x), y) = \begin{cases} -\varphi(x) & \text{if } y = 1 \quad \text{(i.e. if } x \text{ comes from } P\text{)} \\ f^*(\varphi(x)) & \text{if } y = 0 \quad \text{(i.e. if } x \text{ comes from } Q\text{).} \end{cases}$$

f-divergences are particularly strong dissimilarity measures, because of their huge set of test functions $\mathcal{F}$. As we will see in a moment, that often makes them too discriminative. So we may want to deliberately reduce the set of test functions $\mathcal{F}$. In a sense, we already did so with IPMs. They are weak total variation dissimilarities, that replace the set of all measurable functions $\varphi : \mathcal{X} \to [-1,1]$ by a smaller one. Doing the same, not just with total variation, but with any f-divergence leads to *restricted f-divergences*.

They can be Weakened by Reducing the Set of Test Functions $\mathcal{F}$

Restricted (or approximate) f-divergences have the same objective than f-divergences, but are optimized over a smaller set of test functions $\mathcal{F}$:

$$D_{f,\mathcal{F}}(P\,\|\,Q) = \sup_{\varphi\in\mathcal{F}} P(\varphi) - Q(f^*(\varphi))\,.$$

IPMs and f-divergences are thus particular kinds of *restricted* f-divergences: IPMs fix f and vary $\mathcal{F}$, while f-divergences keep $\mathcal{F}$ as big as possible and vary f. Restricted f-divergences are in particular

classifier-based dissimilarities that restrict the capacity of their score-function set to $\mathcal{F}$. They are hence weaker than f-divergence, in the sense that

$$D_{f,\mathcal{F}}(P \,\|\, Q) \leqslant D_f(P \,\|\, Q) \ .$$

The lower bound converges to the upper limit when $\mathcal{F}$ grows towards $(\operatorname{dom} f^*)^{\mathcal{X}}$. But in practice, we may actually prefer the lower bound; the weaker dissimilarity. Let us explain why.

0.1.3 Why Weak Distribution Dissimilarities?

Probability theory and statistics traditionally work with strong distribution-dissimilarities; probably because they mainly deal with continuous probability distributions and because stronger assumptions can considerably simplify a proof. But applied machine learning (ML) is mainly about samples, i.e. empirical measures that have no density. We may for example want to do a two sample test, to evaluate if two samples come from a same distribution; or a sample-quality test, that checks if a sample could come from some predefined, usually continuous reference distribution. Such tests can be used in the context of MCMC sampling or of generative modeling, such as in GANs. For all of them, distribution-dissimilarities seem the right tool.

But here is the catch. Two random samples typically yield disjoint measures; and f-divergences typically saturate on such measures: total variation for example simply takes its maximal value2; the KL-divergence is even infinite. In a sense, that is not surprising. In theory, the two measures are perfectly distinguishable. So the average reward $D_f(P \,\|\, Q)$ of an optimally classifying score-function φ is as high as it can get: the dissimilarity saturates. This saturation masks a lot of relevant comparative information. Two distinct Dirac measures δ_x and δ_y, how ever close they are, always lie at the same, maximal TV-distance, 2. Of course, being distinct, they are theoretically perfectly distinguishable. But it would nevertheless help if the dissimilarity converged to 0 when $y \to x$. That is where weaker dissimilarities come into play.

Consider the total variation again, and imagine that, instead of choosing the test-function φ among all functions with output in $[-1, 1]$, we additionally required that it be 1-Lipschitz. $D_{\mathcal{F}}(\delta_x \,\|\, \delta_y)$ gets maximized only if there exists a test-function $\varphi \in \mathcal{F}$ such that (s.t.) $\varphi(x) = 1$ and $\varphi(y) = -1$. As illustrated in Figure 0.1, when $\mathcal{F} = [-1, 1]^{\mathcal{X}}$, this is the case as soon as $x \neq y$. But if φ is to be 1-Lipschitz, $|\varphi(x) - \varphi(y)| \leqslant |x - y|$. So, not only can't the condition be met if $|x - y| < 2$, but the new dissimilarity[4] $D_{\widetilde{\mathrm{BL}}}(\delta_y \,\|\, \delta_x)$ converges

[4] Technically, this new dissimilarity is not exactly the bounded Lipschitz (Dudley) one defined in Table 0.1, which is why we denote it $D_{\widetilde{\mathrm{BL}}}$ rather than D_{BL}. But both define equivalent norms, i.e. are "almost the same".

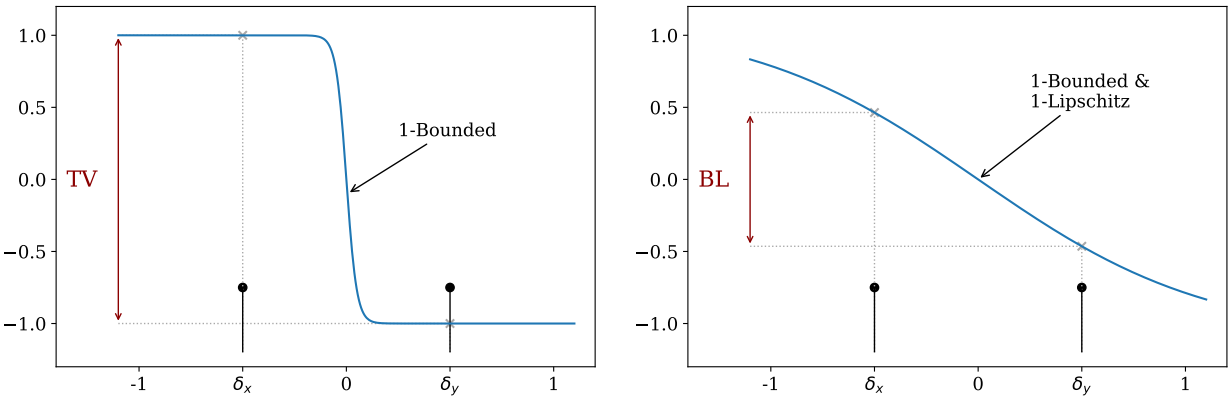


Figure 0.1: Two test functions φ that (almost) achieve $\sup_{\varphi \in \mathcal{F}} \delta_x(\varphi) - \delta_y(\varphi) = D_{\mathcal{F}}(\delta_x \| \delta_y)$ when $\mathcal{F}$ contains resp. all 1-bounded functions (**left**) and when it contains only 1-bounded 1-Lipschitz functions (**right**). The first dissimilarity (TV) is bigger than the second (bounded Lipschitz).

to 0 when $y \to x$. We say that $D_{\widetilde{BL}}$ metrizes a new, weaker topology:[5] while in total variation, $\delta_{y_n} \to \delta_x$ if and only if (iff) $y_n = x$ for all large enough n, in the new topology $\delta_{y_n} \to \delta_x$ iff $y_n \to x$. This new topology is much more informative, because, in a sense, it respects the underlying topology of $\mathcal{X}$.

Instead of simply imposing φ to be 1-Lipschitz, we could add more restrictions, decreasing $\mathcal{F}$ even further. That might ease the training procedure, i.e. the optimization of φ in $\mathcal{F}$. But it may also weaken the dissimilarity too much. Imagine for instance that we kept only one single function in $\mathcal{F}$. Then, no need to optimize φ anymore. But all distributions are at equal distance: the dissimilarity becomes totally uninformative. We hence have to find the right balance between a set $\mathcal{F}$ that is sufficiently small to get a computable dissimilarity measure, but sufficiently large to stay informative.

Of course, what it means to "be informative" depends on our goals. If our only goal is to distinguish true from fake image datasets, we may be content with a dissimilarity that sees no difference between any two different sets of fake images. But if we intend to generate realistic images, we'd better have a dissimilarity that sees differences even in between fake images, and can tell us which one looks more realistic. So the real question, at the core of this thesis, is not so much: how much functions can I take our of $\mathcal{F}$? But rather: *given my goals*, how should I choose $\mathcal{F}$; how should I choose my distribution-dissimilarity? Our three parts focus on three different aspects of this question.

[5] Formally, TV metrizes the *strong* converges of probability measures, while the bounded Lipschitz dissimilarity metrizes the *weak* one.

0.2 Thesis Outline

Our first part focuses on two questions: 1) if I want my dissimilarity to be perfectly discriminative, how should I choose the set of test-functions $\mathcal{F}$? 2) same question, if I want my dissimilarity to metrize a topology known as *weak convergence*? We study these questions on a particular family of IPM dissimilarities called maximum mean

discrepancies (MMDs). Those are weak total variation dissimilarities where $\mathcal{F}$ is the unit ball of a reproducing kernel Hilbert space (RKHS) $\mathcal{H}_k$. RKHSs are function spaces that are entirely determined by a (Mercer) kernel (function) k. That gives flexibility: by changing the kernel, we change the RKHS. It also makes RKHSs very rigid, which can be both good and bad. Good, because it can considerably easen computations: the MMD of two samples for example can always be expressed in closed form. Bad, because RKHS functions inherit many properties of the kernel, which can make them relatively small: if k is bounded, continuous, differentiable and/or smooth, then so are all the functions in the respective RKHS. Some RKHSs get so small, that their MMD is far from being perfectly discriminative. Others, however, are still big enough to be dense in the set of bounded continuous functions $\mathscr{C}^b$. Intuitively, that seems enough to preserve perfect discrimination. Chapter 1 confirms this intuition: it shows[6] that the MMD of a bounded continuous kernel is perfectly discriminative iff $\mathcal{H}_k$ is dense in $\mathscr{C}^b$. More surprising however is that this condition also happens to be necessary and sufficient to metrize weak-convergence. That is the main theorem of Chapter 1, Theorem 1.3.4, and possibly the main theorem of this thesis. Finally, Chapter 1 also shows that MMDs can be extended from probability measures to generalized measures called *Schwartz-distributions*. That will turn out very useful later on. Chapter 2 and 3 then propose two applications: one related to the weak-convergence metrization, and another that makes use of Schwartz-distribution embeddings.

[6] Theorem 1.2.2

Our second part focuses on distribution-dissimilarities in the context of generative models, such as in GANs and variational auto-encoders (VAEs). It implicitly asks: how can I build effective distribution-dissimilarities to learn to generate image data? To do so, Chapter 4 first lists, compares and links the objectives of existing generative algorithms and shows that they all do approximate f-divergence minimization. It then proposes a new, related objective, which implements approximate OT minimization: Wasserstein auto-encoders (WAE). Chapter 5 then remedies a deficiency of GANs and similar generative algorithms known as *mode collapse*, where the generator suddenly produces only one kind of datapoints, ignoring the true data diversity. The solution we propose – an algorithm called AdaGAN – can be applied to any generative method that minimizes a restricted f-divergence.

Our third and final part focuses on an appealing deficiency of all state-of-the-art network-based image-classifiers: their adversarial vulnerability. By adding imperceptible, but targeted perturbations to the input images, the accuracy of almost any such classifier can be drastically turned down. The perturbed and original sample would hence look almost the same to humans, but entirely different to the

classifier-based dissimilarity. Understanding the origins of adversarial vulnerability therefore seems a prerequisite for building dissimilarity measures that capture more human-like perception. But we make worrying findings. For humans, higher resolutions can but help. But for feed-forward networks, independently of their architecture, adversarial vulnerability increases as the square-root of the input dimension. This we show both theoretically and empirically. It strongly suggests that, despite their fantastic accuracies, we may want to rethink our network architectures, if they are to imitate, one day, human-like perception.

0.3 Underlying Material, Co-Workers and Contributions

This work relies on the following papers, which have all been co-written with colleagues and friends. I would like to acknowledge and thank them here, and briefly outline the extent of my personal contributions.

Chapter 1

C.-J. Simon-Gabriel and B. Schölkopf. *Kernel Distribution Embeddings: Universal Kernels, Characteristic Kernels and Kernel Metrics on Distributions*. In: *JMLR* (2018). to appear

Contributions: main ideas, and major part of theory and text.

Chapter 2

C.-J. Simon-Gabriel, A. Scibior, I. O. Tolstikhin, and B. Schölkopf. *Consistent Kernel Mean Estimation for Functions of Random Variables*. In: *NIPS*. 2016, pp. 1732–1740

Contributions: a major part of the theory (especially the proofs of the main Theorems 2.2.1 & 2.2.3) and a substantial part of the text. The code for the little experiment of Figure 2.1 was graciously provided by Krikamol Muandet. Compared with the paper, I also shifted the focus from convergence of KME estimators to the convergence of discrete random samples (see Section 2.1.2).

Chapter 3

C.-J. Simon-Gabriel and L. Mackey. *Targeted Convergence Characteristics of Maximum Mean Discrepancies and Kernel Stein Discrepancies*. In: *preprint* (2018)

Contributions: the original idea of targeted convergence and its application to kernel Stein discrepancies came from L. Mackey, but I worked out most of the theory, proofs and text.

Chapter 4

O. Bousquet, S. Gelly, I. Tolstikhin, C.-J. Simon-Gabriel, and

B. Schölkopf. *From optimal transport to generative modeling: the VEGAN cookbook*. 2017. arXiv: 1705.07642

Contributions: I was mainly involved in the initial ideas and experiments, and afterwards in the analysis discussions. While I contributed only little to the original text, I significantly modified it for this thesis. The proofs however were mainly O. Bousquet's work. I also included experimental results and images of [116], which is a follow-up paper that further implements and tests our main algorithm.

[116] Tolstikhin et al., *WAE*, 2018

Chapter 5

I. O. Tolstikhin, S. Gelly, O. Bousquet, C.-J. Simon-Gabriel, and B. Schölkopf. *AdaGAN: Boosting Generative Models*. In: *NIPS*. 2017, pp. 5424–5433

Contributions: I contributed to the original idea, the discussions, did a significant part of the experiments, and co-analyzed the results. The proofs however were mainly worked out by I. Tolstikhin and O. Bousquet. The initial idea came from B. Schölkopf during a joint discussion.

Chapter 6

C.-J. Simon-Gabriel, Y. Ollivier, B. Schölkopf, L. Bottou, and D. Lopez-Paz. *Adversarial Vulnerability of Neural Networks Increases With Input Dimension*. 2018. arXiv: 1802.01421

Contributions: The main ideas resulted from my work and discussions at Facebook AI Research during my internship there. I coded all experiments, and a major part of the theorems and proofs. The idea to generalize Theorem 6.2.1 and Corollary 6.2.3 from fully-connected and standard convolutional nets to general feed-forward nets (Theorem 6.2.2) came from Y. Ollivier. I worked out its proof under his guidance.

This thesis will not study the following co-authored papers:

B. Schölkopf, D. W. Hogg, D. Wang, D. Foreman-Mackey, D. Janzing, C.-J. Simon-Gabriel, and J. Peters. *Removing Systematic Errors for Exoplanet Search via Latent Causes*. In: *ICML*. 2015. arXiv: 1505.03036

B. Schölkopf, D. W. Hogg, D. Wang, D. Foreman-Mackey, D. Janzing, C.-J. Simon-Gabriel, and J. Peters. *Modeling Confounding by Half-Sibling Regression*. In: *PNAS* 113.27 (2016), pp. 7391–7398

H. Huang, G. M. Peloso, D. Howrigan, B. Rakitsch, C. J. Simon-Gabriel, J. I. Goldstein, M. J. Daly, K. Borgwardt, and B. M.

Neale. *Bootstrat: Population Informed Bootstrapping for Rare Variant Tests*. In: (2016). bioRxiv: 10.1101/068999

The German abstract (Zusammenfassung) was originally translated from English with the DeepL Translator [23], and significantly modified afterwards.

[23] DeepL, *DeepL Translator*, 2018

Part I

Maximum Mean Discrepancies

THIS FIRST PART focuses on a very particular class of dissimilarity measures: maximum mean discrepancies (MMDs). MMDs are IPMs, where the test functions $\varphi \in \mathcal{F}$ are the unit ball of a reproducing kernel Hilbert space (RKHS) $\mathcal{H}_k$ with kernel k. We assume that the reader is familiar with the basics of RKHS theory. For an introduction, see [94] or [79]. MMDs have many attractive features – such as being computable in closed form for any two samples – which gave them quick popularity after their introduction in the ML community in the early 2000s. But compared to more classical spaces such as the continuous bounded and/or Lipschitz functions, RKHS are relatively small, as they inherit many properties of their kernel: if k is continuously differentiable in each variable, so are all the functions of $\mathcal{H}_k$; if $x \mapsto k(x,x)$ is L^p-integrable, so are the functions of $\mathcal{H}_k$. This makes the resulting MMD weaker than more classical IPMs such as the total variation, Wasserstein-1 or Dudley distances. Chapter 1 studies how weak (or strong) MMDs actually are.

MMDs: A Case Study for Dissimilarity Weakening

[94] Schölkopf and Smola, *Learning with Kernels*, 2001
[79] Muandet et al., *Kernel Mean Embedding of Distributions*, 2017

A first weakness is that MMDs may not be perfectly discriminative – meaning that $D(P\|Q) = 0$ may not imply $P = Q$ – in which case the MMD is a semi-metric, but not a metric. Chapter 1 hence first focuses on identifying necessary and sufficient conditions for MMDs to be metrics. A second weakness is that the convergence defined by an MMD is weaker even than the one defined by the Dudley metric, which is already known as the *weak* (or weak-*) convergence in probability theory. But MMD-convergence need not be *strictly* weaker than this weak-convergence: some MMDs are known to *metrize* weak-convergence. Chapter 1 establishes an iff condition which, astonishingly, happens to coincide with the previous iff conditions: a bounded continuous kernel metrizes weak-convergences iff its MMD is a metric. Chapter 2 then develops a straight-forward application of this result: consistent estimation of function of random variables and kernel probabilistic programming.

MMDs, Perfect Discrimination & Weak Convergence

To establish the iff conditions in Chapter 1, we order, redefine and generalize different concepts that were introduced gradually and often independently in the ML literature, such as the definition of kernel mean embeddings (KMEs), and of universal, characteristic and strictly positive definite (spd) kernels. As a byproduct of this reorganization, we will see that KMEs and MMDs can be extended not only to signed measures (as opposed to probability measures), but also to generalized measures called *Schwartz-distributions*. Chapter 1 therefore also investigates some properties of these extensions. It shows in particular KME and differentiation operators commute (essentially, because both are linear). This means that KMEs (and MMDs) can take advantage of the greatest strength of Schwartz-distributions (and the reason they were originally introduced): unlike usual functions and measures, they are all indefinitely differen-

MMDs, KMEs & Extensions to Schwartz-Distributions

tiable. Chapter 3 will show how these insights can be used to prove consistency results for a special kind of MMD called a kernel Stein discrepancies (KSD). Remarkably, the main result, Theorem 3.2.3, is essentially a statement on probability distributions, but its proof explicitly makes use of Schwartz-distributions. Moreover, this proof is almost a copy-paste of a previous proof by [21], but where the introduction of Schwartz-distributions avoids strong assumptions and significantly generalizes the theorem. Chapter 3 therefore serves as a first illustration of the power of Schwartz-distribution combined with KMEs and MMDs.

KMEs of Schwartz-Distributions Apply to KSDs

[21] Chwialkowski et al., *A Kernel Test of Goodness of Fit*, 2016

1 Kernel Distribution Embeddings

The following chapter focuses on Maximum Mean Discrepancies (MMDs). Although MMDs are IPMs, they can – and will – also be introduced via kernel mean embeddings (KMEs), which are defined and studied in Section 1.1. KMEs linearly map signed measures to functions in the RKHS $\mathcal{H}_k$. The MMD of two measures then coincides with the RKHS distance between their embeddings. The advantage of this approach is that it gives access to the huge toolbox of linear functional analysis.

Like all IPMs, MMDs are semi-metrics. Our primary goal here is two-fold. First, determine when this semi-metric is a metric (Theorem 1.2.2). Second, determine when it metrizes the weak convergence of probability measures. Astonishingly, Theorem 1.3.4 will show that, for bounded continuous kernels, this metrization happens exactly when the MMD is a metric.

At the heart of these theorems are the notions of universal, characteristic and spd kernels. While originally introduced in very different contexts and with many variants, they were soon found to be connected in many ways as summarized by Figure 1 in [109]. But by handling separately all the many variants of these notions, the ML community overlooked the general duality principle that underlies all these connections. Our second contribution here is the unification of those many variants, which makes their link explicit, easy to remember, and immediate to generalize.

[109] Sriperumbudur et al., *Universality, Characteristic Kernels and RKHS Embedding of Measures*, 2011

As a byproduct, we will see that KMEs – and therefore MMDs – naturally extend, not only to signed measures, but also to generalized measures called *Schwartz-distributions*, which Section 1.4 will focus on. To avoid confusion, in this chapter, ***distribution*** designates any Schwartz-distribution, while ***measure*** specifically designates usual (signed) measures. For a short introduction to Schwartz-distributions and generalized differentiation see Appendix A.1.

In This Chapter:
$\{\textit{Measures}\} \subsetneq \{\textit{Distributions}\}$

Section 1.4 first proves the following calculus rules:

$$\left\langle f, \int k(.,x)\,dD(x) \right\rangle_k = \int \langle f, k(.,x)\rangle_k \,dD(x) \qquad \text{(Definition of KME)}$$

$$\left\langle \int k(.,y)\,dD(y), \int k(.,x)\,dT(x) \right\rangle_k = \int k(x,y)\,dD(y)\,d\bar{T}(x) \qquad \text{(Fubini)}$$

$$\int k(.,x)\,d(\partial^p S)(x) = (-1)^{|p|} \int \partial^{(0,p)} k(.,x)\,dS(x). \qquad \text{(Differentiation)}$$

While the first two lines extend standard KME formulae, the third one is specific to Schwartz-distributions. It uses the distributional derivative ('∂') which extends the usual derivative to measures and distributions (see Appendix A.1). Second, Section 1.4 proves that, for smooth and translation-invariant kernels, extending an injective KME from probability measures to distributions preserves the injectivity (Theorems 1.4.5 & 1.4.6). Thus, if the associated MMD is a probability-metric, then it is automatically a metric over these bigger distribution spaces.

The structure of this chapter roughly follows this exposition. After fixing our notations, Section 1.1 introduces KMEs of measures and distributions. In Section 1.2 we define the concepts of universal, characteristic and spd kernels and prove their equivalence. Section 1.3 compares convergence in MMD with other modes of convergence for measures and distributions. Section 1.4 focuses specifically on KMEs of Schwartz-distributions, and Section 1.5 concludes.

Definitions and Notations

The input set $\mathcal{X}$ of all considered kernels and functions will be locally compact and Hausdorff. This includes any Euclidian spaces or smooth manifolds, but no infinite-dimensional Banach-space. Whenever referring to differentiable functions or to distributions of order $\geqslant 1$, we will *implicitly* assume that $\mathcal{X}$ is an open subset of $\mathbb{R}^d$ for some $d > 0$.

A *kernel* $k : \mathcal{X} \times \mathcal{X} \to \mathbb{C}$ is a positive definite function, meaning that for all $n \in \mathbb{N}\backslash\{0\}$, all $\lambda_1, \dots, \lambda_n \in \mathbb{C}$, and all $x_1, x_2, \dots x_n \in \mathcal{X}$, $\sum_{i,j=1}^{n} \lambda_i k(x_i, x_j)\overline{\lambda_j} \geqslant 0$. For $p = (p_1, p_2, \dots, p_d) \in \mathbb{N}^d$ and $f : \mathcal{X} \to \mathbb{C}$, we define $|p| := \sum_{i=1}^{d} p_i$ and $\partial^p f := \frac{\partial^{|p|} f}{\partial^{p_1} x_1 \partial^{p_2} x_2 \cdots \partial^{p_d} x_d}$. For $m \in \mathbb{N} \cup \{\infty\}$, we say that f (resp. k) is m-times (resp. (m, m)-times) continuously differentiable and write $f \in \mathscr{C}^m$ (resp. $k \in \mathscr{C}^{(m,m)}$), if for any p with $|p| = m$, $\partial^p f$ (resp. $\partial^{(p,p)} k$) exists and is continuous. $\mathscr{C}_b^m$ (resp. $\mathscr{C}_{\to 0}^m$, $\mathscr{C}_c^m$) is the subsets of $\mathscr{C}^m$ for which $\partial^p f$ is bounded (resp. converges to 0 at infinity, resp. has compact support) whenever $|p| \leqslant m$. Whenever $m = 0$, we may drop the superscript m. By default, we equip $\mathscr{C}_*^m$ ($* \in \{\emptyset, b, 0, c\}$) with their natural topologies (see Introduction of [99] or [119]). We write $k \in \mathscr{C}_0^{(m,m)}$ whenever k is bounded, (m, m)-times continuously differentiable and for all $|p| \leqslant m$ and $x \in \mathcal{X}$, $\partial^{(p,p)} k(., x) \in \mathscr{C}_{\to 0}$.

We call *space of functions* and denote by $\mathcal{F}$ any locally convex (loc. cv.) topological vector space (TVS) of functions (see Appendix A.2). Loc. cv. TVSs include all Banach- or Fréchet-spaces and all function spaces defined in this chapter.

[99] Simon-Gabriel and Schölkopf, *Kernel Distribution Embeddings - arXiv*, 2016
[119] Treves, *Topological Vector Spaces, Distributions and Kernels*, 1967

The dual $\mathcal{F}'$ of a space of functions $\mathcal{F}$ is the space of *continuous* linear forms over $\mathcal{F}$. We denote $\mathcal{M}_\delta$, $\mathscr{E}^m$, $\mathcal{D}^m_{L^1}$ and $\mathcal{D}^m$ the duals of $\mathbb{C}^\mathcal{X}$, $\mathscr{C}^m$, $\mathscr{C}^m_{\to 0}$ and $\mathscr{C}^m_c$ respectively. By identifying each signed measure μ with a linear functional of the form $f \mapsto \int f \, d\mu$, the Riesz-Markov-Kakutani representation theorem (see Appendix A.2) identifies $\mathcal{D}^0$ (resp. $\mathcal{D}^0_{L^1}$, $\mathscr{E}^0$ and $\mathcal{M}_\delta$) with the set $\mathcal{M}_r$ (resp. $\mathcal{M}_f$, $\mathcal{M}_c$, $\mathcal{M}_\delta$) of signed regular Borel measures (resp. with finite total variation, with compact support, with finite support). By definition, $\mathcal{D}^\infty$ is the set of all Schwartz-distributions, but all duals defined above can be seen as subsets of $\mathcal{D}^\infty$ and are therefore sets of Schwartz-distributions. Any element μ of $\mathcal{M}_r$ will be called a measure, any element of $\mathcal{D}^\infty$ a distribution. We extend the usual notation $\mu(f) := \int f(x) \, d\mu(x)$ for measures μ to distributions D: $D(f) =: \int f(x) \, dD(x)$. Given a KME Φ_k and two embeddable distributions D, T (see Definition 1.1.1), we define

$$\langle D \, , \, T \rangle_k := \langle \Phi_k(D) \, , \, \Phi_k(T) \rangle_k \quad \text{and} \quad \|D\|_k := \|\Phi_k(D)\|_k \ .$$

where $\langle . \, , \, . \rangle_k$ is the inner product of the RKHS $\mathcal{H}_k$ of k. To avoid introducing a new name, we call $\|D\|_k$ the maximum mean discrepancy (MMD) of D, even though the term "discrepancy" usually specifically designates a distance between two distributions rather than the norm of a single one. Given two topological sets $\mathcal{S}_1, \mathcal{S}_2$, we write

$$\mathcal{S}_1 \hookrightarrow \mathcal{S}_2$$

$\mathcal{S}_1$ *Continuously Contained in* $\mathcal{S}_2$

and say that $\mathcal{S}_1$ *is continuously contained in* $\mathcal{S}_2$ if $\mathcal{S}_1 \subset \mathcal{S}_2$ and if the topology of $\mathcal{S}_1$ is stronger than the topology induced by $\mathcal{S}_2$. For a general introduction to topology, TVSs and distributions, we recommend [119].

[119] Treves, *Topological Vector Spaces, Distributions and Kernels*, 1967

1.1 Kernel Mean Embeddings of Distributions

In this section, we show how to embed general distribution spaces into an RKHS. To do so, we redefine the integral $\int k(.,x) \, d\mu(x)$ so as to be well-defined even if μ is a distribution. It is often defined as a Bochner-integral; here we instead use the *Pettis-* (or *weak-*) integral:

Definition 1.1.1. *Let D be a linear form over a space of functions $\mathcal{F}$. Let $\varphi : \mathcal{X} \to \mathcal{H}_k$ be an RKHS-valued function s.t. for any $f \in \mathcal{H}_k$, $x \mapsto \langle f \, , \, \varphi(x) \rangle_k \in \mathcal{F}$. Then $\varphi : \mathcal{X} \to \mathcal{H}_k$ is weakly integrable with respect to (wrt) D if there exists a function in $\mathcal{H}_k$, written $\int \varphi(x) \, dD(x)$, s.t.*

$$\forall f \in \mathcal{H}_k, \quad \left\langle f \, , \int \varphi(x) \, dD(x) \right\rangle_k = \int \langle f \, , \, \varphi(x) \rangle_k \, d\bar{D}(x) \, , \tag{1.1}$$

Pettis Integral and KME

where the right-hand-side stands for $\bar{D}\,(\boldsymbol{x} \mapsto \langle f\,,\, \boldsymbol{\varphi}(\boldsymbol{x})\rangle_k)$ *and* $\bar{D}$ *denotes the complex-conjugate of* D. *If* $\boldsymbol{\varphi}(\boldsymbol{x}) = k(.,\boldsymbol{x})$, *we call* $\int k(.,\boldsymbol{x})\,dD(\boldsymbol{x})$ *the kernel mean embedding (KME) of* D *and say that* D *embeds into* $\mathcal{H}_k$. *We denote* $\Phi_{\boldsymbol{\varphi}}$ *the map* $\Phi_{\boldsymbol{\varphi}} : D \mapsto \int \boldsymbol{\varphi}(\boldsymbol{x})\,dD(\boldsymbol{x})$.

This definition extends the usual Bochner-integral: if $\boldsymbol{\varphi}$ is Bochner-integrable wrt a measure $\mu \in \mathcal{M}_+$, then $\boldsymbol{\varphi}$ is weakly integrable wrt μ and the integrals coincide [95]. In particular, if $\boldsymbol{x} \mapsto \|\boldsymbol{\varphi}(\boldsymbol{x})\|_k$ is Lebesgue-integrable, then $\boldsymbol{\varphi}$ is Bochner integrable, thus weakly integrable.

[95] Schwabik, *Topics in Banach Space Integration*, 2005, Prop.2.3.1

The general definition with $\boldsymbol{\varphi}$ instead of $k(.,\boldsymbol{x})$ will be useful in Section 1.4. But for now, let us concentrate on KMEs where $\boldsymbol{\varphi}(\boldsymbol{x}) = k(.,\boldsymbol{x})$. Kernels satisfy the so-called *reproducing property*: for any $f \in \mathcal{H}_k$, $f(\boldsymbol{x}) = \langle f\,,\, k(.,\boldsymbol{x})\rangle_k$. Therefore, the condition for all $f \in \mathcal{H}_k$ $\boldsymbol{x} \mapsto \langle f\,,\, \boldsymbol{\varphi}(\boldsymbol{x})\rangle_k \in \mathcal{F}$ reduces to $\mathcal{H}_k \subset \mathcal{F}$, and Equation (1.1) reads:

$$\forall f \in \mathcal{H}_k, \qquad \left\langle f\,,\, \int k(.,\boldsymbol{x})\,dD(\boldsymbol{x}) \right\rangle_k = \bar{D}(f)\,. \tag{1.2}$$

Reproducing Property and KMEs

Thus, by the Riesz representation theorem (see Appendix A.2), D embeds into $\mathcal{H}_k$ iff it defines a continuous linear form over $\mathcal{H}_k$. And in that case, its KME $\int k(.,\boldsymbol{x})\,dD(\boldsymbol{x})$ is the Riesz-representer of $\bar{D}$ restricted to $\mathcal{H}_k$. Thus, for an embeddable space of distributions $\mathcal{D}$, the embedding Φ_k can be decomposed as follows:

$$\Phi_k : \left\{ \begin{array}{ccccc} \mathcal{D} & \longrightarrow & \mathcal{H}_k' & \longrightarrow & \mathcal{H}_k \\ & \text{Conjugate restriction} & & \text{Riesz representer} & \\ D & \longmapsto & \bar{D}\big|_{\mathcal{H}_k} & \longmapsto & \int k(.,\boldsymbol{x})\,dD(\boldsymbol{x}) \end{array} \right. . \tag{1.3}$$

To know if D is continuous over $\mathcal{H}_k$, we use the following lemma, and its applications.

Lemma 1.1.2. *If* $\mathcal{H}_k \hookrightarrow \mathcal{F}$, *then* $\mathcal{F}'$ *embeds into* $\mathcal{H}_k$.

Proof. Suppose that $\mathcal{H}_k \hookrightarrow \mathcal{F}$. Let $D \in \mathcal{F}'$ and let $f, f_1, f_2, \ldots \in \mathcal{H}_k$. If $f_n \to f$ in $\mathcal{H}_k$ then $f_n \to f$ in $\mathcal{F}$, thus $D(f_n) \to D(f)$. Thus D is a continuous linear form over $\mathcal{H}_k$. □

In practice we typically use one of the following two corollaries (proofs in Appendices C.1.1 and C.1.2). The space $(\mathscr{C}_b)_c$ that they mention will be introduced in the discussions following Theorem 1.2.2. It has the same elements as $\mathscr{C}_b$, but carries a weaker topology.

Corollary 1.1.3. $\mathcal{H}_k \subset \mathscr{C}_{\to 0}$ *(resp.* $\mathcal{H}_k \subset \mathscr{C}_b$, *resp.* $\mathcal{H}_k \subset \mathscr{C}$*) iff the two following conditions hold.*

(i) For all $\boldsymbol{x} \in \mathcal{X}$, $k(.,\boldsymbol{x}) \in \mathscr{C}_{\to 0}$ *(resp.* $k(.,\boldsymbol{x}) \in \mathscr{C}_b$, *resp.* $k(.,\boldsymbol{x}) \in \mathscr{C}$*).*

Embedding Conditions for Measures

(ii) $x \mapsto k(x,x)$ *is bounded (resp. bounded, resp.* locally *bounded, meaning that, for each* $y \in \mathcal{X}$*, there exists a (compact) neighborhood of* y *on which* $x \mapsto k(x,x)$ *is bounded.).*

If so, then $\mathcal{H}_k \hookrightarrow \mathscr{C}_{\to 0}$ *(resp.* $\mathcal{H}_k \hookrightarrow \mathscr{C}_b$*, thus* $\mathcal{H}_k \hookrightarrow (\mathscr{C}_b)_c$*, resp.* $\mathcal{H}_k \hookrightarrow \mathscr{C}$*) and* $\mathcal{M}_f$ *(resp.* $\mathcal{M}_f$*, resp.* $\mathcal{M}_c$*) embeds into* $\mathcal{H}_k$*.*

Corollary 1.1.4.
If $k \in \mathscr{C}^{(m,m)}$*, then* $\mathcal{H}_k \hookrightarrow \mathscr{C}^m$*, thus* $\mathscr{E}^m$ *embeds into* $\mathcal{H}_k$*.*
If $k \in \mathscr{C}_0^{(m,m)}$*, then* $\mathcal{H}_k \hookrightarrow \mathscr{C}^m_{\to 0}$*, thus* $\mathcal{D}^m_{L^1}$ *embeds into* $\mathcal{H}_k$*.*
If $k \in \mathscr{C}_b^{(m,m)}$*, then* $\mathcal{H}_k \hookrightarrow \mathscr{C}^m_b$*, thus* $\mathcal{H}_k \hookrightarrow (\mathscr{C}^m_b)_c$*, thus* $\mathcal{D}^m_{L^1}$ *embeds into* $\mathcal{H}_k$*.*

Embedding Conditions for Distributions

Corollary 1.1.3 applied to $\mathscr{C}_b$ shows that $\mathcal{H}_k$ is (continuously) contained in $\mathscr{C}_b$ iff k is bounded and separately continuous. As discovered by Lehtö [63], there also exist kernels which are not continuous but whose RKHS $\mathcal{H}_k$ is contained in $\mathscr{C}_b$. So the conditions in Corollary 1.1.4 are sufficient, but in general not necessary. Concerning Lemma 1.1.2, note that it not only requires $\mathcal{H}_k \subset \mathcal{F}$, but also that $\mathcal{H}_k$ carries a stronger topology than $\mathcal{F}$. Otherwise there might exist a continuous form over $\mathcal{F}$ that is defined but non-continuous over $\mathcal{H}_k$. However, Corollary 1.1.3 shows that this cannot happen for $\mathscr{C}_*$, because if $\mathcal{H}_k \subset \mathscr{C}_*$ then $\mathcal{H}_k \hookrightarrow \mathscr{C}_*$. Although this also holds for $m = \infty$ [99], we do not know whether it extends to any $m > 0$.

[63] Lehtö, *Some Remarks on the Kernel Function in Hilbert Function Space*, 1952

[99] Simon-Gabriel and Schölkopf, *Kernel Distribution Embeddings - arXiv*, 2016, Prop.4 & Comments

1.2 Universal, Characteristic and SPD Kernels

The literature distinguishes various variants of universal, characteristic and spd kernels, such as c-, cc− or c_0-universal kernels, spd and integrally strictly positive definite ($\int$spd) kernels. They are all special cases of the following unifying definitions.

Definition 1.2.1. *Let* k *be a kernel,* $\mathcal{F}$ *be a space of functions s.t.* $\mathcal{H}_k \subset \mathcal{F}$*, and* $\mathcal{D}$ *be an embeddable subset of* $\mathcal{F}'$ *(e.g. an embeddable set of distributions). We say that* k *is*

- ▷ universal over $\mathcal{F}$ *if* $\mathcal{H}_k$ *is dense in* $\mathcal{F}$*.*
- ▷ characteristic to $\mathcal{D}$ *if the KME* Φ_k *is injective over* $\mathcal{D}$*.*
- ▷ strictly positive definite (spd) over $\mathcal{D}$ *if, for all* $D \in \mathcal{D}$*,*

$$\|\Phi_k(D)\|_k^2 = 0 \Rightarrow D = 0.$$

Universal, Characteristic, SPD

A universal kernel over $\mathscr{C}^m$ *(resp.* $\mathscr{C}^m_{\to 0}$*) will be said* c^m*- (resp.* c_0^m*-) universal (without the superscript when* $m = 0$*). A characteristic kernel to the set* $\mathcal{P}$ *of probability measures will simply be called characteristic.*

Importantly, note that k is characteristic to $\mathcal{P}$ iff the MMD of k can perfectly discriminate two probability measures. Studying when

k is Characteristic iff its MMD is Perfectly Discriminative on $\mathcal{P}$

k is characteristic hence amounts to determining when the associated MMD is perfectly discriminative. In general, instead of writing $\|\Phi_k(D)\|_k$ and $\langle \Phi_k(D), \Phi_k(T)\rangle_k$, we will write $\|D\|_k$ and $\langle D, T\rangle_k$. The previous definitions encompass the usual spd definitions. Denoting δ_x the Dirac measure concentrated on x, what is usually called

Specific SPD Instances

- ▷ spd corresponds to $\mathcal{D} = \mathcal{M}_\delta$, i.e.

$$\forall \mu = \sum_{i=1}^n \lambda_i \delta_{x_i} \in \mathcal{M}_\delta : \begin{cases} \|\mu\|_k^2 = \sum_{i,j=1}^n \lambda_i k(x_i, x_j)\bar{\lambda}_j = 0 \\ \qquad \Longrightarrow \quad \lambda_1 = \ldots = \lambda_n = 0 \,. \end{cases}$$

- ▷ conditionally spd corresponds to $\mathcal{D} = \mathcal{M}_\delta^0$ where $\mathcal{M}_\delta^0 := \{\mu \in \mathcal{M}_\delta : \mu(\mathcal{X}) = 0\}$, i.e.:

$$\forall \mu = \sum\nolimits_{i=1}^n \lambda_i \delta_{x_i} \in \mathcal{M}_\delta \quad \text{s.t.} \quad \sum\nolimits_{i=1}^n \lambda_i = 0 : \\ \|\mu\|_k^2 = \sum_{i,j=1}^n \lambda_i k(x_i, x_j)\bar{\lambda}_j = 0 \quad \Longrightarrow \quad \lambda_1 = \ldots = \lambda_n = 0.$$

- ▷ $\int$spd corresponds to $\mathcal{D} = \mathcal{M}_f$, i.e. :

$$\forall \mu \in \mathcal{M}_f : \quad \|\mu\|_k^2 = \iint k(x, y)\, d\mu(x)\, d\bar{\mu}(y) = 0 \quad \Rightarrow \quad \mu = 0 \,.$$

Let us now state the general link between universal, characteristic and spd kernels, which is the key that underlies Figure 1 in [109].

[109] Sriperumbudur et al., *Universality, Characteristic Kernels and RKHS Embedding of Measures*, 2011

Theorem 1.2.2. *If $\mathcal{H}_k \hookrightarrow \mathcal{F}$, then the following statements are equivalent.*

Equivalence of Universality, Characteristicness and SPD

(i) k *is universal over* $\mathcal{F}$.
(ii) k *is characteristic to* $\mathcal{F}'$.
(iii) k *is strictly positive definite (spd) over* $\mathcal{F}'$.

Proof. Equivalence of (ii) & (iii): Saying that $\|\Phi_k(D)\|_k = 0$ is equivalent to saying $\Phi_k(D) = 0$. Thus Φ_k is spd over $\mathcal{F}'$ iff the $\mathrm{Ker}(\Phi_k)$ (meaning the vector space that is mapped to 0 via Φ_k) is reduced to $\{0\}$, which happens iff Φ_k is injective over $\mathcal{F}'$.

Equivalence of (i) & (ii): Φ_k is the conjugate restriction operator $|_{\mathcal{H}_k} : D \mapsto \bar{D}|_{\mathcal{H}_k}$ composed with the Riesz representer mapping (Diagram Eq.1.3). The Riesz representer map is injective, so Φ_k is injective iff $|_{\mathcal{H}_k}$ is injective. Now, if $\mathcal{H}_k$ is dense in $\mathcal{F}$, then, by continuity, any $D \in \mathcal{F}'$ is uniquely defined by its values taken on $\mathcal{H}_k$. Thus $|_{\mathcal{H}_k}$ is injective. Reciprocally, if $\mathcal{H}_k$ is not dense in $\mathcal{F}$, then, by the Hahn-Banach theorem [119], there exists two different elements in $\mathcal{F}'$ that coincide on $\mathcal{H}_k$ but not on the entire space $\mathcal{F}$. So $|_{\mathcal{H}_k}$ is not injective. Thus $|_{\mathcal{H}_k}$ is injective iff $\mathcal{H}_k$ is dense in $\mathcal{F}$. □

[119] Treves, *Topological Vector Spaces, Distributions and Kernels*, 1967, Thm.18.1, Cor.3

To apply this theorem it suffices to find so-called *duality pairs* $(\mathcal{F}, \mathcal{F}')$ s.t. $\mathcal{H}_k \hookrightarrow \mathcal{F}$. Table 1.1 lists several such pairs. It shows in

Universal	Characteristic	S.P.D.	Name
$\mathcal{F}$	$\mathcal{F}'$	$\mathcal{F}'$	/
$\mathbb{C}^{\mathcal{X}}$	$\mathcal{M}_\delta$	$\mathcal{M}_\delta$	spd
$\mathbb{C}^{\mathcal{X}}/\mathbb{1}$	$\mathcal{M}_\delta^0$	$\mathcal{M}_\delta^0$	conditionally spd
$\mathscr{C}$	$\mathcal{M}_c$	$\mathcal{M}_c$	c-universal (or cc-universal)
$\mathscr{C}_{\to 0}$	$\mathcal{M}_f$	$\mathcal{M}_f$	c_0-universal
$(\mathscr{C}_b)_c$	$\mathcal{M}_f$	$\mathcal{M}_f$	$\int$spd
$((\mathscr{C}_b)_c)/\mathbb{1}$	$\mathcal{P}$ (or $\mathcal{M}_f^0$)	$\mathcal{M}_f^0$	characteristic
$\mathscr{C}^m$	$\mathscr{E}^m$	$\mathscr{E}^m$	c^m-universal
$\mathscr{C}_{\to 0}^m$	$\mathcal{D}_{L^1}^m$	$\mathcal{D}_{L^1}^m$	c_0^m-universal
$(\mathscr{C}_b^m)_c$	$\mathcal{D}_{L^1}^m$	$\mathcal{D}_{L^1}^m$	/

Table 1.1: Equivalence between the notions of universal, characteristic and spd kernels as implied by Thm. 1.2.2 or Prop. 1.2.3.

particular the well-known equivalence between c- (resp. c_0-) universal kernels and characteristic kernels to $\mathcal{M}_c$ (resp. $\mathcal{M}_f$) [110]. But we now discover that spd kernels over $\mathcal{M}_\delta$ can also be characterized in terms of universality over $\mathbb{C}^{\mathcal{X}}$, because $(\mathbb{C}^{\mathcal{X}})' = \mathcal{M}_\delta$ [26]. And we directly get the generalization to distributions and c_*^m-universality.

However, Theorem 1.2.2 leaves open the important case where k is characteristic (to $\mathcal{P}$). Of course, as $\mathcal{P}$ is contained in $\mathcal{M}_f$, it shows that a c_0-universal kernel must be characteristic. But to really characterize characteristic kernels in terms of universality, we would need to find a predual of $\mathcal{P}$, meaning a space $\mathcal{F}$ s.t. $\mathcal{F}' = \mathcal{P}$. This is hardly possible, as $\mathcal{P}$ is not even a vector space. However, we will see in Theorem 1.2.4 that k is characteristic iff k is characteristic to the vector space $\mathcal{M}_f^0 := \{\mu \in \mathcal{M}_f : \mu(\mathcal{X}) = 0\}$. So if we find a predual of $\mathcal{M}_f^0$, then we get an analog of Theorem 1.2.2 applied to $\mathcal{P}$. Let us do so now.

[110] Sriperumbudur et al., *Injective Hilbert Space Embeddings of Probability Measures*, 2008

[26] Duc-Jacquet, *Approximation des Fonctionnelles Linéaires sur les Espaces Hilbertiens Autoreproduisants*, 1973, p.II.35

Predual of $\mathcal{P}$ and $\mathcal{M}_f^0$?

As $\mathcal{M}_f^0$ is the hyperplane of $\mathcal{M}_f$ that is given by the equation $\int 1 \, d\mu = 0$, our idea is to take a predual $\mathcal{F}$ of $\mathcal{M}_f$ and consider the quotient $\mathcal{F}/\mathbb{1}$ of $\mathcal{F}$ divided by the constant function $\mathbb{1}$. Proposition 35.5 of [119] would then show that $(\mathcal{F}/\mathbb{1})' = \mathcal{M}_f^0$. But if we take the usual predual of $\mathcal{M}_f$, $\mathcal{F} = \mathscr{C}_{\to 0}$, then $\mathbb{1} \notin \mathcal{F}$, so the quotient $\mathcal{F}/\mathbb{1}$ is undefined. However, preduals are not unique, so let us try with another space $\mathcal{F}$ that contains $\mathbb{1}$, for example $\mathcal{F} = \mathscr{C}_b$. This time $\mathbb{1} \in \mathcal{F}$, but now the problem is that $\mathcal{F}'$ is in general strictly bigger than $\mathcal{M}_f$ [28] whereas we want $\mathcal{F}' = \mathcal{M}_f$. The trick now is to keep $\mathscr{C}_b$, but equip it with a weaker topology than the usual one, so that

[119] Treves, *Topological Vector Spaces, Distributions and Kernels*, 1967

[28] Fremlin et al., *Bounded Measures on Topological Spaces*, 1972, Sec.2 §2

$\mathcal{F}'$ becomes smaller. Intuitively, the reason for this decrease of $\mathcal{F}'$ is that, by weakening the topology of $\mathcal{F}$, we let more sequences converge in $\mathcal{F}$. This makes it more difficult for a functional over $\mathcal{F}$ to be continuous, because for any converging sequence in $\mathcal{F}$, its images need to converge. Thus some of the linear functionals that were continuous for the original topology of $\mathcal{F}$ get "kicked out" of $\mathcal{F}'$ when $\mathcal{F}$ carries a weaker topology. Now the only remaining step is to find a topology s.t. that $\mathcal{F}'$ shrinks exactly to $\mathcal{M}_f$. There are at least two such topologies: one defined by [96] and another, called the strict topology, whose definition can be found in [28]. Denoting τ_c either of these topologies, and $(\mathscr{C}_b)_c$ the space $\mathscr{C}_b$ equipped with τ_c, we finally get $((\mathscr{C}_b)_c)' = \mathcal{M}_f$, and thus:

[96] Schwartz, *Espaces de fonctions différentiables à valeurs vectorielles*, 1954, p.100-101

[28] Fremlin et al., *Bounded Measures on Topological Spaces*, 1972

Universal-Characteristic Equivalence for $\mathcal{P}$

Proposition 1.2.3. *$((\mathscr{C}_b)_c/\mathbb{1})' = \mathcal{M}_f^0$. Thus, if $\mathcal{H}_k \hookrightarrow (\mathscr{C}_b)_c$, then k is characteristic to $\mathcal{P}$ iff k is universal over the quotient space $((\mathscr{C}_b)_c/\mathbb{1})$.*

Proof. That $((\mathscr{C}_b)_c)' = \mathcal{M}_f$ is proven in [28] or [96]. Prop. 35.5 of [119] then implies $((\mathscr{C}_b)_c/\mathbb{1})' = \mathcal{M}_f^0$ (because $\mathcal{M}_f^0$ is the so-called *polar set* of $\mathbb{1}$; see [119]). Thm 1.2.2 implies the rest. □

[28] Fremlin et al., *Bounded Measures on Topological Spaces*, 1972, Thm.1

[96] Schwartz, *Espaces de fonctions différentiables à valeurs vectorielles*, 1954, p.100-101

[119] Treves, *Topological Vector Spaces, Distributions and Kernels*, 1967

For our purposes, the exact definition of τ_c does not matter. What matters more is that τ_c is weaker than the usual topology of $\mathscr{C}_b$, so that if $\mathcal{H}_k \hookrightarrow \mathscr{C}_b$, then $\mathcal{H}_k \hookrightarrow (\mathscr{C}_b)_c$. Proposition 1.2.3 thus applies every time that $\mathcal{H}_k \subset \mathscr{C}_b$ (see Corollaries 1.1.3 and 1.1.4). However, we do not know of any practical application of Proposition 1.2.3, except that it completes our overall picture of the equivalences between universal, characteristic and spd kernels. Let us also mention that, similarly to Proposition 1.2.3, as $(\mathbb{C}^{\mathcal{X}})' = \mathcal{M}_\delta$, we also have $(\mathbb{C}^{\mathcal{X}}/\mathbb{1})' = \mathcal{M}_\delta^0$. So conditionally spd kernels (meaning spd over $\mathcal{M}_\delta^0$) are universal to $\mathbb{C}^{\mathcal{X}}/\mathbb{1}$.

We now prove what we announced and used earlier: a kernel is characteristic to $\mathcal{P}$ iff it is characteristic to $\mathcal{M}_f^0$. We add a few other characterizations which are probably more useful in practice. They rely on the following observation: as $\mathcal{M}_f^0$ is a hyperplane of $\mathcal{M}_f$, saying that k is characteristic to $\mathcal{P}$ is almost the same as saying that it is characteristic to $\mathcal{M}_f$, i.e. $\int$spd (Thm. 1.2.2): after all, there is only one dimension needed to go from $\mathcal{M}_f^0$ to $\mathcal{M}_f$. Thus there should be a way to construct an $\int$spd kernel out of any characteristic kernel. This is what is described here and proven in Appendix C.1.3.

Theorem 1.2.4. *Let k_0 be a kernel. The following is equivalent.*

(i) k_0 *is characteristic to* $\mathcal{P}$.
(ii) k_0 *is characteristic to* $\mathcal{M}_f^0$.
(iii) There exists $\epsilon \in \mathbb{R}$ *s.t. the kernel* $k(x,y) := k_0(x,y) + \epsilon^2$ *is* $\int$*spd.*
(iv) For all $\epsilon \in \mathbb{R}\backslash\{0\}$, *the kernel* $k(x,y) := k_0(x,y) + \epsilon^2$ *is* $\int$*spd.*
(v) There exists an RKHS $\mathcal{H}_k$ *with kernel* k *and a measure* $\nu_0 \in \mathcal{M}_f \backslash \mathcal{M}_f^0$ *s.t.* k *is characteristic to* $\mathcal{M}_f$ *and* $k_0(x,y) = \langle \delta_x - \nu_0 \,,\, \delta_y - \nu_0 \rangle_k$.

From Characteristic to $\int$SPD Kernels and Back

Under these conditions, k_0 *and* k *induce the same MMD semi-metric in* $\mathcal{M}_f^0$ *and in* $\mathcal{P}$.

We will use this theorem to prove Theorem 1.3.4. Intuitively, a characteristic kernel guarantees that two signed measures μ_1, μ_2 with same total mass get mapped to two different functions in the RKHS. This is captured by (ii) which arbitrarily focuses on the special case where the total mass is 0. When they have different total masses however, they may get mapped to a same function f, except if, like in (iii) and (iv), we add a positive constant to the kernel. In that case, μ_1 and μ_2 get mapped to the functions $f + \mu_1(\mathcal{X})\mathbb{1}$ and $f + \mu_2(\mathcal{X})\mathbb{1}$ which are now different, because $\mu_1(\mathcal{X}) \neq \mu_2(\mathcal{X})$. Intuitively, by adding a positive constant to our kernel, we added one dimension to the RKHS (carried by the function $\mathbb{1}$) that explicitly 'checks' if two measures have the same mass. Finally, (v) tells us that, out of any $\int$spd kernel k, we can construct a characteristic kernel k_0 that is not $\int$spd anymore and vice-versa.

1.3 Topology Induced by k

Remember that for any distribution D of a set of embeddable distributions $\mathcal{D}$ we defined $\|D\|_k := \|\Phi_k(D)\|_k$ and called $\|D\|_k$ the maximum mean discrepancy (MMD) of D. Doing this defines a new topology on $\mathcal{D}$, in which a net D_α converges to D iff $\|D_\alpha - D\|_k$ converges to 0. (A reader unfamiliar with nets may think of them as sequences where the index α can be continuous; see [9].) In this section, we investigate how convergence in MMD compares with other types of convergences defined on $\mathcal{D}$ that we now shortly present.

[9] Berg et al., *Harmonic Analysis on Semigroups Theory of Positive Definite and Related Functions*, 1984

We defined $\mathcal{D}$ as a subset of a dual space $\mathcal{F}'$, so $\mathcal{D}$ will carry the topology induced by $\mathcal{F}'$. Many topologies can be defined on dual spaces, but the two most prominent ones, which we will consider here, are the *weak-*$*$ and the *strong* topology, denoted $w(\mathcal{F}', \mathcal{F})$ and $b(\mathcal{F}', \mathcal{F})$ respectively, or simply $w*$ and b. The weak-$*$ topology is the topology of pointwise convergence (where by 'point', we mean a function in $\mathcal{F}$), while the strong topology corresponds to the uniform convergence over the bounded subsets of $\mathcal{F}$ (see Eq. 1.2). Bounded sets of a TVS are defined in Appendix A.2 (Definition A.2.4). By default, we equip $\mathcal{F}'$ with the strong topology and sometimes write $\mathcal{F}'_b$ to emphasize it. When $\mathcal{F}$ is a Banach space, the strong topology of $\mathcal{F}'$ is the topology of the operator norm $\|D\|_{\mathcal{F}'} := \sup_{\|f\|_{\mathcal{F}} \leqslant 1} |D(f)|$. In particular, strong convergence in $\mathcal{M}_f = (\mathscr{C}_{\to 0})'$ means convergence in total variation (TV) norm and weak-$*$ convergence in $\mathcal{M}_f$ means convergence for any function $f \in \mathscr{C}_{\to 0}$. On $\mathcal{M}_f$, we will also consider the topology of pointwise convergence over $\mathscr{C}_b$ (instead of $\mathscr{C}_{\to 0}$). It is widely used in probability theory where it is known as the *weak* (or

Convergences on Dual Spaces: Weak-$*$ and Strong Convergence

narrow) convergence topology. We will denote it by σ. Importantly, the weak and weak-$*$ topologies of $\mathcal{M}_\mathfrak{f}$ coincide on $\mathcal{P}$ (but not on $\mathcal{M}_\mathfrak{f}$) [9]. Finally, we define the weak RKHS convergence of embeddable distributions, denoted by $w{-}k$, as the pointwise convergence over $\mathcal{H}_k$. Note that D_α converges in $w{-}k$ to D iff their embeddings converge weakly (or equivalently weakly-$*$) in $\mathcal{H}_k$, in the sense that, for any $f \in \mathcal{H}_k$, $\langle f, \Phi_k(D_\alpha)\rangle_k$ converges to $\langle f, \Phi_k(D)\rangle_k$. The following summarizes the different convergence types.

Weak-$*$ Convergence Coincides with Weak Convergence on $\mathcal{P}$

[9] Berg et al., *Harmonic Analysis on Semigroups Theory of Positive Definite and Related Functions*, 1984, Chap. 2, Cor. 4.3

$D_\alpha \xrightarrow{b} D :=$	$\sup_{f\in\mathcal{B}} \lvert D_\alpha(f) - D(f)\rvert \longrightarrow 0$	$\forall$ bounded $\mathcal{B} \subset \mathcal{F}$	$D_\alpha \in \mathcal{F}'$
$D_\alpha \xrightarrow{w*} D :=$	$\lvert D_\alpha(f) - D(f)\rvert \longrightarrow 0$	$\forall f \in \mathcal{F}$	$D_\alpha \in \mathcal{F}'$
$\mu_\alpha \xrightarrow{\sigma} \mu :=$	$\lvert \mu_\alpha(f) - \mu(f)\rvert \longrightarrow 0$	$\forall f \in \mathscr{C}_b$	$\mu_\alpha \in \mathcal{M}_\mathfrak{f}$
$D_\alpha \xrightarrow{w-k} D :=$	$\lvert D_\alpha(f) - D(f)\rvert \longrightarrow 0$	$\forall f \in \mathcal{H}_k$	D_α embeddable
$D_\alpha \xrightarrow{\lVert\cdot\rVert_k} D :=$	$\lVert D_\alpha - D\rVert_k \longrightarrow 0$		D_α embeddable

Table 1.2: Different convergence types, summarized. Note that RKHS-convergence (last line) is a special case of bounded convergence (first line), where $\mathcal{F} = \mathcal{H}_k$.

1.3.1 Embeddings of Dual Spaces are Continuous

In this section, we show that the MMD topology is often weaker than other topologies τ defined on $\mathcal{D}$, meaning that if D_α converges to D in τ, then it also converges to D in MMD. Note that this is equivalent to saying that the KME of $\mathcal{D}_\tau$ (read '$\mathcal{D}$ equipped with τ') is continuous. We start with the following pretty coarse, yet very general result.

Proposition 1.3.1. *If $\mathcal{H}_k \hookrightarrow \mathcal{F}$, then* $\begin{cases} D_\alpha \xrightarrow{b} D \;\Rightarrow\; D_\alpha \xrightarrow{\lVert\cdot\rVert_k} D \\ D_\alpha \xrightarrow{w*} D \;\Rightarrow\; D_\alpha \xrightarrow{w-k} D \end{cases}$.

Relating Strong to Strong and Weak to Weak Convergences

Proof. Proposition 1.3.1 states that the KME is continuous when both $\mathcal{F}'$ and $\mathcal{H}_k$ carry their strong or their weak-$*$ topology, which we now show. From Diagram Eq.(1.3), we know that the KME is the composition of the conjugate restriction operator with the Riesz representer map. The Riesz representer map is a topological (anti-)isomorphism between $\mathcal{H}_k'$ and $\mathcal{H}_k$, thus continuous (see Appendix A.2). And the restriction map is the adjoint (or transpose) of the canonical embedding map $\iota : \mathcal{H}_k \to \mathcal{F},\ f \mapsto f$, thus continuous when both $\mathcal{F}'$ and $\mathcal{H}_k'$ carry their weak-$*$ or strong topologies [119]. □

[119] Treves, *Topological Vector Spaces, Distributions and Kernels*, 1967, Prop.19.5 & Cor

Let us briefly comment on this result. The statement $D_\alpha \xrightarrow{w*} D \;\Rightarrow\; D_\alpha \xrightarrow{w-k} D$ is actually obvious, because $\mathcal{H}_k \subset \mathcal{F}$. Concerning strong convergence, Proposition 1.3.1 implies that, if $\mathcal{F}$ is a Banach space, then any net that converges for the dual norm $\lVert\cdot\rVert_{\mathcal{F}'}$ converges in MMD. Applying this with $\mathcal{F} = \mathscr{C}_{\to 0}$ and $\mathcal{F}' = \mathcal{M}_\mathfrak{f}$ shows that convergence in TV norm implies convergence in MMD, or equivalently, that the TV norm is stronger than the MMD. Similar reasoning can be used to show that the MMD is weaker than the so-called Kantorovich(-Wasserstein) and the Dudley norms [99]. These

[99] Simon-Gabriel and Schölkopf, *Kernel Distribution Embeddings - arXiv*, 2016, Ex.1

results can also be found in [111]. However, the authors there directly bounded the MMD semi-norm by the target norm. This has the advantage of giving concrete bounds, but is more difficult to generalize if $\mathcal{F}$ is not a Banach space.

[111] Sriperumbudur et al., *Hilbert Space Embeddings and Metrics on Probability Measures*, 2010

Though very general, Proposition 1.3.1 is pretty weak, as it only compares a strong with a strong and a weak-$*$ with a weak(-$*$) topology. But how does the weak-$*$ topology on $\mathcal{F}'$ compare with the strong topology of $\mathcal{H}_k$: does weak-$*$ convergence imply convergence in MMD? This question is discussed in details in [99]. The short answer is: not always, but sometimes; it depends on the space $\mathcal{F}'$. For example, if $k \in \mathscr{C}^{(m,m)}$, then weak-$*$ convergence in $\mathscr{E}^m$ implies convergence in MMD; but weak-$*$ convergence in $\mathcal{D}^m_{L^1}$ usually does not imply MMD convergence when $\mathcal{X}$ is non-compact. For us, the only thing we will need later is to know what happens on $\mathcal{M}_+$, the set of finite positive measures. The following lemma shows that weak convergence in $\mathcal{M}_+$ usually implies MMD convergence.

[99] Simon-Gabriel and Schölkopf, *Kernel Distribution Embeddings - arXiv*, 2016, Sec.7

Relating Weak-Probability- to Strong-Kernel Convergence

Lemma 1.3.2. *A bounded kernel k is continuous iff:* $\forall\, \mu_\alpha, \mu \in \mathcal{M}_+$, $\mu_\alpha \xrightarrow{\sigma} \mu \implies \mu_\alpha \xrightarrow{\|.\|_k} \mu$.

Proof. We assume k bounded to ensure that any probability measure is embeddable. Now, suppose that weak convergence implies MMD convergence and take $x, y, \xi, \zeta \in \mathcal{X}$ s.t. that $x \to \xi$ and $y \to \zeta$. Then $\delta_x \xrightarrow{\sigma} \delta_\xi$ and $\delta_y \xrightarrow{\sigma} \delta_\zeta$, so $\Phi_k(\delta_x) \to \Phi_k(\delta_\xi)$ and $\Phi_k(\delta_y) \to \Phi_k(\delta_\zeta)$ in $\mathcal{H}_k$. And by continuity of the inner product:

$$k(x,y) = \langle \Phi_k(\delta_y)\,,\, \Phi_k(\delta_x)\rangle_k \to \langle \Phi_k(\delta_\zeta)\,,\, \Phi_k(\delta_\xi)\rangle_k = k(\xi,\zeta)\,,$$

so k is continuous. Conversely, suppose that k is continuous, and let $\mu_\alpha \xrightarrow{\sigma} \mu$ in $\mathcal{M}_+$. The tensor-product mapping $\mathcal{M}_+(\mathcal{X}) \to \mathcal{M}_+(\mathcal{X}\times\mathcal{X})$, $\mu \mapsto \mu\otimes\mu$ is weakly continuous [9]. So by applying $\bar\mu_\alpha \otimes \mu_\alpha$ to a bounded continuous kernel k, we get

[9] Berg et al., *Harmonic Analysis on Semigroups Theory of Positive Definite and Related Functions*, 1984, Chap.2 Thm.3.3

$$\begin{aligned}\|\Phi_k(\mu_\alpha) - \Phi_k(\mu)\|_k^2 &= \iint k(x,y)\,\mathrm{d}(\mu_\alpha-\mu)(y)\,\mathrm{d}(\bar\mu_\alpha-\bar\mu)(x)\\ &= \left\{\begin{array}{l}[\bar\mu_\alpha\otimes\mu_\alpha](k) - [\bar\mu\otimes\mu_\alpha](k)\\ -[\bar\mu_\alpha\otimes\mu](k) + [\bar\mu\otimes\mu](k)\end{array}\right\} \longrightarrow 0. \quad \square\end{aligned}$$

1.3.2 When Does k Metrize the Topology of $\mathcal{F}'$?

So far we focused on the question: when does convergence in $\mathcal{D}$ imply convergence in MMD? We now seek the opposite: when does MMD-convergence imply convergence in $\mathcal{D}$?

First, the kernel *must* be characteristic to $\mathcal{D}$. Otherwise, the MMD does not define a distance but only a semi-distance, so that the induced topology would not be Hausdorff. Second, we will suppose

that $\mathcal{F}$ is barreled. This is a technical, yet very general assumption that we use in the next theorem. The definition of a barreled space is given in Appendix A.2 for completeness, but all that the reader should remember is that all Banach, Fréchet, Limit-Fréchet and *all function spaces encountered in this chapter are barreled* [1] , except $(\mathscr{C}^m_b)_c$.

[1] $\mathbb{C}^{\mathcal{X}}$ is barreled, because it is a topological product $\prod_{\mathcal{X}} \mathbb{C}$ of barreled spaces. All other mentioned spaces are either Banach, Fréchet or Limit-Fréchet spaces, thus barreled [119, Prop.33.2 & Cor.1-3].

Lemma 1.3.3. *Suppose that $\mathcal{F}$ is barreled, k is universal over $\mathcal{F}$, $\mathcal{H}_k \hookrightarrow \mathcal{F}$ and let $(D_\alpha)_\alpha$ be a bounded net in $\mathcal{F}'_b$. Then $D_\alpha \overset{w-k}{\longrightarrow} D$ iff $D_\alpha \overset{w*}{\longrightarrow} D$. Hence $D_\alpha \overset{\|.\|_k}{\longrightarrow} D \;\Rightarrow\; D_\alpha \overset{w*}{\longrightarrow} D$.*

Relating Weak-Kernel- to Weak-$*$ Convergence

Proof. Prop. 32.5 of [119] shows that the weak topologies of $\mathcal{F}'$ and of $\mathcal{H}'_k$ coincide on so-called *equicontinuous* sets of $\mathcal{F}'$, and the Banach-Steinhaus theorem (see Appendix A.2) states that if $\mathcal{F}$ is barreled, then the equicontinuous sets of $\mathcal{F}'$ are exactly its bounded sets. This precisely means that if the net $(D_\alpha)_\alpha$ is bounded in $\mathcal{F}'$, then $D_\alpha(f) \to D(f)$ for all $f \in \mathcal{F}$ iff it converges for all $f \in \mathcal{H}_k$. Now, if $\|D_\alpha - D\|_k \to 0$, then, by continuity of the inner product, $D_\alpha(f) - D(f) = \langle f, D_\alpha - D\rangle_k \to 0$ for any $f \in \mathcal{H}_k$. □

[119] Treves, *Topological Vector Spaces, Distributions and Kernels*, 1967

Lemma 1.3.3 says that the weak-$*$ topologies of $\mathcal{F}'$ and of $\mathcal{H}_k$ coincide on subsets of $\mathcal{F}'$ that are bounded in the strong topology. By the Banach-Steinhaus theorem (see App. A.2), the net $(D_\alpha)_\alpha$ of Lemma 1.3.3 is bounded iff[2] $\sup_\alpha |D_\alpha(f)| < \infty$ for all $f \in \mathcal{F}$. It is however not enough in general to show that $\sup_\alpha \|D_\alpha\|_k < \infty$. A bounded set in $\mathcal{M}_f$ is also a set whose measures have uniformly bounded total variation. The total variation of any probability measure being 1, $\mathcal{P}$ is bounded. So Lemma 1.3.3 shows that for continuous c_0-universal kernels, convergence of probability measures in MMD distance implies weak-$*$ convergence, which on $\mathcal{P}$ is the same as weak-convergence. But by Lemma 1.3.2 the reverse is true as well. Thus, *for a continuous c_0-universal kernel k, probability measures converge weakly iff they converge in MMD distance.* Such kernels are said to *metrize* the weak convergence on $\mathcal{P}$.

[2] Banach-Steinhaus: on barreled spaces, strongly bounded ≡ weakly bounded

However, the condition that k be c_0-universal seems slightly too restrictive. Indeed, it is needed in Lemma 1.3.3 to ensure that the KME be characteristic to $\mathcal{M}_f$ (by Thm. 1.2.2 applied to $\mathcal{F} = \mathscr{C}_{\to 0}$) so that the MMD be a metric over $\mathcal{M}_f$ (not only a semi-metric). But, to be a metric over $\mathcal{P}$, it would suffice that k be characteristic to $\mathcal{P}$, which is a slightly coarser assumption than c_0-universality. Is this condition enough to guarantee the metrization of weak-convergence in $\mathcal{P}$? The following theorem shows that it is.

Theorem 1.3.4. *A bounded kernel over a locally compact Hausdorff space metrizes the weak convergence of probability measures iff it is continuous and characteristic (to $\mathcal{P}$).*

Bounded Characteristic Kernels Metrize Weak Convergence

Proof. [Theorem 1.3.4] If k metrizes the weak convergence over $\mathcal{P}$, then, by Lemma 1.3.2, k is continuous, and, for $\|.\|_k$ to be a norm,

k needs to be characteristic. Conversely, if k is continuous, then by Lemma 1.3.2 weak convergence implies convergence in MMD. So it remains to show that MMD convergence implies weak convergence. To do so, we use Lemma C.1.2 of the appendix, which states that for an $\int$spd kernel, MMD convergence of probability measures implies their weak convergence. Now k might not be $\int$spd, but using Theorem 1.2.4(iv), we can transform it to a kernel $k_1 := k + 1$ which induces the same MMD metric over probability measures than k, but which is $\int$spd This concludes. From Theorem 1.2.4 it follows that k_1 is $\int$spd, thus Lemma C.1.2 (Appendix C.1.4) shows that the weak topology $w(\mathcal{P}, \mathcal{H}_{k_1})$ induced by k_1 in $\mathcal{P}$, and which is weaker than the MMD topology, coincides with the weak-$*$ topology $w(\mathcal{P}, \mathscr{C}_{\to 0})$. Thus convergence of probability measures in the MMD distance of k_1 implies weak convergence. But k_1 and k induce the same metric in $\mathcal{P}$ (Thm. 1.2.4), which concludes. □

To the best of our knowledge, this is the first characterization of the class of kernels that metrize the weak-convergence of probability measures. For example Gaussian, Laplace, inverse-multiquadratic or Matérn kernels are continuous and characteristic, so they all metrize the weak convergence over $\mathcal{P}$. In general however, even if a kernel metrizes the weak convergence over $\mathcal{P}$, it usually does not metrize weak convergence over $\mathcal{M}_+$ or $\mathcal{M}_f$ [99].

[99] Simon-Gabriel and Schölkopf, *Kernel Distribution Embeddings - arXiv*, 2016

1.4 Kernel Mean Embeddings of Schwartz-Distributions

We extended KMEs of measures to Schwartz-distributions and showed that they are continuous, but we hardly said anything about what to do and how to work with distributions. We will now catch up by focusing on distributions only. In Section 1.4.1, we discuss and prove the Fubini and the Differentiation formulae featured in the introduction. In Section 1.4.2 we provide sufficient conditions for a translation-invariant kernel to be c_*^m-universal.

1.4.1 Distributional Calculus

Proposition 1.4.1. *Let D, T be two embeddable distributions into $\mathcal{H}_k$.*

Fubini for Distributions

$$\langle D\,,\,T\rangle_k = \iint k(x,y)\,dD(y)\,d\bar{T}(x) = \iint k(x,y)\,d\bar{T}(x)\,dD(y) \qquad (1.4)$$
$$\|D\|_k^2 = \iint k(x,y)\,dD(y)\,d\bar{D}(x) = \iint k(x,y)\,d\bar{D}(x)\,dD(y)\,,$$

where $\iint k(x,y)\,dD(y)\,d\bar{T}(x)$ is to be understood as $\bar{T}(\mathcal{I})$ with $\mathcal{I}(x) = \int k(x,y)\,dD(y)$.

Proof. Definition 1.1.1 of a KME, together with the property that $k(y,x) = \overline{k(x,y)}$ leads to:

$$\begin{aligned}
\langle D\,,\,T\rangle_k &= \int_x \left\langle \int_y k(.,y)\,dD(y)\,,\,k(.,x)\right\rangle_k d\bar{T}(x)\\
&= \int_x \overline{\left\langle k(.,x)\,,\,\int_y k(.,y)\,dD(y)\right\rangle_k} d\bar{T}(x)\\
&= \int_x \int_y \overline{\langle k(.,x)\,,\,k(.,y)\rangle_k}\,d\bar{D}(y)\,d\bar{T}(x)\\
&= \iint k(x,y)\,dD(y)\,d\bar{T}(x).
\end{aligned}$$

To prove the right-most part of (1.4), use $\langle D\,,\,T\rangle_k = \overline{\langle T\,,\,D\rangle}_k$. □

These formulae are well-known when D and T are probability measures. They show that if you know how to integrate a function (the kernel) wrt a measure or a distribution, then you can compute its MMD norm. However, integrating wrt a distribution that is not a measure can be tedious. But the following proposition gives us a way to convert an integration wrt a distribution into an integration wrt a measure.

Embedding of Derivatives

Proposition 1.4.2. *Let* $k \in \mathscr{C}^{(m,m)}$ *and* $p \in \mathbb{N}^d$ *s.t.* $|p| \leqslant m$. *A distribution* D *embeds into* $\mathcal{H}_k$ *via* $\partial^{(0,p)}k$ *iff* $\partial^p D$ *embeds into* $\mathcal{H}_k$ *via* k. *In that case,*

$$\Phi_k(\partial^p D) = (-1)^{|p|}\int [\partial^{(0,p)}k](.,x)\,dD(x) = (-1)^{|p|}\,\Phi_{\partial^{(0,p)}k}(D)\,. \tag{1.5}$$

If moreover k *is translation-invariant, then*

$$\Phi_k(\partial^p D) = \partial^p[\Phi_k(D)]. \tag{1.6}$$

Proof. The proof holds in the following equalities. For any $f \in \mathcal{H}_k$,

$$\begin{aligned}
\left\langle f\,,\,\int k(.,x)\,d[\partial^p D](x)\right\rangle_k &= \int \langle f\,,\,k(.,x)\rangle_k\,d[\partial^p\bar{D}](x) = [\partial^p\bar{D}](f)\\
&= (-1)^{|p|}\bar{D}(\partial^p f)\\
&= (-1)^{|p|}\bar{D}(\left\langle f\,,\,\partial^{(0,p)}k(.,x)\right\rangle_k)\\
&= (-1)^{|p|}\int \left\langle f\,,\,\partial^{(0,p)}k(.,x)\right\rangle_k d\bar{D}(x)\\
&= \left\langle f\,,\,(-1)^{|p|}\int \partial^{(0,p)}k(.,x)\,dD(x)\right\rangle_k .
\end{aligned}$$

The first line uses the definition of KMEs (1.1), the second uses the definition of distributional derivatives (see App. A.1), the third uses Lemma C.1.1, the fourth line rewrites the previous line with our notational convention, and the fifth one uses again the definition of a weak integral (1.1). □

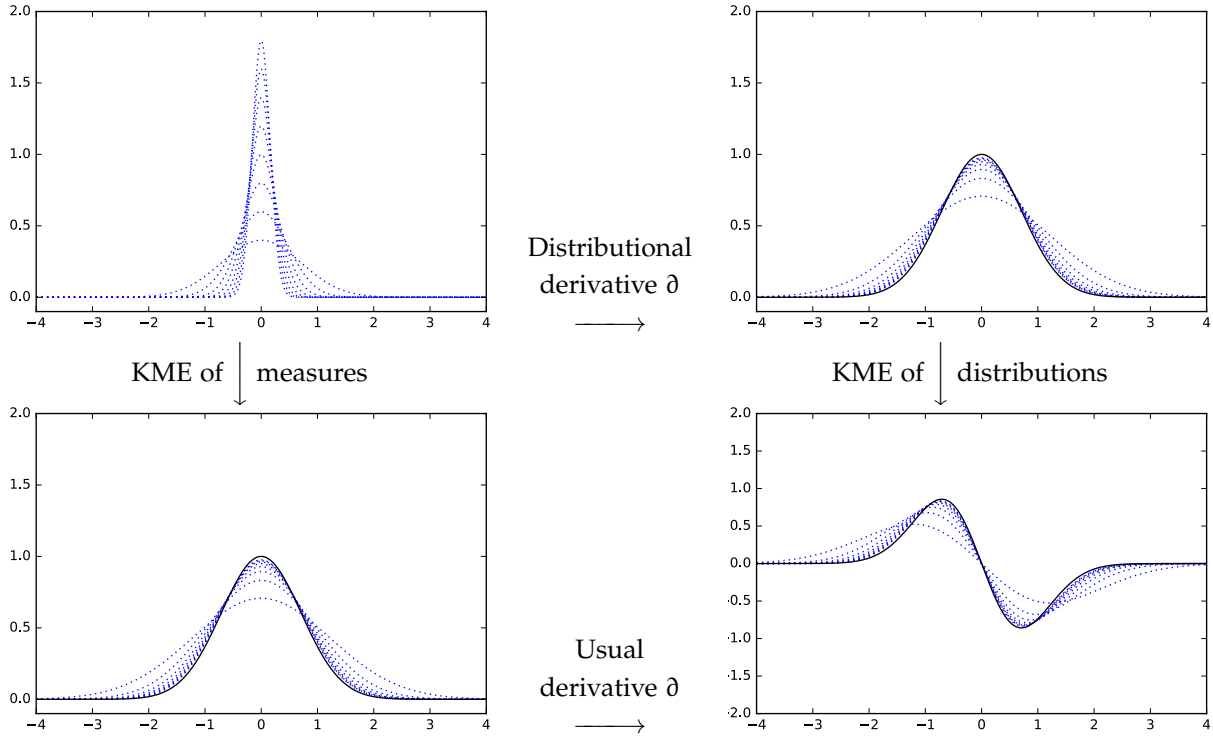


Figure 1.1: Densities of more and more peaked Gaussian probability measures μ_σ (top left) with their derivatives (top right) and their embeddings (below) using a Gaussian kernel (see Example 1). Equation (1.6) states that the diagram is commutative. When σ goes to 0, the Gaussians converge (weakly) to a Dirac mass δ_0, which has no density, but who's embedding is the solid black line (bottom left). The derivatives converge (weakly) to the Schwartz-distribution $\partial\delta_0$, which is not even a signed measure, but whose embedding (bottom right, black solid line) can easily be computed using (1.5) or (1.6). Moreover, the embeddings of μ_σ and $\partial\mu_\sigma$ converge (weakly) to the embeddings of μ_0 and $\partial\mu_0$, which illustrates Proposition 1.3.1.

Equation (1.6) describes a commutative diagram pictured in Figure 1.1: it states that with translation-invariant kernels, it is equivalent to take the (distributional) derivative of a distribution and embed it, or to embed it and take the (usual) derivative of the embedding. See Appendix A.1 for an introduction to distributional derivatives. Note that for a signed measure μ with a $|\mathbf{p}|$-times differentiable density q, the distributional derivative $\partial^{\mathbf{p}}\mu$ is the signed measure with density $\partial_u^{\mathbf{p}} q$, where $\partial_u^{\mathbf{p}}$ is the usual partial derivative operator. However, Proposition 1.4.2 becomes most useful when μ has no differentiable density, for example when μ is an empirical measure. Then there is no analytical formula for the derivative of μ, but we can still compute its KME anlytically by using (1.5) or (1.6).

Example 1. Let us illustrate Proposition 1.4.2 on KMEs of Gaussian probability measures μ_σ with density $q_\sigma(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-x^2/\sigma^2}$ using a Gaussian kernel $k(x, y) = e^{-(x-y)^2}$. When σ goes to zero, μ_σ gets more and more peaked around 0 and converges weakly to the Dirac measure $\mu_0 := \delta_0$. The KME of μ_σ is easy to compute and using (1.6) we get

$$\Phi_k(\mu_\sigma)(x) = \frac{1}{\sqrt{1+2\sigma^2}}\, e^{-\frac{x^2}{1+2\sigma^2}}$$
$$\Phi_k(\partial\mu_\sigma)(x) = \partial[\Phi_k(\mu_\sigma)] = -\frac{2x}{(1+2\sigma^2)^{3/2}}\, e^{-\frac{x^2}{1+2\sigma^2}}\,,$$

where the formulae still hold when $\sigma = 0$. Figure 1.1 plots these embeddings for different σ's. Note that contrary to $\partial\mu_\sigma$ with $\sigma >$

0, $\partial\mu_0$ is not a signed measure (but a Schwartz-distribution) but it has a KME which, moreover, can easily be computed using (1.6). Notice also that on Figure 1.1 both the embeddings of μ_σ and $\partial\mu_\sigma$ converge (weakly) to the embeddings of μ_0 and $\partial\mu_0$. This illustrates Proposition 1.3.1.

Theoretically, (1.5) can be used to convert the KME of *any* distribution into a sum of KMEs of measures. In other words, the integral wrt a distribution appearing in (1.1) can be converted into a sum of integrals wrt to signed measures. Here is how. Given a measure $\mu \in \mathcal{M}_f = \mathcal{D}^0_{L^1}(\mathbb{R})$, we may differentiate μ and get a new distribution $\partial\mu$ which may or may not be itself a measure[3]. But in any case, what will follow shows that $\partial\mu$ is in $\mathcal{D}^1_{L^1}(\mathbb{R})$. Thus the space of distributions that can be written as a sum $\mu_0 + \partial\mu_1$ of two finite measures μ_1, μ_2 is a subspace of $\mathcal{D}^1_{L^1}(\mathbb{R})$ and we may wonder how big exactly it is. Schwartz [96] showed that it is exactly the space $\mathcal{D}^1_{L^1}(\mathbb{R})$. More generally, he showed:

[3] Think for example of the Dirac measure: it is a measure, but not its derivative. See App. A.1.

[96] Schwartz, *Espaces de fonctions différentiables à valeurs vectorielles*, 1954, around p.100

Lemma 1.4.3 (Schwartz). *For any $m \leqslant \infty$ and any distribution in $T \in \mathcal{D}^m_{L^1}$ (resp. $T \in \mathscr{E}^m$) there exists a* finite *family of measures $\mu_p \in \mathcal{M}_f$ (resp. $\mu_p \in \mathcal{M}_c$) s.t. $T = \sum_{|p|\leqslant m} \partial^p \mu_p$.*

$\mathcal{D}^\infty_{L^1}$: smallest vector space containing $\mathcal{M}_f$ and all its derivatives

Using (1.5), this means that the KME can be computed as $\sum_{|p|\leqslant m} \int \partial^{(0,p)} k(.,x)\, d\mu_p(x)$, which gives a way to numerically compute the KME of distributions. As most distributions encountered in practice happen to be defined as measures or derivatives of some measures, this method is highly relevant in practice.

By combining Propositions 1.4.1 and 1.4.2, we get the following corollary.

Corollary 1.4.4. *Let $k \in \mathscr{C}^{(m,m)}$, $p \in \mathbb{N}^d$ with $|p| \leqslant m$, and let S, T be two distributions s.t. $\partial^p S$ and $\partial^p T$ embed into $\mathcal{H}_k$. Then:*

RKHS-Norm of Derivatives

$$\langle \partial^p S\,,\, \partial^p T\rangle_k = \langle S\,,\, T\rangle_{\partial^{(p,p)}k} \quad \textit{and} \quad \|\partial^p S\|_k = \|S\|_{\partial^{(p,p)}k}\ .$$

Proof. The proof reduces to the following equations.

$$\begin{aligned}
\langle \partial^p S\,,\, \partial^p T\rangle_k &\overset{(a)}{=} \left\langle \int \partial^{(0,p)} k(.,x)\, dS(x)\,,\, \int \partial^{(0,p)} k(.,y)\, dT(y) \right\rangle_k \\
&\overset{(b)}{=} \int \left\langle \partial^{(0,p)} k(.,y)\,,\, \partial^{(0,p)} k(.,x) \right\rangle_k dS(y)\, d\bar{T}(x) \\
&\overset{(c)}{=} \int \partial^{(p,p)} k(x,y)\, dS(y)\, d\bar{T}(x) \\
&\overset{(d)}{=} \langle S\,,\, T\rangle_{\partial^{(p,p)}k}\ ,
\end{aligned}$$

Equality (a) uses Proposition 1.4.2, (b) uses twice (on the left and on the right of the inner product) the definition of a weak integral (1.1), (c) uses Equation (C.2) proven in Appendix C.1 which states

that $\left\langle \partial^{(0,p)}k(.,y)\,,\,\partial^{(0,p)}k(.,x)\right\rangle_k = \partial^{(p,p)}k(x,y)$, and (d) uses (1.4) applied to the kernel $\partial^{(p,p)}k$. □

Corollary 1.4.4 tells us that if we use $\partial^{(p,p)}k$ – which is a kernel – to compute the MMD distance between two probability distributions D, T, then we are actually computing the MMD distance between their derivatives $\partial^p D$ and $\partial^p T$ with the kernel k. One could extend this corollary from (p,p) to (p,q) when $|q| \leqslant m$, yielding $\langle \partial^p D\,,\,\partial^q T\rangle_k = \int \partial^{(q,p)}k(x,y)\,dD(x)\,d\bar{T}(y)$. But in that case, $\partial^{(q,p)}k$ might not be a kernel anymore.

1.4.2 c^m- and c_0^m-Universal Kernels

Theorem 1.2.2 shows the equivalence between c_*^m-universality and characteristicness over $\mathcal{D}^m_{L^1}$ or $\mathscr{E}^m$. But neither the universality, nor the characteristic assumption seems easy to check in general. However, for translation invariant kernels, meaning kernels that can be written as $k(x,y) = \psi(x-y)$ for some function ψ, we will now show that being characteristic to $\mathcal{P}$ or to $\mathcal{D}^m_{L^1}$ is one and the same thing, provided that $k \in \mathscr{C}_b^{(m,m)}$. Thus, any technique to prove that a kernel is characteristic may also be used to prove that it is characteristic to the much wider space $\mathcal{D}^m_{L^1}$. One of these techniques consists in verifying that the distributional Fourier transform $\mathscr{F}\psi$ has full support. The reader unfamiliar with distributional Fourier transforms may think of them as an extension of the usual Fourier transform – which is usually only defined on L^1, L^2 or $\mathcal{M}_f$ – to wider function and distribution spaces. Let us mention that $\mathscr{F}\psi$ is exactly the unique positive, symmetric, finite measure appearing in Bochner's theorem [125], and whose (usual) Fourier transform is ψ. We now successively present the result for $\mathcal{D}^m_{L^1}$ and then their pendant for $\mathscr{E}^m$.

[125] Wendland, *Scattered Data Approximation*, 2004, Thm.6.6

When Perfect Discrimination Extends from $\mathcal{P}$ to $\mathcal{D}^m_{L^1}$

Theorem 1.4.5. *Let $k \in \mathscr{C}^{(m,m)}$ be a translation-invariant kernel $k(x,y) = \psi(x-y)$ over $\mathcal{X} = \mathbb{R}^d$, and $\mathscr{F}\psi$ its distributional Fourier transform. Then $\mathcal{D}^m_{L^1}$ embeds into $\mathcal{H}_k$ and the following are equivalent.*

(i) k is characteristic (to $\mathcal{P}$).
(ii) k is characteristic to $\mathcal{D}^m_{L^1}$.
(iii) $\mathscr{F}\psi$ has full support.

If moreover $\psi \in \mathscr{C}^m_{\to 0}$, then k is c_0^m-universal iff it is c_0-universal.

Perfect Discrimination over $\mathscr{E}^m$

Theorem 1.4.6. *Let $k \in \mathscr{C}^{(m,m)}$ be a translation-invariant kernel $k(x,y) = \psi(x-y)$ with $\mathcal{X} = \mathbb{R}^d$. If the support of $\mathscr{F}\psi$ has Lebesgue-measure > 0, then k is characteristic to $\mathscr{E}^m$.*

Proof. (of Theorem 1.4.5) First, note that $\partial^{(p,p)}k(x,y) \leqslant \partial^{(p,p)}k(x,x)\partial^{(p,p)}k(y,y) = (\partial^{2p}\psi(0))^2$ for any $|p| \leqslant m$ (see

Lemma C.1.1 in Appendix C.1). Hence $k \in \mathscr{C}_b^{(m,m)}$, which, by Corollary 1.1.4, proves that $\mathcal{D}_{L^1}^m$ embeds into $\mathcal{H}_k$. Now suppose that (i) and (ii) are equivalent, then they are also equivalent to k being characteristic to $\mathcal{M}_f$. Using Theorem 1.2.2, we thus proved the last sentence. Now, (ii) clearly implies (i) and Theorem 9 in [111] states that (i) and (iii) are equivalent. So it remains to show that (iii) implies (ii). We now sketch its proof and relegate the details to Appendix C.1.5. Let Λ be the finite positive measure from Bochner's theorem, s.t. $\psi = \mathscr{F}\Lambda$ and let $D \in \mathcal{D}_{L^1}^m$. Then

[111] Sriperumbudur et al., *Hilbert Space Embeddings and Metrics on Probability Measures*, 2010

$$\begin{aligned}
\|D\|_k^2 &= \iint \left(\int e^{i(x-y)\xi} \, d\Lambda(\xi) \right) d\bar{D}(x) \, dD(y) \\
&\overset{(a)}{=} \int \left(\iint (e^{i(x-y)\xi}) \, d\bar{D}(x) \, dD(y) \right) d\Lambda(\xi) \\
&\overset{(b)}{=} \int |[\mathscr{F}D](\xi)|^2 \, d\Lambda(\xi).
\end{aligned}$$

Λ being positive, if it has full support, then $[\mathscr{F}D](\xi) = 0$ for almost all $\xi \in \mathcal{X}$. Thus $D = 0$. Thus, assuming that (a) and (b) indeed hold, we just showed that if (iii), then $\|D\|_k = 0$ implies $D = 0$, meaning that k is spd to $\mathcal{D}_{L^1}^m$, which, with Theorem 1.2.2, proves (ii). We relegate the proof of (a) and (b) to Appendix C.1.5. □

Proof. (of Theorem 1.4.6) For any $D \in \mathscr{E}^m$, we can write, like before: $\|D\|_k^2 = \int |[\mathscr{F}D](\xi)|^2 \, d\Lambda(\xi)$. But now, the Paley-Wiener-Schwartz theorem [119] states that $\mathscr{F}D$ is an analytical function, so if its set of zeros has Lebesgue-measure > 0, then $\mathscr{F}D$ is the 0 function,so $D = 0$, showing that Φ_k is injective over $\mathscr{E}^m$. □

[119] Treves, *Topological Vector Spaces, Distributions and Kernels*, 1967, Thm.29.2

These theorems show for example that Gaussian kernels are c_0^∞-universal and that the sinc kernel, defined on $\mathcal{X} = \mathbb{R}$ by $k(x, y) = \sin(x - y)/(x - y)$ (and 1 on the diagonal), is c^∞- but not c_0^∞-universal. When $\mathcal{X} = \mathbb{R}$, one can refine the conditions on the Fourier transform in Theorem 1.4.6 so that they become necessary and sufficient [99].

[99] Simon-Gabriel and Schölkopf, *Kernel Distribution Embeddings - arXiv*, 2016, Thm.41

1.5 Chapter Conclusion

This chapter grouped various notions of universal, characteristic and spd kernels into three fundamental definitions – one for each – and showed that they are essentially equivalent: they describe the same family of kernels, but from dual perspectives. Using this duality link, we could systematically recover most of the previously known links, but also discovered new ones, such as the equivalence between characteristicness to $\mathcal{P}$ and universality over $(\mathscr{C}_b)_c/\mathbb{1}$; or between strict positive definiteness (over $\mathcal{M}_\delta$) and universality over $\mathbb{C}^\mathcal{X}$. We then compared the convergence in MMD with other convergence types of

distributions and measures. Importantly, we showed that a bounded kernel metrizes the weak convergence of probability measures iff it is continuous and characteristic. Said differently, the MMD of a bounded continuous kernel perfectly discriminates probability measures iff it metrizes weak convergence. Incidentally, we also showed that KMEs over signed measures can be extended to generalized measures called Schwartz-distributions. For translation-invariant kernels, this extension preserves perfect discrimination, in the sense that a perfectly discriminative MMD over $\mathcal{P}$ typically stays perfectly discriminative over $\mathcal{D}_{L^1}^m$.

Here, we always assumed $\mathcal{X}$ to be locally compact. Although this assumption fits many very general spaces, unfortunately, it does not contain any infinite-dimensional Banach space. So a main open question of this chapter is whether our characterization of kernels that metrize the weak convergence of probability measures also applies to more general spaces, such as so-called Polish spaces, which are very standard spaces in probability theory. Finally, we proved a few results that are specific to KMEs of distributions. Proposition 1.4.2 and its Corollary 1.4.4 already show that these KMEs of distributions naturally appear when considering KMEs wrt derivatives of kernels. We hope that they will in future lead to more insights and applications in machine learning.

History and Related Machine Learning Literature

Universal and characteristic kernels play an essential role in kernel methods and their theory. Universal kernels ensure consistency of many RKHS-based estimators in the context of regression and classification [112, 113], whereas characteristic kernels are of prime interest in any MMD-based algorithm, such as kernel two-sample tests [41, 42], HSIC independence tests [34, 43, 44], kernel density estimators [107] and MMD-type GANs [27, 66]. The ML community gradually introduced more and more variants of universal kernels [16, 17, 76, 112], but instead of also introducing variants of characteristic kernels, it stuck to the original definition given by [31] which considered only characteristicness to $\mathcal{P}$. As a result, the literature started proving various links between the various variants of universal kernels and the only notion of characteristic kernels that it had. Eventually these notions were linked to $\int$spd and conditionally $\int$spd kernels [31, 32, 33, 34, 42, 108, 110, 111] and all known relations got summarized in a superb overview article by Sriperumbudur et al. [109]. However, by not introducing the notion of a characteristic kernel to something else than $\mathcal{P}$, the literature oversaw the fundamental dual link between universal, characteristic and spd kernels shown in Theorem 1.2.2, which

[112] Steinwart, *On the Influence of the Kernel on the Consistency of Support Vector Machines*, 2001; [113] Steinwart and Christmann, *Support Vector Machines*, 2008;

[42] Gretton et al., *A Kernel Method for the Two-Sample-Problem*, 2007; [41] Gretton et al., *A Kernel Two-Sample Test*, 2012;

[43] Gretton et al., *A Kernel Statistical Test of Independence*, 2008; [44] Gretton and Györfi, *Consistent Nonparametric Tests of Independence*, 2010; [34] Fukumizu et al., *Kernel Measures of Conditional Dependence*, 2008;

[107] Sriperumbudur, *On the Optimal Estimation of Probability Measures in Weak and Strong Topologies*, 2016

[66] Li et al., *Generative Moment Matching Networks*, 2015; [27] Dziugaite et al., *Training Generative Neural Networks via MMD Optimization*, 2015;

[112] Steinwart, *On the Influence of the Kernel on the Consistency of Support Vector Machines*, 2001; [76] Micchelli et al., *Universal Kernels*, 2006; [17] Carmeli et al., *Vector Valued Reproducing Kernel Hilbert Spaces of Integrable Functions and Mercer Theorem*, 2006; [16] Caponnetto et al., *Universal Multi-Task Kernels*, 2008;

[31] Fukumizu et al., *Kernel Dimensionality Reduction for Supervised Learning*, 2004

[31] Fukumizu et al., *Kernel Dimensionality Reduction for Supervised Learning*, 2004; [34] Fukumizu et al., *Kernel Measures of Conditional Dependence*, 2008; [33] Fukumizu et al., *Characteristic Kernels on Groups and Semigroups*, 2009; [32] Fukumizu et al., *Kernel Choice and Classifiability for RKHS Embeddings of Probability Distributions*, 2009; [42] Gretton et al., *A Kernel Method for the Two-Sample-Problem*, 2007; [110] Sriperumbudur et al., *Injective Hilbert Space Embeddings of Probability Measures*, 2008; [108] Sriperumbudur et al., *On the Relation between Universality, Characteristic Kernels and RKHS Embedding of Measures*, 2010; [111] Sriperumbudur et al., *Hilbert Space Embeddings and Metrics on Probability Measures*, 2010;
[109] Sriperumbudur et al., *Universality, Characteristic Kernels and RKHS Embedding of Measures*, 2011

easily explains all the previously reported links and unravels new ones.

Concerning the study of kernels that metrize the weak convergence of probability measures, in mathematics it dates back at least to Guilbart [46], but it got introduced into the machine learning community only many years later by [111]. They gave new sufficient conditions to metrize the weak convergence, which then got improved by [107]. However, by generalizing these sufficient conditions even further, Theorem 1.3.4 is the first to provide conditions that are both sufficient *and necessary*, and that holds on any locally compact Hausdorff space $\mathcal{X}$ (which is more general than in the existing literature).

[46] Guilbart, *Etude des Produits Scalaires sur l'Espace des Mesures*, 1978
[111] Sriperumbudur et al., *Hilbert Space Embeddings and Metrics on Probability Measures*, 2010
[107] Sriperumbudur, *On the Optimal Estimation of Probability Measures in Weak and Strong Topologies*, 2016, Thm.2

2

Kernel Mean Estimation for Functions of Random Variables

The following chapter uses our previous insights on the MMD-topology to provide theoretical foundations to a recent probabilistic programming framework proposed by Schölkopf et al. [93] called kernel probabilistic programming. Probabilistic programs are usual programming languages with a few additional operations designed to facilitate the coding of and the inference from probabilistic models. Without going into details (for an overview, see [38, 73]), one key challenge of probabilistic programs is their ability to correctly represent the distribution of $f(\mathbf{X})$, when $f : \mathcal{X} \to \mathcal{Z}$ is an arbitrary measurable function between measurable spaces $\mathcal{X}, \mathcal{Z}$ of one or several random variables $\mathbf{X}$ ($\mathbf{X}$ could be multi-dimensional). This can sometimes be done analytically. For example, if $f(x) = ax + b$ is linear and $X \sim \mathcal{N}(\mu, \sigma)$ is Gaussian, then $f(X) \sim \mathcal{N}(a\mu + b, a\sigma)$ is Gaussian again. But outside these textbook cases, $f(\mathbf{X})$ has typically no standard distribution and a more general distribution-representation is needed.

[93] Schölkopf et al., *Computing Functions of Random Variables via RKHS Representations*, 2015

[38] Gordon et al., *Probabilistic Programming*, 2014; [73] McKinley, *Programming the World of Uncertain Things (Keynote)*, 2016

Distributions of Functions of R.V. are Difficult to Compute

Many alternatives have been proposed. Finite integer distributions are usually encoded as lists of $(x_i, p(x_i))$ pairs. Real valued distributions are sometimes represented using integral transforms [106], mixtures of Gaussians [77], or Laguerre- [126] or Chebyshev-polynomial [59] approximations of their cumulative distribution functions. Probabilistic finite automata sometimes represent string variables. All those approaches have their merits, but they always assume specific distributions, functions or input types.

[106] Springer, *The Algebra of Random Variables*, 1979
[77] Milios, *Probability Distributions as Program Variables*, 2009
[126] Williamson, *Probabilistic Arithmetic*, 1989
[59] Korzeń and Jaroszewicz, *PaCAL*, 2014
[53] Kalos and Whitlock, *Monte Carlo Methods*, 2008

A more general framework is Monte Carlo sampling [53], which simply represents $\mathbf{X}$ by a weighted sample $\hat{\mathbf{X}} := \{(\mathbf{x}_i, w_i)\}_{i=1}^N$ (with $w_i \geqslant 0$). This representation has several advantages: it works for any input type, the sample size controls the time-accuracy trade-off, and applying functions to random variables reduces to applying the functions pointwise to the sample: $f(\hat{\mathbf{X}}) := \{(f(\mathbf{x}_i), w_i)\}$ represents $f(\mathbf{X})$. A key challenge however is to assess the quality of this representation to handle error propagation and allow representation optimizations at fixed sample size. To do so, [93] proposed to use MMDs, as they are easy to compute on samples and can be tailored to focus on different properties of $\mathbf{X}$, depending on the user's needs and prior assumptions. To do so, simply define the MMD between random

Monte-Carlo Representation are Particularly Adapted

[93] Schölkopf et al., *Computing Functions of Random Variables via RKHS Representations*, 2015

MMDs Provide a Distance to Assess Representation Quality

variables $\mathbf{X}, \mathbf{X}' \in \mathcal{X}$ as the MMD between their probability measures:

$$\mathrm{MMD}_k(\mathbf{X}, \mathbf{X}') := \|P_{\mathbf{X}} - P_{\mathbf{X}'}\|_k \ ,$$

MMD of Random Variables

and use empirical measures $P_{\hat{\mathbf{X}}} := \sum_{i=1}^{N} w_i \delta_{x_i}$ for discrete samples $\hat{\mathbf{X}}$.

Here we build on this work and provide general theoretical guarantees for the proposed estimators. Our goal is to prove MMD-consistency of $f(\hat{\mathbf{X}})$ given MMD-consistency of $\hat{\mathbf{X}}$. Said differently, under appropriate assumptions, we will prove that if a sequence of samples[1] $\hat{\mathbf{X}}_n$ converges to $\mathbf{X}$ in MMD, then $f(\hat{\mathbf{X}}_n)$ also converges to $f(\hat{\mathbf{X}})$. And we will sometimes be able to provide finite finite-sample guarantees. Importantly, our results make no assumption on the origin of $\hat{\mathbf{X}}$, and in particular no independent and identically distributed (iid) assumption. This makes them a powerful tool not only to work with MCMC-, kernel herding- [20, 61] or other typically non-iid samples, but also for sample compression or privacy preservation. We may indeed replace $\hat{\mathbf{X}}$ by any new sample $\hat{\mathbf{X}}'$ and still keep guarantees on the quality of $f(\hat{\mathbf{X}}')$ as long as we appropriately control the MMD between $\hat{\mathbf{X}}$ and $\hat{\mathbf{X}}'$.

[1] Note that each sample n has a sample size N_n that depends on n. For convenience we may nevertheless write N instead of N_n.

[20] Chen et al., *Super-Samples from Kernel Herding*, 2010; [61] Lacoste-Julien et al., *Sequential Kernel Herding : Frank-Wolfe Optimization for Particle Filtering*, 2015

The next section presents a few use-cases of this proposed framework, while Section 2.2 contains our main results: it proves MMD-consistency in a general setting (Section 2.2.1), and finite sample guarantees when Matérn kernels are used (Section 2.2.2). Section 2.3 then shows how our results apply to functions of multiple variables, both interdependent and independent and Section 2.4 concludes.

2.1 Motivating Examples

This section first illustrates the MMD's flexibility to focus on some distribution-differences and ignore others. It then shows how MMDs can be used for compression and privacy preservation, and finishes with an example to illustrate the need and gains of compression when dealing with functions of several variables.

2.1.1 Tailoring the MMD to Our Needs

One advantage of MMDs is that, by choosing the kernel wisely, we can tailor the MMD to emphasize some distribution-differences more than others. For example if on $\mathcal{X} = \mathbb{R}$ we choose $k(x, x') := x \cdot x'$, then one can easily show that $\mathrm{MMD}_k(X, X') = |\mathbb{E}[X] - \mathbb{E}[X']|$: the MMD only focuses on the difference of means. If instead we prefer the MMD to consider all first p moments, we could use the kernel $k(x, x') := (x \cdot x' + 1)^p$. Or we could directly aim for a characteristic kernel – Gaussian, Laplacian or Matérn f.ex. – so that the MMD

be zero iff all moments coincide. This does not mean we cannot change the MMD's focus anymore. For instance, any Gaussian kernel $k(x, x') := \exp(-\|x-x'\|^2/(2\sigma^2))$ with positive bandwidth $\sigma^2 > 0$ is characteristic. But the larger the bandwidth σ, the more the MMD focuses on low frequency differences. If it is too large, all samples will essentially look the same, while being too small makes them look all equally different. In both cases, we are back to the usual Monte Carlo setting, equivalent to no dissimilarity measure.

2.1.2 Two Interpretations of the Statement $\hat{\mathbf{X}}_n \overset{k}{\to} \mathbf{X}$

What does it mean for a sequence of samples $\hat{\mathbf{X}}_n$ to converges to $\mathbf{X}$ in MMD? We see at least two interpretations. When the weights of each sample $\hat{\mathbf{X}}_n$ are non-negative and sum to one, then we may see $\hat{\mathbf{X}}_n$ as a random variable whose distribution is given by $P_{\hat{\mathbf{X}}_n}$. The convergence $\hat{\mathbf{X}}_n \to_k \mathbf{X}$ can then be interpreted as a usual convergence statement between random variables. Alternatively, we may identify $\hat{\mathbf{X}}_n$ and $\mathbf{X}$ with the KMEs of their associated measures $\Phi_k(P_{\hat{\mathbf{X}}_n})$ and $\Phi_k(P_{\mathbf{X}})$, which we will henceforth simply denote $\hat{\mu}^k_{\hat{\mathbf{X}}_n}$ and $\mu^k_{\mathbf{X}}$:

$$\begin{aligned} \hat{\mu}^k_{\hat{\mathbf{X}}} &:= \Phi_k(P_{\hat{\mathbf{X}}}) := \textstyle\sum_{i=1}^N w_i k(\cdot, x_i) \\ \mu^k_{\mathbf{X}} &:= \Phi_k(P_{\mathbf{X}}) := \textstyle\int_{\mathcal{X}} k(\cdot, x)\, dP_{\mathbf{X}}(x) \end{aligned} \quad . \tag{2.1}$$

KME of Random Variables

The statement $\hat{\mathbf{X}}_n \to_k \mathbf{X}$ then becomes a convergence-statement on RKHS-functions: $\hat{\mu}^k_{\hat{\mathbf{X}}_n} \to \mu^k_{\mathbf{X}}$ in RKHS-norm. This second point of view will be useful in some proofs. It also allows the weights of $\hat{\mathbf{X}}_n$ to neither sum to one, nor even be positive, which may be handy when used in conjunction with reduced expansion set methods (see Section 2.1.3). But if they don't, then it becomes unclear how to sample from $\hat{\mathbf{X}}_n$. That is why, contrary to [93], we favor the first interpretation. In both cases $\hat{\mathbf{X}}$ estimates $\mathbf{X}$, but we prefer to think of it in terms of random variables rather than RKHS functions.

Two Interpretations: Convergence of R.V. or Convergence of KMEs

[93] Schölkopf et al., *Computing Functions of Random Variables via RKHS Representations*, 2015

2.1.3 MMD for Compression and Privacy Preservation

Compression. Using MMDs gives a principled way to reduce sample sizes, i.e. do compression. Indeed, given $\hat{\mathbf{X}}$, we could compress it by choosing a smaller sample $\hat{\mathbf{X}}'$ that minimizes $\mathrm{MMD}(\hat{\mathbf{X}}', \hat{\mathbf{X}})$. Such algorithms are known as *reduced expansion set methods* and have been studied by the ML community [94]. Of course, the resulting sample points (and their weights) will typically be mutually dependent. Now, when $\hat{\mathbf{X}}$ is an iid sample of $\mathbf{X}$, then $f(\hat{\mathbf{X}})$ is also an iid sample of $f(\mathbf{X})$, and as such, it is known to MMD-converge to $f(\mathbf{X})$ in $1/\sqrt{N}$ [105]. But this argument breaks if $\hat{\mathbf{X}}$ first gets compressed into a new,

[94] Schölkopf and Smola, *Learning with Kernels*, 2001, Chapter 18

[105] Smola et al., *A Hilbert Space Embedding for Distributions*, 2007

non-iid sample $\hat{X}'$. While [93] advocates the use of reduced expansion set methods to save computational resources, they do not prove that preserving the MMD-consistency when compressing $\hat{X}$ also preserves the MMD-consistency of $f(\hat{X}')$. That is the goal of Section 2.2.

[93] Schölkopf et al., *Computing Functions of Random Variables via RKHS Representations*, 2015

Privacy preservation. Another potentially significant application is privacy preservation. Imagine that one has a large database of user data. If we transform the original data into new, *synthetic* expansion points using a reduced expansion set method, then we can pass on these (weighted) synthetic expansion points to a third party without revealing the original data. Using our results (Sections 2.2&2.3), the third party can nevertheless perform arbitrary continuous functional operations on the synthetic data in a consistent manner. In a follow-up work, Balog et al. [6] even show that this protocol is differentially private.

[6] Balog et al., *Differentially Private Database Release via KMEs*, 2018

The next subsection shows how useful compression can become when working with functions of several variables.

2.1.4 Functions of Two Variables and Compression

Suppose that we want to construct a sample-representation of $f(X, Y)$ given iid samples $\hat{X} = \{x_i, 1/N\}_{i=1}^N$ and $\hat{Y} = \{y_j, 1/N\}_{j=1}^N$ from two independent random variables $X \in \mathcal{X}$ and $Y \in \mathcal{Y}$ respectively. Let Q be the distribution of $Z = f(X, Y)$.

A first option is to consider what we call the *diagonal representation* $\widehat{\Delta} := \{f(x_i, y_i), 1/N\}_{i=1}^N$. Since $\widehat{\Delta}$ is an iid sample of $f(X, Y)$, it is $1/\sqrt{N}$-consistent [105]. Another option is the *U-statistic* representation $\widehat{\mathcal{U}} := \{f(x_i, y_j), 1/N^2\}_{i,j=1}^N$, which is also $1/\sqrt{N}$-consistent. Experiments show that $\widehat{\mathcal{U}}$ is more accurate and has lower variance than $\widehat{\Delta}$ (see Figure 2.1), but it needs $O(n^2)$ memory instead of $O(n)$. For this reason [93] proposes to use a reduced expansion set method both on $\hat{X}$ and $\hat{Y}$ to get new, smaller samples $\hat{X}' := \{x_i', w_i\}_{i=1}^n$ and $\hat{Y}' := \{y_j, v_j\}_{j=1}^n$ of size $n \ll N$, and then represent $f(X, Y)$ using the compressed U-statistic $\widehat{\mathcal{U}}' := \{f(x_i', y_j'), w_i v_j\}_{i,j=1}^n$.

[105] Smola et al., *A Hilbert Space Embedding for Distributions*, 2007

[93] Schölkopf et al., *Computing Functions of Random Variables via RKHS Representations*, 2015

Use MMDs to Reduce the Sample-Size of U-Statistics

We ran experiments on synthetic data to show how accurately $\widehat{\Delta}$, $\widehat{\mathcal{U}}$ and $\widehat{\mathcal{U}}'$ approximate $f(X, Y)$ with growing sample size N. We considered three basic arithmetic operations: multiplication $X \cdot Y$, division X/Y, and exponentiation X^Y, with $X \sim \mathcal{N}(3, 0.5)$ and $Y \sim \mathcal{N}(4, 0.5)$. As the MMD distance to the true $f(X, Y)$ is unknown, we approximated it by the MMD distance to the U-statistic estimator computed on a large sample (125 points). For $\widehat{\mathcal{U}}'$, we used the simplest possible reduced expansion set method: we randomly sampled subsets of size $n = 0.01 \cdot N$ of the x_i, and optimized the weights w_i and v_i to best approximate $\hat{X}$ and $\hat{Y}$. The results are summarized in Figure 2.1 and corroborate our expectations: (i) all estimators converge, (ii) $\widehat{\mathcal{U}}$ con-

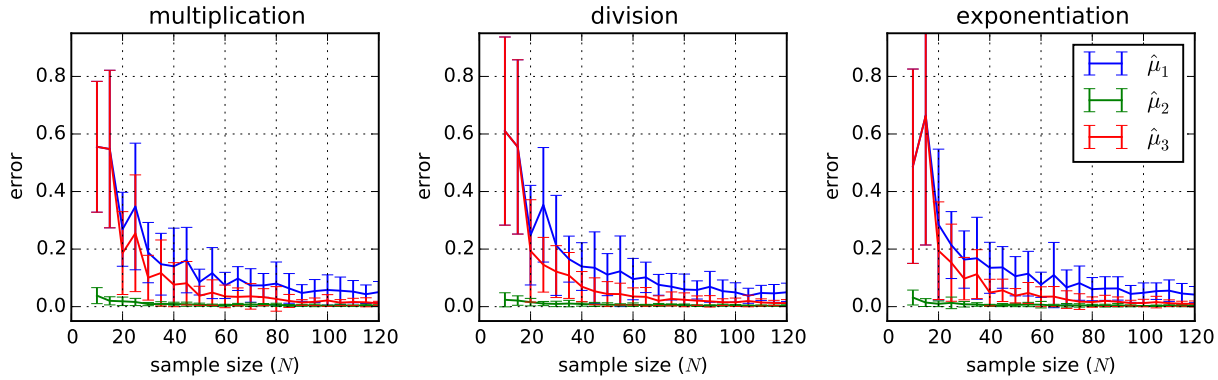


Figure 2.1: MMD-assessed quality of three sample-based estimators of basic arithmetic functions of two variables, $X \cdot Y$, X/Y and X^Y, as a function of sample size N. The U-statistic estimator $\widehat{U}$ works best, closely followed by the proposed estimator $\widehat{U}'$, which outperforms the diagonal estimator $\widehat{\Delta}$.

verges fastest and has the lowest variance, and (iii) $\widehat{U}'$ is worse than $\widehat{U}$, but much better than the diagonal estimator $\widehat{\Delta}$. Note that unlike the U-statistic estimator $\widehat{U}$, the reduced set based estimator $\widehat{U}'$ can be used with a fixed storage budget even if we perform a sequence of function applications – a situation naturally appearing in the context of probabilistic programming.

Schölkopf et al. [93] prove the consistency of $\widehat{U}$ only for a rather limited case, when the reduced samples $\{x'_i\}_{i=1}^n$ and $\{y'_i\}_{i=1}^n$ are iid copies of X and Y and the weights $\{(w_i, v_j)\}_{i,j=1}^n$ are constant. Using our new results we will prove in Section 2.3 the consistency of $\widehat{U}'$ under fairly general conditions, even when expansion points and weights are interdependent random variables.

[93] Schölkopf et al., *Computing Functions of Random Variables via RKHS Representations*, 2015

2.2 Consistency and Finite-Sample Guarantees

This section contains our main results regarding consistency and finite sample guarantees for the estimator $f(\hat{X})$ of $f(X)$.

2.2.1 Consistency

If k_X is c_0-*universal* (see Section 1.2), consistency of $\hat{\mu}_{f(X)}$ can be shown in a rather general setting.

When Consistency of $\hat{X}$ Implies Consistency of $f(\hat{X})$

Theorem 2.2.1. *Let $\mathcal{X}$ and $\mathcal{Z}$ be locally compact Hausdorff spaces equipped with their Borel σ-algebras, $f : \mathcal{X} \to \mathcal{Z}$ a continuous function, k_X, k_Z bounded continuous kernels on $\mathcal{X}, \mathcal{Z}$ respectively. Assume k_X is c_0-universal. If (i) $\mathcal{X}$ and $\mathcal{Z}$ are compact and there exists C such that $\sum_i |w_i| \leqslant C$ independently of n; or if (ii) all $w_i \geqslant 0$ and $\sum_i w_i = 1$ for all n, then*

$$\text{if} \quad \hat{X} \overset{k_X}{\to} X \quad \text{then} \quad f(\hat{X}) \overset{k_Z}{\to} f(X) \quad \text{as} \quad n \to \infty.$$

Proof. Idea: $\hat{\mathbf{X}} \overset{k_x}{\to} \mathbf{X} \overset{(a)}{\Rightarrow} \hat{\mathbf{X}} \overset{\sigma}{\to} \mathbf{X} \overset{(b)}{\Rightarrow} f(\hat{\mathbf{X}}) \overset{\sigma}{\to} f(\mathbf{X}) \overset{(c)}{\Rightarrow} f(\hat{\mathbf{X}}) \overset{k_z}{\to} f(\mathbf{X})$. *Proof of* (b):[2] if for any $\varphi \in \mathscr{C}_b$, $\mathbb{E}\,\varphi(\hat{\mathbf{X}}) \to \mathbb{E}\,\varphi(\mathbf{X})$ then, f being continuous, $\varphi \circ f \in \mathscr{C}_b$, hence $\mathbb{E}\,\varphi \circ f(\hat{\mathbf{X}}) \to \mathbb{E}\,\varphi \circ f(\mathbf{X})$. For points (a)&(c) we now treat assumptions (i) & (ii) separately.

[2] When the weights of $\hat{\mathbf{X}}$ are non-negative and sum to one (i.e. when $\hat{\mathbf{X}}$ is a random variable), then property (b) is known as the continuous mapping theorem.

Under assumption (i). *Proof of* (a)*:* First, the constraint $\sum_i |w_i| \leqslant C$ (independently of n) means that the measures associated to $\hat{\mathbf{X}}$ are uniformly bounded in total variation: by definition, they hence lie in a bounded subset of $\mathcal{M}_f$. Now, $\mathcal{X}$ being compact, $\mathscr{C}_b = \mathscr{C}$, hence σ- and $w*$-convergence are the same and $\mathcal{M}_f = (\mathscr{C})'$. Hence Lemma 1.3.3 applies and proves (a). *Proof of* (c)*:* apply Theorem 44 (i)&(iii) of [99] which shows that for a bounded continuous kernel (k_z) and on compact spaces (here $f(\mathcal{X})$), a bounded sequence of signed finite measures ($f(\hat{\mathbf{X}})$) converges in the weak-$*$ (i.e. σ-) sense iff it converges in the MMD sense.footnoteThis theorem extends Lemma 1.3.2 from $\mathcal{M}_+$ to $\mathcal{M}_f$ when the input space is compact.

[99] Simon-Gabriel and Schölkopf, *Kernel Distribution Embeddings - arXiv*, 2016

Under assumption (ii). In that case, the measures associated to $\hat{\mathbf{X}}$ are probability measures. Theorem 1.3.4 (equivalence between MMD- and σ-convergence for bounded continuous characteristic kernels) hence proves (a). And Lemma 1.3.2 (for bounded continuous kernels, σ-convergence of unsigned measures implies MMD-convergence) proves (c). □

The continuity assumption is rather nonrestrictive. All kernels and functions defined on a discrete space are continuous with respect to the discrete topology, so the theorem applies in this case. For $\mathcal{X} = \mathbb{R}^d$, many kernels used in practice are continuous, including Gaussian, Laplacian, Matérn and other radial kernels. The slightly limiting factor of this theorem is that k_x must be c_0-universal, which often can be tricky to verify. However, most standard kernels—including all radial, non-constant kernels—are c_0-universal (see Sec. 1.4.2 and [109]). The assumption that the input domain is compact is satisfied in most applications, since any measurements coming from physical sensors are contained in a bounded range. Finally, the assumption that $\sum_i |w_i| \leqslant C$ can be enforced, for instance, by applying a suitable regularization in reduced expansion set methods. However, if we end up with some negative weights, or weights that do not sum to 1, then, despite $\hat{\mathbf{X}}$ being close to $\mathbf{X}$ in the MMD sense, we won't know how to sample from $\hat{\mathbf{X}}$. That is why we prefer the second condition, (ii), which imposes that $\hat{\mathbf{X}}$ corresponds to a probability measure. Of course, this requires to use reduced expansion set methods that incorporate the constraints on the weights of $\hat{\mathbf{X}}$.

[109] Sriperumbudur et al., *Universality, Characteristic Kernels and RKHS Embedding of Measures*, 2011

2.2.2 Finite sample guarantees

Theorem 2.2.1 guarantees that the estimator $f(\hat{\mathbf{X}})$ converges to $f(\mathbf{X})$ when $\hat{\mathbf{X}}$ converges to $\mathbf{X}$. However, it says nothing about the speed of convergence. In this section we provide a convergence rate when working with Matérn kernels, which are of the form

$$k_\chi^s(x,x') = \frac{2^{1-s}}{\Gamma(s)} \left\|x-x'\right\|_2^{s-d/2} \mathcal{B}_{s-d/2}\left(\left\|x-x'\right\|_2\right) , \tag{2.2}$$

Matérn Kernel

where $\mathcal{B}_\alpha$ is a modified Bessel function of the third kind of order α [125], Γ is the Gamma function and $s > \frac{d}{2}$ is a smoothness parameter. The RKHS induced by k_χ^s is the Sobolev space $\mathcal{W}_2^s(\mathbb{R}^d)$ [125] containing s-times differentiable functions. The finite-sample bound of Theorem 2.2.3 is based on the analysis of [54], which requires the following assumptions:

[125] Wendland, *Scattered Data Approximation*, 2004, Definition 5.10
[125] Wendland, *Scattered Data Approximation*, 2004, Theorem 6.13 & Chap.10
[54] Kanagawa et al., *Convergence Guarantees for Kernel-Based Quadrature Rules in Misspecified Settings*, 2016

Assumptions 2.2.2. *Let X be a random variable over $\mathcal{X} = \mathbb{R}^d$ with distribution P and let $\hat{X} = \{(x_i, w_i)\}_{i=1}^n$ be random variables over $\mathcal{X}^n \times \mathbb{R}^n$ with joint distribution S. There exists a probability distribution Q with full support on $\mathbb{R}^d$ and a bounded density, satisfying the following properties:*

(i) P has a bounded density function w.r.t. Q;
(ii) there is a constant $A > 0$ independent of n, such that

$$\mathop{\mathbb{E}}_{S}\left[\frac{1}{n}\sum_{i=1}^n g^2(x_i)\right] \leqslant A \left\|g\right\|_{L^2(Q)}^2 , \qquad \forall g \in L^2(Q) .$$

These assumptions were shown to be fairly general and we refer to [54] for various examples where they are met. Next we state the main result of this section.

[54] Kanagawa et al., *Convergence Guarantees for Kernel-Based Quadrature Rules in Misspecified Settings*, 2016, Section 4.1

Theorem 2.2.3. *Let $\mathcal{X} = \mathbb{R}^d$, $\mathcal{Z} = \mathbb{R}^{d'}$, and $f : \mathcal{X} \to \mathcal{Z}$ be an α-times differentiable function ($\alpha \in \mathbb{N}_+$). Take $s > d/2$ and $t > d'$ such that $s, t/2 \in \mathbb{N}_+$. Let k_χ^s and k_z^t be Matérn kernels over $\mathcal{X}$ and $\mathcal{Z}$ respectively as defined in (2.2). Assume $\mathbf{X} \sim P$ and $\hat{\mathbf{X}} = \{(x_i, w_i)\}_{i=1}^n \sim S$ satisfy 2.2.2. Moreover, assume that P and the marginals of $x_1, \dots x_n$ have a common compact support. Suppose that, for some constants $b > 0$ and $0 < c \leqslant 1/2$:*

(i) $\mathbb{E}_S\left[\mathrm{MMD}^2_{k_\chi^s}(\hat{\mathbf{X}}, \mathbf{X})\right] = O(n^{-2b})$;
(ii) $\sum_{i=1}^n w_i^2 = O(n^{-2c})$ (with probability 1) .

Propagating Finite Sample Guarantees from $\hat{\mathbf{X}}$ to $f(\hat{\mathbf{X}})$

Let $\theta = \min(\frac{t}{2s}, \frac{\alpha}{s}, 1)$ and assume $\theta b - (1/2 - c)(1-\theta) > 0$. Then

$$\mathop{\mathbb{E}}_{S}\left[\mathrm{MMD}^2_{k_z^t}(f(\hat{\mathbf{X}}), f(\mathbf{X}))\right] = O\left((\log n)^{d'} n^{-2(\theta b - (1/2-c)(1-\theta))}\right) .$$

Before we provide a short sketch of the proof, let us briefly comment on this result. As a benchmark, remember that when $x_1, \dots x_n$ are iid observations from $\mathbf{X}$ and $\hat{\mathbf{X}} = \{(x_i, 1/N)\}_{i=1}^N$, we

get $|\mathrm{MMD}(f(\hat{\mathbf{X}}), f(\mathbf{X}))|^2 = O_P(1/n)$, which was recently shown minimax optimal [118]. How do we compare to this benchmark? In this case we have $b = c = 1/2$ and our rate is defined by θ. If f is smooth enough, say $\alpha > d/2 + 1$, and by setting $t > 2s = 2\alpha$, we recover the $O(1/n)$ rate up to an extra $(\log n)^{d'}$ factor.

[118] Tolstikhin et al., *Minimax Estimation of Kernel Mean Embeddings*, 2017

However, Theorem 2.2.3 applies to much more general settings. Importantly, it makes no iid assumptions on the data points and weights, allowing for complex interdependence. Instead, it asks the MMD-convergence of the estimator $\hat{\mathbf{X}}$ to $\mathbf{X}$ to be sufficiently fast. On the downside, the upper bound is affected by the smoothness of f, even in the iid setting: if $\alpha < d/2$ the rate will become slower, as $\theta = \alpha/s$. Also, the rate depends both on d and d'. Whether these are artifacts of our proof remains an open question.

Proof. Here we sketch the main ideas of the proof and develop the details in Appendix C.2.1. Throughout the proof, C will designate a constant that depends neither on the sample size n nor on the variable R (to be introduced). C may however change from line to line. It will be convenient to reason on KME rather than on MMDs. To do so, we will reason on the KMEs $\mu_X^{k_x}$ and $\hat{\mu}_X^{k_x}$ of $\mathbf{X}$ and $\hat{\mathbf{X}}$ (see Eq. 2.1). Hence remember that: $\mathrm{MMD}_{k_x}(\hat{\mathbf{X}}, \mathbf{X}) = \left\| \hat{\mu}_X^{k_x} - \mu_X^{k_x} \right\|_{k_x}$. We start by showing that:

$$\mathop{\mathbb{E}}_{S}\left[\left\|\hat{\mu}^{k_z}_{f(\mathbf{X})} - \mu^{k_z}_{f(\mathbf{X})}\right\|^2_{k_z}\right] = (2\pi)^{\frac{d'}{2}} \int_{\mathcal{Z}} \mathop{\mathbb{E}}_{S}\left[\left([\hat{\mu}^{h}_{f(\mathbf{X})} - \mu^{h}_{f(\mathbf{X})}](z)\right)^2\right] dz, \tag{2.3}$$

where h is Matérn kernel over $\mathcal{Z}$ with smoothness parameter $t/2$. Second, we upper bound the integrand by roughly imitating the proof idea of Theorem 1 in [54]. This eventually yields:

[54] Kanagawa et al., *Convergence Guarantees for Kernel-Based Quadrature Rules in Misspecified Settings*, 2016

$$\mathop{\mathbb{E}}_{S}\left[\left([\hat{\mu}^{h}_{f(\mathbf{X})} - \mu^{h}_{f(\mathbf{X})}](z)\right)^2\right] \leqslant C n^{-2\nu}, \tag{2.4}$$

where $\nu := \theta b - (1/2 - c)(1 - \theta)$. Unfortunately, this upper bound does not depend on z and can not be integrated over the whole $\mathcal{Z}$ in (2.3). Denoting B_R the ball of radius R, centered on the origin of $\mathcal{Z}$, we thus decompose the integral in (2.3) as:

$$\begin{aligned}\int_{\mathcal{Z}} \mathbb{E}\left[\left([\hat{\mu}^{h}_{f(\mathbf{X})} - \mu^{h}_{f(\mathbf{X})}](z)\right)^2\right] dz = &\int_{B_R} \mathbb{E}\left[\left([\hat{\mu}^{h}_{f(\mathbf{X})} - \mu^{h}_{f(\mathbf{X})}](z)\right)^2\right] dz \\ &+ \int_{\mathcal{Z}\setminus B_R} \mathbb{E}\left[\left([\hat{\mu}^{h}_{f(\mathbf{X})} - \mu^{h}_{f(\mathbf{X})}](z)\right)^2\right] dz.\end{aligned}$$

On B_R we upper bound the integral by (2.4) times the ball's volume (which grows like R^d):

$$\int_{B_R} \mathbb{E}\left[\left([\hat{\mu}^{h}_{f(\mathbf{X})} - \mu^{h}_{f(\mathbf{X})}](z)\right)^2\right] dz \leqslant C R^d n^{-2\nu}. \tag{2.5}$$

On $\mathcal{Z}\backslash B_R$, we upper bound the integral by a value that decreases with R, which is of the form:

$$\int_{\mathcal{Z}\backslash B_R} \mathbb{E}\left[\left([\hat{\mu}^h_{f(\mathbf{X})} - \mu^h_{f(\mathbf{X})}](z)\right)^2\right] dz \leqslant Cn^{1-2c}(R-C')^{t-2}e^{-2(R-C')} \tag{2.6}$$

with $C' > 0$ being a constant smaller than R. In essence, this upper bound decreases with R because $[\hat{\mu}^h_{f(\mathbf{X})} - \mu^h_{f(\mathbf{X})}](z)$ decays with the same speed as h when $\|z\|$ grows indefinitely. We are now left with two rates, (2.5) and (2.6), which respectively increase and decrease with growing R. We complete the proof by balancing these two terms, which results in setting $R \approx (\log n)^{1/2}$. □

2.3 Functions of Multiple Arguments

The previous section applies to functions f of one single but possibly multi-dimensional variable $\mathbf{X}$. We now concentrate on the specific case where $\mathbf{X}$ is multi-dimensional, i.e. when f is a function of several scalar random variables. For ease of notations, we concentrate on the case of two input variables from spaces $\mathcal{X}$ and $\mathcal{Y}$ respectively, but the results also apply to more inputs. To be precise, our input space changes from $\mathcal{X}$ to $\mathcal{X} \times \mathcal{Y}$, input random variable from $\mathbf{X}$ to (X, Y), and the kernel on the input space from k_x to k_{xy}.

Constructing a Consistent Estimator of the Joint (X, Y)

To apply our results from Section 2.2, all we need is an MMD-consistent estimator $\widehat{(X,Y)}$ of the joint distribution (X, Y). There are different ways to get such an estimator. One way is to sample (x_i, y_i) iid from the joint distribution of (X, Y) and construct the usual empirical estimator, or approximate it using reduced expansion set methods. Alternatively, we may want to construct $\widehat{(X,Y)}$ based on separately consistent estimators of X and Y, like we did with the U-statistics $\widehat{u}$ and $\widehat{u}'$ in Section 2.1.4. Below we show that this can indeed be done consistently, *provided that X and Y be independent*.

Given two sample estimators $\hat{X}$ and $\hat{Y}$, we denote $\hat{X} \otimes \hat{Y}$ their associated U-statistic estimator (see Section 2.1.4). Similarly, given kernels k_x and k_y on $\mathcal{X}$ and $\mathcal{Y}$ respectively, we denote k_{xy} their (tensor-) product kernel: $k_{xy}\big((x,y),(x',y')\big) := k_x \otimes k_y((x,y),(x',y')) := k_x(x,x')k_y(y,y')$. Note that the KME of any random variables (X, Y) then becomes the tensor product of the KME of their marginals: $\mu_{(X,Y)} = \mu_X \otimes \mu_Y$.

Lemma 2.3.1. *Let $(s_n)_n$ be any positive sequence converging to zero. Then*

From Separate to Joint Consistency of IID Variables

$$\left.\begin{aligned} \mathrm{MMD}_{k_x}(\hat{X}, X) &= O(s_n) \\ \mathrm{MMD}_{k_y}(\hat{Y}, Y) &= O(s_n) \end{aligned}\right\} \implies \mathrm{MMD}_{k_{xy}}(\hat{X} \otimes \hat{Y}, X \otimes Y) = O(s_n)\,.$$

Proof. For a detailed expansion of the first inequality see App. C.2.2.

$$\begin{aligned}\mathrm{MMD}_{k_{xy}}(\hat{X}\otimes\hat{Y}, X\otimes Y) &\leqslant \|\mu_X\|_{k_x}\|\hat{\mu}_Y-\mu_Y\|_{k_y}+\|\mu_Y\|_{k_y}\|\hat{\mu}_X-\mu_X\|_{k_x}\\ &+\|\hat{\mu}_X-\mu_X\|_{k_x}\|\hat{\mu}_Y-\mu_Y\|_{k_y} = O(s_n)+O(s_n)+O(s_n^2) = O(s_n). \quad \square\end{aligned}$$

Corollary 2.3.2. *If* $\hat{X} \xrightarrow{k_x} X$ *and* $\hat{Y} \xrightarrow{k_y} Y$*, then* $\hat{X}\otimes\hat{Y} \xrightarrow{k_{xy}} X\otimes Y$.

Noting that when X and Y are independent, by definition, $(X,Y) = X\otimes Y$, we can combine Corollary 2.3.2 with the MMD-consistency Theorem 2.2.1 and get

Corollary 2.3.3. *If* X, Y *are independent random variables and if* $\hat{X}, \hat{Y}$ *satisfy either both condition (i) or both condition (ii) of Theorem 2.2.1, then*

Propagating Separate Consistency of IID Variables X, Y *to* $f(X,Y)$

$$\left.\begin{aligned}\hat{X} &\xrightarrow{k_x} X\\ \hat{Y} &\xrightarrow{k_y} Y\end{aligned}\right\} \quad\Longrightarrow\quad f(\hat{X},\hat{Y}) \xrightarrow{k_{xy}} f(X,Y).$$

Unfortunately, we cannot apply Theorem 2.2.3 to get the speed of convergence, because a product of Matérn kernels is not a Matérn kernel anymore. Another downside of this overall approach is that the number of expansion points used for the estimation of the joint increases exponentially with the number of arguments of f. This can lead to prohibitively large computational costs, especially if the result of such an operation is used as an input to another function of multiple arguments. That is why we may use the reduced expansion set methods mentioned earlier, either before or after applying f.

To conclude this section, let us summarize the implications of our results for two practical scenarios that should be distinguished. First scenario: if we have separate samples from two random variables X and Y, then our results justify how to provide an estimate of the mean embedding of $f(X,Y)$ provided that X and Y are *independent*. The samples themselves need not be iid– we can also work with weighted samples computed, for instance, by a reduced expansion set method. Second scenario: How about *dependent* random variables? For instance, imagine that $Y = -X$, and $f(X,Y) = X+Y$. Clearly, in this case the distribution of $f(X,Y)$ is a delta measure on 0, and there is no way to predict this from separate samples of X and Y. However, it should be stressed that our results (consistency and finite sample bound) apply even to the case where X and Y are dependent. In that case, however, they require a consistent estimator of the joint embedding $\mu_{(X,Y)}$.

Sec 2.2: any (X,Y) *but joint conv.*
Sec 2.3: separate conv., but $X \perp\!\!\!\perp Y$.

2.4 Chapter Conclusion

This chapter provides convergence guarantees when representing random variables by finite samples and taking arbitrary continuous

functions of them. When working with bounded, continuous and characteristic kernels, we show that any MMD-consistent estimator of $\mathbf{X}$ leads to an MMD-consistent estimator of $f(\mathbf{X})$ provided that f be continuous. For Matérn kernels and smooth enough functions f, we corroborate these results with convergence rate bounds. Importantly, our results make no iid assumption on the sample-points and therefore also apply to estimators with interdependent expansion points and weights, such as MCMC, compressed or privatized samples. One interesting future direction is to improve the finite-sample bounds and extend them to general radial and/or translation-invariant kernels.

Our work is motivated by the field of probabilistic programming. Using our theoretical results, kernel mean embeddings can be used to generalize functional operations (which lie at the core of all programming languages) to distributions over data types in a principled manner, by applying the operations to the points or approximate kernel expansions. This is in principle feasible for any data type provided a suitable kernel function can be defined. We believe that the approach holds significant potential for future probabilistic programming systems.

3

Kernel Stein Discrepancies

Our two previous chapters established sufficient conditions for a kernel k to metrize weak convergence. Those kernels guarantee that, *whatever* probability measure P we choose, a sequence $(P_n)_n$ converges weakly to P iff they converge in MMD metric. But in the context of sample quality measurement and goodness-of-fit testing, we only care about convergence to a known and fixed target measure P. We may then accept a kernel k that is not characteristic to all probability measures $\mathcal{P}$, as long as its MMD can at least distinguish P from all other probability measures and ensure that if $P_n \to_k P$, then $P_n \to_\sigma P$. Said differently, we may be content with a kernel satisfying (B) but not (A), where

From Global To Targeted Characteristicness & Convergence

(A) **target-indep. metr.:** $\forall P \in \mathcal{P}, \forall P_n \in \mathcal{P}$: $P_n \xrightarrow{k} P$ iff $P_n \xrightarrow{\sigma} P$;
(B) **targeted metrization:** $P \in \mathcal{P}, \forall P_n \in \mathcal{P}$, $P_n \xrightarrow{k} P$ iff $P_n \xrightarrow{\sigma} P$.

Of course (A) implies (B), so why not choose any usual characteristic kernel? One reason is that, typically, the MMD between a discrete measure P_n (the sample) and an arbitrary target P is analytically untractable, because it involves integrating k wrt P:

MMDs with Arbitrary Targets are Typically Untractable

$$\mathrm{MMD}_k^2(P_n, P) = \|\Phi_k(P_n)\|_k^2 + \underbrace{\|\Phi_k(P)\|_k^2 + \langle \Phi_k(\hat{P}), \Phi_k(P) \rangle_k}_{\text{analytically untractable}}$$

Most usual kernels are thus inappropriate for goodness-of-fit tests. Given a target P, Liu et al. [68] and Chwialkowski et al. [21] however proposed an elegant trick to transform any given kernel k into a new kernel κ called the *Stein kernel* of k, such that the KME of P wrt κ be the null function: $\Phi_\kappa(P) = 0$. The resulting MMD_κ – called the *Kernel Stein Discrepancy* KSD_k wrt k – is easily computable on any sample, since for a sample $P_n = 1/N \sum_{i=1}^N \delta_{x_i}$ it reduces to

[68] Liu et al., *A Kernelized Stein Discrepancy for Goodness-of-fit Tests*, 2016
[21] Chwialkowski et al., *A Kernel Test of Goodness of Fit*, 2016

KSD: Easy to Compute, but not Globally Characteristic

$$\mathrm{KSD}_{k,P}(P_n) := \mathrm{MMD}_\kappa(P_n, P) = \|P_n\| = \frac{1}{N^2} \sum_{i,j=1}^N \kappa(x_i, x_j) \,.$$

It is hence very-well suited for sample quality assessment and goodness-of-fit tests. But, despite the freedom to choose k, the resulting Stein kernel κ will typically be unbounded, yielding an MMD that is not even defined on all $\mathcal{P}$, let alone characteristic to $\mathcal{P}$. That is why we will here both focus on targeted metrization and on replacing the kernel's boundedness assumption by an assumption on

But Targeted Properties Suffice

its moments of order α. This will lead us to consider not only usual weak, but also Wasserstein-α convergence results, which, as we will see, is essentially weak convergence plus a moment convergence condition.

Contributions & outline. Theorem 3.1.3 & 3.1.4 first establish sufficient conditions for a kernel k to ensure targeted metrization (B). We then apply those results to KSDs and substantially generalize all existing results. Incidentally, these results and their proofs will naturally lead to use Schwartz-distributions, yielding a first application of the Schwartz-distribution specific results of Chapter 1. Contrary to most of the existing literature, our results also apply to unbounded kernels.

3.1 From Global to Targeted Weak Convergence

After a few preliminaries on Wasserstein convergence and uniform integrability, this section proves necessary and sufficient conditions for a kernel k to metrize targeted weak convergence of tight sequences and targeted Wasserstein-convergence of uniformly integrable sequences.

3.1.1 Preliminaries

$\mathcal{X}$ is now a Polish Space

Input and function spaces. Contrary to Chapter 1 & 2, ***letter $\mathcal{X}$ now designates a Polish space*** (separable, metric and complete) and $|\mathbf{x}|$ denotes the distance of $\mathbf{x} \in \mathcal{X}$ to an arbitrary reference point $\mathbf{x}_0 \in \mathcal{X}$. Starting from Section 3.2, we will take $\mathcal{X} = \mathbb{R}^d$ and $|\mathbf{x}|$ will be any norm on $\mathbb{R}^d$. For a non-negative real $\alpha \geqslant 0$, we say that a real-valued function f on $\mathcal{X}$ has α-growth and write $f(\mathbf{x}) = O(|\mathbf{x}|^\alpha)$ if there exists constants A, B such that $|f(\mathbf{x})| \leqslant A + B|\mathbf{x}|^\alpha$. The sets $\mathscr{C}$, $\mathscr{C}_\alpha$, $\mathscr{C}_{\to 0}$ respectively designate the continuous functions, the continuous functions with α-growth and the continuous functions that vanish at infinity. In addition to $\mathscr{C}^m$, $\mathscr{C}^m_{\to 0}$, $\mathscr{C}^{(m,m)}$, $\mathscr{C}^{(m,m)}_{\to 0}$ defined in Chapter 1, we now also introduce the set $\mathscr{C}_\alpha$ of continuous functions with α-growth. On $\mathcal{X} = \mathbb{R}^d$, the functions with 0-growth are exactly the continuous bounded ones, yielding $\mathscr{C}_0 = \mathscr{C}_b$.

α-Growth

α-Moments

Probability spaces and weak- and α-convergence. $\mathcal{P}_\alpha$ denotes the set of probability measures with bounded α-moments, i.e.

$$\mathcal{P}_\alpha := \{P \in \mathcal{P} : \int |\mathbf{x}|^\alpha \, dP(\mathbf{x}) < \infty\}.$$

Tightness & Uniform Integration

A sequence $P_n \in \mathcal{P}$ is *tight* iff for any $\epsilon > 0$ there exists a compact $K \subset \mathcal{X}$ such that $\sup_n P_n(\mathcal{X} \backslash K) \leqslant \epsilon$. It has *uniformly integrable α-moments*

iff for any $\epsilon > 0$ there exists $r > 0$ such that $\sup_n \int_{|x| \geqslant r} |x|^\alpha \, dP_n(x) \leqslant \epsilon$. Note that a tight sequence has uniformly integrable 0-moments; and vice-versa if $\mathcal{X} = \mathbb{R}^d$.

By definition, for $P_n, P \in \mathcal{P}$, the sequence $(P_n)_n$ converges weakly to P, denoted $P_n \to_\sigma P$, if for any $f \in \mathscr{C}_b$, $P_n(f) \to P(f)$. For $P_n, P \in \mathcal{P}_\alpha$, we say that P_n α-converges to P and write $P_n \to_\alpha P$, if for any $f \in \mathscr{C}_\alpha$, $P_n(f) \to P(f)$. While α-convergence is stronger than weak convergence, the following proposition, taken from [121] shows that α-convergence is essentially the same as weak convergence plus a constraint on the convergence of the moments (which incidentally ensures that the target P be in $\mathcal{P}_\alpha$).

[121] Villani, *Optimal Transport: Old and New*, 2009, Def 6.8 & Thm 6.9

Proposition 3.1.1. *For any $\alpha > 0$, $P_n, P \in \mathcal{P}_\alpha$, the following are equivalent:*

(i) $P_n \xrightarrow{\alpha} P$
(ii) $P_n \xrightarrow{\sigma} P$ *and* $(P_n)_n$ *has uniformly integrable α-moments*
(iii) $P_n \xrightarrow{\sigma} P$ *and* $\int_{\mathcal{X}} |x|^\alpha \, dP_n(x) \to \int_{\mathcal{X}} |x|^\alpha \, dP(x)$
(iv) P_n *converges to* P *in Wasserstein-α distance.*

$$\alpha\text{-conv} \Leftrightarrow \begin{cases} \text{weak-convergence} \\ \text{moment condition} \end{cases}$$

As mentioned in previous chapters, weak convergence can be metrized by the Dudley metric. Similarly, point (iv) states that α-convergence is metrized by the so-called Wasserstein-α distance (see Section 0.1 and [121]). Finally, note that, similarly to Corollary 1.1.3, if $k \in \mathscr{C}_{2\alpha}$, i.e. if $k(x,x) = O(|x|^{2\alpha})$, then $\mathcal{P}_\alpha$ embeds into $\mathcal{H}_k$.[1]

[121] Villani, *Optimal Transport: Old and New*, 2009, Def 6.1

[1] Use Bochner's integration criterion: $\int \|k(\cdot,x)\|_k \, dP(x) = \int \sqrt{k}(x,x) \, dP(x) < \infty$.

3.1.2 Targeted Metrization of Weak and α-Convergence

In this section we establish sufficient conditions on k to metrize targeted weak convergence. For target-independent metrization, Theorem 1.3.4 showed that a bounded continuous kernel metrize weak convergence iff it is characteristic to $\mathcal{P}$. Up to an additional tightness assumption, it will be the same for targeted metrization, but with targeted characteristicness instead of characteristicness.

Definition 3.1.2. *Given a kernel k, let $\mathcal{D}$ be any set of embeddable distributions. We say that a kernel k is* characteristic to P in $\mathcal{D}$ *(or $P \in \mathcal{D}$) iff for any $Q \in \mathcal{D}$, $\mathrm{MMD}_k(Q,P) = 0 \Rightarrow Q = P$.*

Targeted Characteristicness

If k is characteristic to all measures P in $\mathcal{P}$, then we recover Definition 1.2.1: k is characteristic to $\mathcal{P}$. With targeted characteristicness, it is crucial to specify the target P *and* the surrounding set $\mathcal{D}$, because k could be characteristic to $P \in \mathcal{P}_\alpha$ without being characteristic to $P \in \mathcal{P}$. Now, if k is not characteristic to $P \in \mathcal{P}$, then MMD_k obviously cannot metrize weak convergence to P. The next theorem shows that conversely, up to a tightness assumption, this condition is actually

sufficient to metrize weak convergence to P. Theorem 3.1.4 then generalizes this to α-convergence, which also applies for unbounded kernels. The proofs – very different in flavor from those for untargeted metrization – are in Appendix C.3.

Targeted Metrization of Weak Convergence

Theorem 3.1.3. *Let k be a bounded continuous kernel. Then k is characteristic to $P \in \mathcal{P}$ iff, for a sequence of probability measures P_n*

$$P_n \xrightarrow{\sigma} P \iff \begin{cases} (a) & P_n \xrightarrow{k} P \\ (b) & (P_n)_n \text{ is tight.} \end{cases}$$

Targeted Metrization of α-Convergence

Theorem 3.1.4. *Let k be a continuous kernel such that $k(x,x) = O(|x|^{2\alpha})$. Then k is characteristic to $P \in \mathcal{P}_\alpha$ iff, for any sequence $P_n \in \mathcal{P}_\alpha$*

$$P_n \xrightarrow{\alpha} P \iff \begin{cases} (a) & P_n \xrightarrow{k} P \\ (b) & (P_n)_n \text{ is tight} \\ (c) & (P_n)_n \text{ has uniformly integrable } \alpha\text{-moments.} \end{cases}$$

Theorem 3.1.3 is itself a particular case of Theorem 3.1.4 with $\alpha = 0$. Indeed, $\mathcal{P}_0 = \mathcal{P}$, and a tight sequence automatically has uniformly integrable 0-moments. Now, some comments on conditions (a)-(c). First, when $\mathcal{X}$ is $\mathbb{R}^d$ equipped with any norm, the tightness assumption (b) in Theorem 3.1.4 is already implied by the uniform integrability (c) and can therefore be dropped. Second, note how well Theorem 3.1.4 reflects the equivalence between points (i) and (ii) of Theorem 3.1.1: (a) and (b) ensure that $P_n \to_\sigma P$ as in Theorem 3.1.3, while (c) "promotes" the weak-convergence to α-convergence. Intuitively, this uniform integrability property (c) guarantees that the α-moments of $(P_n)_n$ all stay localized in space $\mathcal{X}$, so that the measures P_n and P stay confined in $\mathcal{P}_\alpha$. As for the tightness assumption (b), it ensures that no mass can "escape to infinity", so that if P_n has a weak limit (in the set of finite measures), than this limit is in $\mathcal{P}$. An example of mass escaping to infinity is the following. Take $\mathcal{X} = \mathbb{R}$ and $P_n := \frac{1}{n}\sum_{i=1}^n \delta_i$. Then P_n weakly converges to the null measure, which is outside of $\mathcal{P}$. This can only happen because $(P_n)_n$ is not tight [2]. Similar phenomena may happen with MMDs. For example, for any bounded continuous kernel, weak convergence of finite nonnegative measures implies MMD-convergence (Lemma 1.3.2. Hence if P_n weakly converges to the null measure, then $\mathrm{MMD}_k(P_n, 0) \to 0$.

Tightness Prevents Mass-Escape to ∞

[2] Ambrosio et al., *Gradient Flows*, 2005, Prokhorov's Thm 5.1.3

As a short detour, let us now "aggregate" the results of Theorems 3.1.3 (and of Theorem 3.1.4 respectively (resp.)) over different targets P, to get a new corollary on target-independent metrization. To do so let us say that k *enforces tightness* (resp. and *uniform α-integrability*) over $\mathcal{P}$ (resp. over $\mathcal{P}_\alpha$) if, for any $P_n, P \in \mathcal{P}$ (resp. $\in \mathcal{P}_\alpha$), if $\mathrm{MMD}_k(P_n, P) \to 0$ then $(P_n)_n$ is tight (resp. and has uniformly integrable α-moments).

Corollary 3.1.5. *A bounded kernel* k *metrizes weak convergence over* $\mathcal{P}$ *iff it is continuous, characteristic to* $\mathcal{P}$ *and enforces tightness in* $\mathcal{P}$.

A kernel such that $k(x,x) = O(|x|^{2\alpha})$ *enforces* α*-convergence over* $\mathcal{P}_\alpha$ *iff it is continuous, characteristic to* $\mathcal{P}_\alpha$ *and enforces tightness and uniform* α*-integrability over* $\mathcal{P}_\alpha$.

Targeted Convergence Everywhere Implies Untargeted Convergence

This corollary is very similar to our main theorem on untargeted weak-convergence metrization in Chapter 1, Theorem 1.3.4. There are notable differences though. On the one hand, Corollary 3.1.5 also handles unbounded kernels and Wasserstein convergence. More importantly perhaps, it holds on Polish spaces rather than locally compact ones, which makes it more relevant to abstract modern probability theory, which is now mainly based on Polish spaces. On the other hand however, Theorem 1.3.4 makes no tightness assumption, which not only shows that the present corollary adds a superfluous assumption (at least on locally compact spaces), but is also of great advantage in practice. It guarantees that the kernel automatically enforces tightness: a property that many common kernel Stein discrepancies lack [39].

[39] Gorham and Mackey, *Measuring Sample Quality with Kernels*, 2017, Thm 6

Proof. (of Corollary 3.1.5) Let k be bounded. Suppose that k metrizes weak convergence. Then k is characteristic to $\mathcal{P}$ (otherwise MMD_k is not even a proper metric), and it enforces tightness, because any weakly converging sequence is tight by Prokhorov's theorem [2]. We now show that k is continuous. Let $x_n, y_n, x, y \in \mathcal{X}$ such that $x_n \to x$ and $y_n \to y$. Then the Dirac masses δ_{x_n} weakly converge to δ_x and similarly with y. But Dirac masses are embeddable into $\mathcal{H}_k$. Hence their embeddings converge and give: $k(x_n, y_n) = \langle \Phi_k(\delta_{x_n}), \Phi_k(\delta_{y_n})\rangle_k \to \langle \Phi_k(\delta_x), \Phi_k(\delta_y)\rangle_k = k(x,y)$. Thus k is continuous. Conversely, suppose that k is continuous, characteristic to $\mathcal{P}$ and enforces tightness in $\mathcal{P}$. Then by Theorem 3.1.3, k metrizes weak convergence to any target $P \in \mathcal{P}$, thus metrizes weak convergence over $\mathcal{P}$. The proof for the α-convergence case goes almost exactly the same way. Only add point (ii) of Proposition 3.1.1 to show that metrizing α-convergence implies enforcing uniform α-integrability. □

[2] Ambrosio et al., *Gradient Flows*, 2005, Prokhorov's Thm 5.1.3

3.2 When are Stein Kernels Characteristic?

This section applies the previous results to kernel Stein discrepancies. We will determine when the KSD is characteristic to a target $P \in \mathcal{P}$, thereby metrizing weak-convergence to P. Doing so, we generalize to any measure $Q \in \mathcal{P}$ a result by Chwialkowski et al. [21] which holds only for measures with a continuously differentiable density q. The trick will be to mimic their proof, which uses the gradient $\partial_x q$ of q,

[21] Chwialkowski et al., *A Kernel Test of Goodness of Fit*, 2016, Thm. 2.2

but replace usual differentiation with Schwartz-differentiation. We now first formally introduce KSDs in Section 3.2.1 and then proceed with the desired results in Section 3.2.2. In this section we take $\mathcal{X} = \mathbb{R}^d$. For relevant work on compact domains, see [83].

[83] Oates et al., *Convergence Rates for a Class of Estimators Based on Stein's Method*, 2018

3.2.1 Introduction to Kernel Stein Discrepancies

As mentioned earlier, $\mathrm{MMD}(Q, P)$ is easy to compute when P and Q are two sample measures. But when either one is non-discrete, the MMD usually becomes analytically intractable. The KSD is an MMD that circumvents this intractability when the target P has a continuously differentiable and fully supported density p, *known up to a normalizing constant*, which we will always assume from now on. Indeed, out of any given base kernel k, we can construct a new kernel κ – which depends on both k and P – that maps P to the null function of the RKHS $\mathcal{H}_\kappa$ of κ. The new MMD, i.e. the KSD, thus ends up being the RKHS distance between the embedding of Q and the null function. More formally:

Proposition & Definition 3.2.1. *Let k be a kernel in $\mathscr{C}^{(1,1)}$ and P be a probability measure with a continuously differentiable, fully-supported density p. Let $s_P^i(x) := \partial_{x_i} \log p(x)$ denote the i-th coordinate of the score function s_P of p. Define a new kernel κ that depends both on P and k as*

$$\kappa(x, y) := \sum_{i=1}^d \kappa^i(x, y) \quad \textit{where}$$
$$\kappa^i(x, y) := \begin{cases} s_P^i(x) s_P^i(y) k(x, y) + s_P^i(x) \partial_{y_i} k(x, y) \\ + s_P^i(y) \partial_{x_i} k(x, y) + \partial_{x_i} \partial_{y_i} k(x, y) \,. \end{cases}$$

Stein Kernel κ of k

Then P embeds into $\mathcal{H}_\kappa$ and its embedding is the null function of $\mathcal{H}_\kappa$. Therefore, $d_\kappa(Q, P) = \|\Phi_\kappa(Q)\|_\kappa$, and we define the kernel Stein discrepancy (KSD) between Q and P with kernel k as

$$\mathrm{KSD}^2_{k,P}(Q) := \mathrm{MMD}^2_\kappa(Q, P) = \int_{\mathcal{X}\times\mathcal{X}} \kappa(x, y) \, dQ(x) Q(y) \,.$$

KSD Definition

For the proof of this proposition, see Theorem 2.1 in [21] and preceding discussions therein. Now two remarks on this definition. First, let us once again emphasize that contrary to usual MMDs, where the kernel does not specifically target P, computing the KSD needs no explicit integration with respect to P. Hence, when Q is the empirical measure associated to a sample S of size n, the KSD simply reduces to $\mathrm{KSD}^2_{k,P}(Q) = \frac{1}{n^2} \sum_{x,y \in S} \kappa(x, y)$. That makes it an easily computable sample quality measure. Second, the new kernel κ is independent of the normalization constant of P, which means that the KSD can be computed even without knowing this constant. Both these properties – computability and normalization-independence –

[21] Chwialkowski et al., *A Kernel Test of Goodness of Fit*, 2016

make the KSD typically very-well adapted to evaluate the quality of MCMC-samples and construct powerful stopping-criteria.

Finally, let us mention that the KSD can also be seen as the IPM distance between Q and P (see Eq. 0.3), where the set of test-functions $\mathcal{F} = \mathcal{T}(\mathcal{H}_k)$ is $\mathcal{H}_k$ mapped through a so-called Langevin Stein operator $\mathcal{T}$, which has the property that, for any $f \in \mathcal{T}(\mathcal{H}_k)$, $P(f) = 0$. Hence $\mathrm{KSD}_{k,P}(Q) = \sup_{f\in\mathcal{T}(\mathcal{H}_k)} |Q(f) - P(f)| = \sup_{f\in\mathcal{T}(\mathcal{H}_k)} |Q(f)|$. While we will not need $\mathcal{T}$, this point of view nevertheless leads to another way to compute the KSD which we will use later, and therefore reproduce here. Again, see Theorem 2.1 in [21] and preceding discussions therein for a proof.

[21] Chwialkowski et al., *A Kernel Test of Goodness of Fit*, 2016

Proposition 3.2.2. *Given k and P as in Def. 3.2.1, define for any $x, y \in \mathbb{R}^d$*

$$\xi_P(x,y) := \sum_{i=1}^d \xi_P^i(x,y) \quad \textit{where} \quad \xi_P^i(x,y) := \begin{cases} s_P^i(x)k(x,y) \\ +\partial_{x_i}k(x,y) \end{cases} .$$

Then for any probability distribution Q, Q is embeddable into $\mathcal{H}_\kappa$ iff for every coordinate i, ξ_P^i is Pettis-integrable by Q, in which case:

KSD as the Integral of a Function in $\mathcal{H}_k$

$$\mathrm{KSD}^2_{k,P}(Q) = \sum_{i=1}^d \left\| \mathop{\mathbb{E}}_{x\sim Q}\left[\xi_P^i(x,\cdot)\right] \right\|_k^2 .$$

3.2.2 Metrizing Targeted Weak Convergence with KSDs

The goal of this section is to characterize the metrization of weak convergence by KSD metrics. Such a property is important for developing effective sample quality measures and constructing powerful goodness-of-fitness tests, as it ensures that the $\mathrm{KSD}_{k,P}(Q)$ is small iff Q is close to P in a traditional, trustworthy sense. Chwialkowski et al. [21] and Liu et al. [68] previously established targeted characteristicness of KSDs under strong assumptions on the approximating distributions Q and base kernel k, while [39] established tight convergence control for KSDs under strong assumptions on the target P and base kernel k. Our first contribution generalizes all of these results, requiring weaker assumptions on the approximating distributions, target, and kernels. Interestingly, the proof (in Appendix C.3.4) is almost literally the same as in [21], but with usual differentiation replaced by Schwartz-differentiation. This swap makes Schwartz-distribution appear naturally and is thereby the first application of kernel Schwartz-distribution embeddings.

[21] Chwialkowski et al., *A Kernel Test of Goodness of Fit*, 2016, Thm 2.2
[68] Liu et al., *A Kernelized Stein Discrepancy for Goodness-of-fit Tests*, 2016, Prop 3.3
[39] Gorham and Mackey, *Measuring Sample Quality with Kernels*, 2017, Thm 7

[21] Chwialkowski et al., *A Kernel Test of Goodness of Fit*, 2016

Theorem 3.2.3. *Let k be a kernel in $\mathscr{C}_b^{(1,1)}$, and κ be the Stein kernel constructed from k and a target P. Let $\mathcal{P}_\kappa$ be the set of probability measures that embed into $\mathcal{H}_\kappa$. Then the following statements are equivalent.*

KSD is perfectly discriminative iff k charact. to some Schwartz-distr.

(i) κ *is characteristic to* P *in* $\mathcal{P}_\kappa$.
(ii) For any $Q \in \mathcal{P}_\kappa$: $\mathrm{KSD}_{k,P}(Q) = 0 \;\Rightarrow\; Q = P$.
(iii) k *is characteristic to the null measure in the set*

$$\mathcal{D} := \{s_P^i Q - \partial_{x_i} Q \,:\, Q \in \mathcal{P}_\kappa, 1 \leqslant i \leqslant d\}.$$

If those conditions are satisfied and if $P_\alpha \subset P_\kappa$ *for some* $\alpha \geqslant 0$*, then, for any sequence* $(P_n)_n$ *in* P_α*:*

KSD-Metrization of Targeted Convergence

$$P_n \to_\alpha P \iff \begin{cases} \text{(a)} & \mathrm{KSD}_{k,P}(P_n) \to 0 \\ \text{(c)} & (P_n)_n \text{ has uniformly integrable } \alpha\text{-moments.} \end{cases}$$

Theorem 3.2.3 gives necessary and sufficient conditions for the KSD to metrize α-convergence to P of κ-embeddable, uniformly integrable sequences. Two remarks are in order. First, the case $\alpha = 0$ corresponds to the usual weak convergence. Second, Theorem 3.2.3 has no equivalent of the tightness condition (b) in Theorem 3.1.4 because in $\mathcal{X} = \mathbb{R}^d$, uniform integrability implies tightness. The following corollary now replaces conditions (i)-(ii) by stronger ones, but which are often easier to check in practice. The idea is simple: First notice that the set $\mathcal{D}$ is a subset of $\mathcal{D}^1_{L^1}$, the set of integrable distributions (defined in Chapter 1), because, by Lemma 1.4.3:

$$\mathcal{D}^1_{L^1} = \{\mu + \partial_{x_i}\nu \,:\, \mu, \nu \in \mathcal{M}_f\}\,. \tag{3.1}$$

Hence, if k is characteristic to $\mathcal{D}^1_{L^1}$ – for example, if k is c_0^1-universal –, it is a fortiori characteristic to $\mathcal{D}$.

k characteristic to $\mathcal{D}^1_{L^1}$ $\Rightarrow$ KSD_k characteristic to $P \in \mathcal{P}$

Corollary 3.2.4. *Let* k *be a kernel in* $\mathscr{C}_b^{(1,1)}$ *and suppose that* k *is characteristic to* $\mathcal{D}^1_{L^1}$ *(see Eq. 3.1), for example* c_0^1*-universal. Let* $Q \in \mathcal{P}$ *be such that* $\mathbb{E}_{x,y\sim Q}[\kappa(x,y)] < \infty$*. Then* $\mathrm{KSD}_{k,P}(Q) = 0$ *iff* $Q = P$.

Proof. (of Corollary 3.2.4) First, Theorem 1.2.2 and Table 1.1 show that a c_0^1-universal kernel is characteristic over $\mathcal{D}^1_{L^1}$, which explains the "for example" part. Next, $\mathbb{E}_{x,y\sim Q}[\kappa(x,y)] < \infty$ shows that $Q \in \mathcal{P}_\kappa$. Now apply Theorem 3.2.3 and notice that condition (iii) is satisfied, because k is characteristic to $\mathcal{D}^1_{L^1}$ and $\mathcal{D} \subset \mathcal{D}^1_{L^1}$. Conclude with (ii). □

Corollary 3.2.4 can be viewed as a direct generalization of [21] and [68] that does not restrict Q to have a continuously differentiable Lebesgue density; this makes it especially suitable for applications with discrete sample-based approximations Q. Moreover, Theorem 1.4.5 shows that many common and not-so-common kernels are characteristic to $\mathcal{D}^1_{L^1}$: smooth translation-invariant kernels only need to be characteristic to $\mathcal{P}$ or c_0-universal.. This covers all of the most

[21] Chwialkowski et al., *A Kernel Test of Goodness of Fit*, 2016, Thm 2.2
[68] Liu et al., *A Kernelized Stein Discrepancy for Goodness-of-fit Tests*, 2016, Prop 3.3

common KSD base kernels including the Gaussian, Matérn, and inverse multiquadric radial kernels. In addition, [19] recently proved that composition kernels of the form $k_b(x, y) = k(b(x), b(y))$ inherit universality properties from the base kernel k under appropriate assumptions on the invertible transformation $b : \mathbb{R}^d \to \mathbb{R}^d$. Our next proposition, proven in Appendix C.3.3 is a mild adaptation to c_0^1-universality of this result.

[19] Chen et al., *Stein Points*, 2018, Thm 6

Proposition 3.2.5. *Suppose k is a c_0^1-universal kernel and $b : \mathbb{R}^d \to \mathbb{R}^d$ is an invertible, Lipschitz, norm-coercive (i.e., $|b(x)|_2 \to \infty$ whenever $|x|_2 \to \infty$) mapping with $\det(\partial_x b(x))$ never zero. Then the composition kernel $k_b(x, y) = k(b(x), b(y))$ is c_0^1-universal.*

c_0^1-*Universal Composition Kernels*

Finally, the following corollary of Corollary 3.2.4 provides a prototypical application of these results to determining weak convergence in the context of sample quality measurement.

Corollary 3.2.6. *Let $k(x, x') = \psi(x - x')$ be a translation-invariant kernel, with $\psi \in \mathscr{C}_b^2$. Let $P \in \mathcal{P}$ and let $(P_n)_n$ be a sequence of empirical probability measures, i.e. convex sums of Dirac measures. If k is characteristic to $\mathcal{P}$, then $P_n \to_\sigma P$ iff $(P_n)_n$ is tight and $\mathrm{KSD}_{k,P}(P_n) \to 0$.*

k *smooth, trans.-inv., charact.* $\Rightarrow \mathrm{KSD}_k$ *characteristic to* $P \in \mathcal{P}$

Proof. (of Corollary 3.2.6) Being characteristic to $\mathcal{P}$ and translation-invariant, Theorem 1.4.5 shows that k is characteristic to $\mathcal{D}_{L^1}^1$. As P_n is an empirical measure, it embeds into $\mathcal{H}_\kappa$. Applying Corollary 3.2.4 proves point (ii) of Theorem 3.2.3 which in turn proves the desired result. □

3.3 Chapter Conclusion

We have established necessary and sufficient conditions for an MMD to metrize tight weak convergence and uniformly integrable Wasserstein convergence to a fixed target of interest. Our conditions parallel those for target-independent metrization but are less stringent and, importantly, are derived for both bounded and unbounded kernels. These results are particularly well-suited for characterizing the convergence properties of kernel Stein discrepancies (KSDs), target-specific MMDs that are popular in sample quality measurement and goodness-of-fit testing and that often involve unbounded kernels. By drawing upon the theory of Schwartz-distributions, we obtain a variety of necessary and sufficient conditions for a KSD to metrize Wasserstein and weak convergence under uniform integrability. This enables us to establish the characteristicness of Stein kernels κ for new combinations of targets P, base kernels k, and approximating distributions Q.

There are many interesting avenues for future work. First, it is of great practical interest to establish weak conditions (e.g., on

a base kernel k and a target P) under which a Stein kernel κ enforces tightness. Second, we suspect that, when $k(\boldsymbol{x},\boldsymbol{x}) = O(|\boldsymbol{x}|^{2\alpha})$, the conditions of Theorem 3.1.4 can be further weakened to grant Wasserstein-α' convergence for $\alpha' < \alpha$ under α'-uniform integrability and a (non-uniform) embeddability assumption for each P_n. But for now, let us widen again the focus from MMDs to general distribution-dissimilarities and see, in Part II, how they are being used in generative models.

Part II

Neural Network Based Restricted $\mathbf{f}$-Divergences

So far, we mainly focused on MMDs, a specific type of classifier-based distribution-dissimilarity. We analyzed how the classifier's capacity – the set of test functions $\mathcal{F}$ – influences the strength of the MMD dissimilarities, and discussed some applications. The following part now widens the scope to general classifier-based distribution-dissimilarities and analyzes their properties in the context of sample generation. Given a sample from a target random variable $X \sim P_X$, the goal there is to generate new sample points $Y \sim P_Y$ that look like samples from X. Here, to "look like" means that P_Y should be close to P_X for some chosen distribution-dissimilarity D. To generate Y, we typically start by sampling a *latent variable* Z from a standard distribution P_Z (such as a multivariate Gaussian). We then squeeze Z through a parametric function G (usually a neural network) to yield a variable $Y = G(Z)$ whose distribution P_Y is the *push-forward* distribution of P_Z through G. To ensure that samples from Y look like samples from X, we then optimize the parameters of G to minimize $D(P_X \| P_Y)$. We hence (approximately) solve

Sample Generation In a Nutshell

$$\inf_{G \in \mathcal{G}} D(P_X \| P_Y)$$

where $\mathcal{G}$ is the set of functions attainable by G. If we replace $D(P_X \| P_Y)$ by the classifier-based dissimilarity (0.1), the minimization problem actually becomes a minimax problem[1]

$$\inf_{g \in \mathcal{G}} \sup_{\varphi \in \mathcal{F}} \mathop{\mathbb{E}}_{X,C} [-\mathcal{L}(\varphi(X), C)]. \tag{3.2}$$

Classifier-Based Dissimilarity Minimization: Minimax Problem

There are many questions to ask about these objectives, such as

Choice of $\mathcal{F}$ and $\mathcal{G}$? Should we use an RKHS ball, as with MMDs, where the supremum can be computed analytically on samples, as with GANs? Or should φ simply be a parametrized neural network optimized with SGD?

Choice of the training procedure? The sup and inf typically get optimized alternatively. But should we, each time, optimize until convergence, or only a few gradient steps.

General Questions with (GAN-like) Minimax Optimization

Choice of $\mathcal{L}$? When $\mathcal{L}$ is the cross-entropy loss, we essentially get a restricted Jensen-Shannon divergence [37]. More generally, by varying $\mathcal{L}$ and $\mathcal{F}$, we can get any restricted f-divergence (see Section 0.1). But are there some better than others?

[37] Goodfellow et al., *Generative Adversarial Nets*, 2014

Choice of the prior probability π? The minimax problem (3.2) implicitly depends on the prior probability π that X be sampled from P_X.[2] Most algorithms choose $\pi = .5$, but why?

[2] See marginnote 1 on p.3

Those questions are obviously deeply intertwined. It is easy to see for example that changing π amounts to changing the loss $\mathcal{L}$.

[1] Recall that (X, C) denotes a sample coming from P_X if $C = 1$ and P_Y if $C = 0$, and $-\mathcal{L}(\varphi(X), C)$ is the reward for classifying (or scoring) it as $\varphi(X)$.

But there are many more fascinating links between π and the choice of f-divergence, as discussed in [67]. Our goal here is not to answer all these questions. They were given only for context and illustrative purposes. Our approach here is rather simply to review the main generative algorithms, see how they fit into the previous dissimilarity-minimization framework, and try to fix some empirically noticed deficiencies.

[67] Liese and Vajda, *On Divergences and Informations in Statistics and Information Theory*, 2006

GAN, VAE & WAE as Dissimilarity Minimization Algos

More precisely, Chapter 4 first shows that many standard generative algorithms – GANs, VAEs, and variations – exactly fit in this framework of dissimilarity-minimization. Most of these algorithms, we will see, use restricted f-divergence dissimilarities to compare P_X and P_Y. As an alternative, we propose to use optimal transportation (OT) based measures. They lead to new generative models called Wasserstein Auto-Encoders (WAEs) and give new insights on the relation between GAN- and VAE-like algorithms. Chapter 4 hence focuses on training objectives, and leaves aside the problem of actual training. Now, the training of GANs turns out to be especially unstable. Among other issues, it is indeed not rare that, after some training, the generator suddenly focuses on one image-type only, which it mimics quite accurately, but it completely fails to capture the overall diversity of images contained in P_X. This problem, known as *mode collapse*, will be the focus of Chapter 5. We will not try to understand the origin of the problem, but rather remedy it by combining several GANs (or other generative models), focusing each on different distribution modes in P_X. This leads to an algorithm similar in spirit to AdaBoost, which is why we called it *AdaGAN*. Overall, this second part really focuses on the generative procedure as a whole, rather than specifically on the dissimilarities used. The next and last part, Part III will then focus again on distribution-dissimilarities only, and specifically on those typically used in generative algorithms: network-based dissimilarities.

AdaGAN to Fix Mode-Collapse

4

Generative Models & Dissimilarity Minimization

Contrary to the previous chapters which focused on MMDs only, we now get back to distribution-dissimilarities in general, and concentrate on their currently most trendy application: generative models. Traditionally, those models are not introduced via distribution-dissimilarities: GAN papers usually highlight the adversarial game perspective, while VAEs generally insist more on the empirical likelihood and variational inference aspects. But viewing GANs as a minimization problem of a distribution-dissimilarity recently lead to powerful new algorithms, such as Wasserstein- and f-GANs [4, 82]. Following this path, Section 4.1 describes a set of popular, recent generative models and how they relate to distribution-dissimilarity minimization, and in particular to f-divergence minimization. Section 4.2 then proposes to minimize optimal transport (OT) distances, which will lead to a new algorithm now known as Wasserstein Auto-Encoders (WAEs). The accompanying analysis unveils further links between the previous generative algorithms, which are discussed in Section 4.3. In particular, for the squared Euclidian transportation cost c, WAE is a so-called Adversarial Auto-Encoders (AAE), an algorithm proposed by [71] without strong theoretical justification. Section 4.4 concludes with possible future work and related literature.

[4] Arjovsky et al., *Wasserstein GAN*, 2017; [82] Nowozin et al., *f-GAN*, 2016

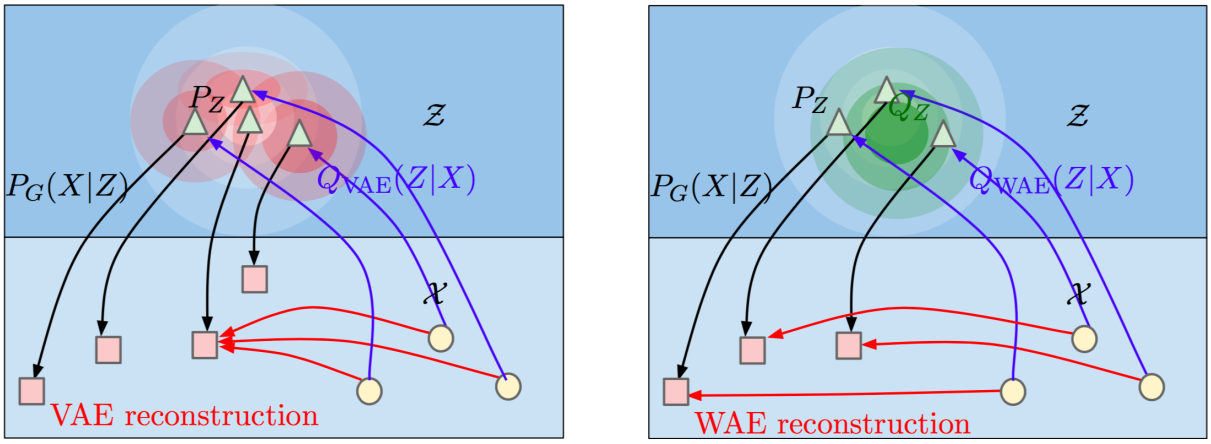
[71] Makhzani et al., *Adversarial Autoencoders*, 2016

4.1 Generative Models and Dissimilarity Minimization

The goal of generative modeling is to sample points Y from a model with distribution P_Y that look like samples from a target random variable $X \sim P_X$, where $X, Y \in \mathcal{X}$. Almost all such generative models rely on *latent variable models*, which means that Y is generated in two steps: first we sample a latent variable or *latent code* $Z \in \mathcal{Z}$ from a known, arbitrary distribution P_Z (usually a multidimensional Gaussian); then we squeeze it through a (possibly random) function $G : \mathcal{Z} \to \mathcal{X}$ to get Y, i.e. $Y = G(Z)$. Thus, if Y has a density p_Y, it can be decomposed as:

Latent Generative Models

$$p_Y(y) := \int_{\mathcal{Z}} p_Y(y|z)\, dP_Z(z), \quad \forall y \in \mathcal{X}\,. \tag{4.1}$$



(a) VAE and AVB

(b) WAE and AAE

Figure 4.1: Both VAE and WAE consist of an encoder $Q_{Z|X}$ and a decoder $G_{X|Z}$ which map inputs $X \in \mathcal{X}$ to latent codes $Z \in \mathcal{Z}$ and vice-versa. Both minimize two terms: the reconstruction cost, and a regularizer penalizing differences between encoded distributions and a predefined latent distribution P_Z. But while VAE forces each conditional distribution $Q_{Z|X=x}$ (the red balls) to match the same distribution P_Z for all x, WAE only constrains the marginal $Q_Z = \int Q_{Z|X=x}\, dP_X(x)$ (the green distribution) to match P_Z. Contrary to WAE, VAE hence promotes overlap of different conditionals $Q_{Z|X}$, which can lead to blurry reconstructions.

Those models have two major advantages: they are easy to sample from and P_Y can be optimized using SGD as soon as G can be differentiated analytically wrt its parameters. The field of generative modeling is growing rapidly. Here we introduce and compare several new algorithms – essentially GAN and VAE variations –, and show how they fit in the distribution-dissimilarity minimization procedure described in the introduction of Part II.

Generative adversarial nets (GAN). The original GAN approach [37] minimizes

$$D_{\text{GAN}}(P_X \,\|\, P_Y) = \sup_{T \in \mathcal{F}} \mathop{\mathbb{E}}_{X \sim P_X} [\log T(X)] + \mathop{\mathbb{E}}_{Z \sim P_Z} \big[\log\big(1 - T(G(Z))\big)\big] \quad (4.2)$$

wrt a deterministic *generator* $G\colon \mathcal{Z} \to \mathcal{X}$, where $\mathcal{F}$ is a predefined class of test functions. Each function $T \in \mathcal{F}$ is usually seen as a *discriminator* which discriminates between true points $X \sim P_X$ and fake points $G(Z) \sim P_Y$. The original GAN authors already showed that $D_{\text{GAN}}(P_X \,\|\, P_Y) \leqslant 2 \cdot D_{\text{JS}}(P_X \,\|\, P_Y) - \log(4)$ with an equality in the *nonparametric limit*, i.e. when the class $\mathcal{F}$ becomes rich enough to represent *all* functions mapping $\mathcal{X}$ to $(0, 1)$. Hence GANs minimize a restricted JS-divergence, a *lower bound* on the JS-divergence. In practice however, both generator G and discriminator T are trained with alternating SGD steps on finite samples from P_X and P_Z. Stopping criteria as well as adequate evaluation of the trained GAN models remain open questions.

[37] Goodfellow et al., *Generative Adversarial Nets*, 2014

f-GANs. Instead of using a restricted JS-divergence, Nowozin et al. [82] showed that one can use any restricted f-divergence simply by replacing (4.2) by a new objective:

$$D_{\text{f,GAN}}(P_X \,\|\, P_Y) = \sup_{T \in \mathcal{F}} \mathop{\mathbb{E}}_{X \sim P_X} [T(X)] + \mathop{\mathbb{E}}_{Z \sim P_Z} [f^*(G(Z)]\,,$$

[82] Nowozin et al., *f-GAN*, 2016

where f^* is the Fenchel conjugate of f (see Section 0.1). They showed that the original GAN simply corresponds to a specific function f.

Wasserstein GAN (WGAN). A year later, Arjovsky et al. [4] advocated to focus on the 1-Wasserstein distance, meaning a "restricted Total Variation" where all test functions $\mathcal{W}$ are constrained to be 1-Lipschitz (see Section 0.1):

[4] Arjovsky et al., *Wasserstein GAN*, 2017

$$D_{\text{WGAN}}(P_X \,\|\, P_Y) = \sup_{T \in \mathcal{W}} \mathop{\mathbb{E}}_{X \sim P_X}[T(X)] - \mathop{\mathbb{E}}_{Z \sim P_Z}[T(G(Z))].$$

{∅, f, W, MMD}-GANs: Generator-Discrim. Architecture Minimize Restr. f-Divergence

The 1-Lipschitz constraint bounds the gradients wrt G, which greatly stabilizes the training.

MMD-GANs and moment matching networks. Instead of optimizing a network-based discriminator T to compute a restricted f-divergence, [27, 66] independently proposed to use an MMD. That way, computing $D_{\text{MMD}}(P_X \,\|\, P_Y) = \text{MMD}_k(P_X, P_Y)$ on samples needs no optimization. Except of course if the kernel k itself has parameters which need optimization. The kernel k then spans a family $\mathcal{K}$ typically given by $k(x, y) = k_0(h(x), h(y))$ where k_0 is some fixed predefined kernel and h is a parametrized neural network. And the dissimilarity becomes a supremum over a family of MMDs (see [11, 64] and references therein):

[66] Li et al., *Generative Moment Matching Networks*, 2015; [27] Dziugaite et al., *Training Generative Neural Networks via MMD Optimization*, 2015

[64] Li et al., *MMD GAN*, 2017; [11] Bińkowski et al., *Demystifying MMD GANs*, 2018

$$D_{\text{MMDs}}(P_X \,\|\, P_Y) = \sup_{k \in \mathcal{K}} \text{MMD}_k(P_X, P_Y).$$

Variational auto-encoder (VAE). Kingma and Welling [56] introduced VAEs as a generative model trained by minimizing the negative log-likelihood $\mathbb{E}_{P_X}[\log p_Y(X)]$. This being equivalent to KL minimization, VAEs are indeed approximate f-divergence minimization algorithms, too. But their main distinctive feature is to decompose this negative log-likelihood into two parts, yielding the objective

[56] Kingma and Welling, *Auto-Encoding Variational Bayes*, 2014

$$D_{\text{VAE}}(P_X \,\|\, P_Y) = \inf_{Q(Z|X) \in \mathcal{Q}} \mathop{\mathbb{E}}_{P_X}\left[D_{\text{KL}}(Q(Z|X) \,\|\, P_Z) - \mathop{\mathbb{E}}_{Q(Z|X)}[\log p_Y(X|Z)] \right]. \quad (4.3)$$

VAE, AVB: Encoder-Decoder Architecture Minimize Neg. Log-Lik., i.e. KL

If we think of Z as being a latent code for X, then $Q(Z|X)$ and $p_Y(X|Z)$ can be seen respectively as a random *encoder* and *decoder* (rather than *generator*) of X. $P_Y(X|Z)$ is usually a function G of Z parametrized by a deep net, but with some randomness "added on top". A typical choice is to use Gaussians $P_Y(X|Z) = \mathcal{N}(X;\, G(Z),\, \sigma^2 \cdot I)$. Now, if $\mathcal{Q}$ is the set of *all* conditional probability distributions $Q(Z|X)$, this VAE objective indeed coincides with the negative marginal log-likelihood $-\mathbb{E}_{P_X}[\log P_Y(X)]$. However, in order to make the D_{KL} term of (4.3) tractable in closed form, the original implementation of VAE uses

a standard normal P_Z and restricts $\mathcal{Q}$ to a class of Gaussian distributions $Q(Z|X) = \mathcal{N}(Z; \mu(X), \Sigma(X))$ with mean μ and diagonal covariance Σ parametrized by deep nets. VAE hence minimizes an *upper bound* on the negative log-likelihood or, equivalently, on the KL-divergence $D_{KL}(P_X \| P_Y)$.

Adversarial variational Bayes (AVB). Mescheder et al. [74] also start from (4.3). But they avoid the restriction of $\mathcal{Q}$ to Gaussians, which loosens the bound between (4.3) and the actual negative log-likelihood. Instead, they notice that samples from $Q(Z|X = x)$ can be written as samples from a random variable $e(x, \epsilon)$ where ϵ follows a standard normal and $e : \mathcal{X} \times \mathbb{R} \to \mathcal{Z}$ is some appropriate function. They then parametrize e by a neural network, replace the intractable term $D_{KL}(Q_e(Z|X) \| P_Z)$ in (4.3) by the adversarial approximation $D_{f,GAN}$ corresponding to the KL-divergence, and essentially minimize

[74] Mescheder et al., *Adversarial Variational Bayes*, 2017

$$D_{AVB}(P_X \| P_Y) = \inf_{Q_e(Z|X) \in \mathcal{Q}} \mathop{\mathbb{E}}_{X \sim P_X} \left[\begin{array}{l} D_{f,GAN}(Q_e(Z|X) \| P_Z) \\ -\mathbb{E}_{Z \sim Q_e(Z|X)}[\log p_Y(X|Z)] \end{array} \right]. \quad (4.4)$$

Adversarial auto-encoders (AAE). The D_{KL} term in (4.3) can be viewed as a mere regularizer: when dropped, VAE actually reduces to the classical (unregularized) auto-encoder. This regularizer happens to be crucial for sample generation [7]. But weirdly enough, it pushes each encoded variable to follow the same distribution P_Z, whatever the original X value was. Instead, Makhzani et al. [71] replace the D_{KL} term in (4.3) by a regularizer that enforces a constraint on the marginal – rather than the conditional – distribution of the X-encodings:

[7] Bengio et al., *Representation Learning: A Review and New Perspectives*, 2013

[71] Makhzani et al., *Adversarial Autoencoders*, 2016

$$D_{AAE}(P_X \| P_Y) = \inf_{Q(Z|X) \in \mathcal{Q}} D_{GAN}(Q_Z \| P_Z) - \mathop{\mathbb{E}}_{P_X} \mathop{\mathbb{E}}_{Q(Z|X)} [\log p_Y(X|Z)]. \quad (4.5)$$

Q_Z is called the *aggregated posterior*. It is the marginal distribution of Z when X is sampled from P_X and then Z is sampled from $Q(Z|X)$. Similarly to AVB, there is no clear link to log-likelihood. The authors report an equally good performance for different types of conditional distributions $Q(Z|X)$, including Gaussians as in VAEs, implicit models Q_e as in AVB, and *deterministic* encoder mappings.

AAE: Encoder-Decoder Arch. Justification of D_{AAE}?

Summarizing the progress so far, we interpreted GANs, f-GANs, MMD-GANs, VAEs and AVB as approximate f-divergence minimizations, but not AAE. Indeed, we derived AAE from VAE, but gave no interpretation in terms of dissimilarity minimization. To do so, we now take a step back and show how to use OT based dissimilari-

ties instead of f-divergences. They yield an algorithm that we call Wasserstein Auto-Encoders (WAEs), of which AAE is a special case.

4.2 From Optimal Transport to WAE

Our introduction, Section 0.1, introduced OT dissimilarities from the *dual* perspective. The original, *primal* perspective defines the OT dissimilarity D_c between measures P_X and P_Y as

$$D_c(P_X \| P_Y) := \inf_{\gamma \in \Gamma(P_X, P_Y)} \mathop{\mathbb{E}}_{(X,Y)\sim\gamma} [c(X,Y)], \tag{4.6}$$

where $\Gamma(P_X, P_Y)$ denotes the set of *couplings* between P_X and P_Y, i.e. the set of joint distributions whose marginals are P_X and P_Y. The non-negative number $c(x,y)$ is usually understood as the cost for transporting a (Dirac) unit mass from x to y. Then $D(P_X \| P_Y)$ is the minimal cost to redistribute a unit mass with distribution P_X to a distribution P_Y. Kantorovich proved the equivalence between the primal and dual perspectives. When $c = d^\alpha$ where d is a distance on $\mathcal{X}$ and $1 \leqslant \alpha < \infty$, then $D_c^{1/\alpha}$ is the (d-based) α-Wasserstein distance (see Section 3.1.1). WGANs rely on the dual formulation when $c = d$. Instead, we now show how to approximately solve the primal formulation (for a general cost c). To do so, we first reparametrize the space of couplings Γ (Section 4.2.1) and then relax the constraints on the coupling's margins (Section 4.2.2).

OT: Primal, Dual & Mass Redistribution at Minimal Cost

4.2.1 Reparametrization of the Couplings

Remember that Y is obtained via a latent generative model from $\mathcal{Z}$ to $\mathcal{Y} = \mathcal{X}$. Our goal here is to make this model appear in the OT equation (4.6). To do so, let us for now assume that this model is deterministic, i.e. that $Y = G(Z)$ for some function G. We will relax this assumption in Section 4.2.3. Then $\mathbb{E}_{(X,Y)}[c(X,Y)] = \mathbb{E}_{(X,Z)}[c(X, G(Z))]$, which shows that:

Deterministic Case: $Y = G(Z)$

$$\inf_{P_{X,Y} \in \Gamma(P_X, P_Y)} \mathop{\mathbb{E}}_{P_{X,Y}} [c(X,Y)] = \inf_{P_{X,Z} \in \Gamma(P_X, P_Z)} \mathop{\mathbb{E}}_{P_{X,Z}} [c(X, G(Z))] \tag{4.7}$$

From OT btw. P_X & P_Y to OT btw. P_X & P_Z

Said differently, we just transformed the OT problem between P_X and P_Y with cost c into an OT problem between P_X and P_Z with cost $c_G(x,z) := c(x, G(z))$: $D_c(P_X \| P_Y) = D_{c_G}(P_X \| P_Z)$. Next, notice that $\mathbb{E}_{P_{X,Z}}[c(X, G(Z))] = \mathbb{E}_{P_X} \mathbb{E}_{P_{Z|X}}[c(X, G(Z))]$. P_X being a fixed distribution, the optimization over couplings $P_{X,Y}$ can thus be turned into an optimization over the set of conditional distributions $Q_{Z|X}$ whose marginal distribution of Z is P_Z. More formally:

Theorem 4.2.1. *If* $Y = G(Z)$ *for some function* $G : \mathcal{Z} \to \mathcal{X}$*, then*

$$D_c(P_X \,\|\, P_Y) = \inf_{P_{X,Z} \in \Gamma(P_X, P_Z)} \mathbb{E}_{P_{X,Z}}\big[c\big(X, G(Z)\big)\big] \tag{4.8}$$

$$= \inf_{Q_{Z|X}\colon Q_Z = P_Z} \mathbb{E}_{P_X} \mathbb{E}_{Q_{Z|X}}\big[c\big(X, G(Z)\big)\big], \tag{4.9}$$

where Q_Z *is the marginal distribution of* Z *when* $X \sim P_X$ *and* $Z \sim Q_{Z|X}$.

From Optimal Couplings to Optimal Conditionals

4.2.2 Relaxing the Constraints on the Coupling's Margins

If we think of $Q_{Z|X}$ as being a random encoder of X, then (4.9) says that we can compute the OT cost from P_X to P_Y by optimizing over the set of encoders whose aggregated posterior Q_Z is P_Z. But the optimization under the latter constraint may seem even more complicated than an optimization over couplings (as in Eq. 4.8). That is why, we propose to relax this hard constraint using the following classical regularization technique. Namely, choose $\lambda > 0$ and a convex *penalty* $F : \mathcal{P} \to \mathbb{R}_+$ such that $F(Q) = 0$ if and only if $Q = P_Z$, and replace the constrained optimization of $D_c(P_X \,\|\, P_Y)$ by the following relaxed version:

Hard Constraint on $Q_{Z|X}$*:* $Q_Z = P_Z$

$$D_c^\lambda(P_X \,\|\, P_Y) := \inf_{Q_{Z|X}} \mathbb{E}_{P_X} \mathbb{E}_{Q_{Z|X}}\big[c\big(X, G(Z)\big)\big] + \lambda F(Q_Z), \tag{4.10}$$

It is well known [13] that under mild conditions adding a penalty as in (4.10) is equivalent to adding a constraint of the form $F(Q) \leqslant \mu_\lambda$ for some $\mu_\lambda > 0$. As λ increases, the corresponding μ_λ decreases, and as $\lambda \to \infty$, the solutions of (4.10) reach the feasible region where $P_Z = Q_Z$. Hence $D_c^\lambda(P_X \,\|\, P_Y) \leqslant D_c(P_X \,\|\, P_Y)$ for all $\lambda \geqslant 0$ and the gap reduces with increasing λ.

[13] Borwein and Lewis, *Convex Analysis and Nonlinear Optimization*, 2006

We may now be tempted to choose a KL-, JS- or any other f-divergence as a regularizer $F(Q_Z) = D_f(Q_Z \,\|\, P_Z)$. But this would result in an intractable penalty F. Instead, we propose two alternatives: either use an adversarial approximation $F(Q_Z) = D_{f,\mathrm{GAN}}(Q_Z \,\|\, P_Z)$; or, following [116], use the MMD of a characteristic kernel k, i.e. $F(Q_Z) = \mathrm{MMD}_k(Q_Z, P_Z)$. We hence end up with the following objective, which we originally called the *penalized optimal transport* or POT objective, and was later re-baptized as the *Wasserstein Auto-Encoder* (WAE) objective:

Soft Constraint on $Q_{Z|X}$*:* $D(Q_Z \,\|\, P_Z) \leqslant \mathrm{Cst}$

[116] Tolstikhin et al., *WAE*, 2018

$$D_{\mathrm{WAE}}(P_X \,\|\, P_Y) := \inf_{Q_{Z|X} \in \mathcal{Q}} \mathbb{E}_{P_X} \mathbb{E}_{Q_{Z|X}}\big[c\big(X, G(Z)\big)\big] + \lambda D(Q_Z \,\|\, P_Z) \tag{4.11}$$

WAE-GAN: $D = D_{\mathrm{GAN}}$
WAE-MMD: $D = \mathrm{MMD}$

where D is D_{GAN} or MMD_k, and $\mathcal{Q}$ a set of conditionals parametrized by an (encoder) network. When the cost function c is differentiable,

Algorithm 4.1: *Wasserstein Auto-Encoder with GAN-based penalty (WAE-GAN).*

Require: Regularization coefficient $\lambda > 0$.
Initialize the parameters of the encoder Q_ϕ, decoder G_θ, and latent discriminator D_γ.
while (ϕ, θ) not converged **do**
Sample $\{x_1, \dots, x_n\}$ from the training set
Sample $\{z_1, \dots, z_n\}$ from the prior P_Z
Sample $\tilde{z}_i$ from $Q_\phi(Z|x_i)$ for $i = 1, \dots, n$
Update D_γ by ascending:

$$\frac{\lambda}{n} \sum_{i=1}^{n} \log D_\gamma(z_i) + \log\big(1 - D_\gamma(\tilde{z}_i)\big)$$

Update Q_ϕ and G_θ by descending:

$$\frac{1}{n} \sum_{i=1}^{n} c\big(x_i, G_\theta(\tilde{z}_i)\big) - \lambda \cdot \log D_\gamma(\tilde{z}_i)$$

end while

Algorithm 4.2: *Wasserstein Auto-Encoder with MMD-based penalty (WAE-MMD).*

Require: Regularization coefficient $\lambda > 0$, characteristic positive-definite kernel k.
Initialize the parameters of the encoder Q_ϕ, decoder G_θ, and latent discriminator D_γ.
while (ϕ, θ) not converged **do**
Sample $\{x_1, \dots, x_n\}$ from the training set
Sample $\{z_1, \dots, z_n\}$ from the prior P_Z
Sample $\tilde{z}_i$ from $Q_\phi(Z|x_i)$ for $i = 1, \dots, n$
Update Q_ϕ and G_θ by descending:

$$\frac{1}{n} \sum_{i=1}^{n} c\big(x_i, G_\theta(\tilde{z}_i)\big) + \frac{\lambda}{n(n-1)} \sum_{\ell \neq j} k(z_\ell, z_j) + \frac{\lambda}{n(n-1)} \sum_{\ell \neq j} k(\tilde{z}_\ell, \tilde{z}_j) - \frac{2\lambda}{n^2} \sum_{\ell, j} k(z_\ell, \tilde{z}_j)$$

end while

this objective can be trained with SGD using Algorithm 1 or 2 (borrowed from [116]) for the f-GAN and MMD alternatives respectively.

[116] Tolstikhin et al., *WAE*, 2018

4.2.3 Random Generators $P_{Y|Z}$

What happens when Y is not given by a deterministic, but a random generator? I.e. not by a function $Y = G(Z)$, but a set of conditional distributions $(P_{Y|Z=z})_{z \in \mathcal{Z}}$, such as in VAE where $Y \sim \mathcal{N}\big(G(Z), \sigma^2\big)$? Then Equation (4.8) does not hold anymore in general [15]. But if instead of optimizing over all $\Gamma(P_X, P_Y)$, we optimized only over the couplings that verify $(Y \perp\!\!\!\perp X) \,|\, Z$, then we can recover a theorem much like Theorem 4.2.1. More formally:

[15] Bousquet et al., *From optimal transport to generative modeling*, 2017, Prop 2

Theorem 4.2.2. *Let*

$$D_c^\dagger(P_X \,\|\, P_Y) := \inf_{P_{X,Y} \in \Gamma^\dagger(P_X, P_Y)} \mathbb{E}_{P_{X,Y}}[c(X,Y)] \quad \textit{where}$$

$$\Gamma^\dagger(P_X, P_Y) := \{P_{X,Y} \in \Gamma(P_X, P_Y) \,:\, (Y \perp\!\!\!\perp X) \,|\, Z\}. \quad \textit{Then}$$

$$D_c(P_X \,\|\, P_Y) \leqslant D_c^\dagger(P_X \,\|\, P_Y) = \inf_{Q_{Z|X} \,:\, Q_Z = P_Z} \mathbb{E}_{P_X} \mathbb{E}_{Q_{Z|X}} \mathbb{E}_{P_{Y|Z}} \big[c(X,Y)\big].$$

Random Decoders: OT-Cost from P_X to P_Y $\leqslant$ OT-Cost from P_X to P_Y "via P_Z"

Proof. The inequality stems from the inclusion $\Gamma^\dagger(P_X, P_Y) \subset \Gamma(P_X, P_Y)$. The equality immediately follows from the tower rule. □

Intuitively, $\Gamma^\dagger(P_X, P_Y)$ is exactly the set of couplings that can be obtained with an encoder-decoder structure with latent space variable Z. We are now ready to compare the OT and WAE objectives with the algorithms presented in Section 4.1.

4.3 WAE and VAE-style Algorithms Compared

This section compares the OT and WAE objectives with the GAN and VAE style algorithms described earlier, first in theory (Section 4.3.1), then in practice (Section 4.3.2).

4.3.1 Theoretical Comparison

WAE and WGAN. Although GANs and WAE end up having two conceptually quite different structures – a generator-discriminator as opposed to an encoder-decoder –, both algorithms are obviously linked via WGANs. Indeed, WGANs approximate the Wasserstein-1 distance $W_1(P_X, P_Y)$ as does WAE when choosing a non-squared transportation (i.e. reconstruction) cost $c(x, y) = \|x - y\|_2$. However, WAE uses the primal OT formulation, while WGAN uses the dual one. The latter formulation happens to be especially handy with this non-squared ℓ_2-cost. It is not clear how to generalize this dual approach (WGAN) for other costs, whereas WAE does not assume any particular cost function. Other differences between WGAN and WAE are discussed in [15].

WAE Relies on Primal
WGAN Relies on Dual

[15] Bousquet et al., *From optimal transport to generative modeling*, 2017

Gaussian decoder and relation to VAE and AVB. Following the examples of VAE, AVB and AAE, let us focus on Gaussian decoders $Y \sim \mathcal{N}((, G)(Z), \sigma^2 \cdot I_d)$, whose distribution we denote P_Y^σ, with a quadratic reconstruction cost $c(x, y) = \|x - y\|_2^2$ on $\mathcal{X} = \mathbb{R}^d$. Then D_c is the squared Wasserstein-2 distance, $D_c = W_2^2$, and Theorem 4.2.2 shows that

$$
\begin{aligned}
&W_2^2(P_X, P_Y^\sigma) \leqslant \\
&\quad D_c^\dagger(P_X \,\|\, P_Y^\sigma) = d \cdot \sigma^2 + \inf_{Q_{Z|X} : Q_Z = P_Z} \mathop{\mathbb{E}}_{P_X} \mathop{\mathbb{E}}_{Q_{Z|X}} \left[\|X - G(Z)\|_2^2 \right] \\
&\qquad\qquad = d \cdot \sigma^2 + D_c^\dagger \left(P_X \,\middle\|\, P_Y^0 \right) = d \cdot \sigma^2 + W_2^2(P_X, P_Y^0),
\end{aligned}
$$

Argmin of R.H.S.
$\perp\!\!\!\perp$ *Decoder-Noise* σ

where the last equality stems from Theorem 4.2.1. Hence, the minimization of the upper-bound $D_c^\dagger \left(P_X \,\middle\|\, P_Y^\sigma \right)$ wrt the decoder G is independent of the decoder's noise-level[1] σ, and coincides with the noiseless minimization of W_2. This is good news, because in practice we always generate new samples Y as $Y = G(Z)$, i.e. without adding Gaussian noise. Having the optimization of $D_c^\dagger \left(P_X \,\middle\|\, P_Y^\sigma \right)$ be independent of σ thus avoids having a mismatch between inference

[1] contrary to the minimization of $W_2^2(P_X, P_Y^\sigma)$, which usually depends on σ. See [15, Prop 2].

and training. This contrasts with the VAE and AVB losses, where σ acts as a balancing factor between the reconstruction error and the latent KL-divergence, and which get ill-defined when we are most interested in: when $\sigma = 0$.

Relation to AAE. Substituting $\log p_Y(x|z)$ by its analytical form in (4.5), we see that D_{AAE} *coincides* with the GAN-version of D_{WAE} – up to additive terms independent of the encoder $Q(Z|X)$ and decoder G – when the regularization coefficient λ is set to $2\sigma^2$. For $0 < \sigma^2 < \infty$ this means that *AAE is minimizing the penalized relaxation* D_{WAE} *of the constrained optimization problem corresponding to* $D_c^\dagger\left(P_X \,\|\, P_Y^\sigma\right)$. The gap between D_{WAE} and $D_c^\dagger$ depends on the choice of λ, i.e. on σ^2. When $\lambda = 2\sigma^2 \to 0$, the upper bound $D_c^\dagger$ converges to the OT cost D_c, but its relaxation D_{WAE} gets looser. AAE then approaches the classical unregularized auto-encoder, and has less and less connection to OT. When $\lambda = 2\sigma^2 \to \infty$, the solution of D_{WAE} gets closer and closer to the hard-constrained objective (4.9). Hence the solution of AAE then converges more and more to the solution of $D_x^\dagger\left(P_X \,\|\, P_Y^\sigma\right)$, which, we know, is also the solution of $D_c\left(P_X \,\|\, P_Y^0\right)$. Said differently, at the limit where $\sigma^2 \to \infty$, AAE minimizes the Wasserstein-2 distance between P_X and $G(Z)$. Interestingly, the authors of [74] only connected AAE to log-likelihood maximization (VAE and AVB). They argued that AAE is "a crude approximation" to AVB. Our results instead suggest that AAE is actually attempting to minimize the 2-Wasserstein distance between P_X and P_Y^σ, which may explain its good empirical performance reported in [71].

AAE = WAE-GAN with $c(x,y) = \|x-y\|_2^2$, $\lambda = 2\sigma^2$, $Y \sim \mathcal{N}\big(G(Z),\, \sigma^2 \cdot I_d\big)$

[74] Mescheder et al., *Adversarial Variational Bayes*, 2017

[71] Makhzani et al., *Adversarial Autoencoders*, 2016

Blurriness of VAE and AVB. Compared to GANs, VAE are much easier to train, but their samples are known to be blurrier. We think this blurriness has two origins. The first one comes from the fact that the "$\log p_Y(Y|X)$" part of VAE tries to minimize the ℓ_2 distance between an image X and its reconstruction Y. From ℓ_2-regression, we know that the solution should thus be a (conditional) average over different images Y, and hence blurry. The second origin of this blurriness comes from the KL-divergence term, which enforces the conditional distribution of $Z|X$ to match a distribution which is independent of Z. See Figure 4.1. If $Q(Z|X)$ could be anything (the non-parametric limit), the encoder could still ensure that the distributions $Q(Z|X=x)$ do not overlap for different values of x. But VAE specifically restricts $Q(Z|X)$ to be Gaussians. Hence all of them overlap, meaning that different values x_1 and x_2 of X can be mapped to a same code Z. To minimize the reconstruction error, the decoder $G(Z)$ will hence yield some average between x_1, x_2 and the other possible values of X: $G(Z)$ will be blurry. This contrasts with AAE and WAE, which, by imposing a constraint on the margin Q_Z rather

VAE Blurry, because: ℓ_2*-Reconstruction = Cond. Avg. & Overlap of Conditionals* $\{Z|x\}_x$

than the conditional $Q(Z|X)$, do not promote any overlap between the conditionals. See Figure 4.1 for an illustration.

4.3.2 Experimental Comparison

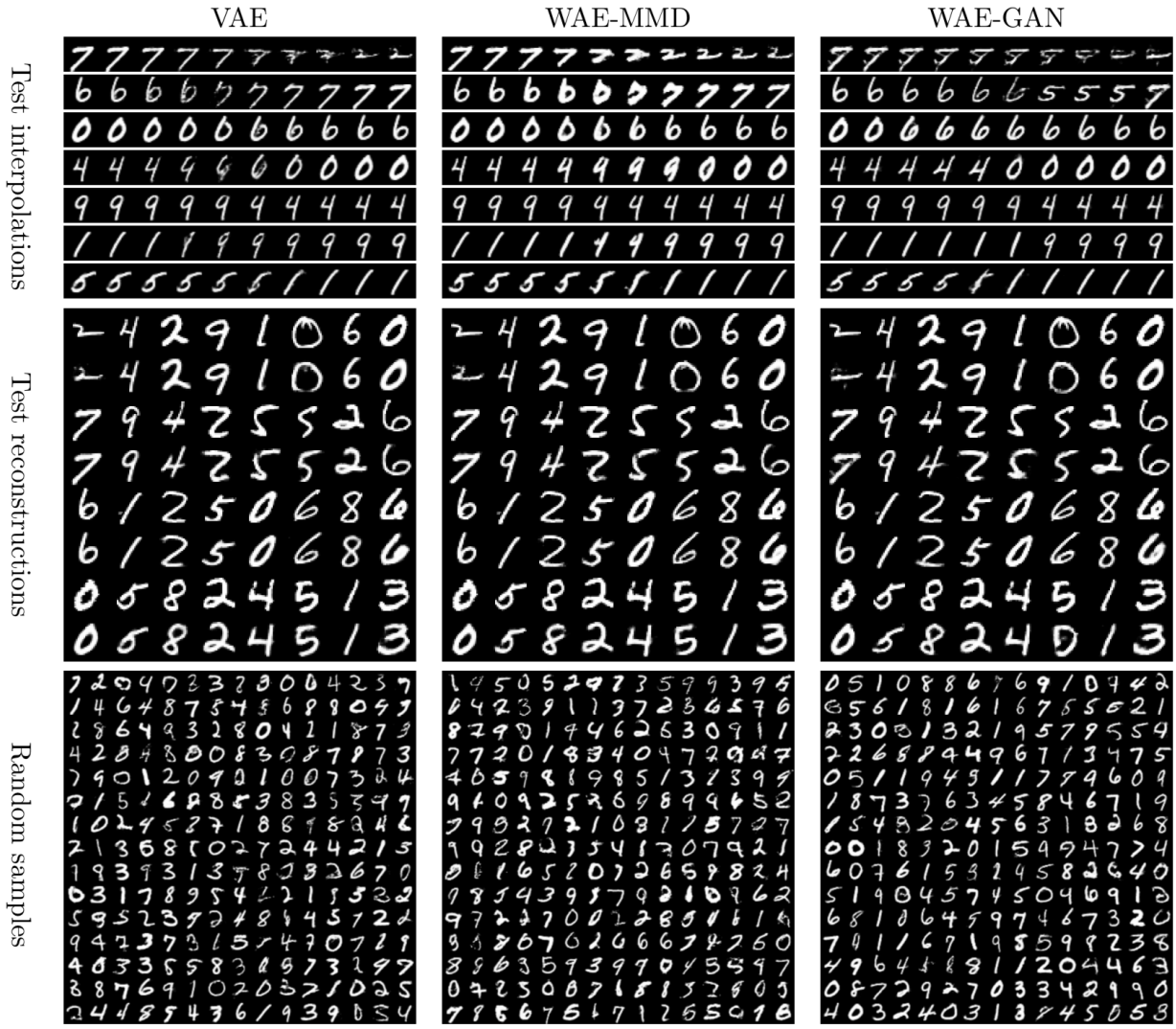


Figure 4.2: VAE (left column), WAE-MMD (middle column), and WAE-GAN (right column) trained on MNIST dataset. In "test reconstructions" odd rows correspond to the real test points.

In a follow-up work, Tolstikhin et al. [116] compared VAE with the WAE-MMD and WAE-GAN algorithms on two well-known datasets: MNIST [62] and CelebA [69], containing respectively a set of handwritten-digits and celebrity faces. They used a deterministic encoder-decoder (Q-G) pair with DCGAN-like architectures [86]. Their results are plotted in Figures 4.2 and 4.3. They feature images $G(Z)$ for linear interpolations values of Z between two encoded images; test reconstructions $G(Q(X))$; and random samples $G(Z)$ with Z sampled from the latent target $P_Z = \mathcal{N}((,0), \sigma \cdot I_{d_Z})$, where $d_Z = 8$

[116] Tolstikhin et al., *WAE*, 2018
[62] LeCun et al., *Gradient-Based Learning Applied to Document Recognition*, 1998
[69] Liu et al., *Deep Learning Face Attributes in the Wild*, 2015
[86] Radford et al., *Unsupervised Representation Learning with Deep Convolutional GANs*, 2016

Figure 4.3: VAE (left column), WAE-MMD (middle column), and WAE-GAN (right column) trained on CelebA dataset. In "test reconstructions" odd rows correspond to the real test points.

for MNIST and $d_{\mathcal{Z}} = 64$ for CelebA. As expected, the WAE-GAN turned out more unstable to train than WAE-MMD, because of the unstable GAN training. But interestingly, they observed that in WAE-MMD, Gaussian kernels failed to adequately penalize encoded outliers in the latent space $\mathcal{Z}$. The authors incriminate the Gaussian's quick decay, and therefore resorted to multi-quadratic kernels, which are characteristic and have very heavy tails. For further details, see the original paper.

4.4 Chapter Conclusion

This chapter presented several recent generative algorithms and showed that they all approximately do f-divergence minimization. As an alternative, we proposed WAE, an algorithm that approximately minimizes an optimal transport problem. It relies on an

encoder-decoder approach similar to VAE-style algorithms, with a reconstruction (or transportation) cost $c(x, y)$, and a constraint on the encoded distribution, which we enforce using a GAN- or MMD-based dissimilarity. When taking $c(x, y) = \|x - y\|_2$, WAE approximates the same objective than WGAN, but from the primal rather than dual perspective; when $c(x, y) = \|x - y\|_2^2$, WAE essentially coincides with AAE, which was originally derived as a VAE-variation. WAE hence links the VAE and GAN approaches to sample generation. So far however, we barely mentioned the problems one may encounter during actual training. Let us discuss an ominous one now: mode-collapse.

5

AdaGAN: Boosting Generative Models

Imagine we have a large corpus, containing unlabeled pictures of animals, and our task is to build a generative probabilistic model of the data. We run a recently proposed algorithm and end up with a model which produces impressive pictures of cats and dogs, but not a single giraffe. A natural way to fix this is to manually remove all cats and dogs from the training set and run the algorithm on the updated corpus. The algorithm has no choice but to produce new animals. By iterating this process until there's only giraffes left in the training set, we hence end up with a model generating giraffes (assuming sufficient sample size). At the end, we aggregate the models obtained by building a mixture model. Unfortunately, the described meta-algorithm requires manual work for removing certain pictures from the *unlabeled* training set at every iteration.

Let us turn this into an automatic approach. Rather than including or excluding a picture, we continuously weight their appearance probability. To do so, we train a binary classifier to separate "true" pictures of the original corpus from the set of "synthetic" pictures generated by the mixture of *all the models* trained so far. The classifier should then make confident predictions for the true pictures of animals missed by the model (giraffes), because there are no synthetic pictures nearby to be confused by. By a similar argument, the classifier should make less confident predictions for the true pictures containing animals already generated by one of the trained models (cats and dogs). For each picture in the corpus, we can thus use the classifier's confidence to re-weight the picture's probability to appear in next iteration's dataset.

The present chapter provides a principled way to perform this re-weighting, with theoretical guarantees showing that the resulting mixture models indeed approach the true data distribution.[1] Our algorithm, called *AdaGAN*, for Adaptive GAN, originally aimed to solve a well-known GAN-training issue known as *mode-collapse* or *missing-mode*, where the generator suddenly converges to only one or a few data modes – our cats and dogs – and ignores the overall data diversity of the training set – the remaining giraffes. But AdaGAN can actually be used with any generative model: Gaussian mixture models, VAEs, WGANs, or even unrolled [75] or mode-regularized

[1] Note that the term "mixture" should not be interpreted to imply that each component models only one mode: the models to be combined into a mixture can themselves cover multiple modes.

[75] Metz et al., *Unrolled GANs*, 2017

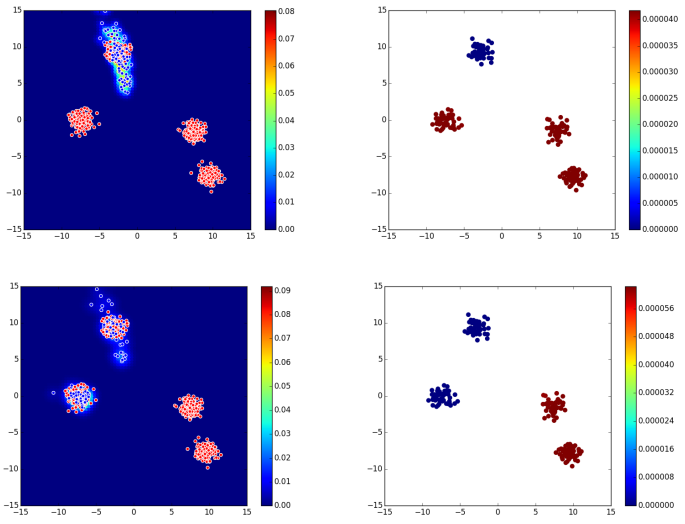


Figure 5.1: A 2D toy illustration of our meta-algorithm: AdaGAN. The **top left** shows a sample from a 2D target distribution (red dots) together with a sample produced by a single GAN (blue dots): the GAN obviously covers only one out of four modes. We thus (automatically) reweight the target sample (**top right**), train a new GAN on a sample of this re-weighted dataset, and build a mixture model out of the new and old GANs, which now covers two modes (**bottom right**). We can iterate this process, until all modes are covered.

AdaGAN Meta-Algorithm

Input: Training sample $S_N := \{X_1, \dots, X_N\}$.
Output: Mixture generative model $G = G_T$.

Train vanilla GAN $G_1 = \text{GAN}(S_N, W_1)$ with a uniform weight $W_1 = (1/N, \dots, 1/N)$ over the training points
for $t = 2, \dots, T$ **do**
#Choose the overall weight of the next mixture component
$\beta_t = \text{ChooseMixtureWeight}(t)$
#Update the weight of each training example
$W_t = \text{UpdateTrainingWeights}(G_{t-1}, S_N, \beta_t)$
#Train t*-th "weak" component generator* G_t^c
$G_t^c = \text{GAN}(S_N, W_t)$
#Update the overall generative model:
#Form a mixture of G_{t-1} *and* G_t^c.
$G_t = (1 - \beta_t) G_{t-1} + \beta_t G_t^c$
end for

Algorithm 5.1: AdaGAN, a meta-algorithm to construct a "strong" mixture of T individual generative models (f.ex. GANs), trained sequentially.

GANs [18], which already try to avoid mode-collapse. Thus, we do not aim at improving the original GAN or any other generative algorithm. We rather propose and analyze a meta-algorithm that can be used on top. This meta-algorithm is similar in spirit to AdaBoost in the sense that each iteration corresponds to learning a "weak" generative model (e.g., GAN) with respect to a re-weighted data distribution. The weights change over time to focus on the "hard" examples, i.e. those that the mixture has not been able to properly generate so far. The main steps of the AdaGAN algorithm are described in Algorithm 5 and its two first iterations are illustrated on a toy example in Figure 5.1.

The chapter is organized as follows. Section 5.1 presents our main theoretical results when iteratively building mixture models that minimize an arbitrary f-divergence. Section 5.1.3 shows that

[18] Che et al., *Mode Regularized GANs*, 2017

if the optimization at each step is perfect, the process converges to the true data distribution at an exponential rate (or even in a *finite number of steps*, for which we provide a necessary and sufficient condition). Section 5.1.4 then shows that imperfect solutions still lead to the exponential convergence-rate under certain "weak learnability" conditions. These results naturally lead to a new boosting-style iterative procedure for constructing generative models. When used with GANs, it results in our *AdaGAN* algorithm, detailed in Section 5.2. Finally, we report initial empirical results in Section 5.3, where we compare AdaGAN with several benchmarks, including original GAN and uniform mixture of multiple independently trained GANs. Proofs are to be found in Appendix C.4.

Notations and reminders. In this chapter, all densities are assumed to exist with respect to some dominating positive finite measure μ. For a probability measure P, we denote this density dP. f-divergences were already introduced in introduction (Section 0.1 from the dual perspective. They can also be (and usually are) introduced from an equivalent, primal perspective, in which case they are defined as

Primal Definition of f-Divergences

$$D_f(Q\,\|\,P) := \int f\left(\frac{dQ}{dP}(x)\right) dP(x)$$

The reader may want to keep this primal definition in mind for some proofs. In this chapter, we always assume that f is convex, defined on $(0,\infty)$ and satisfies $f(1)=0$. $\mathcal{F}_{\text{div}}$ designates the set of such functions. Finally, several commonly used symmetric f-divergences are *Hilbertian* [30, 49]: the Jensen-Shannon divergence, Hellinger distance and the total variation among others. This implies in particular that their square-root satisfies the triangular inequality:

[30] Fuglede and Topsoe, *Jensen-Shannon Divergence and Hilbert Space Embedding*, 2004; [49] Hein and Bousquet, *Hilbertian Metrics and Positive Definite Kernels on Probability Measures*, 2005

Hilbertian f-Divergences

$$\sqrt{D_f(P\,\|\,Q)} \leqslant \sqrt{D_f(P\,\|\,R)} + \sqrt{D_f(R\,\|\,Q)}.$$

5.1 Minimizing f-divergence with Mixtures

5.1.1 Incremental Mixture Building

Our goal is to construct a generative model P_Y of a fixed target distribution P_X. To do so, assume that we are given a set of "weak" generative models $Q \in \mathcal{G}$ – for example GANs – for which we can approximately solve

$$\arg\min_{Q\in\mathcal{G}} D_f(Q\,\|\,R)\,. \tag{5.1}$$

for any given distribution R. We could of course just pick the best weak model that minimizes (5.1) when $R = P_X$. But we prefer to build a stronger model P_Y^T that combines T weak models Q_t into a mixture model

$$P_Y^T = \sum_{t=1}^{T} \alpha_t^T Q_t, \quad \text{where} \quad \alpha_t^T \geqslant 0, \ \sum_t \alpha_t^T = 1. \tag{5.2}$$

Mixture of T Generators Q_t

However, imagine we cannot directly minimize (5.1) over the complete mixture. Then we may instead train each Q_t one after another so as to maximize its "added-value" to the current mixture model P_Y^{t-1}. More precisely, each new component $Q_t \in \mathcal{G}$ with weight $\beta_t \in [0, 1]$ should minimize

$$\arg\min_{Q \in \mathcal{G}} D_f\left((1-\beta)P_Y + \beta Q \,\|\, P_X\right), \tag{5.3}$$

Optimizing the New Component

where we dropped the dependence on t for readability. In practice, we do not expect to find the optimal Q that minimizes (5.3) at each step. But our algorithm will still converge even if Q only slightly improves our current approximation of P_X, i.e. if for some $c < 1$, each new Q satisfies

$$D_f\left((1-\beta)P_Y + \beta Q \,\|\, P_X\right) \leqslant c \cdot D_f\left(P_Y \,\|\, P_X\right). \tag{5.4}$$

Weak Mixture-Improvement by New Component

To build Q we may be tempted to minimize (5.3) directly. But this approach has a significant drawback in practice. As we build up the mixture, we need to make β decrease: after all, if P_Y^t approximates P_X better and better, the correction at each step should get smaller and smaller. But since we approximate (5.3) using samples only, the sample from the mixture will contain only a fraction β of examples from Q. So, as t increases, getting meaningful information from a sample so as to tune Q becomes harder and harder (the information is "diluted"). To solve this issue, we propose to upper bound (5.3) by a problem of the form (5.4) where the distribution R can be computed as a re-weighting of the original data distribution P_X. This procedure is reminiscent of the AdaBoost algorithm [29], which combines multiple *weak* predictors into one *strong* composition. On each step AdaBoost adds new predictor to the current composition, which is trained to minimize the binary loss on the re-weighted training set. The weights are constantly updated to bias the next weak learner towards "hard" examples, which were incorrectly classified during previous stages.

Train Q on Reweighted Data R
Optimize $D_f(Q \,\|\, R)$, not (5.3)

[29] Freund and Schapire, *A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting*, 1997

In the following we will analyze the properties of (5.3) and derive upper bounds that provide practical optimization criteria to build the mixture. We will also show that under certain assumptions, the min-

imization of the upper bound leads to the optimum of the original criterion.

5.1.2 Upper Bounds

We provide two upper bounds on the divergence of the mixture in terms of the divergence of the additive component Q with respect to some reference distribution R.

Lemma 5.1.1. *Let* P_X, P_Y, Q, R *be four probability distributions,* $\beta \in [0,1]$, $f \in \mathcal{F}_{\mathrm{div}}$. *If the* f*-divergence* D_f *is Hilbertian, then*

$$\sqrt{D_f\left((1-\beta)P_Y + \beta Q \,\|\, P_X\right)} \leqslant \sqrt{\beta D_f\left(Q \,\|\, R\right)} + \sqrt{D_f\left((1-\beta)P_Y + \beta R \,\|\, P_X\right)}. \quad (5.5)$$

More generally, for any f*-divergence* D_f *s.t.* $f \in \mathcal{F}_{\mathrm{div}}$, *if* $\beta\, dR \leqslant dP_X$, *then*

$$D_f\left((1-\beta)P_Y + \beta Q \,\|\, P_X\right) \leqslant \beta D_f\left(Q \,\|\, R\right) + (1-\beta) D_f\left(P_Y \,\middle\|\, \frac{P_X - \beta R}{1-\beta}\right). \quad (5.6)$$

We can thus exploit those bounds by introducing some well-chosen distribution R and then minimizing them with respect to Q. A natural choice for R is a distribution that minimizes the last term of the upper bound (which does not depend on Q). The next theorem, this chapter's main result, indicates the shape of the distributions minimizing the right-most terms in those bounds.

Theorem 5.1.2. *For any differentiable* f*-divergence function* $f \in \mathcal{F}_{\mathrm{div}}$, *any fixed probability distributions* P_X, P_Y, *and any* $\beta \in (0,1]$, *the problem*

$$\arg\min_{R \in \mathcal{P}} D_f\left((1-\beta)P_Y + \beta R \,\|\, P_X\right)$$

is minimized by a distribution R^*_β *with density*

$$dR^*_\beta(x) = \frac{1}{\beta}\left(\lambda^* dP_X(x) - (1-\beta) dP_Y(x)\right)_+ = \frac{dP_X}{\beta}\left(\lambda^* - (1-\beta)\frac{dP_Y}{dP_X}\right)_+. \quad (5.7)$$

Optimal Reweighting R^*_β

for the unique $\lambda^* \in [\beta, 1]$ *that satisfies* $\int dR^*_\beta = 1$. *Also,* $\lambda^* = 1$ *iff* $P_X((1-\beta)dP_Y > dP_X) = 0$, *i.e. iff* $\beta\, dR^*_\beta = dP_X - (1-\beta)dP_Y$.

Theorem 5.1.3. *Let* $P_X, P_Y \in \mathcal{P}$, $\beta \in (0,1]$, *and* f *be defined over* $(0,\infty)$. *Assume that* $P_X(dP_Y = 0) < \beta$. *Then*

$$\arg\min_{R:\, \beta\, dR \leqslant dP_X} D_f\left(P_Y \,\middle\|\, \frac{P_X - \beta R}{1-\beta}\right)$$

has a solution with density $dR^\dagger_\beta(x) = \frac{1}{\beta}\big(dP_X(x) - \lambda^\dagger(1-\beta)dP_Y(x)\big)_+$ *for the unique* $\lambda^\dagger \geqslant 1$ *that satisfies* $\int dR^\dagger_\beta = 1$.

Optimal Reweighting $R^\dagger_\beta$

Surprisingly, in both Theorems 5.1.2 and 5.1.3, the solutions do not depend on the choice of the function f, which means that the solution is the same for *any* f-divergence[2]. Note that λ^* is implicitly defined by a fixed-point equation. In Section 5.2 we will show how it can be computed efficiently in the case of empirical distributions.

[2] In particular, by replacing f with $f^\circ(x) := xf(1/x)$, we get the same solution for the criterion written in the other direction. Hence the order in which we write the divergence does not matter and the optimal solution is optimal for both orders.

5.1.3 Convergence Analysis for Optimal Updates

In the previous section we derived the analytical expressions of the distributions R that minimize the last term of (5.5) and (5.6) respectively. Assuming for now that Q can perfectly match R, i.e. $D_f(Q\,\|\,R) = 0$, we are now interested in the convergence of the mixture (5.2) to the true data distribution P_X when $Q = R^*_\beta$ or $Q = R^\dagger_\beta$. We start with simple results showing that adding R^*_β or $R^\dagger_\beta$ to the current mixture yields a strict improvement of the divergence, as targeted by (5.4).

Lemma 5.1.4. *Under the conditions of Theorem 5.1.2, we have*

$$D_f\Big((1-\beta)P_Y + \beta R^*_\beta \,\Big\|\, P_X\Big) \leqslant D_f\left((1-\beta)P_Y + \beta P_X \,\|\, P_X\right) \leqslant (1-\beta)D_f\left(P_Y \,\|\, P_X\right).$$

Under the conditions of Theorem 5.1.3, we have

$$D_f\left(P_Y \,\middle\|\, \frac{P_X - \beta R^\dagger_\beta}{1-\beta}\right) \leqslant D_f\left(P_Y \,\|\, P_X\right) \quad \textit{and}$$

$$D_f\Big((1-\beta)P_Y + \beta R^\dagger_\beta \,\Big\|\, P_X\Big) \leqslant (1-\beta)D_f\left(P_Y \,\|\, P_X\right).$$

Property 5.4
Optimal Reweightings R^*_β *&* $R^\dagger_\beta$ *Improve Mixture Exponentially*

Imagine repeatedly adding T new components to the current mixture P_Y, where on every step we use the same weight β and choose the components described in Theorem 5.1.2. Then Lemma 5.1.4 guarantees that the original objective value $D_f(P_Y\,\|\,P_X)$ gets reduced at least to $(1-\beta)^T D_f(P_Y\,\|\,P_X)$. This exponential convergence rate may look surprisingly good. But it relies on the optimal component target R^*_β, which depends on P_X, which we actually only know via a sample.

Lemma 5.1.4 also suggests setting β as large as possible since we assume that we can compute the optimal mixture component (which for $\beta = 1$ is P_X). However, in practice we may prefer to keep β relatively small to preserve what we learned so far through P_Y: for instance, when P_Y already covered a few modes of P_X, we only need Q to cover the remaining ones. Section 5.2 provides further discussions on how to choose β. Additionally, Corollary 1 in [117] establishes

[117] Tolstikhin et al., *AdaGAN - arXiv*, 2017

necessary and sufficient conditions for the mixture to converge, not just exponentially, but even in a finite number of steps.

5.1.4 Suboptimal Updates: Weak to Strong Learnability

In practice the component Q that we add to the mixture is not exactly R^*_β or $R^\dagger_\beta$, but rather an approximation. In this section we show that if this approximation is good enough, then we retain the property (5.4) (exponential improvements).

Looking again at Lemma 5.1.1, we notice that the first upper bound is less tight than the second one. Indeed, take the optimal distributions provided by Theorems 5.1.2 and 5.1.3 and plug them back as R into the upper bounds of Lemma 5.1.1. Assume that Q can match R exactly, i.e. $D_f(Q \| R) = 0$. In this case both sides of (5.5) are equal to $D_f\left((1-\beta)P_Y + \beta R^*_\beta \,\middle\|\, P_X\right)$, which is the optimal value for the original objective (5.3). On the other hand, (5.6) does not become an equality and the r.h.s. is not the optimal one for (5.3). Nevertheless, Corollaries 5.1.5 and 5.1.6 will show that our more modest goal (5.4) – which guarantees exponential improvements – may still be reached. They provide sufficient conditions for strict improvements when we use the upper bounds (5.6) and (5.5) respectively.

Corollary 5.1.5. *Let* $P_X, P_Y \in \mathcal{P}$, $\beta \in (0,1]$ *and assume that* $P_X\left(\frac{dP_Y}{dP_X} = 0\right) < \beta$. *Let* $R^\dagger_\beta$ *be as defined in Theorem 5.1.3. If* Q *satisfies*

$$D_f\left(Q \,\middle\|\, R^\dagger_\beta\right) \leqslant \gamma D_f(P_Y \| P_X) \tag{5.8}$$

for some $\gamma \in [0,1]$, *then*

$$D_f((1-\beta)P_Y + \beta Q \| P_X) \leqslant (1-\beta(1-\gamma))\, D_f(P_Y \| P_X).$$

Imperfect Reweightings May Lead to Exponential Improvement

Corollary 5.1.6. *Let* D_f *be a Hilbertian f-divergence. Take any* $\beta \in (0,1]$, P_X, P_Y, *and let* R^*_β *be as defined in Theorem 5.1.2. If* Q *satisfies*

$$D_f\left(Q \,\middle\|\, R^*_\beta\right) \leqslant \gamma D_f(P_Y \| P_X) \tag{5.9}$$

for some $\gamma \in [0,1]$, *then* $D_f((1-\beta)P_Y + \beta Q \| P_X) \leqslant C_{\gamma,\beta} \cdot D_f(P_Y \| P_X)$, *where* $C_{\gamma,\beta} = \left(\sqrt{\gamma\beta} + \sqrt{1-\beta}\right)^2$ *is strictly smaller than* 1 *as soon as* $\gamma < \beta/4$ *(and* $\beta > 0$*).*

Conditions (5.8) and (5.9) may be compared to the "weak learnability" condition of AdaBoost. As long as our weak learner is able to solve the surrogate problem (5.1) of matching respectively $R^\dagger_\beta$ or R^*_β accurately enough, the original objective (5.3) is guaranteed to decrease as well. Note however that when $\gamma < \beta/4$, Condition (5.9) might be too strong to be called "weak". Indeed, as already men-

(5.8) & (5.9) = *Weak Learnability*

tioned, the weight β usually decreases as the number of mixture components T increases, which makes it harder and harder to meet Condition (5.9). This obstacle may be partially resolved by the fact that we will use a GAN to fit Q, which corresponds to a relatively rich[3] class of models $\mathcal{G}$ in (5.1). In other words, our weak learner is not so weak. As to Condition (5.8), it is milder. No matter what $\gamma \in [0, 1]$ and $\beta \in (0, 1]$ are, the new component Q is guaranteed to strictly improve our objective. This comes at the price of the additional condition $P_X(dP_Y/dP_X = 0) < \beta$, which asserts that β should be larger than the mass of true data P_X missed by the current model P_Y. This is a rather reasonable condition: if P_Y still missed many modes of P_X we would prefer to assign a relatively large weight β to the new component Q. However both Conditions (5.8) and (5.9) are difficult to check in practice.

[3] How hard it is to meet Condition (5.9) of course depends on the class of models $\mathcal{G}$ used to fit Q in (5.1). We leave this analysis for future research.

5.2 AdaGAN

We now describe the functions *ChooseMixtureWeight* and *UpdateTrainingWeights* of Algorithm 5. The complete AdaGAN meta-algorithm with the details of *UpdateTrainingWeight* and *ChooseMixtureWeight*, is summarized in Algorithm B.2 of Appendix B.2.1.

UpdateTrainingWeights. At each iteration we add a new component Q with weight β to the current mixture P_Y which yields a new mixture $(1-\beta)P_Y + \beta Q$. Q should approach the "optimal target" R^*_β provided by (5.7) in Theorem 5.1.2. But R^*_β depends on the density ratio dP_Y/dP_X, which is not directly accessible. We can however estimate this ratio using adversarial training as follows. We train a separate *mixture discriminator* D_M to approximately minimize the f-divergence between P_X and the current mixture P_Y (using samples only):

Reweighting Data Using R^*_β Requires dP_Y/dP_X

$$\arg\min_{D_M} \mathop{\mathbb{E}}_{X\sim P_X}[D_M(X)] - \mathop{\mathbb{E}}_{Y\sim P_Y}[f^*(Y)]$$

But from Nowozin et al. [82], we know that to each f-divergence corresponds a function h such that the values of the optimal discriminator D_M are related to the density ratio by

[82] Nowozin et al., *f-GAN*, 2016

$$\frac{dP_Y}{dP_X}(x) = h\big(D_M(x)\big). \tag{5.10}$$

Approximate dP_Y/dP_X With a P_X-P_Y-Discriminator

We can replace $dP_Y(x)/dP_X(x)$ in (5.7) with $h\big(D_M(x)\big)$. For the Jensen-Shannon divergence, used by the original GAN algorithm,

$h(z) = \frac{1-z}{z}$. In practice, when we compute dR^*_β on the training sample $S_N = (X_1, \ldots, X_N)$, each example X_i receives weight

$$w_i = \frac{1}{\beta N}\big(\lambda^* - (1-\beta)h(d_i)\big)_+ , \quad \text{where} \quad d_i = D_M(X_i) . \tag{5.11}$$

The only remaining task is to determine λ^*. As the weights w_i in (5.11) must sum to 1, we get:

$$\lambda^* = \frac{\beta}{\sum_{i\in\mathcal{I}(\lambda^*)} p_i}\left(1 + \frac{(1-\beta)}{\beta}\sum_{i\in\mathcal{I}(\lambda^*)} p_i h(d_i)\right) \tag{5.12}$$

Determining the λ^* in R^*

where $\mathcal{I}(\lambda) := \{i : \lambda > (1-\beta)h(d_i)\}$. To find $\mathcal{I}(\lambda^*)$, we sort $h(d_i)$ in increasing order: $h(d_1) \leqslant \ldots \leqslant h(d_N)$. Then $\mathcal{I}(\lambda^*)$ is a set consisting of the first k indices. We then successively test all k-s until the λ given by (5.12) verifies $(1-\beta)h(d_k) < \lambda \leqslant (1-\beta)h(d_{k+1})$. This procedure is guaranteed to converge by Theorem 5.1.2. It is summarized in Algorithm B.1 of Appendix B.2.1.

ChooseMixtureWeight. For every β there is an optimal reweighting scheme with weights given by (5.11). If the GAN could perfectly approximate its target R^*_β, then choosing $\beta = 1$ would be optimal, because $R^*_1 = P_X$. But in practice, GANs cannot do that. So we propose to choose β heuristically by imposing that each generator of the final mixture model has same weight. This yields $\beta_t = 1/t$, where t is the iteration index. Other heuristics are proposed in Appendix B.2.2, but did not lead to any significant difference.

β Heuristic: $\beta_t = 1/t$

The optimal discriminator. In practice it is of course hard to find the optimal discriminator D_M achieving the global maximum of the variational representation for the f-divergence and verifying (5.10). For the JS-divergence this would mean that D_M is the classifier achieving minimal expected cross-entropy loss in the binary classification between P_Y and P_X. In practice, we observed that the reweighting (5.11) leads to the desired property of emphasizing at least some of the missing modes as long as D_M distinguishes reasonably between data points already covered by the current model P_Y and those which are still missing. We found an early stopping (while training D_M) sufficient to achieve this. In the *worst case*, when D_M overfits and returns 1 for all true data points, the reweighting simply leads to the uniform distribution over the training set.

Restrict the Power of the P_X-P_Y-Discriminator

5.3 Experiments

We ran AdaGAN[4] on toy datasets, for which we can interpret the missing modes in a clear and reproducible way, and on MNIST, which is a high-dimensional dataset. The goal of these experiments

[4] Code available online at https://github.com/tolstikhin/adagan

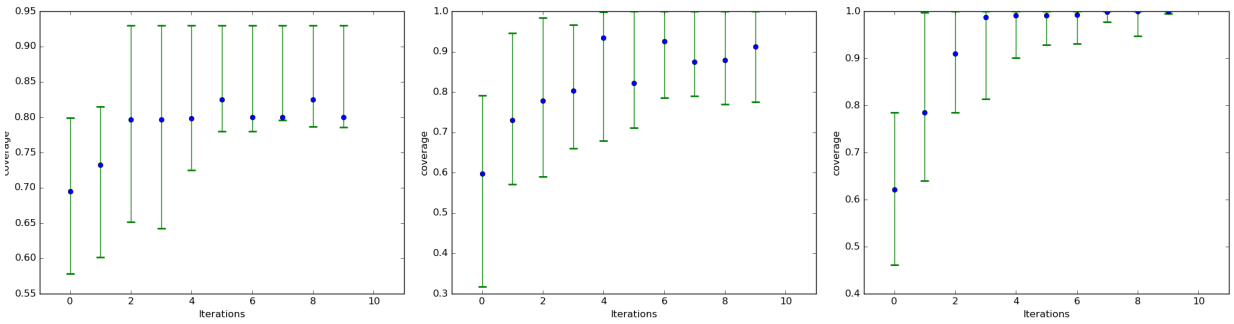


Figure 5.2: Coverage C of the true data by the model distribution P_Y^T, as a function of iterations T. Experiments correspond to the data distribution with 5 modes. Each blue point is the median over 35 runs. Green intervals are defined by the 5% and 95% percentiles (see Section 5.3). Iteration 0 is equivalent to one vanilla GAN. The **left** plot corresponds to taking the best generator out of T runs. The **middle** plot is an "ensemble" GAN, simply taking a uniform mixture of T independently trained GAN generators. The **right** plot corresponds to our boosting approach (AdaGAN), with $\beta_t = 1/t$.

was not to evaluate the visual quality of individual sample points, but to demonstrate that the re-weighting scheme of AdaGAN promotes diversity and effectively covers the missing modes.

5.3.1 Toy Datasets

Recovering 2D Mixture of 5 Gaussians

Our target distribution is a mixture of isotropic Gaussians over $\mathbb{R}^2$, as in Figure 5.1. The distances between the means are large enough to roughly avoid overlaps between different Gaussian components. We vary the number of modes to test how well each algorithm performs when there are fewer or more expected modes. We compare the baseline GAN algorithm with AdaGAN variations, and with other meta-algorithms that all use the same underlying GAN procedure. For details on these algorithms and on the architectures of the underlying generator and discriminator, see Appendix B.2.2.

To evaluate how well the generated distribution matches the target distribution, we use a *coverage* metric C. We compute the probability mass of the true data "covered" by the model P_Y. More precisely, we compute $C := P_X(dP_Y > t)$ with t such that $P_Y(dP_Y > t) = 0.95$. This metric is more interpretable than the likelihood, making it easier to assess the difference in performance of the algorithms. To approximate the density of P_Y we use a kernel density estimation, where the bandwidth is chosen by cross validation. We repeat the run 35 times with the same parameters (but different random seeds). For each run, the learning rate is optimized using a grid search on a validation set. We report the median over those multiple runs, and the interval corresponding to the 5% and 95% percentiles.

Figure 5.2 summarizes the performance of algorithms as a function of the number of iterations T. Both the ensemble and the boosting approaches significantly outperform the vanilla GAN and the "best of T" algorithm. Interestingly, the improvements are significant even after just one or two additional iterations ($T = 2$ or 3). Our boosting approach converges much faster. In addition, its variance is much lower, improving the likelihood that a given run gives good

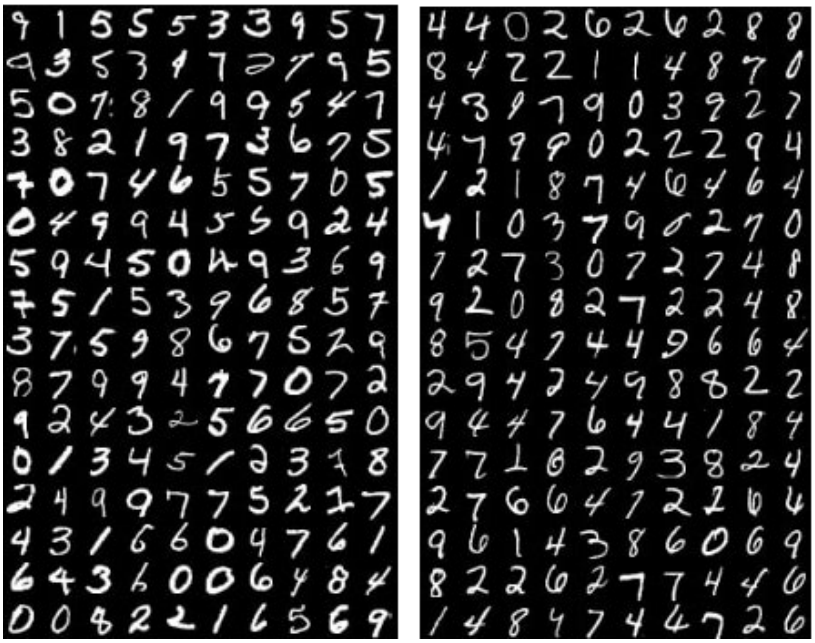


Figure 5.3: Digits from the MNIST dataset corresponding to the smallest (**left**) and largest (**right**) weights, obtained by the AdaGAN procedure (see Section 5.2) in one of the runs. Bold digits (left) are already covered and next GAN will concentrate on thin (**right**) digits.

results. On this setup, the vanilla GAN approach has a significant number of catastrophic failures (visible in the lower bounds of the intervals). Further empirical results are available in Appendix B.2.2, where we compared AdaGAN variations to several other baseline meta-algorithms in more details (Table B.1) and combined AdaGAN with the unrolled GANs (UGAN) [75] (Figure B.1). Interestingly, Figure B.1 shows that AdaGAN ran with UGAN outperforms the vanilla UGAN on the toy datasets, demonstrating the advantage of using AdaGAN as a way to further improve the mode coverage of any existing GAN implementations.

[75] Metz et al., *Unrolled GANs*, 2017

5.3.2 MNIST and MNIST3

We ran experiments both on the original MNIST and on the 3-digit MNIST (MNIST3) [18, 75] dataset, obtained by concatenating 3 randomly chosen MNIST images to form a 3-digit number between 0 and 999. According to [18, 75], MNIST contains 10 modes, while MNIST3 contains 1000 modes, and these modes can be detected using the pre-trained MNIST classifier. We combined AdaGAN both with simple MLP GANs and DCGANs [86]. We used $T \in \{5, 10\}$, tried models of various sizes and performed a reasonable amount of hyperparameter search.

[18] Che et al., *Mode Regularized GANs*, 2017; [75] Metz et al., *Unrolled GANs*, 2017;

[18] Che et al., *Mode Regularized GANs*, 2017; [75] Metz et al., *Unrolled GANs*, 2017;

[86] Radford et al., *Unsupervised Representation Learning with Deep Convolutional GANs*, 2016

Similarly to [75] we failed to reproduce the missing modes problem for MNIST3 reported in [18] and found that simple GAN architectures are capable of generating all 1000 numbers. The authors of [75] proposed to artificially introduce the missing modes again by limiting the generators' flexibility. In our experiments, GANs trained with the architectures reported in [75] were often generating poorly

[75] Metz et al., *Unrolled GANs*, 2017, Sec 3.3.1

[18] Che et al., *Mode Regularized GANs*, 2017

[75] Metz et al., *Unrolled GANs*, 2017

[75] Metz et al., *Unrolled GANs*, 2017

looking digits. As a result, the pre-trained MNIST classifier was outputting random labels, which again led to full coverage of the 1000 numbers. We tried to threshold the confidence of the pre-trained classifier, but decided that this metric was too ad-hoc.

Reweighted Data Emphasizes Specific Strokes

For MNIST we noticed that the re-weighted distribution was often concentrating its mass on digits having very specific strokes: on different rounds it could highlight thick, thin, vertical, or diagonal digits, indicating that these traits were underrepresented in the generated samples (see Figure 5.3). This suggests that AdaGAN does a reasonable job at picking up different modes of the dataset, but also that there are more than 10 modes in MNIST (and more than 1000 in MNIST3). It is not clear how to evaluate the quality of generative models in this context.

We also tried to use the "inversion" metric discussed in [75]. For MNIST3 we noticed that a single GAN was capable of reconstructing most of the training points *very* accurately both visually and in the ℓ_2-reconstruction sense. The "inversion" metric tests whether the trained model can generate certain examples or not, but unfortunately it does not take into account *the probabilities* of doing so.

[75] Metz et al., *Unrolled GANs*, 2017, Sec 3.4.1

5.4 Chapter Conclusion

We studied the problem of minimizing general f-divergences with additive mixtures of distributions. Our detailed theoretical analysis naturally leads to a greedy iterative procedure. On every iteration the mixture is updated with a new component trained on a reweighted target distribution. We provided conditions under which this procedure is guaranteed to converge to the target distribution at an exponential rate. While our results can be combined with any generative modeling techniques, we focused on GANs and provided a boosting-style algorithm *AdaGAN*. Preliminary experiments show that AdaGAN successfully produces a mixture which iteratively recovers the missing modes.

This closes our second part, which concentrated on distribution-dissimilarities in the context of generative algorithms. Almost all those algorithms relied on classifier-based dissimilarities with a neural network as a classifier. We now turn specifically towards those "network-based dissimilarities", and one of their striking deficiencies: adversarial vulnerability.

Related Literature

Several authors [45, 120, 124] proposed to use boosting techniques in the context of density estimation by incrementally adding com-

[124] Welling et al., *Self Supervised Boosting*, 2002; [120] Tu, *Learning Generative Models via Discriminative Approaches*, 2007; [45] Grover and Ermon, *Boosted Generative Models*, 2018;

ponents in the log domain. This idea was even applied to GANs in [45]. A major downside of these approaches is that the resulting mixture is a product of components. Sampling from such models is nontrivial (at least when applied to GANs where the model density is not expressed analytically) and requires techniques such as Annealed Importance Sampling [81] for the normalization.

[45] Grover and Ermon, *Boosted Generative Models*, 2018

[81] Neal, *Annealed Importance Sampling*, 2001

When the log likelihood can be computed, [90] proposed to use an additive mixture model. They derived the update rule by computing the steepest descent direction when adding a component with infinitesimal weight. However, their results do not apply once the weight β becomes non-infinitesimal. In contrast, for any fixed weight of the new component our approach gives the overall optimal update (rather than just the best direction) for a specified f-divergence. In both theories, improvements of the mixture are guaranteed only if the new "weak" learner is still good enough (see Conditions 5.8 & 5.9)

[90] Rosset and Segal, *Boosting Density Estimation*, 2002

Similarly, [65] studied the construction of mixtures minimizing the KL divergence and proposed a greedy procedure to do so. They also proved that under certain conditions, finite mixtures can approximate arbitrary mixtures at a rate $1/k$ where k is the number of components in the mixture when the weight of each newly added component is $1/k$. These KL-specific results are consistent with our more general results.

[65] Li and Barron, *Mixture Density Estimation*, 1997

An additive procedure similar to ours was proposed in [123] but with a different re-weighting scheme, which is not motivated by a theoretical analysis of optimality conditions. On every new iteration the authors run a GAN on the k training examples with maximal values of the discriminator from the last iteration.

[123] Wang et al., *Ensembles of GANs*, 2016

Finally, many papers address mode-collapse issues by directly modifying the training objective of an individual GAN. For instance, [18] adds an auto-encoding cost to the training objective of GAN, while [75] allows the generator to "look a few steps ahead" when making a gradient step.

[18] Che et al., *Mode Regularized GANs*, 2017

[75] Metz et al., *Unrolled GANs*, 2017

Part III

Machine versus Human Perception

ALMOST ALL generative models encountered in the previous part appear to minimize a classifier-based, and more precisely a *network*-based distribution-dissimilarity. They compute approximate f-divergences (or optimal transportation costs) by optimizing a neural-network classifier, either in form of a GAN-like discriminator, or of a VAE-like encoder.[1] The properties of these classifiers largely determine those of the distribution-dissimilarities. Our introduction and first part for example stress the importance of the classifier's capacity. Too high capacity yields too strong dissimilarities, like f-divergences that routinely saturate on empirical measures. But if the capacity gets too low, the dissimilarity may miss essential differences. For MMDs (and more generally IPMs), if we want guaranteed perfect discrimination, then the classifier, we saw, needs enough capacity to approximate any bounded continuous function.[2]

[1] Both share very similar architectures.

From Classifier Properties to Dissimilarity Properties

[2] cf. Thm 1.2.2

However, perfect discrimination just means that the dissimilarity between P and Q is minimized only if $P = Q$. It does not say anything about other equi-dissimilarity curves. Two distributions may look much more alike to humans than what the computed dissimilarity value may suggest, despite being perfectly discriminative; and vice-versa. Especially for image generation, where we want to get realistically looking images, it seems natural to seek distribution-dissimilarities with "human-like perception". What we mean is that their scores should reflect the human perception of dissimilarity: the lower their score, the more the scored distributions should look alike to us. At a pinch, if two samples look so much alike that we, as humans, are unable to tell if their images came from one or the other sample, then we may as well let the distribution-dissimilarity give them the best possible score, even though their empirical measures may actually not be equal. So it seems that we may want to stop worrying so much about perfect discrimination and start focusing more on the general shape and properties of the equi-dissimilarity curves, i.e. the general scoring patterns of the dissimilarity and not just at the minimal score.

Perfect Discrimination Ignores Other Important Properties

That is what neural network based dissimilarities seem so good at. Neural network's startling curve-fitting abilities ensure enough capacity for (almost) perfect discrimination. But more importantly, the network's architecture gives much freedom to emphasize some distributional differences over others. Convolutional layers for example ensure local translation and morphing invariances [72], so that the dissimilarity be naturally inclined to attribute low scores to samples that differ only by small image-shifts and -morphings. For high-dimensional images, GANs heavily rely on such layers; and their ability to generate sometimes impressively good artificial images suggests that their discriminator may indeed reflect human perception.

[72] Mallat, *Understanding Deep Convolutional Networks*, 2016

Network Architecture Can Favor Some Dissimilarity Invariances

Another way to test the discriminator's ability to capture human perception is to train it on a usual binary or multi-class classification task and compare its decisions with human ones. Now, state-of-the-art convolutional neural nets (CNNs) can achieve almost human-like performance. So everything would seem fine, if it weren't for the recent discovery of *adversarial examples*. Indeed, Goodfellow et al. [36] noticed that invisible but targeted perturbations of input images can lead to drastically different predictions. Two indistinguishable samples may hence look radically different to the classifier, or equivalently, to the associated distribution-dissimilarity. That shows that their similarity perception is actually radically different from human perception. In this last, short part, we try to understand why. We show in particular that the adversarial vulnerability *of almost all current feedforward architectures* increases with the input-dimension. This explains why even state-of-the-art nets are so sensitive to adversarial input perturbations, and shows that our networks are vulnerable *by construction*. It also reveals an essential difference with humans. For humans, the higher the input resolution, the better. At some point, we even stop noticing any difference. For our current neural networks, it is all the contrary: the higher the input resolution, the more adversarially vulnerable they become. That suggests that neural networks rely much more than humans on high-frequency patterns for their decisions; something that needs to change, if our network-based distribution-dissimilarities are too capture, one day, human perception.

[36] Goodfellow et al., *Explaining and Harnessing Adversarial Examples*, 2015

Adversarial Vulnerability Reveals Discrepancy btw. Human and Machine Perception/Dissimilarity

6

Adversarial Vulnerability of Network Dissimilarities

Following the work of Goodfellow et al. [36], Convolutional Neural Networks (CNNs) have been found vulnerable to adversarial examples: an adversary can drive the performance of state-of-the art CNNs down to chance level with imperceptible changes of the inputs. A number of studies have tried to address this issue, but only few have stressed that, because adversarial examples are essentially small input changes that create large output variations, they are inherently caused by large gradients of the neural network with respect to its inputs. Of course, this view, which we will focus on here, assumes that the network and loss are differentiable. It has the advantage to yield a large body of specific mathematical tools, but might not be easily extendable to masked gradients, non-smooth models or the 0-1-loss. Nevertheless, our conclusions might even hold for non-smooth models, given that the latter can often be viewed as smooth at a coarser level.

[36] Goodfellow et al., *Explaining and Harnessing Adversarial Examples*, 2015

More specifically, we provide theoretical and empirical arguments supporting the existence of a monotonic relationship between the gradient norm of the training objective (of a differentiable classifier) and its adversarial vulnerability. Evaluating this norm based on the weight statistics at initialization, we show that CNNs and most feed-forward networks, *by design*, exhibit increasingly large gradients with input dimension d, almost independently of their architecture. That leaves them increasingly vulnerable to adversarial noise. We corroborate our theoretical results by extensive experiments. Although some of those experiments involve adversarial regularization schemes, our goal is not to advocate a new adversarial defense (these schemes are already known), but to show how their effect can be explained by our first order analysis. We do not claim to explain all aspects of adversarial vulnerability, but we claim that our first order argument suffices to explain a significant part of the empirical findings on adversarial vulnerability. This calls for researching the design of neural network architectures with inherently smaller gradients and provides useful guidelines to practitioners and network designers.

6.1 From Adversarial Examples to Large Gradients

Suppose that a given classifier φ classifies an image $\mathbf{x}$ as being in category $\varphi(\mathbf{x})$. An adversarial image is a small modification of $\mathbf{x}$, barely noticeable to the human eye, that suffices to fool the classifier into predicting a class different from $\varphi(\mathbf{x})$. It is a *small* perturbation of the inputs, that creates a *large* variation of outputs. Adversarial examples thus seem inherently related to large gradients of the network. A connection, that we will now clarify. Note that visible adversarial examples sometimes appear in the literature, but we deliberately focus on imperceptible ones.

Adversarial vulnerability and adversarial damage. In practice, an adversarial image is constructed by adding a perturbation δ to the original image $\mathbf{x}$ such that $\|\delta\| \leqslant \epsilon$ for some (small) number ϵ and a given norm $\|\cdot\|$ over the input space. We call the perturbed input $\mathbf{x}+\delta$ an ϵ-sized $\|\cdot\|$-attack and say that the attack was successful when $\varphi(\mathbf{x}+\delta) \neq \varphi(\mathbf{x})$. This motivates

Definition 6.1.1. *Given a distribution* P *over the input-space, we call* adversarial vulnerability *of a classifier* φ *to an* ϵ*-sized* $\|\cdot\|$*-attack the probability that there exists a perturbation* δ *of* $\mathbf{x}$ *such that*

$$\|\delta\| \leqslant \epsilon \quad \textit{and} \quad \varphi(\mathbf{x}) \neq \varphi(\mathbf{x}+\delta)\,. \tag{6.1}$$

Adversarial Vulnerability $\geqslant$ *Adversarial Damage*

We call the average increase-after-attack $\mathbb{E}_{\mathbf{x}\sim P}[\Delta\mathcal{L}]$ *of a loss* $\mathcal{L}$ *the (*$\mathcal{L}$*-)* adversarial damage *(of the classifier* φ *to an* ϵ*-sized* $\|\cdot\|$*-attack).*

When $\mathcal{L}$ is the 0-1-loss $\mathcal{L}_{0/1}$, adversarial damage is the accuracy-drop after attack. The 0-1-loss damage is always smaller than adversarial vulnerability, because vulnerability counts all class-changes of $\varphi(\mathbf{x})$, whereas some of them may be neutral to adversarial damage (e.g. a change between two wrong classes). The $\mathcal{L}_{0/1}$-adversarial damage thus lower bounds adversarial vulnerability. Both are even equal when the classifier is perfect (before attack), because then every change of label introduces an error. It is hence tempting to evaluate adversarial vulnerability with $\mathcal{L}_{0/1}$-adversarial damage.

From $\Delta\mathcal{L}_{0/1}$ to $\Delta\mathcal{L}$ and to $\partial_x\mathcal{L}$. In practice however, we do not train our classifiers with the non-differentiable 0-1-loss but use a smoother loss $\mathcal{L}$, such as the cross-entropy loss. For similar reasons, we will now investigate the adversarial damage $\mathbb{E}_{\mathbf{x}}[\Delta\mathcal{L}(\mathbf{x},c)]$ with loss $\mathcal{L}$ rather than $\mathcal{L}_{0/1}$. Like for [36, 70, 104] and many others, a classifier φ will hence be robust if, on average over $\mathbf{x}$, a small adversarial perturbation δ of $\mathbf{x}$ creates only a small variation $\delta\mathcal{L}$ of the loss. Now, But if $\|\delta\| \leqslant \epsilon$, then a first order Taylor expansion in ϵ shows that

From the 0-1-Loss to a Smoother Loss

[36] Goodfellow et al., *Explaining and Harnessing Adversarial Examples*, 2015; [70] Lyu et al., *A Unified Gradient Regularization Family for Adversarial Examples*, 2015; [104] Sinha et al., *Certifiable Distributional Robustness with Principled Adversarial Training*, 2018

$$\delta\mathcal{L} = \max_{\delta\,:\,\|\delta\|\leqslant\epsilon} |\mathcal{L}(x+\delta, c) - \mathcal{L}(x, c)| \approx \max_{\delta\,:\,\|\delta\|\leqslant\epsilon} |\partial_x\mathcal{L}\cdot\delta| = \epsilon\,\|\partial_x\mathcal{L}\|_*, \tag{6.2}$$

where $\partial_x\mathcal{L}$ denotes the gradient of $\mathcal{L}$ with respect to x, and where the last equality stems from the definition of the dual norm $\|\cdot\|_*$ of $\|\cdot\|$. Now two remarks. First: the dual norm only kicks in because we let the input noise δ optimally adjust to the coordinates of $\partial_x\mathcal{L}$ within its ϵ-constraint. This is the brand mark of *adversarial* noise: the different coordinates add up, instead of statistically canceling each other out as they would with random noise. For example, if we impose that $\|\delta\|_2 \leqslant \epsilon$, then δ will strictly align with $\partial_x\mathcal{L}$. If instead $\|\delta\|_\infty \leqslant \epsilon$, then δ will align with the sign of the coordinates of $\partial_x\mathcal{L}$. Second remark: while the Taylor expansion in (6.2) becomes exact for infinitesimal perturbations, for finite ones it may actually be dominated by higher-order terms. Our experiments (Figure 6.1) however strongly suggest that in practice the first order term dominates the others. Now, remembering that the dual norm of an ℓ_p-norm is the corresponding ℓ_q-norm, and summarizing, we have proven

Lemma 6.1.2. *At first order approximation in ϵ, an ϵ-sized adversarial attack generated with norm $\|\cdot\|$ increases the loss $\mathcal{L}$ at point x by $\epsilon\,\|\partial_x\mathcal{L}\|_*$, where $\|\cdot\|_*$ is the dual norm of $\|\cdot\|$. In particular, an ϵ-sized ℓ_p-attack increases the loss by $\epsilon\,\|\partial_x\mathcal{L}\|_q$ where $1 \leqslant p \leqslant \infty$ and $\frac{1}{p} + \frac{1}{q} = 1$.*

Adversarial Damage $\propto$ Loss-Gradient-Norm

Consequently, the adversarial damage of a classifier to ϵ-sized attacks generated with norm $\|\cdot\|$ is approximately $\epsilon\,\mathbb{E}_x\,\|\partial_x\mathcal{L}\|_*$. Since adversarial damage is always smaller than adversarial vulnerability, this also estimates a lower-bound of the latter. This is valid only at first order, but it proves that *at least* this kind of first-order vulnerability is present. We will see that the first-order predictions closely match the experiments, and that this insight helps protect even against iterative (non-first-order) attack methods (Figure 6.1).

Calibrating the threshold ϵ to the attack-norm $\|\cdot\|$. Going back to Lemma 6.1.2, we see that adversarial vulnerability depends on three main factors: (i) $\|\cdot\|$, the norm chosen for the attack (ii) ϵ, the size of the attack, and (iii) $\mathbb{E}_x\,\|\partial_x\mathcal{L}\|_*$, the expected *dual* norm of $\partial_x\mathcal{L}$. We could see Point (i) as a measure of our sensibility to image perturbations, (ii) as our sensibility threshold, and (iii) as the classifier's expected marginal sensibility to a unit perturbation. $\mathbb{E}_x\,\|\partial_x\mathcal{L}\|_*$ hence intuitively captures the discrepancy between our perception (as modeled by $\|\cdot\|$) and the classifier's perception for an input-perturbation of small size ϵ. Of course, this viewpoint supposes that we actually found a norm $\|\cdot\|$ (or more generally a metric) that faithfully reflects human perception – a project in its own right, far beyond the scope of this chapter. However, it is clear that the threshold ϵ that we choose

Threshold ϵ Should Depend on Norm $\|\cdot\|$

should depend on the norm $\|\cdot\|$ and hence on the input-dimension d. In particular, for a given pixel-wise order of magnitude of the perturbations δ, the ℓ_p-norm of the perturbation will scale like $d^{1/p}$. This suggests to write the threshold ϵ_p used with ℓ_p-attacks as:

$$\epsilon_p = \epsilon_\infty \, d^{1/p} \,, \tag{6.3}$$

Constant Perception Threshold

where ϵ_∞ denotes a dimension-independent constant. In Appendix B.3.3 we show that this scaling also preserves the average signal-to-noise ratio $\|x\|_2 / \|\delta\|_2$, both across norms and dimensions, so that ϵ_p could correspond to a constant human perception-threshold. With this in mind, the impatient reader may already jump to Section 6.2, which contains our main contributions: the estimation of $\mathbb{E}\, x \|\partial_x \mathcal{L}\|_q$ for standard feed-forward nets. Meanwhile, the rest of this section shortly discusses two straightforward defenses that we will use later and that further illustrate the role of gradients.

A new old regularizer. Lemma 6.1.2 shows that the loss of the network after an $\epsilon/2$-sized $\|\cdot\|$-attack is

$$\mathcal{L}_{\epsilon,|||\cdot|||}(x, c) := \mathcal{L}(x, c) + \frac{\epsilon}{2} |||\partial_x \mathcal{L}||| \,. \tag{6.4}$$

Linearized Loss-After-Attack

It is thus natural to take this loss-after-attack as a new training objective. Here we introduced a factor 2 for reasons that will become clear in a moment. Incidentally, for $\|\cdot\| = \|\cdot\|_2$, this new loss reduces to an old regularization-scheme proposed by Drucker and LeCun [25] called *double-backpropagation*. At the time, the authors argued that slightly decreasing a function's or a classifier's sensitivity to input perturbations should improve generalization. In a sense, this is exactly our motivation when defending against adversarial examples. It is thus not surprising to end up with the same regularization term. Note that our reasoning only shows that training with one specific norm $|||\cdot|||$ in (6.4) helps to protect against adversarial examples generated from $\|\cdot\|$. A priori, we do not know what will happen for attacks generated with other norms; but our experiments suggest that training with one norm also protects against other attacks (see Figure 6.2 and Section 6.3.1).

$\|\cdot\|_2$ Yields Double-Backprop

[25] Drucker and LeCun, *Double Back-propagation Increasing Generalization Performance*, 1991

Link to adversarially-augmented training. In (6.1), ϵ designates an attack-size threshold, while in (6.4), it is a regularization-strength. Rather than a notation conflict, this reflects an intrinsic duality between two complementary interpretations of ϵ, which we now investigate further. Suppose that, instead of using the loss-after-attack, we augment our training set with ϵ-sized $\|\cdot\|$-attacks $x + \delta$, where for each training point x, the perturbation δ is generated on the fly to

locally maximize the loss-increase. Then we are effectively training with

$$\tilde{\mathcal{L}}_{\epsilon,\|\cdot\|}(x, c) := \frac{1}{2}(\mathcal{L}(x, c) + \mathcal{L}(x + \epsilon\, \delta, c)), \tag{6.5}$$

Adversarially Augmented Loss

where by construction δ satisfies (6.2). We will refer to this technique as *adversarially augmented training*. It was first introduced in [36] with $\|\cdot\| = \|\cdot\|_\infty$ under the name of FGSM[1]-augmented training. Using the first order Taylor expansion in ϵ of (6.2), this 'old-plus-post-attack' loss of (6.5) simply reduces to our loss-after-attack, which proves

[36] Goodfellow et al., *Explaining and Harnessing Adversarial Examples*, 2015

[1] FGSM = *F*ast *G*radient *S*ign *M*ethod

Proposition 6.1.3. *Up to first-order approximations in* ϵ, $\tilde{\mathcal{L}}_{\epsilon,\|\cdot\|} = \mathcal{L}_{\epsilon,|||\cdot|||}$. *Said differently, for small enough* ϵ, *adversarially-augmented training with* ϵ*-sized* $\|\cdot\|$*-attacks amounts to penalizing the* dual *norm* $|||\cdot|||$ *of* $\partial_x\mathcal{L}$ *with weight* $\epsilon/2$. *In particular, double-backpropagation corresponds to training with* ℓ_2*-attacks, while FGSM-augmented training corresponds to an* ℓ_1*-penalty on* $\partial_x\mathcal{L}$.

Adversarially Augmented Loss ≈ Linearized Loss-After-Attack

This correspondence between training with perturbations and using a regularizer can be compared to Tikhonov regularization: Tikhonov regularization amounts to training with *random* noise [12], while training with *adversarial* noise amounts to penalizing $\partial_x\mathcal{L}$. Section 6.3.1 verifies the correspondence between adversarial augmentation and gradient regularization empirically, which also strongly suggests the empirical validity of the first-order Taylor expansion in (6.2).

Tikhonov Regu. ≈ Random Noise
Gradient Regu. ≈ Advers. Noise

[12] Bishop, *Training with noise is equivalent to Tikhonov regularization*, 1995

6.2 Gradient and Adversarial Vulnerability Estimation

In this section, we evaluate the size of $\|\partial_x\mathcal{L}\|_q$ for standard neural network architectures. We start with fully-connected networks, and finish with a much more general theorem that, not only encompasses CNNs (with or without strided convolutions), but also shows that the gradient-norms are essentially independent of the network topology. We start our analysis by showing how changing q affects the size of $\|\partial_x\mathcal{L}\|_q$. Suppose for a moment that the coordinates of $\partial_x\mathcal{L}$ have typical magnitude $|\partial_x\mathcal{L}|$. Then $\|\partial_x\mathcal{L}\|_q$ scales like $d^{1/q}|\partial_x\mathcal{L}|$. Consequently

$$\epsilon_p \|\partial_x\mathcal{L}\|_q \;\propto\; \epsilon_p\, d^{1/q}\, |\partial_x\mathcal{L}| \;\propto\; d\,|\partial_x\mathcal{L}|\,. \tag{6.6}$$

Adv. Damage ∝ Input-Dim d & Avg. Gradient-Coord. Size

This equation carries two important messages. First, we see how $\|\partial_x\mathcal{L}\|_q$ depends on d and q. The dependence seems highest for $q = 1$. But once we account for the varying perceptibility threshold $\epsilon_p \propto d^{1/p}$, we see that adversarial vulnerability scales like $d \cdot |\partial_x\mathcal{L}|$, whatever ℓ_p-norm we use. Second, (6.6) shows that to be robust against any type of ℓ_p-attack at any input-dimension d, the average

absolute value of the coefficients of $\partial_x \mathcal{L}$ must grow slower than $1/d$. Now, here is the catch, which brings us to our core insight.

6.2.1 Core Idea: One Neuron with Many Inputs

In order to preserve the activation variance of the neurons from layer to layer, the neural weights are usually initialized with a variance that is inversely proportional to the number of inputs per neuron. Imagine for a moment that the network consisted only of one output neuron o linearly connected to all input pixels. For the purpose of this example, we assimilate o and $\mathcal{L}$. Because we initialize the weights with a variance of $1/d$, their average absolute value $|\partial_x o| \equiv |\partial_x \mathcal{L}|$ grows like $1/\sqrt{d}$, rather than the required $1/d$. By (6.6), the adversarial vulnerability $\epsilon \|\partial_x o\|_q \equiv \epsilon \|\partial_x \mathcal{L}\|_q$ therefore increases like $d/\sqrt{d} = \sqrt{d}$.

For Linear Nets, Average Gradient-Coord. Size $\propto 1/\sqrt{d}$

This toy example shows that the standard initialization scheme, which preserves the variance from layer to layer, causes the average coordinate-size $|\partial_x \mathcal{L}|$ to grow like $1/\sqrt{d}$ instead of $1/d$. When an ℓ_∞-attack tweaks its ϵ-sized input-perturbations to align with the coordinate-signs of $\partial_x \mathcal{L}$, all coordinates of $\partial_x \mathcal{L}$ add up in absolute value, resulting in an output-perturbation that scales like $\epsilon\sqrt{d}$ and leaves the network increasingly vulnerable with growing input-dimension.

6.2.2 Generalization to Deep Networks

Our next theorems generalize the previous toy example to a very wide class of feedforward nets with ReLU activation functions. For illustration purposes, we start with fully connected nets and only then proceed to the broader class, which includes any succession of (possibly strided) convolutional layers. In essence, the proofs iterate our insight on one layer over a sequence of layers. They all rely on the following set ($\mathcal{H}$) of hypotheses:

Assumptions on Network's Weight-Distribution

- H1 Non-input neurons are followed by a ReLU killing half of its inputs, independently of the weights.
- H2 Neurons are partitioned into layers, meaning groups that each path traverses at most once.
- H3 All weights have 0 expectation and variance $2/(\text{in-degree})$ ('He-initialization').
- H4 The weights from different layers are independent.
- H5 Distinct weights w, w' from a same node satisfy $\mathbb{E}[w\,w'] = 0$.

If we follow common practice and initialize our nets as proposed by He et al. [47], then H3-H5 are satisfied at initialization by design, while H1 is usually a very good approximation [5]. Note that

[47] He et al., *Delving Deep into Rectifiers*, 2015
[5] Balduzzi et al., *Neural Taylor Approximations*, 2017

such i.i.d. weight assumptions have been widely used to analyze neural nets and are at the heart of very influential and successful prior work (e.g., equivalence between neural nets and Gaussian processes as pioneered by [80]). Nevertheless, they do not hold after training. That is why all our statements in this section are to be understood as *orders of magnitudes* that are very well satisfied at initialization in theory and in practice, and that we will confirm experimentally after training in Section 6.3. Said differently, while our theorems rely on the statistics of neural nets at initialization, our experiments confirm their conclusions after training.

[80] Neal, *Bayesian Learning for Neural Networks*, 1996

Theorem 6.2.1. *Consider a succession of fully connected layers with ReLU activations which takes inputs x of dimension d, satisfies assumptions (H), and outputs logits $f_k(x)$ that get fed to a final cross-entropy-loss layer $\mathcal{L}$. Then the coordinates of $\partial_x f_k$ grow like $1/\sqrt{d}$, and*

Vulnerability of Fully Connected Nets

$$\|\partial_x \mathcal{L}\|_q \propto d^{\frac{1}{q}-\frac{1}{2}} \quad \text{and} \quad \epsilon_p \|\partial_x \mathcal{L}\|_q \propto \sqrt{d}\,. \tag{6.7}$$

These networks are thus increasingly vulnerable to ℓ_p-attacks with growing input-dimension.

Theorem 6.2.1 is a special case of the next theorem, which will show that the previous conclusions are essentially independent of the network-topology. We will use the following symmetry assumption on the neural connections. For a given path p, let the *path-degree* d_p be the multiset of encountered in-degrees along path p. For a fully connected network, this is the unordered sequence of layer-sizes preceding the last path-node, including the input-layer. Now consider the multiset $\{d_p\}_{p\in\mathcal{P}(x,o)}$ of all path-degrees when p varies among all paths from input x to output o. The symmetry assumption (relatively to o) is

(S) All input nodes x have the same multiset $\{d_p\}_{p\in\mathcal{P}(x,o)}$ of path-degrees from x to o.

General Symmetry Assumption

Intuitively, this means that the statistics of degrees encountered along paths to the output are the same for all input nodes. This symmetry assumption is exactly satisfied by fully connected nets, almost satisfied by CNNs (up to boundary effects, which can be alleviated via periodic or mirror padding) and exactly satisfied by strided layers, if the layer size is a multiple of the stride.

Theorem 6.2.2. *Consider any feed-forward network with linear connections and ReLU activation functions. Assume the net satisfies assumptions (H) and outputs logits $f_k(x)$ that get fed to the cross-entropy-loss $\mathcal{L}$. Then $\|\partial_x f_k\|_2$ is independent of the input dimension d and $\epsilon_2 \|\partial_x \mathcal{L}\|_2 \propto \sqrt{d}$. Moreover, if the net satisfies the symmetry assumption (S), then $|\partial_x f_k| \propto 1/\sqrt{d}$ and (6.7) still holds: $\|\partial_x \mathcal{L}\|_q \propto d^{\frac{1}{q}-\frac{1}{2}}$ and $\epsilon_p \|\partial_x \mathcal{L}\|_q \propto \sqrt{d}$.*

Vulnerability of General Feedforward Nets

Theorems 6.2.1 and 6.2.2 are proven in Appendix C.5. The main proof idea is that in the gradient norm computation, the He-initialization exactly compensates the combinatorics of the number of paths in the network, so that this norm becomes independent of the network topology. In particular, we get

Corollary 6.2.3. *In any succession of convolution and dense layers, strided or not, with ReLU activations, that satisfies assumptions (H) and outputs logits that get fed to the cross-entropy-loss $\mathcal{L}$, the gradient of the logit-coordinates scale like $1/\sqrt{d}$ and (6.7) is satisfied. In particular, it is increasingly vulnerable with growing input-resolution to attacks generated with any ℓ_p-norm.*

Vulnerability of CNNs

Appendix B.3.1 shows that the network gradient are dampened when replacing strided layers by average poolings, essentially because average-pooling weights do not follow the He-init assumption H3.

6.3 Empirical Results

In Section 6.3.1, we empirically verify the validity of the first-order Taylor approximation made in (6.2) (Fig.6.1) and check the correspondence between loss-gradient regularization and adversarially-augmented training (Fig.6.2). Section 6.3.2 then empirically verifies that both the average ℓ_1-norm of $\partial_x\mathcal{L}$ and the adversarial vulnerability grow like $\sqrt{d}$ as predicted by Corollary 6.2.3. For all experiments, we approximate adversarial vulnerability using various attacks of the Foolbox-package [87]. We use an ℓ_∞ attack-threshold of size $\epsilon_\infty = 0.005$ which, for pixel-values ranging from 0 to 1, is completely imperceptible, but suffices to fool the classifiers on a significant proportion of examples. This ϵ_∞-*threshold* should not be confused with the *regularization-strengths* ϵ appearing in (6.4) and (6.5), which will be varied in some experiments.

[87] Rauber et al., *Foolbox v0.8.0*, 2017

6.3.1 First-Order Approximation, Gradient Penalty and Adversarial Augmentation

We train several CNNs with same architecture to classify CIFAR-10 images [60]. For each net, we use a specific training method with a specific regularization value ϵ. The training methods used were ℓ_1- and ℓ_2-penalization of $\partial_x\mathcal{L}$ (Eq. 6.4), adversarial augmentation with ℓ_∞- and ℓ_2- attacks (Eq. 6.5) and the cross-Lipschitz regularizer (Eq. B.4 in Appendix B.3.2). All networks have 6 'strided convolution $\to$ batchnorm $\to$ ReLU' layers with strides [1, 2, 2, 2, 2, 2] respectively and 64 output-channels each, followed by a final fully-connected lin-

[60] Krizhevsky, *Learning Multiple Layers of Features from Tiny Images*, 2009

Training Same Networks with Different Adversarial Regularizers

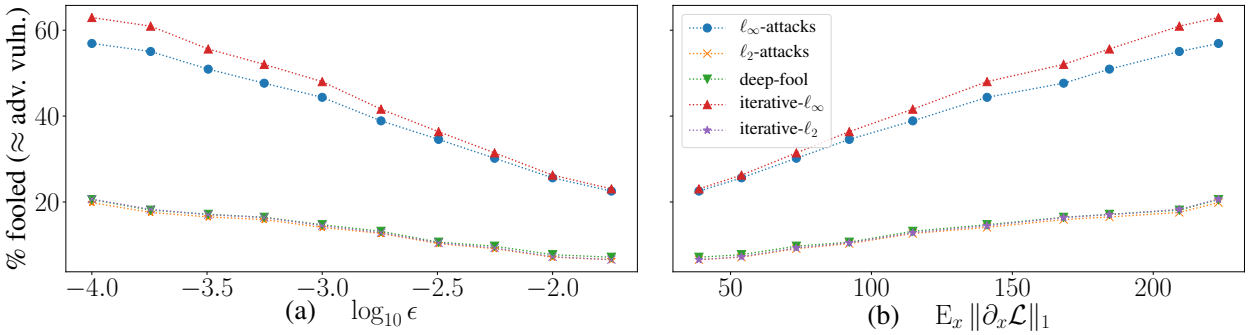


Figure 6.1: Adversarial vulnerability approximated by different attack-types for 10 trained networks as a function of (a) the ℓ_1 gradient regularization-strength ϵ used to train the nets and (b) the average gradient-norm. These curves confirm that the first-order expansion term in (6.2) is a crucial component of adversarial vulnerability.

ear layer. Results are summarized in Figures 6.1 and 6.2. Figure 6.1 fixes the training method – gradient ℓ_1-regularization – and plots the obtained adversarial vulnerabilities for various attacks types. Figure 6.2 fixes the attack type – iterative ℓ_1-attack – but plots the curves obtained for various training methods. Note that our goal here is not to advocate one defense over another, but rather to check the validity of the Taylor expansion, and empirically verify that first order terms (i.e., gradients) suffice to explain much of the observed adversarial vulnerability. Similarly, our goal in testing several attacks (Figure 6.1) is not to present a specifically strong one, but rather to verify that for all attacks, the trends are the same: the vulnerability grows with increasing gradients.

Validity of first order expansion. The efficiency of the first-order defense against iterative (non-firs-order) attacks (Fig.6.1a) strongly suggest that the first-order Taylor expansion in (6.2) is indeed a crucial component of adversarial vulnerability. This is further confirmed by the functional-like dependence between any approximation of adversarial vulnerability and $\mathbb{E}_x \|\partial_x \mathcal{L}\|_1$ (Fig.6.1b), and its independence on the training method (Fig.6.2d). Said differently, adversarial examples seem indeed to be primarily caused by large gradients of the classifier as captured via the induced loss. [2]

Illustration of Proposition 6.1.3. The upper row of Figure 6.2 plots $\mathbb{E}_x \|\partial_x \mathcal{L}_1\|$, adversarial vulnerability and accuracy as a function of $\epsilon\, d^{1/p}$. The excellent match between the adversarial augmentation curve with $p = \infty$ ($p = 2$) and its gradient-regularization dual counterpart with $q = 1$ (resp. $q = 2$) illustrates the duality between ϵ as a threshold for adversarially-augmented training and as a regularization constant in the regularized loss (Proposition 6.1.3). *It also supports the validity of the first-order Taylor expansion in* (6.2).

Confirmation of (6.3). Still on the upper row, the curves for $p = \infty, q = 1$ have no reason to match those for $p = 2, q = 2$ when plotted against ϵ, because ϵ is a threshold that is relative to a specific attack-

[2] On Figure 6.1, the two ℓ_∞-attacks seem more efficient than the others, because we chose an ℓ_∞ perturbation threshold (ϵ_∞). With an ℓ_2-threshold it is the opposite (see Figure B.4, Appendix B.3.4).

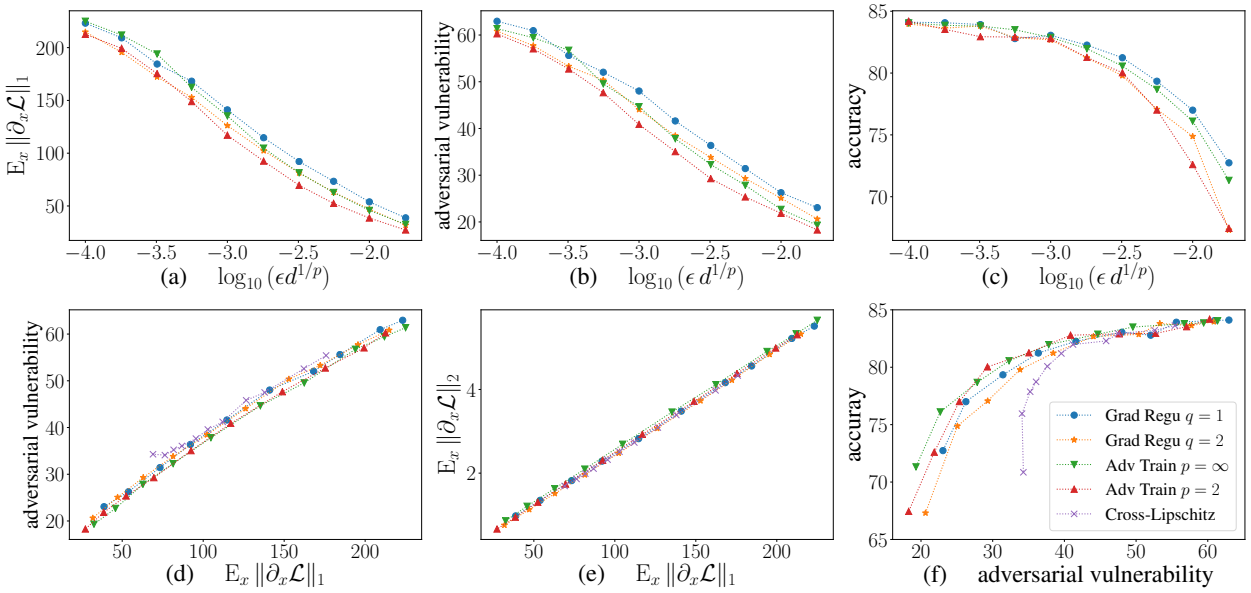


Figure 6.2: Average norm $\mathbb{E}_x \|\partial_x \mathcal{L}\|$ of the loss-gradients, adversarial vulnerability and accuracy (before attack) of various networks trained with different adversarial regularization methods and regularization strengths ϵ. Each point represents a trained network, and each curve a training-method. *Upper row*: A priori, the regularization-strengths ϵ have different meanings for each method. The near superposition of all upper-row curves illustrates (i) the duality between adversarial augmentation and gradient-regularization (Prop. 6.1.3) and (ii) confirms the rescaling of ϵ proposed in (6.3). (d): near functional relation between adversarial vulnerability and average loss-gradient norms. (e): the near-perfect linear relation between the $\mathbb{E}\|\partial_x \mathcal{L}\|_1$ and $\mathbb{E}\|\partial_x \mathcal{L}\|_2$ suggests that protecting against a given attack-norm also protects against others. (f): Merging 6.2band 6.2c shows that all adversarial augmentation and gradient-regularization methods achieve similar accuracy-vulnerability trade-offs.

norm. However, (6.3) suggested that the rescaled thresholds $\epsilon d^{1/p}$ may approximately correspond to a same 'threshold-unit' across ℓ_p-norms and across dimension. This is well confirmed by the upper row plots: by rescaling the x-axis, the $p = q = 2$ and $q = 1$,$p = \infty$ curves get almost super-imposed.

Accuracy-vs-vulnerability trade-off. Merging Figures 6.2b and 6.2c by taking out ϵ, Figure 6.2f shows that all gradient regularization and adversarial training methods yield equivalent accuracy-vulnerability trade-offs. For higher penalization values, these trade-offs are much better than those given by cross Lipschitz regularization.

The penalty-norm does not matter. We were surprised to see that on Figures 6.2d and 6.2f, the $\mathcal{L}_{\epsilon,q}$ curves are almost identical for $q = 1$ and 2. This indicates that both norms can be used interchangeably in (6.4) (modulo proper rescaling of ϵ via (6.3)), and suggests that protecting against a specific attack-norm also protects against others. (6.6) may provide an explanation: if the coordinates of $\partial_x \mathcal{L}$ behave like centered, uncorrelated variables with equal variance –which follows from assumptions ($\mathcal{H}$) –, then the ℓ_1- and ℓ_2-norms of $\partial_x \mathcal{L}$ are simply proportional. Plotting $\mathbb{E}_x \|\partial_x \mathcal{L}(x)\|_2$ against $\mathbb{E}_x \|\partial_x \mathcal{L}(x)\|_1$ in Figure 6.2e confirms this explanation. The slope is independent of the training method. Therefore, penalizing $\|\partial_x \mathcal{L}(x)\|_1$ during training will not only decrease $\mathbb{E}_x \|\partial_x \mathcal{L}\|_1$ (as shown in Figure 6.2a), but also drive down $\mathbb{E}_x \|\partial_x \mathcal{L}\|_2$ and vice-versa.

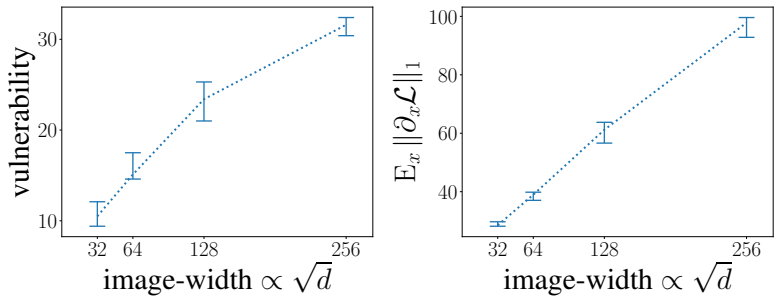


Figure 6.3: Both adversarial vulnerability (left) and $\mathbb{E}_x \|\partial_x \mathcal{L}\|_1$ (right) increase linearly with the square-root of the image-resolution d, as predicted by Corollary 6.2.3. Adversarial vulnerability gets slightly dampened at higher dimension, probably because the first-order approximation made in (6.2) becomes less and less valid.

6.3.2 Vulnerability Grows with Input Resolution

Theorems 6.2.1-6.2.2 and Corollary 6.2.3 predict a linear growth of the average ℓ_1-norm of $\partial_x \mathcal{L}$ with the square root of the input dimension d, and therefore also of adversarial vulnerability (Lemma 6.1.2). To test these predictions, we created a 12-class dataset of approximately $80,000$ $256 \times 256 \times 3$-sized RGB-images by merging similar ImageNet-classes [24], resizing the smallest image-edge to 256 pixels and center-cropping the result. We then downsized the images to 32, 64, 128 and 256 pixels per edge, and trained 10 CNNs on each of these downsized datasets. We then computed adversarial vulnerability (with iterative ℓ_∞-attacks) and average $\|\partial_x \mathcal{L}\|_1$ for each network on a same held-out test-dataset. Figure 6.3 summarizes the results. The dashed-line follows the median of each group of 10 networks; the errorbars show the 10^{th} and 90^{th} quantiles. As predicted by our theorems, both $\|\partial_x \mathcal{L}\|_1$ and adversarial vulnerability grow approximately linearly with $\sqrt{d}$. As the gradients get much larger at higher dimensions, the first order approximation in (6.2) becomes less and less valid, which may explain the little inflection of the adversarial vulnerability curve. For smaller ϵ-thresholds, we verified that the inflection disappears.

[24] Deng et al., *ImageNet: A Large-Scale Hierarchical Image Database*, 2009

Training Similar Nets but with Different Input Sizes

Gradients Indeed Increase as $\sqrt{d}$

All networks had exactly the same amount of parameters and very similar structure across the various input-resolutions. The CNNs were a succession of 8 'convolution $\to$ batchnorm $\to$ ReLU' layers with 64 output channels, followed by a final full-connection to the 12 logit-outputs. We used 2×2-max-poolings after layers 2,4 and 6, and a final max-pooling after layer 8 that fed only 1 neuron per channel to the fully-connected layer. To ensure that the convolution-kernels cover similar ranges of the images across each of the 32, 64, 128 and 256 input-resolutions, we respectively dilated all convolutions ('à trous') by a factor 1, 2, 4 and 8.

6.4 Chapter Conclusion

For differentiable classifiers and losses, we showed that adversarial vulnerability increases with the gradients $\partial_x \mathcal{L}$ of the loss, which is confirmed by the near-perfect functional relationship between gradient norms and vulnerability (Figures 6.1&6.2d). We then evaluated the size of $\|\partial_x \mathcal{L}\|_q$ and showed that usual feed-forward nets (convolutional or fully connected) are increasingly vulnerable to ℓ_p-attacks with growing input dimension d (the image-size), almost independently of their architecture. Our theorems rely on the statistical weight distribution at initialization, but our experiments confirm the conclusions also for the tested networks after training. Our results rely on a first order analysis that assumes a differentiable loss and architecture: they may not cover every aspect of adversarial vulnerability nor easily extend to non-differentiable structures (even though such structures are often smooth at a coarser scale). Nevertheless, they show that at least this type of first-order vulnerability is present, common, and firmly rooted in our current network architectures. They hence suggest to tackle adversarial vulnerability by designing new architectures (or new architectural building blocks) rather than by new regularization techniques.

These findings also question the quality of the network-based dissimilarity measures used in GAN- and VAE-style algorithms. Those algorithms indeed all rely on a network-based classifier (either in the form of a GAN discriminator or of a VAE encoder). GANs for example use this classifier to compute a dissimilarity measure between true and fake samples. This measure should quantify how realistic the generated samples looks like. But if even real samples can be altered by invisible perturbations that fool the classifier into being certain that they are fake, one may seriously question the reliability of these network-based distribution-dissimilarities. This chapter shows that if we want to overcome these pitfalls and get network-dissimilarities with human-like perception, we must at least seriously rethink our architectures.

Adv. Vulnerability: Mismatch btw. Human & Machine Perception

Related Literature

Goodfellow et al. [36] already stressed that adversarial vulnerability increases with growing dimension d. Their argument relies on a 'one-output-to-many-inputs'-model with dimension-independent weights. They therefore conclude on a linear growth of adversarial vulnerability with d and accuse our networks of being "too linear-like". Although this linear dependence becomes $\sqrt{d}$ when adjusting for a dimension-dependent weight-initialization, our theory and ex-

[36] Goodfellow et al., *Explaining and Harnessing Adversarial Examples*, 2015

periments nevertheless confirm this point of view, in the sense that a first-order Taylor expansion is indeed sufficient to explain the adversarial vulnerability of neural networks. As suggested by the one-output-to-many-inputs model, the culprit is that growing dimensionality gives the adversary more and more room to 'wriggle around' with the noise and adjust to the gradient of the output neuron. This wriggling, we show, is still possible when the output is connected to all inputs only indirectly, even when no neuron is directly connected to all inputs, like in CNNs. This explanation of adversarial vulnerability is independent of the *intrinsic* dimensionality or geometry of the data (compare to [3, 35]).

Incidentally, Goodfellow et al. [36] also already relate adversarial vulnerability to large gradients of the loss $\mathcal{L}$, an insight at the very heart of their FGSM-algorithm. They however do not propose any explicit penalizer on the gradient of $\mathcal{L}$ other than indirectly through adversarially-augmented training. Conversely, [89] propose the old double-backpropagation to robustify networks but make no connection to FGSM and adversarial augmentation. Lyu et al. [70] discuss and use the connection between gradient-penalties and adversarial augmentation, but never actually compare both in experiments. This comparison however is essential to test the validity of the first-order Taylor expansion in (6.2), as confirmed by the similarity between the gradient-regularization and adversarial-augmentation curves in Figure 6.2.Hein and Andriushchenko [48] derived yet another gradient-based penalty –the *cross-Lipschitz*-penalty– by considering (and proving) formal guarantees on adversarial vulnerability itself, rather than adversarial damage. While both penalties are similar in spirit, focusing on the adversarial damage rather than vulnerability has two main advantages. First, it achieves better accuracy-to-vulnerability ratios, both in theory and practice, because it ignores class-switches between misclassified examples and penalizes only those that reduce the accuracy. Second, it allows to deal with one number only, $\Delta\mathcal{L}_{0/1}$ or $\Delta\mathcal{L}$, whereas the cross-Lipschitz regularizer [48] and theoretical guarantees explicitly involve *all* K logit-functions (and their gradients). See Appendix B.3.2. Penalizing network-gradients is also at the heart of contractive auto-encoders as proposed in [88], where it is used to regularize the encoder-features. Seeing adversarial training as a generalization method, let us also mention Hochreiter and Schmidhuber [50], who propose to enhance generalization by searching for parameters in a "flat minimum region" of the loss. This leads to a penalty involving the gradient of the loss, but taken with respect to the weights, rather than the inputs. In the same vein, a gradient-regularization of the loss of generative models also appears in Proposition 6 of [84], where it stems from a code-length bound on the data (minimum description length). More generally, the gradient regu-

[3] Amsaleg et al., *The Vulnerability of Learning to Adversarial Perturbation Increases with Intrinsic Dimensionality*, 2017; [35] Gilmer et al., *Adversarial Spheres*, 2018

[36] Goodfellow et al., *Explaining and Harnessing Adversarial Examples*, 2015

[89] Ross and Doshi-Velez, *Improving the Adversarial Robustness and Interpretability of Deep Neural Networks by Regularizing Their Input Gradients*, 2018

[70] Lyu et al., *A Unified Gradient Regularization Family for Adversarial Examples*, 2015

[48] Hein and Andriushchenko, *Formal Guarantees on the Robustness of a Classifier against Adversarial Manipulation*, 2017

[48] Hein and Andriushchenko, *Formal Guarantees on the Robustness of a Classifier against Adversarial Manipulation*, 2017

[88] Rifai et al., *Contractive Auto-Encoders*, 2011

[50] Hochreiter and Schmidhuber, *Simplifying Neural Nets by Discovering Flat Minima*, 1995

[84] Ollivier, *Auto-Encoders: Reconstruction versus Compression*, 2014

larized objective (6.4) is essentially the first-order approximation of the robust training objective $\max_{\|\delta\| \leqslant \epsilon} \mathcal{L}(\mathbf{x} + \delta, c)$ which has a long history in math [122], machine learning [127] and now adversarial vulnerability [104]. Finally, [22] proposes new network-architectures that have small gradients by design, rather than by special training: an approach that makes all the more sense, considering the conclusion of Theorems 6.2.1 and 6.2.2. For further details and references on adversarial attacks and defenses, we refer to [128].

[22] Cisse et al., *Parseval Networks*, 2017

[128] Yuan et al., *Adversarial Examples*, 2017

Conclusion

Statistics traditionally use strong dissimilarities. They are handy for proofs, but not for applied work on samples. Practitioners therefore usually resort to unjustified tricks, such as smoothing the samples by adding Gaussian white noise. Instead, they could systematically use the "f-GAN-trick", that is, move from an f-divergence to a restricted f-divergence; from an unconstrained set of test function to a constrained one; from a strong dissimilarity to a weaker one.

Weakening Dissimilarities Only Helps Up to A Certain Point

This weakening often helps: bounding the Lipschitz norm of the test functions transforms the total variation metric into the bounded-Lipschitz one, which saturates no more on samples. But reducing the set of test functions too much can also make the dissimilarity blind to relevant changes. At the limit, if we keep only one single test function, all distributions would look the same. Hence the question: how much weakening is actually good?

Of course, it depends on our goals: there is no universal answer. However, we may set some prerequisites or constraints. We may for example want our dissimilarity to be perfectly discriminative, in the sense that it minimizes the distance between two distributions only if both are equal. We may also formulate these requirements as constraints on the topology generated by our dissimilarity, by asking that it be no weaker than some other weak topology. These are precisely the kind of prerequisites that we considered in our first chapter, on a particular kind of "weak total variation" dissimilarities: maximum mean discrepancies (MMD). Recall that both TV and MMDs can be written as

$$D_{\mathcal{F}}(P \,\|\, Q) = \sup_{\varphi \in \mathcal{F}} |P(\varphi) - Q(\varphi)|. \tag{7.1}$$

Perfectly Discriminative MMD Requires Big Enough RKHS

Only TV uses $\mathcal{F} = \mathcal{B}(\mathscr{C}_b)$, the unit ball of $\mathscr{C}_b$, while MMDs use a smaller set $\mathcal{F} = \mathcal{B}(\mathcal{H}_k)$, thereby weakening the distribution-dissimilarity. Now, the results we got in Chapter 1 clearly illustrate the idea that weakening a strong dissimilarity helps, but only up to a certain point. By reducing the set of test functions to an RKHS unit ball, we move from a dissimilarity that almost systematically saturates on samples, to one that is always well defined, easy to compute and never saturates there. But to stay perfectly discriminative and/or strong enough to metrize weak-convergence, we must ensure that our RKHS still be large enough to approximate any bounded, continuous function: that is the content of Theorem 1.2.2 on the equivalence of characteristic, universal and spd kernels, and of Theorem 1.3.4 on the metrization of weak convergence.

Choosing a characteristic kernel, i.e. a perfectly discriminative MMD, can be handy: Chapter 2 shows for example how characteristic kernels ensure the consistency of kernel mean estimators of functions of random variables, with applications to probabilistic

programming or privacy aware data releases. But requiring perfect discrimination between any two measures can also be an overkill. Goodness-of-fit tests and sample quality measures for example need to assess the dissimilarity only between some samples and a fixed, known target measure. There is no need to discriminate between all pairs of measures, because one of them, the target, is always fixed anyway. This led us to introduce the notion of *targeted* characteristic kernels and targeted convergence, which we then applied to kernel Stein discrepancies (KSD), a special kind of MMD. As a side remark, let us note that this application also gave us the first opportunity to illustrate the power of the KMEs of generalized measures, Schwartz-distribution: they can treat and embed the derivative of any measure, even discrete and empirical ones, which was key to our main theorem on the discriminative power of KSDs, Theorem 3.2.3.

Perfect Discrimination Can Be Too Much

Those insights in MMDs and KSDs were possible mostly because RKHSs have a very rigid structure that makes them easy to analyze: they are Hilbert spaces, meaning that they enjoy many properties of Euclidian spaces; they are completely defined by a single kernel function, meaning that many properties of the space and the MMD can be cast into properties of the kernel; their reproducing property simplifies many computations, in particular the MMD between two samples; and yet, despite all this rigidness, RKHSs are large enough to approximate any continuous function, which is key to a perfectly discriminative MMD. RKHSs hence offered a perfect opportunity to shape our intuitions and gain insights into weakened f-divergences, and weak total variations in particular.

RKHS Rigidity Eased MMD & KSD Analysis

In practice MMDs are still state-of-the-art for various settings and data types, such as independence or sample-quality tests or on some genomic, biological and discrete datasets. However, when Gaussian MMDs are used to train an image generator, as in Generative Moment Matching Networks [27, 66], they perform significantly worse than GANs. In both cases, a generator network minimizes a distribution-dissimilarity similar to (7.1). But instead of using an RKHS ball, GANs define the test functions $\mathcal{F}$ to be all functions attainable by a discriminative network. And on image date, GANs work better, even though Gaussian MMDs ensure both perfect discrimination and weak convergence, while GAN divergences don't. Understanding why could certainly make a whole thesis. However, it strongly suggests that perfect discrimination and weak convergence metrization, as theoretically pleasing as they may sound, are not so relevant for good empiric results. More relevant seems to be the architecture of those discriminators. That is not surprising, because this architecture profoundly shapes the properties of $\mathcal{F}$. Architectures that strongly rely on convolutional layers (CNNs) for instance ensure local translation and morphing invariances [72] that directly

[66] Li et al., *Generative Moment Matching Networks*, 2015; [27] Dziugaite et al., *Training Generative Neural Networks via MMD Optimization*, 2015

Network-Based Dissimilarities Perform Better on Image-Data

Network Architecture Encodes Dissimilarity-Invariances

[72] Mallat, *Understanding Deep Convolutional Networks*, 2016

propagate to $D_{\mathcal{F}}$. In a sense, those invariances are the opposite of perfect discrimination, the "holy grail" of our first part: they actively ensure *imperfect* discrimination between distributions that only differ by certain transformations. This imperfection does not harm GANs. Actually it helps: GANs with fully connected networks perform much worse than CNN-based GANs on image data. Once again, we are back to our original claim: weakening the distribution-dissimilarity can help. But now we see that this is even true when the dissimilarity is weakened beyond perfect discrimination.

Dissimilarity Invariances Better Than Perfect Discrimination

The way we like to think of GANs is that $\mathcal{F}$ or $D_{\mathcal{F}}$ represents our eye that tries to evaluate how similar the generated samples are to the original samples. If we do not want the generated samples to be a one-to-one copy of the original samples, we need to let the generator incorporate any change that we consider irrelevant. That is the role of the network invariances: they make our artificial eye, the dissimilarity $D_{\mathcal{F}}$, blind to those changes. Of course, all the art is to target specifically those irrelevant changes, without loosing discrimination power on other features; to reduce $\mathcal{F}$, but not too much; to weaken $D_{\mathcal{F}}$, but not too much. Once again, we are back to our central trade-off. That is what neural networks seem to be so good at: they can incorporate invariances and still keep a very expressive set of test functions.

But contrary to RKHSs, these sets of network-based test functions are much more complicated to analyze. While we may get some guarantees on the local invariances that we explicitly incorporate, it seems much more difficult to get theoretical guarantees on the global expressivity or capacity of $\mathcal{F}$. And as effective as it might be, switching to network-based test functions brings its own bunch of new problems. One of them is mode collapse, which we addressed with our algorithm AdaGAN in Chapter 5. Another is adversarial vulnerability, which illustrates how difficult it is in practice to incorporate all relevant invariances. Indeed, one reasonable requirement seems that if we want to produce realistically looking images, the dissimilarity $D_{\mathcal{F}}$ should be similar to our own, human perception dissimilarity. It should see differences when we do, and ignore those we don't. But adversarial vulnerability shows that this is far from being the case: two samples can look absolutely the same to us, and get two completely different classifications. We may be tempted to think that this vulnerability occurs only for a few particularly ill-chosen network architectures. But one of our startling conclusions in Chapter 6 is that this vulnerability is not specific to one or the other network architecture: it concerns almost all our usual feed-forward architectures. To design an accurate and robust classifier it hence won't suffice to slightly fiddle existing architectures: if such robust networks are to be found, we will likely need to incorporate at least

Network-Dissimilarity Issues: Mode-Collapse & Adv. Vuln.

Adversarial Vulnerability Not Architecture Specific

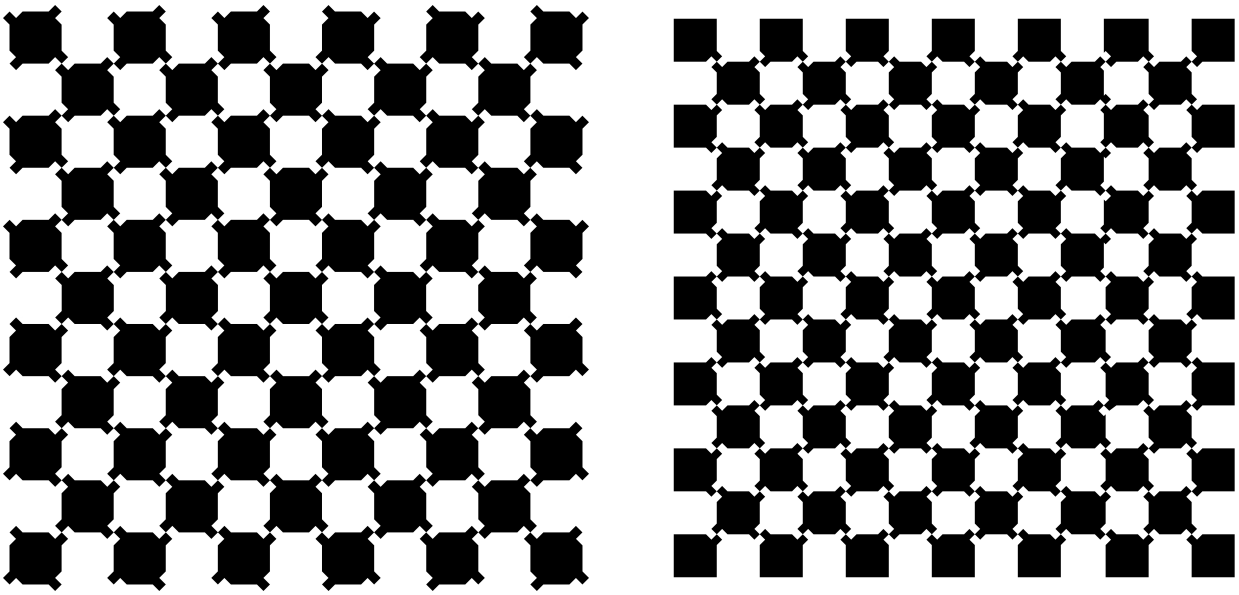


Figure 7.1: *Turtles* and *Cultured Turtles*: two optical illusions by [58]. The straight lines are parallel. But tilted edges suffice to make them appear curved. Human adversarial examples?

one truly new building block. Of course, one major open question is whether humans themselves are adversarially robust. So far, adversarial examples that fool our neural networks do not fool humans. That shows that these classifiers do not use the same features than us and that the associated dissimilarities do not yet capture human perception accurately. But it does not prove that no small perturbations can fool humans. On the contrary, recent research by Kitaoka on optical illusions [57] actually hints that humans are also prone to adversarial perturbations. See in particular the startling optical illusions of Figure 7.1, taken from [58]. Therefore, if we do not find both accurate and adversarially robust classifiers soon, we may want in future to turn to the question: is it actually possible to robustly classify the kind of data that we have? Future work might tell us. And even if it turned out to be impossible, it would not hinder us from trying to build a classifier that has a human-like perception; with its weaknesses and with its strengths.

Humans Too Are Adversarially Vulnerable

Adversarial Vulnerability Unavoidable?

[57] Kitaoka, *Tilt Illusions after Oyama (1960): A Review*, 2007

[58] Kitaoka, *Akiyoshi's Illusion Page: Illusion of Frindged Edges*, 2018

Appendix

A

Background Material

A.1 Schwartz-Distributions

To introduce Schwartz-distributions, the first step is to notice that any continuous function f is uniquely characterized by the values taken by $f(\varphi) := \int \varphi(\mathbf{x}) f(\mathbf{x})\, d\mathbf{x}$ when φ goes through $\mathscr{C}_c$. Rather than seeing f as a function that acts on points $\mathbf{x}$ in $\mathcal{X}$, we could thus equivalently see f as a linear functional that acts on other functions φ in $\mathscr{C}_c$ and takes its values in $\mathbb{C}$. Such functionals are called *linear forms*. We could do the same for measures: a signed measure μ is also characterized by the values of $\mu(\varphi) := \int \varphi(\mathbf{x})\, d\mu(\mathbf{x})$. So we could also see it as a linear functional that acts on functions φ in $\mathscr{C}_c$. Doing so effectively identifies f with the signed measure μ_f that has density f, because both define the same linear form $\varphi \mapsto \int \varphi(\mathbf{x}) f(\mathbf{x})\, d\mathbf{x}$. So from this perspective, a function f becomes a particular kind of measure, and a measure μ a sort of 'generalized function'. Moreover, seen as linear forms over $\mathscr{C}_c$, f and μ are continuous in the sense that if φ_α converges to φ, then $\mu(\varphi_\alpha)$ converges to $\mu(\varphi)$. Thus, by definition, we just identified f and μ with elements of the dual of $\mathscr{C}_c$.

We may now ask whether there are other continuous linear forms over $\mathscr{C}_c$. The answer is negative and is given by the Riesz-Markov-Kakutani representer theorem (see Appendix A.2). It states that the dual of $\mathscr{C}_c$ is exactly the set of signed regular Borel measures $\mathcal{M}_r$, meaning that any continuous linear form over $\mathscr{C}_c$ can be written as $\varphi \mapsto \int \varphi\, d\mu(\mathbf{x})$ for some $\mu \in \mathcal{M}_r$, and can thus be identified with a measure μ. So it seems that our generalization of functions to measures using continuous linear forms is as general as it can get. But this is forgetting the following detail. To distinguish a measure μ from all the others in $\mathcal{M}_r$, we do not need to know the values $\mu(\varphi)$ for *all* functions φ of $\mathscr{C}_c$. Actually, it suffices to know them for all φ in $\mathscr{C}_c^\infty$. This is because $\mathscr{C}_c^\infty$ is a dense subset of $\mathscr{C}_c$. Thus for any $\varphi \in \mathscr{C}_c$, even if $\varphi \notin \mathscr{C}_c^\infty$, we can reconstruct the value $\mu(\varphi)$ by taking a sequence φ_α in $\mathscr{C}_c^\infty$ that converges to φ and noticing that, by continuity, $\mu(\varphi)$ is the limit of $\mu(\varphi_\alpha)$. So instead of seeing a function or a measure as an element of $(\mathscr{C}_c)'$, we could also see it as an element of $(\mathscr{C}_c^\infty)'$.

But do we gain anything from it? Yes indeed, because now, we can define linear functionals over $\mathscr{C}_c^\infty$ that we could not define over

$\mathscr{C}_c$. For example, suppose that $\mathcal{X} = \mathbb{R}$ and consider the linear form d_x that, to each function φ associates its derivative $\partial\varphi(x)$ evaluated at x. This is a valid (continuous) linear form over $\mathscr{C}_c^\infty$ – called a dipole in x – but it cannot be defined over $\mathscr{C}_c$, because not all continuous functions are differentiable. This example shows that, although each measure in $(\mathscr{C}_c)'$ can be seen as an element of $(\mathscr{C}_c^\infty)'$, the latter space contains many more linear forms which do not correspond to a signed measure. This bigger set of linear forms, which we denote $\mathcal{D}^\infty$, is called the set of Schwartz-distributions.

Now, why are distributions useful? First of all, because they can all be seen as limits of functions [97][Thm XV, Chap III]. As an example, consider the sequence of functions

[97] Schwartz, *Théorie des Distributions*, 1978

$$f_\sigma : \quad x \quad \longmapsto \quad \tfrac{1}{\sigma} g(\tfrac{x+\sigma}{\sigma}) - \tfrac{1}{\sigma} g(\tfrac{x-\sigma}{\sigma}) \ ,$$

where g is a Gaussian. f_σ is the difference of two Gaussians that get closer and closer and more and more peaked with decreasing σ. Now, applying f_σ to a function $\varphi \in \mathscr{C}_c^\infty$, it is not difficult to see that $f_\sigma(\varphi)$ converges to $\partial\varphi(0) = d_0(\varphi)$ when $\sigma \to 0$. The dipole d_0 can thus be seen as a weak limit of the functions f_σ, although it is itself neither a function nor even a signed measure. See Figure A.1 for plots of f_σ and its KMEs.

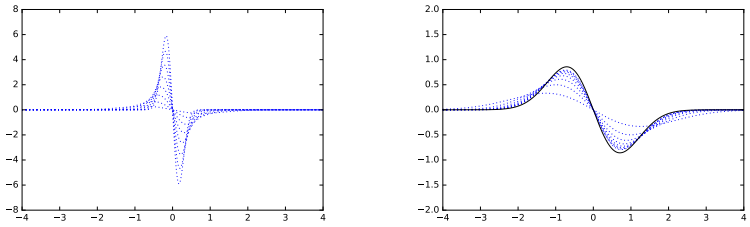


Figure A.1: Left: the difference f_σ of two Gaussians that get closer and closer and more and more peaked with decreasing σ. Right: the KMEs of f_σ. Note the difference in the y-axis scale. f_σ converges to a *dipole*, which is not a measure, but a Schwartz-distribution. It cannot be represented as a function, but its KME can (black solid line). Note that the KMEs of f_σ seem to converge to the KME of the dipole.

Another reason to use distributions is that many common linear operations can be extended to them (or to big subsets of them), such as differentiation, Fourier transformation and convolution. Let us show for example how to extend differentiation. If we want the distributional derivative ∂ to be an extension of the usual derivative, then of course we should require that $\partial\mu_f = \mu_{f'}$ whenever f is a continuously differentiable function over $\mathcal{X} = \mathbb{R}$. Now, by integration by part, we get, for any $\varphi \in \mathscr{C}_c^\infty$:

$$\mu_{f'}(\varphi) = \int f'\varphi = -\int f\varphi' = -\mu_f(\varphi') \ .$$

This suggests to define the derivative of any $D \in \mathcal{D}^\infty$ as $\partial^p D(\varphi) := (-1)^{|p|} D(\partial^p \varphi)$ for any $\varphi \in \mathscr{C}_c^\infty$. Doing so, we just defined a notion of differentiation that is compatible with the usual differentiation and makes *any* distribution infinitely many times differentiable. In

particular, any function and any measure is infinitely differentiable in this distributional sense. Moreover, if a sequence of differentiable functions f_n converges to a distribution D (in the sense that $f_n(\varphi)$ converges to $D(\varphi)$ for any φ), then f_n' converges to ∂D (where we used $'$ to denote the usual differentiation). All this makes distributions of prime interest for solving linear differential equations and more generally for physicists. Note that, by construction, if Q is a probability measure with smooth density q, then $\partial^p Q$ is the signed measure with density $\partial^p q$.

A.2 Topological Vector Spaces

Let us start with the definition of a *barreled* set. In a normed space $(\mathcal{E}, \|.\|)$, the sets $T := \{f \in \mathcal{E} \mid \|f\| \leqslant C\}$ where $C > 0$ are the closed balls centered on the origin of $\mathcal{E}$. A normed space is a particular case of a topological vector space (TVS). In a general locally convex (loc. cv.) TVS $\mathcal{E}$, the topology might not be given by a single norm, but by a family of semi-norms $(\|.\|_\alpha)_{\alpha\in\mathcal{I}}$ (where the index set $\mathcal{I}$ can be uncountable). A so-called *barrel* of $\mathcal{E}$ is then any closed ball centered on the origin and associated to a norm $\|.\|_\alpha$, $\alpha \in \mathcal{I}$. More abstractly, a barrel can be defined as follows.

Definition A.2.1 (Barrel)**.** *A subset* T *of a TVS* $\mathcal{E}$ *is called a* barrel *if it is*

(i) absorbing*: for any* $f \in \mathcal{E}$*, there exists* $c_f > 0$ *such that* $f \in c_f T$*;*
(ii) balanced*: for any* $f \in \mathcal{E}$*, if* $f \in T$ *then* $\lambda f \in T$ *for any* $\lambda \in \mathbb{C}$ *with* $|\lambda| \leqslant 1$ *;*
(iii) convex *;*
(iv) closed*.*

In any loc. cv. space, there exists a basis of neighborhoods of the origin consisting only of barrels. However, in general, there may be barrels that are not a neighborhood of 0. This leads to

Definition A.2.2 (Barrelled spaces)**.** *A TVS is* barreled *if any barrel is a neighborhood of the origin.*

Although many authors include local convexity in the definition, in general, a barreled space need not be loc. cv. Barreled spaces were introduced by Bourbaki, because they were well-suited for the following generalization of the celebrated *Banach-Steinhaus* theorem.

Theorem A.2.3 (Banach-Steinhaus)**.** *Let* $\mathcal{E}$ *be a barreled TVS,* $\mathcal{F}$ *be a loc. cv. TVS, and let* $L(\mathcal{E}, \mathcal{F})$ *be the set of continuous linear maps form* $\mathcal{E}$ *to* $\mathcal{F}$*. For any* $H \subset L(\mathcal{E}, \mathcal{F})$ *the following properties are equivalent:*

(i) H *is equicontinuous.*
(ii) H *is bounded for the topology of pointwise convergence.*
(iii) H *is bounded for the topology of bounded convergence.*

When $\mathcal{E}$ is a normed space and $\mathcal{F} = \mathbb{C}$, then $L(\mathcal{E}, \mathcal{F})$ is by definition $\mathcal{E}'$. With $\|.\|_{\mathcal{E}'}$ being the dual norm in $\mathcal{E}'$, the equivalence of (ii) and (iii) states that

$$\forall f \in \mathcal{E},\ \sup_{h\in H} |h(f)| < \infty \quad \Longrightarrow \quad \sup_{h\in H} \|h\|_{\mathcal{E}'} < \infty\,.$$

Obviously, to understand the content of the Banach-Steinhaus theorem, one needs the definition of a bounded set. Let us define them now.

When $\mathcal{E}$ is a normed space, then a subset B of $\mathcal{E}$ is called *bounded* if $\sup_{f\in B}\|f\|_{\mathcal{E}} < \infty$. In a more general loc. cv. TVS $\mathcal{E}$, where the topology is given by a family of semi-norms $(\|.\|_\alpha)_{\alpha\in\mathcal{I}}$, a subset B of $\mathcal{E}$ is called *bounded* if, for any $\alpha \in \mathcal{I}$, $\sup_{f\in B}\|f\|_\alpha < \infty$. This can be shown equivalent to the following, more usual definition.

Definition A.2.4 (Bounded Sets in a TVS)**.** *A subset B of a TVS $\mathcal{E}$ is* bounded*, if, for any neighborhood $U \subset \mathcal{E}$ of the origin, there exists a real $c_B > 0$ such that $B \subset c_B U$.*

Note that the notion of boundedness depends on the underlying topology. By default, a bounded set of some dual space $\mathcal{E} = \mathcal{F}'$ designates a set that is bounded for the strong dual topology. We now move on to an unrelated topic: the Riesz Representation theorem for Hilbert spaces. Most of Chapter 1 relies on this one theorem.

Theorem A.2.5 (Riesz Representation Theorem for Hilbert Spaces)**.** *A Hilbert space $\mathcal{H}$ and its topological dual $\mathcal{H}'$ are isometrically (anti-) isomorphic via the Riesz representer map*

$$\begin{array}{rccc} \iota: & \mathcal{H} & \longrightarrow & \mathcal{H}' \\ & f & \longmapsto & D_f := \left\{ \begin{array}{ccc} \mathcal{H} & \longrightarrow & \mathbb{C} \\ g & \longmapsto & \langle g\,,\, f\rangle \end{array} \right. \end{array} .$$

In particular, for any continuous linear form $D \in \mathcal{H}'$, there exists a unique element $f \in \mathcal{H}$, called the Riesz representer of D*, such that*

$$\forall g \in \mathcal{H}, \qquad D(g) = \langle g\,,\, f\rangle\ .$$

Note that "anti" in "anti-isomorphic" simply means that, instead of being linear, ι is anti-linear: for any $\lambda \in \mathbb{C}$ and $f \in \mathcal{H}$, $\iota(\lambda f) = \bar{\lambda}\,\iota(f)$. Often, we prefer to say that $\mathcal{H}$ is isometrically isomorphic to $\overline{\mathcal{H}}'$, where $\overline{\mathcal{H}}'$ denotes the conjugate of $\mathcal{H}$, where the scalar multiplication is replaced by $(\lambda, f) \mapsto \bar{\lambda} f$. $\mathcal{H}'_k$ and $\overline{\mathcal{H}_k}'$ are obviously isomorphic via the complex conjugation map $D \mapsto \bar{D}$.

The Riesz representation theorem for Hilbert spaces is not to be confounded with the following theorem, also known as the Riesz — or Riesz-Markov-Kakutani— representation theorem. In Chapter 1, we always refer to the latter as the Riesz-Markov-Kakutani representation theorem. This theorem has numerous variants, depending on which dual pair $(\mathcal{E}, \mathcal{E}')$ one uses. Here we state it for $\mathcal{E} = \mathscr{C}_{\to 0}$.

Theorem A.2.6 (Riesz-Markov-Kakutani). *Let $\mathcal{X}$ be a locally compact Hausdorff space. The spaces $\mathcal{M}_f(\mathcal{X})$ and $(\mathscr{C}_{\to 0}(\mathcal{X}))'$ are isomorphic, both algebraically and topologically via the map*

$$\begin{array}{rccc} \iota: & \mathcal{M}_f(\mathcal{X}) & \longrightarrow & (\mathscr{C}_{\to 0}(\mathcal{X}))' \\ & \mu & \longmapsto & D_\mu := \left\{ \begin{array}{ccc} \mathscr{C}_{\to 0} & \longrightarrow & \mathbb{C} \\ \varphi & \longmapsto & \int \varphi \, d\mu \end{array} \right. \end{array} .$$

In other words, for any continuous linear form D over $\mathscr{C}_{\to 0}(\mathcal{X})$, there exists a unique finite Borel measure $\mu \in \mathcal{M}_f$ such that, for any test function $\varphi \in \mathscr{C}_{\to 0}(\mathcal{X})$, $D(\varphi) = \int \varphi \, d\mu$. Moreover, $\sup_{\|\varphi\|_\infty \leqslant 1} D(\varphi) = |\mu|(\mathcal{X})$, or in short: $\|D\|_{(\mathscr{C}_{\to 0})'} = \|\mu\|_{TV}$, where $\|\mu\|_{TV}$ denotes the total variation norm of μ. This is why, in Chapter 1, we identify $\mathcal{M}_f$ —a space of σ-additive set functions— with $\mathcal{M}_f$ —a space of linear functionals.

In Chapter 1, to embed a space of measures into an RKHS $\mathcal{H}_k$ we successively apply both Riesz representation theorems: If $\mathcal{H}_k$ embeds continuously into $\mathscr{C}_{\to 0}$, then $(\mathscr{C}_{\to 0})'$ embeds continuously into $\overline{\mathcal{H}_k}'$, via the embedding map Φ_k. But $(\mathscr{C}_{\to 0})' = \mathcal{M}_f$ (Riesz-Markov-Kakutani Representation) and $\overline{\mathcal{H}_k}' = \mathcal{H}_k$ (Riesz Representation). Thus Φ_k may also be seen as an embedding of $\mathcal{M}_f$ into $\mathcal{H}_k$.

B

Details and Chapter Complements

B.1 Chapter 3

B.1.1 Statement and Proof of Theorem 2.2 of [21]

[21] Chwialkowski et al., *A Kernel Test of Goodness of Fit*, 2016

Theorem B.1.1 (Chwialkowski et al. [21]). *Let* k *be a* c_0*-universal kernel and let* P, Q *be two measures with differentiable densities* p, q*, such that* $\mathbb{E}_{x,x' \sim Q}[\kappa(x, x')] < \infty$ *and* $\mathbb{E}_Q\left[\partial_x \log \frac{p(x)}{q(x)}\right] < \infty$*. Then* $\mathrm{KSD}_{k,P}(P)$ *is well-defined and* $\mathrm{KSD}_{k,P}(P) = 0$ *if and only if* $P = Q$*.*

Proof. The authors introduce the functional: $\xi_P(x) := s_P(x)k(x, .) + \partial_x k(x, \cdot)$, and show that the Stein discrepancy $\mathrm{KSD}_{k,P}(Q) = \left\|\mathbb{E}_Q[\xi_P]\right\|_{\mathcal{H}^d}$. Decomposing $\log p$ into $\log q + \log p - \log q$, they then write

$$\begin{aligned}
\mathop{\mathbb{E}}_{Q}[\xi_P] &= \int \Big(k(x, \cdot)s_P(x) + \partial_x k(x, \cdot)\Big)\, dQ(x) \\
&= \int \Big(k(x, \cdot)s_Q(x) + \partial_x k(x, \cdot)\Big)\, dQ(x) \\
&+ \int k(x, \cdot)\big(s_P(x) - s_Q(x)\big)Q(x)\, dx \\
&= \mathop{\mathbb{E}}_{Q}\big[\xi_Q\big] + \int k(x, \cdot)\, \partial_x \Big(\log \frac{p(x)}{q(x)}\Big)\, dx\,.
\end{aligned}$$

They then show that $\mathbb{E}_Q\big[\xi_Q\big] = 0$ and note that the second term is the embedding of the function $g(x) := \partial_x \log \frac{p(x)}{q(x)}$. And because they assume that the kernel k is c_0-universal, $\mathrm{KSD}_{k,P}(Q) = 0$ if and only if each component of $g(x)$ is 0, which implies $Q = P$. □

B.2 Chapter 5

B.2.1 Algorithms

Algorithm B.1: Determining λ^*

Sort the values $h(d_i)$ in increasing order
Initialize $\lambda \leftarrow \frac{\beta}{p_1}\left(1 + \frac{1-\beta}{\beta} p_1 h(d_1)\right)$ and $k \leftarrow 1$
while $(1-\beta)h(d_k) \geqslant \lambda$ **do**
  $k \leftarrow k+1$
  $\lambda \leftarrow \frac{\beta}{\sum_{i=1}^{k} p_i}\left(1 + \frac{(1-\beta)}{\beta}\sum_{i=1}^{k} p_i h(d_i)\right)$
end while

Algorithm B.2: AdaGAN, a meta-algorithm to construct a "strong" mixture of T individual GANs, trained sequentially. The mixture weight schedule ChooseMixtureWeight should be provided by the user (see Sec 5.2). This is an instance of the high level Algorithm 5, instantiating UpdateTrainingWeights.

Input: Training sample $S_N := \{X_1, \ldots, X_N\}$.
Output: Mixture generative model $G = G_T$.
Train vanilla GAN: $G_1 = \text{GAN}(S_N)$
for $t = 2, \ldots, T$ **do**
  #Choose a mixture weight for the next component
  $\beta_t = \text{ChooseMixtureWeight}(t)$
  #Compute the new weights of the training examples (UpdateTrainingWeights)
  #Compute the discriminator between the original (unweighted) data and the current mixture G_{t-1}
  $D \leftarrow \text{DGAN}(S_N, G_{t-1})$
  #Compute λ^* *using Algorithm B.1*
  $\lambda^* \leftarrow \lambda(\beta_t, D)$
  #Compute the new weight for each example
  for $i = 1, \ldots, N$ **do**
    $W_t^i = \frac{1}{N\beta_t}\left(\lambda^* - (1-\beta_t)h(D(X_i))\right)_+$
  end for
  #Train t*-th "weak" component generator* G_t^c
  $G_t^c = \text{GAN}(S_N, W_t)$
  #Update the overall generative model
  #Notation below means forming a mixture of G_{t-1} *and* G_t^c.
  $G_t = (1-\beta_t)G_{t-1} + \beta_t G_t^c$
end for

B.2.2 Details on the Toy Experiments

GAN architectures. In all our experiments, the GAN's generator uses the latent space $\mathcal{Z} = \mathbb{R}^5$, and two ReLU hidden layers, of size 10

	Modes : 1	Modes : 2	Modes : 3	Modes : 5	Modes : 10
Vanilla	0.97 (0.9; 1.0)	0.88 (0.4; 1.0)	0.63 (0.5; 1.0)	0.72 (0.5; 0.8)	0.59 (0.2; 0.7)
Best of T (T=3)	0.99 (1.0; 1.0)	0.96 (0.9; 1.0)	0.91 (0.7; 1.0)	0.80 (0.7; 0.9)	0.70 (0.6; 0.8)
Best of T (T=10)	0.99 (1.0; 1.0)	0.99 (1.0; 1.0)	0.98 (0.8; 1.0)	0.80 (0.8; 0.9)	0.71 (0.7; 0.8)
Ensemble (T=3)	0.99 (1.0; 1.0)	0.98 (0.9; 1.0)	0.93 (0.8; 1.0)	0.78 (0.6; 1.0)	0.80 (0.6; 1.0)
Ensemble (T=10)	1.00 (1.0; 1.0)	0.99 (1.0; 1.0)	1.00 (1.0; 1.0)	0.91 (0.8; 1.0)	0.89 (0.7; 1.0)
TopKLast0.5 (T=3)	0.98 (0.9; 1.0)	0.98 (0.9; 1.0)	0.95 (0.9; 1.0)	0.95 (0.8; 1.0)	0.86 (0.6; 0.9)
TopKLast0.5 (T=10)	0.99 (1.0; 1.0)	0.98 (0.9; 1.0)	0.98 (1.0; 1.0)	0.99 (0.8; 1.0)	1.00 (0.8; 1.0)
Boosted (T=3)	0.99 (1.0; 1.0)	0.99 (0.9; 1.0)	0.98 (0.9; 1.0)	0.91 (0.8; 1.0)	0.86 (0.7; 1.0)
Boosted (T=10)	1.00 (1.0; 1.0)	1.00 (1.0; 1.0)	1.00 (1.0; 1.0)	1.00 (1.0; 1.0)	1.00 (1.0; 1.0)

Table B.1: Performance of the different algorithms on varying number of mixtures of Gaussians. The reported score is the coverage C, probability mass of P_X covered by the $5th$ percentile of P_Y defined in Section 5.3. The reported scores are the median and interval defined by the 5% and 95% percentile (in parenthesis) (see Section 5.3), over 35 runs for each setting. Both the ensemble and the boosting approaches significantly outperform the vanilla GAN even with just three iterations (i.e. just two additional components). The boosting approach converges faster to the optimal coverage and with smaller variance.

and 5 respectively. The corresponding discriminator has two ReLU hidden layers of size 20 and 10 respectively. We use 64k training examples, and 15 epochs, which is enough compared to the small scale of the problem. The optimizer is a simple SGD: Adam was also tried but gave slightly less stable results. All networks converge properly and overfitting is never an issue.

Details on the tested algorithms and more tests. In our experiments, we compared the following algorithms:

- ▷ The baseline GAN algorithm, called **Vanilla GAN** in the results.

(a) The best model out of T runs of GAN, that is: run T GAN instances independently, then take the run that performs best on a validation set. This gives an additional baseline with similar computational complexity as the ensemble approaches. Note that the selection of the best run is done on the reported target metric (see below), rather than on the internal metric. As a result this baseline is slightly overestimated. This procedure is called **Best of T** in the results.

(b) A mixture of T GAN generators, trained independently, and combined with equal weights (the "bagging" approach). This procedure is called **Ensemble** in the results.

- ▷ A mixture of GAN generators, trained sequentially with different choices of data re-weighting:

(c) The AdaGAN algorithm (Algorithm 5), with $\beta = 1/t$. Thus each component will have the same weight in the resulting mixture (see Section 5.2). This procedure is called **Boosted** in the results.

- ▷ The AdaGAN algorithm (Algorithm 5), but with a constant β, exploring several values. This procedure is called for example **Beta0.3** for $\beta = 0.3$ in the results. Note that in this setting, not all components of the mixture have the same weight.
- ▷ Reweighting similar to "Cascade GAN" from [123], i.e. keeping the top r fraction of examples, based on the discriminator corresponding to the *previous* generator. This procedure is called for example **TopKLast0.3** for $r = 0.3$.
- ▷ Keep the top r fraction of examples, based on the discriminator corresponding to *the mixture of all previous* generators. This procedure is called for example **TopK0.3** for $r = 0.3$.

[123] Wang et al., *Ensembles of GANs*, 2016

The left, middle, and right panels in Figure 5.2 of Section 5.3 respectively correspond to the settings (a), (b) and (c).

Experiments with unrolled GAN. To illustrate the 'meta-algorithm aspect' of AdaGAN, we also performed experiments with an unrolled GAN (UGAN) [75] instead of a GAN as the base generator. We trained the GANs both with the Jensen-Shannon objective (4.2), and with its modified version proposed in [37] (and often considered as the baseline GAN), where $\log(1 - D(G(Z)))$ is replaced by $-\log(D(G(Z)))$. We use the same network architecture as in the other toy experiments. Figure B.1 illustrates our results. We find that AdaGAN works with all UGAN algorithms. Note that, where the usual GAN updates the generator and the discriminator once, an UGAN with 5 unrolling steps updates the generator once and the discriminator 1 + 5, i.e. 6 times (and then rolls back 5 steps). Thus, in terms of computation time, training 1 single UGAN roughly corresponds to doing 3 steps of AdaGAN with a usual GAN. In that sense, Figure B.1 shows that AdaGAN (with a usual GAN) significantly outperforms a single unrolled GAN ($T = 1$ on bottom pictures). Also note that AdaGAN ran with UGAN outperforms a single UGAN and keeps improving its performance as we increase the number of iterations. Additionally, we note that using the Jensen-Shannon objective (rather than the modified version) seems to have some mode-regularizing effect.

[75] Metz et al., *Unrolled GANs*, 2017

[37] Goodfellow et al., *Generative Adversarial Nets*, 2014

B.2.3 Details for AdaGAN on MNIST

GAN architecture. We ran AdaGAN on MNIST (28x28 pixel images) using (de)convolutional networks with batch normalizations

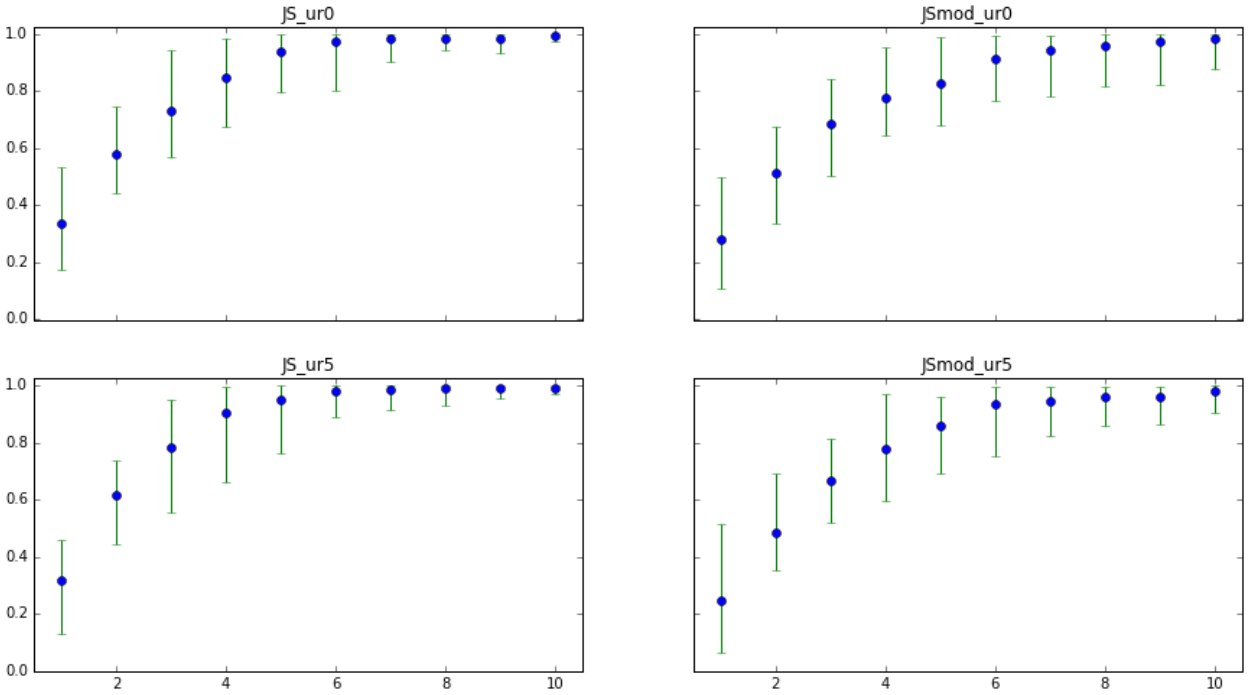


Figure B.1: Comparison of AdaGAN ran with a GAN (top row) and with an unrolled GAN with 5 unrolling steps [75] (bottom). Coverage C of the true data by the model distribution P^T_{model}, as a function of iterations T. Experiments are similar to those of Figure 5.2, but with 10 modes. Left figures used the Jensen-Shannon objective (4.2), while right figures used the modified objective originally proposed by [37]. In terms of computation time, one step of AdaGAN with unrolled GAN corresponds to roughly 3 steps of AdaGAN with a usual GAN.

and leaky ReLUs. The latent space has dimension 100. We used the following architectures:

Generator:100 x 1 x 1 –> FC –> 7 x 7 x 16 –> deconv –>
14 x 14 x 8 –> deconv–> 28 x 28 x 4 –> deconv –> 28 x 28 x 1

Discriminator:28 x 28 x 1 –> conv –> 14 x 14 x 16 –> conv –>
7 x 7 x 32–> FC –> 1

where each arrow consists of a leaky ReLU (with 0.3 leak) followed by a batch normalization, conv and deconv are convolutions and transposed convolutions with 5x5 filters, and fully connected (FC) are linear layers with bias. The distribution over $\mathcal{Z}$ is uniform over the unit box. We use the Adam optimizer with $\beta_1 = 0.5$, with 2 G steps for 1 D step and learning rates 0.005 for G, 0.001 for D, and 0.0001 for the classifier C that does the reweighting of digits. We optimized D and G over 200 epochs and C over 5 epochs, using the original Jensen-Shannon objective (4.2), without the log trick, with no unrolling and with minibatches of size 128.

Empirical observations. Although we could not find any appropriate metric to measure the increase of diversity promoted by AdaGAN, we observed that the re-weighting scheme indeed focuses on digits with very specific strokes. In Figure B.2 for example, we see that after 1 AdaGAN step, the generator produces overly thick digits (top left image). Thus AdaGAN puts small weights on the thick digits of the dataset (bottom left) and high weights on the thin ones (bottom right). After the next step, the new GAN produces both thick and thin digits.

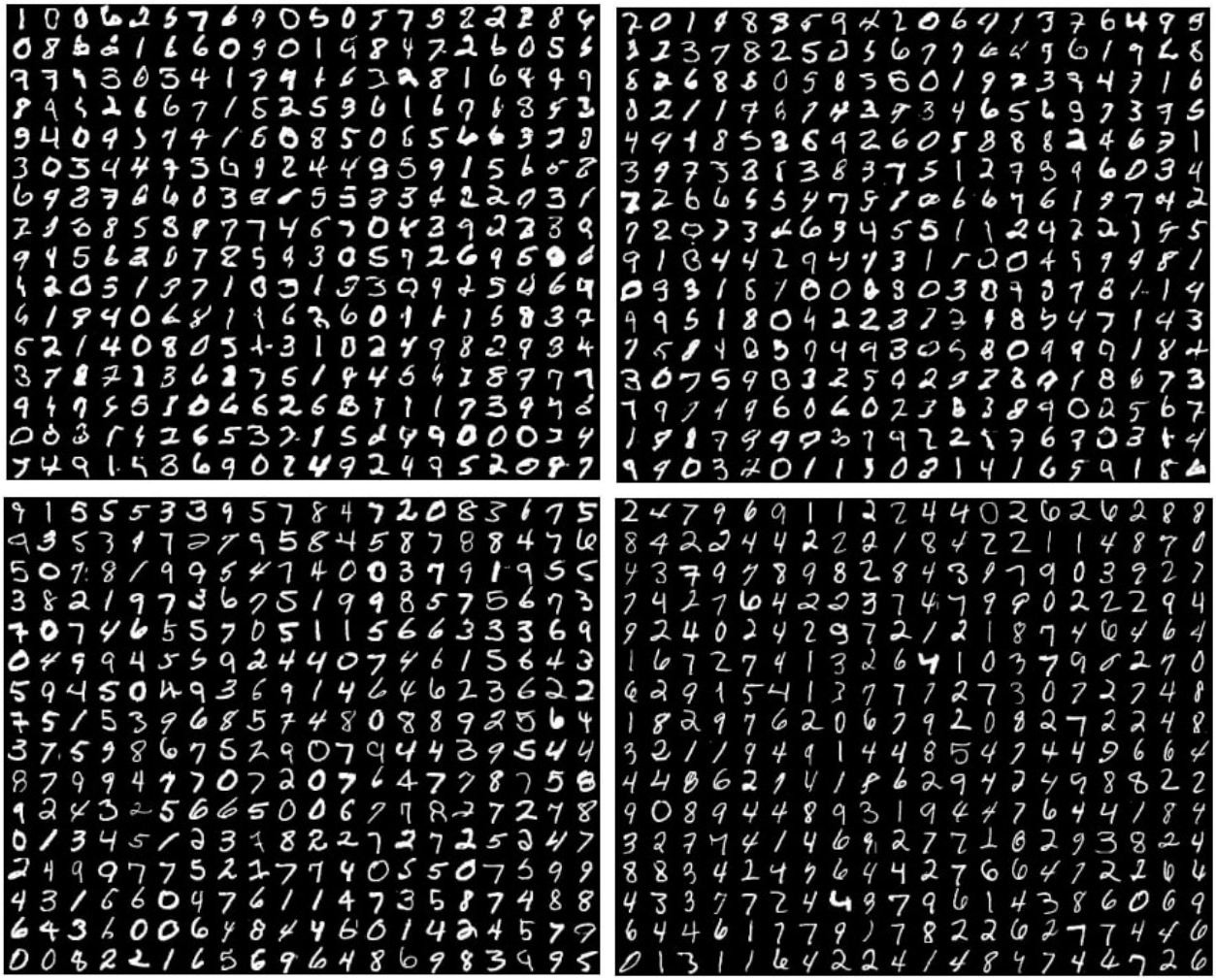


Figure B.2: AdaGAN on MNIST. Bottom row are true MNIST digits with smallest (left) and highest (right) weights after re-weighting at the end of the first AdaGAN step. Those with small weight are thick and resemble those generated by the GAN after the first AdaGAN step (top left). After training with the re-weighted dataset during the second iteration of AdaGAN, the new mixture produces more thin digits (top right).

B.2.4 Refinement of Lemma 5.1.4

If the ratio dP_Y/dP_X is almost surely bounded, the first inequality of Lemma 5.1.4 can be refined as follows.

Lemma B.2.1. *Under the conditions of Theorem 5.1.2*

$$D_f\left((1-\beta)P_Y + \beta R^*_\beta \,\middle\|\, P_X\right) \leqslant f(\lambda^*) + \frac{f(M)(1-\lambda^*)}{M-1}$$

given there exists $M > 1$ *such that* $P_X((1-\beta)dP_Y > M dP_X) = 0$.

This upper bound can be tighter than that of Lemma 5.1.4 when λ^* gets close to 1. Indeed, for $\lambda^* = 1$ the upper bound is exactly 0 and is thus tight, while the upper bound of Lemma 5.1.4 will not be zero in this case.

Proof. We use Inequality (C.25) of Lemma C.4.2 with $X = \beta$, $Y = (1-\beta)dP_Y/dP_X$, and $c = \lambda^*$. We easily verify that $X + Y = ((1-$

$\beta)dP_Y + \beta dP_X)/dP_X$ and $\max(c, Y) = ((1-\beta)dP_Y + \beta\, dR^*_\beta)/dP_X$ and both have expectation 1 with respect to P_X. We thus obtain:

$$D_f\left((1-\beta)P_Y + \beta R^*_\beta \,\middle\|\, P_X\right) \leqslant f(\lambda^*) + \frac{f(M) - f(\lambda^*)}{M - \lambda^*}(1-\lambda^*). \quad \text{(B.1)}$$

Since $\lambda^* \leqslant 1$ and f is non-increasing on $(0, 1)$ we get

$$D_f\left((1-\beta)P_Y + \beta R^*_\beta \,\middle\|\, P_X\right) \leqslant f(\lambda^*) + \frac{f(M)(1-\lambda^*)}{M-1}. \qquad \square$$

B.2.5 Conditions for finite steps convergence

Here we study the convergence of (5.3) to 0 in the case where, while performing the iterations, we use the upper bound (5.5) and the weight β is fixed (i.e. the same value at each iteration). We will provide necessary and sufficient conditions for the iterative process to converge to the data distribution P_X in finite number of steps. The analysis can easily be extended to a non-constant (variable) weight scheduling β. We start with the following result.

Lemma B.2.2. *For any $f \in \mathcal{F}_{\mathrm{div}}$ such that $f(x) \neq 0$ for $x \neq 1$, the following conditions are equivalent:*

(i) $P_X((1-\beta)dP_Y > dP_X) = 0$;
(ii) $D_f\left((1-\beta)P_Y + \beta R^*_\beta \,\middle\|\, P_X\right) = 0$.

Proof. The first condition is equivalent to $\lambda^* = 1$ according to Theorem 5.1.2. In this case, $(1-\beta)P_Y + \beta R^*_\beta = P_X$, hence the divergence is 0. In the other direction, when the divergence is 0, since f is strictly positive for $x \neq 1$ (keep in mind that we can always replace f by f_0 to get a non-negative function which will be strictly positive if $f(x) \neq 0$ for $x \neq 1$), this means that with P_X probability 1 we have the equality $dP_X = (1-\beta)dP_Y + \beta\, dR^*_\beta$, which implies that $(1-\beta)dP_Y > dP_X$ with P_X probability 1 and also $\lambda^* = 1$. $\square$

This result tells that we cannot perfectly match P_X by adding a new mixture component to P_Y as long as there are points in the space where our current model P_Y severely over-samples. As an example, consider an extreme case where P_Y puts a positive mass in a region outside of the support of P_X. Clearly, unless $\beta = 1$, we will not be able to match P_X.

We now provide the conditions for the convergence of the iterative process in a finite number of steps. The criterion is based on the ratio dP_1/dP_X, where P_1 is the first component of our mixture model.

Corollary B.2.3. *Take any $f \in \mathcal{F}_{\mathrm{div}}$ such that $f(x) \neq 0$ for $x \neq 1$. Starting from $P^1_{\mathrm{model}} = P_1$, update the model iteratively according to $P^{t+1}_{\mathrm{model}} = (1-\beta)P^t_{\mathrm{model}} + \beta R^*_\beta$, where on every step R^*_β is as defined in Theorem*

5.1.2 with $P_Y := P^t_{model}$. *In this case* $D_f\left(P^t_{model} \middle\| P_X\right)$ *will reach* 0 *in a finite number of steps if and only if there exists* $M > 0$ *such that*

$$P_X((1-\beta)dP_1 > M dP_X) = 0\,. \tag{B.2}$$

When the finite convergence happens, it takes at most $-\ln \max(M,1)/\ln(1-\beta)$ *steps.*

Proof. From Lemma B.2.2, it is clear that if $M \leqslant 1$ the convergence happens after the first update. So let us assume $M > 1$. Notice that $dP^{t+1}_{model} = (1-\beta)dP^t_{model} + \beta\, dR^*_\beta = \max(\lambda^* dP_X, (1-\beta)dP^t_{model})$ so that if $P_X((1-\beta)dP^t_{model} > M dP_X) = 0$, then $P_X((1-\beta)dP^{t+1}_{model} > M(1-\beta)dP_X) = 0$. This proves that (B.2) is a sufficient condition.

Now assume the process converged in a finite number of steps. Let P^t_{model} be a mixture right before the final step. Note that P^t_{model} is represented by $(1-\beta)^{t-1}P_1 + (1-(1-\beta)^{t-1})P$ for certain probability distribution P. According to Lemma B.2.2 we have $P_X((1-\beta)dP^t_{model} > dP_X) = 0$. Together these two facts immediately imply (B.2). □

It is also important to keep in mind that even if (B.2) is not satisfied the process still converges to the true distribution at exponential rate (see Lemma 5.1.4 as well as Corollaries 5.1.5 and 5.1.6 below)

B.3 Chapter 6

B.3.1 Effects of Strided and Average-Pooling Layers on Adversarial Vulnerability

It is common practice in CNNs to use average-pooling layers or strided convolutions to progressively decrease the number of pixels per channel. Corollary 6.2.3 shows that using strided convolutions does not protect against adversarial examples. However, what if we replace strided convolutions by convolutions with stride 1 plus an average-pooling layer? Theorem 6.2.2 considers only *randomly* initialized weights with typical size $1/\sqrt{\text{in-degree}}$. Average-poolings however introduce *deterministic* weights of size $1/(\text{in-degree})$. These are smaller and may therefore dampen the input-to-output gradients and protect against adversarial examples. We confirm this in our next theorem, which uses a slightly modified version ($\mathcal{H}'$) of ($\mathcal{H}$) to allow average pooling layers. ($\mathcal{H}'$) is ($\mathcal{H}$), but where the He-init H3 applies to all weights *except* the (deterministic) average pooling weights, and where H1 places a ReLU on every non-input *and non-average-pooling* neuron.

Theorem B.3.1 (Effect of Average-Poolings). *Consider a succession of convolution layers, dense layers and n average-pooling layers, in any order, that satisfies ($\mathcal{H}'$) and outputs logits $f_k(x)$. Assume the n average pooling layers have a stride equal to their mask size and perform averages over a_1, ..., a_n nodes respectively. Then $\|\partial_x f_k\|_2$ and $|\partial_x f_k|$ scale like $1/\sqrt{a_1 \cdots a_n}$ and $1/\sqrt{d\, a_1 \cdots a_n}$ respectively.*

Proof in Appendix C.5.4. Theorem B.3.1 suggest to try and replace any strided convolution by its non-strided counterpart, followed by an average-pooling layer. It also shows that if we systematically reduce the number of pixels per channel down to 1 by using only non-strided convolutions and average-pooling layers (i.e. $d = \prod_{i=1}^{n} a_i$), then all input-to-output gradients should become independent of d, thereby making the network completely robust to adversarial exam-

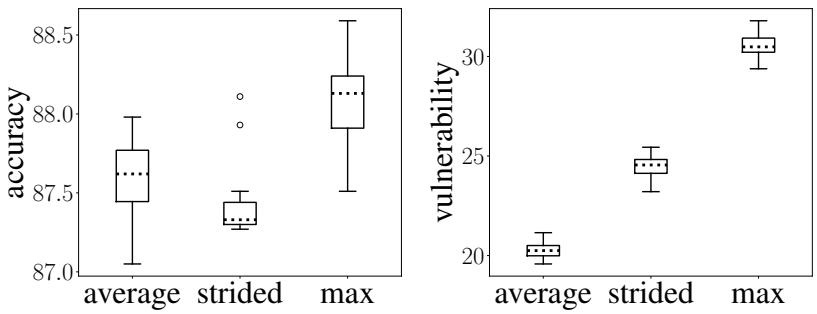


Figure B.3: As predicted by Theorem B.3.1, average-pooling layers make networks more robust to adversarial examples, contrary to strided (and max-pooling) ones. But the vulnerability with average-poolings remains higher than anticipated.

ples. Our following experiments (Figure B.3) show that after training, the networks get indeed robustified to adversarial examples, but remain more vulnerable than suggested by Theorem B.3.1.

Experimental setup. Theorem B.3.1 shows that, contrary to strided layers, average-poolings should decrease adversarial vulnerability. We tested this hypothesis on CNNs trained on CIFAR-10, with 6 blocks of 'convolution $\to$ BatchNorm $\to$ReLU' with 64 output-channels, followed by a final average pooling feeding one neuron per channel to the last fully-connected linear layer. Additionally, after every second convolution, we placed a pooling layer with stride and mask-size $(2,2)$ (thus acting on 2×2 neurons at a time, without overlap). We tested average-pooling, strided and max-pooling layers and trained 20 networks per architecture. Results are shown in Figure B.3. All accuracies are very close, but, as predicted, the networks with average pooling layers are more robust to adversarial images than the others. However, they remain more vulnerable than what would follow from Theorem B.3.1. We also noticed that, contrary to the strided architectures, their gradients after training are an order of magnitude higher than at initialization and than predicted. This suggests that assumptions $(\mathcal{H})$ get more violated when using average-poolings instead of strided layers. Understanding why will need further investigations.

B.3.2 Comparison to the Cross-Lipschitz Regularizer

In their Theorem 2.1, Hein and Andriushchenko [48] show that the minimal $\epsilon = \|\delta\|_p$ perturbation to fool the classifier must be bigger than:

[48] Hein and Andriushchenko, *Formal Guarantees on the Robustness of a Classifier against Adversarial Manipulation*, 2017

$$\min_{k\neq c} \frac{f_c(x) - f_k(x)}{\max_{y\in B(x,\epsilon)} \|\partial_x f_c(y) - \partial_x f_k(y)\|_q}\,. \tag{B.3}$$

They argue that the training procedure typically already tries to maximize $f_c(x) - f_k(x)$, thus one only needs to additionally ensure that $\|\partial_x f_c(x) - \partial_x f_k(x)\|_q$ is small. They then introduce what they call a Cross-Lipschitz Regularization, which corresponds to the case $p = 2$ and involves the gradient differences between *all* classes:

$$\mathcal{R}_{\text{xLip}} := \frac{1}{K^2}\sum_{k,h=1}^{K} \|\partial_x f_h(x) - \partial_x f_k(x)\|_2^2 \tag{B.4}$$

In contrast, using (C.29), (the square of) our proposed regularizer $\|\partial_x \mathcal{L}\|_q$ from (6.4) can be rewritten, for $p = q = 2$ as:

$$\mathcal{R}_{\|\cdot\|_2}(f) = \sum_{k,h=1}^{K} q_k(x) q_h(x) \left(\partial_x f_c(x) - \partial_x f_k(x)\right) \cdot \\ \cdot \left(\partial_x f_c(x) - \partial_x f_h(x)\right) \quad \text{(B.5)}$$

Although both (B.4) and (B.5) consist in K^2 terms, corresponding to the K^2 cross-interaction between the K classes, the big difference is that while in (B.4) all classes play exactly the same role, in (B.5) the summands all refer to the target class c in at least two different ways. First, all gradient differences are always taken with respect to $\partial_x f_c$. Second, each summand is weighted by the probabilities $q_k(x)$ and $q_h(x)$ of the two involved classes, meaning that only the classes with a non-negligible probability get their gradient regularized. This reflects the idea that only points near the margin need a gradient regularization, which incidentally will make the margin sharper.

B.3.3 Perception Threshold

To keep the average pixel-wise variation constant across dimensions d, we saw in (6.3) that the threshold ϵ_p of an ℓ_p-attack should scale like $d^{1/p}$. We will now see another justification for this scaling. Contrary to the rest of this work, where we use a fixed ϵ_p for all images x, here we will let ϵ_p depend on the ℓ_2-norm of x. If, as usual, the dataset is normalized such that the pixels have on average variance 1, both approaches are almost equivalent.

Suppose that given an ℓ_p-attack norm, we want to choose ϵ_p such that the signal-to-noise ratio (SNR) $\|x\|_2 / \|\delta\|_2$ of a perturbation δ with ℓ_p-norm $\leqslant \epsilon_p$ is never greater than a given SNR threshold $1/\epsilon$. For $p = 2$ this imposes $\epsilon_2 = \epsilon \|x\|_2$. More generally, studying the inclusion of ℓ_p-balls in ℓ_2-balls yields

$$\epsilon_p = \epsilon \|x\|_2 \, d^{1/p - 1/2} . \quad \text{(B.6)}$$

Note that this gives again $\epsilon_p = \epsilon_\infty d^{1/p}$. This explains how to adjust the threshold ϵ with varying ℓ_p-attack norm.

Now, let us see how to adjust the threshold of a given ℓ_p-norm when the dimension d varies. Suppose that x is a natural image and that decreasing its dimension means either decreasing its resolution or cropping it. Because the statistics of natural images are approximately resolution and scale invariant [52], in either case the average

[52] Huang, *Statistics of Natural Images and Models*, 2000

squared value of the image pixels remains unchanged, which implies that $\|\mathbf{x}\|_2$ scales like $\sqrt{d}$. Pasting this back into (B.6), we again get:

$$\epsilon_p = \epsilon_\infty \, d^{1/p} \, .$$

In particular, $\epsilon_\infty \propto \epsilon$ is a dimension-free number, exactly like in (6.3) of the main part.

Now, why did we choose the SNR as our invariant reference quantity and not anything else? One reason is that it corresponds to a physical power ratio between the image and the perturbation, which we think the human eye is sensible to. Of course, the eye's sensitivity also depends on the spectral frequency of the signals involved, but we are only interested in orders of magnitude here.

Another point: any image $\mathbf{x}$ yields an adversarial perturbation $\boldsymbol{\delta}_\mathbf{x}$, where by constraint $\|\mathbf{x}\|_2 \,/\, \|\boldsymbol{\delta}_\mathbf{x}\| \leqslant 1/\epsilon$. For ℓ_2-attacks, this inequality is actually an equality. But what about other ℓ_p-attacks: (on average over $\mathbf{x}$,) how far is the signal-to-noise ratio from its imposed upper bound $1/\epsilon$? For $p \notin \{1, 2, \infty\}$, the answer unfortunately depends on the pixel-statistics of the images. But when p is 1 or ∞, then the situation is locally the same as for $p = 2$. Specifically:

Lemma B.3.2. *Let $\mathbf{x}$ be a given input and $\epsilon > 0$. Let ϵ_p be the greatest threshold such that for any $\boldsymbol{\delta}$ with $\|\boldsymbol{\delta}\|_p \leqslant \epsilon_p$, the SNR $\|\mathbf{x}\|_2 \,/\, \|\boldsymbol{\delta}\|_2$ is $\leqslant 1/\epsilon$. Then $\epsilon_p = \epsilon \, \|\mathbf{x}\|_2 \, d^{1/p-1/2}$.*

Moreover, for $p \in \{1, 2, \infty\}$, if $\boldsymbol{\delta}_\mathbf{x}$ is the ϵ_p-sized ℓ_p-attack that locally maximizes the loss-increase i.e. $\boldsymbol{\delta}_\mathbf{x} = \arg\max_{\|\boldsymbol{\delta}\|_p \leqslant \epsilon_p} |\partial_\mathbf{x} \mathcal{L} \cdot \boldsymbol{\delta}|$, then:

$$\mathrm{SNR}(\mathbf{x}) := \frac{\|\mathbf{x}\|_2}{\|\boldsymbol{\delta}_\mathbf{x}\|_2} = \frac{1}{\epsilon} \quad \textit{and} \quad \mathop{\mathbb{E}}_{\mathbf{x}}[\mathrm{SNR}(\mathbf{x})] = \frac{1}{\epsilon} \, .$$

Proof. The first paragraph follows from the fact that the greatest ℓ_p-ball included in an ℓ_2-ball of radius $\epsilon \, \|\mathbf{x}\|_2$ has radius $\epsilon \, \|\mathbf{x}\|_2 \, d^{1/p-1/2}$.

The second paragraph is clear for $p = 2$. For $p = \infty$, it follows from the fact that $\boldsymbol{\delta}_\mathbf{x} = \epsilon_\infty \operatorname{sign} \partial_\mathbf{x} \mathcal{L}$ which satisfies: $\|\boldsymbol{\delta}_\mathbf{x}\|_2 = \epsilon_\infty \sqrt{d} = \epsilon \, \|\mathbf{x}\|_2$. For $p = 1$, it is because $\boldsymbol{\delta}_\mathbf{x} = \epsilon_1 \max_{i=1..d} |(\partial_\mathbf{x} \mathcal{L})_i|$, which satisfies: $\|\boldsymbol{\delta}_\mathbf{x}\|_2 = \epsilon_2 / \sqrt{d} = \epsilon \, \|\mathbf{x}\|_2$. □

Intuitively, this means that for $p \in \{1, 2, \infty\}$, the SNR of ϵ_p-sized ℓ_p-attacks on any input $\mathbf{x}$ will be exactly equal to its fixed upper limit $1/\epsilon$. And in particular, the mean SNR over samples $\mathbf{x}$ is the same $(1/\epsilon)$ in all three cases.

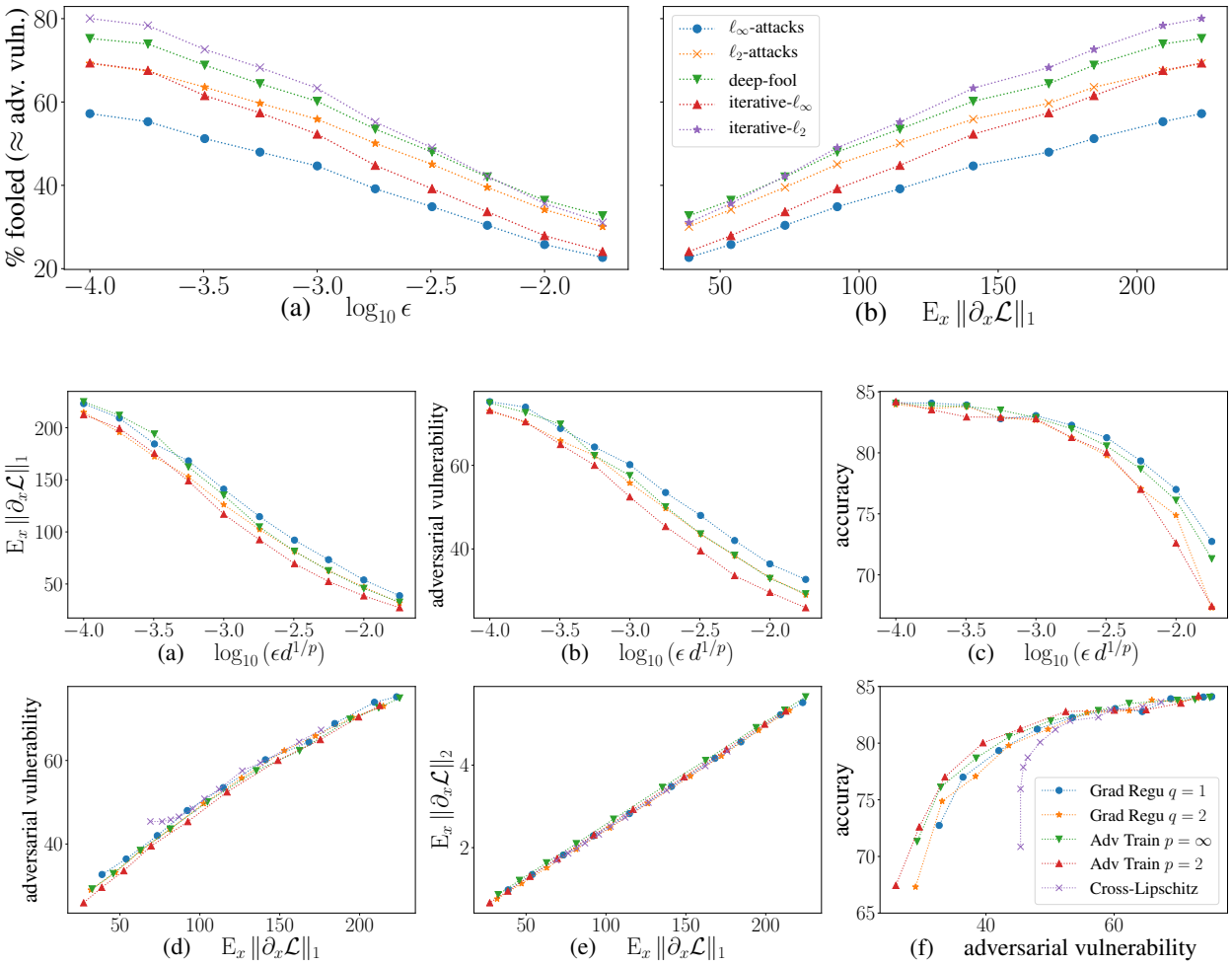


Figure B.4: Same as Figure 6.1 but using an ℓ_2 threshold instead of a ℓ_∞ one. Now the ℓ_2-based methods (deep-fool, and single-step and iterative ℓ_2-attacks) seem more effective than the ℓ_∞ ones.

B.3.4 Figures with an ℓ_2 Perturbation-Threshold and Deep-Fool Attacks

Here we plot the same curves as in the main part, but using an ℓ_2-attack threshold of size $\epsilon_2 = 0.005\sqrt{d}$ instead of the ℓ_∞-threshold, and deep-fool attacks [78] instead of iterative ℓ_∞-ones in Figs. B.5 and B.6. Note that contrary to ℓ_∞-thresholds, ℓ_2-thresholds must be rescaled by $\sqrt{d}$ to stay consistent across dimensions (see Eq.6.3 and Appendix B.3.3). All curves look essentially the same as their counterparts in the main text.

Figure B.5: Same as Figure 6.2, but with an ℓ_2- perturbation-threshold (instead of ℓ_∞) and deep-fool attacks [78] instead of iterative ℓ_∞ ones. All curves look essentially the same than in Fig. 6.2.

[78] Moosavi-Dezfooli et al., *DeepFool*, 2016

B.3.5 A Variant of Adversarially-Augmented Training

In usual adversarially-augmented training, the adversarial image $x + \delta$ is generated on the fly, but is nevertheless treated as a fixed input of the neural net, which means that the gradient does not get backpropagated through δ. This need not be. As δ is itself a function of x, the gradients could actually also be backpropagated through δ.

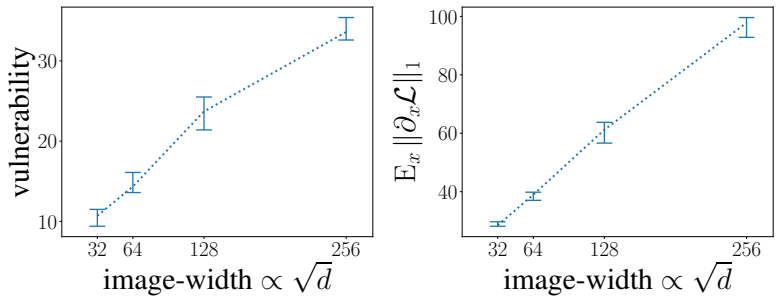


Figure B.6: Same as Figure 6.3 but with an ℓ_2 perturbation-threshold (instead of an ℓ_∞ one) and using deep-fool (instead of iterative-ℓ_∞) attacks to approximate adversarial vulnerability.

As it was only a one-line change of our code, we used this opportunity to test this variant of adversarial training (FGSM-variant in Figure 6.2) and thank Martín Arjovsky for suggesting it. But except for an increased computation time, we found no significant difference compared to usual augmented training.

C

Proofs

C.1 Chapter 1

In this section, we gather all the complements to non fully proved theorems, propositions, corollaries or lemmas appearing in the main text. We start with a Lemma that essentially follows from [113], and which we will need a few times for the proofs.

[113] Steinwart and Christmann, *Support Vector Machines*, 2008, Cor 4.36

Lemma C.1.1. *Let* $k \in \mathscr{C}_b^{(m,m)}$ *and let* $\Phi : \mathcal{X} \to \mathcal{H}_k$, $x \mapsto k(.,x)$. *Then for any* $p \in \mathbb{N}^d$ *with* $|p| \leqslant m$, *the partial derivative* $\partial^p \Phi$ *exists, belongs to* $\mathcal{H}_k$, *is continuous and verifies* $\partial^p \Phi(x) = \partial^{(0,p)} k(.,x)$. *Moreover, for any* $f \in \mathcal{H}_k$, $\partial^p f$ *exists, belongs to* $\mathcal{H}_k$ *and verifies:*

$$\partial^p f(x) = \left\langle f\,,\, \partial^{(0,p)} k(.,x) \right\rangle_k \,. \tag{C.1}$$

Applied with $f = \partial^{(0,q)} k(.,y)$ *where* $|q| \leqslant m$ *also proves that*

$$\partial^{(p,q)} k(x,y) = \left\langle \partial^{(0,q)} k(.,y)\,,\, \partial^{(0,p)} k(.,x) \right\rangle_k \,. \tag{C.2}$$

Proof. This Lemma is essentially proven in Corollary 4.36 and in its proof in [113]. We only added Equation (C.2), which is a straightforward consequence of (C.1), and the part stating that $\partial^p \Phi(x) = \partial^{(0,p)} k(.,x)$. This can be shown as follows. Steinwart and Christmann [113] prove that $\partial^p \Phi$ exists and belongs to $\mathcal{H}_k$. Thus

[113] Steinwart and Christmann, *Support Vector Machines*, 2008

[113] Steinwart and Christmann, *Support Vector Machines*, 2008

$$\begin{aligned}
[\partial^p \Phi(x)](y) &= \langle \partial^p \Phi(x)\,,\, k(.,y) \rangle_k \\
&= \left\langle \lim_{h \to 0} (\Phi(x + h e_i) - \Phi(x))/h\,,\, k(.,y) \right\rangle_k \\
&= \lim_{h \to 0} (k(y, x + h e_i) - k(y,x))/h \\
&= \partial^{(0,p)} k(y,x)\,,
\end{aligned}$$

where we used the continuity of the inner product to swap limit and bracket signs. □

C.1.1 Proof of Corollary 1.1.3

Proof. Suppose that $\mathcal{H}_k \subset \mathscr{C}_{\to 0}$. (i) clearly holds. Suppose (ii) was not met. Then let $x_n \in \mathcal{X}$ such that $k(x_n, x_n) = \|k(.,x_n)\|_k^2 \to \infty$. Thus $k(.,x_n)$ is unbounded. But $\langle f\,,\, k(.,x_n) \rangle_k = f(x_n)$ is bounded

for any $f \in \mathcal{H}_k$, thus $k(.,x_n)$ is bounded (Banach-Steinhaus, see Thm. A.2.3). Contradiction. Thus (ii) is met.

Conversely, suppose that (i) and (ii) hold. Let $\mathcal{H}_k^{\text{pre}} := \text{span}\{k(.,x) \,|\, x \in \mathcal{X}\}$. Then, $\mathcal{H}_k^{\text{pre}} \subset \mathscr{C}_{\to 0}$, and for any $f, g \in \mathcal{H}_k$, $\|f-g\|_\infty \leqslant \|f-g\|_k \, \|k\|_\infty$. Thus $\mathcal{H}_k^{\text{pre}}$ continuously embeds into the *closed* $\mathscr{C}_{\to 0}$, thus so does its $\|.\|_k$-closure, $\mathcal{H}_k$. The proof of the cases $\mathcal{H}_k \subset \mathscr{C}$ and $\mathcal{H}_k \subset \mathscr{C}_b$ are similar (see also [10]). □

[10] Berlinet and Thomas-Agnan, *Reproducing Kernel Hilbert Spaces in Probability and Statistics*, 2004, Thm. 17

C.1.2 Proof of Corollary 1.1.4

Proof. Suppose that $k \in \mathscr{C}_b^{(m,m)}$. Then $\mathcal{H}_k^{\text{pre}} \subset \mathscr{C}_b^m$ [113] and for any $x \in \mathcal{X}$, $f \in \mathcal{H}_k^{\text{pre}}$, and $|p| \leqslant m$, we have $\|\partial^p f\|_\infty \leqslant \|f\|_k \left\| \sqrt{\partial^{(p,p)} k} \right\|_\infty$. Thus $\mathcal{H}_k^{\text{pre}}$ continuously embeds into the *closed* space $\mathscr{C}_b^m$, thus so does its $\|.\|_k$-closure, $\mathcal{H}_k$. But, by definition of $(\mathscr{C}_b^m)_c$ is the space $\mathscr{C}_b$ equipped with a weaker topology (see Section 1.2), thus $\mathscr{C}_b^m \hookrightarrow (\mathscr{C}_b^m)_c$. Thus $\mathcal{H}_k \hookrightarrow (\mathscr{C}_b^m)_c$, which concludes. The proofs when $k \in \mathscr{C}$ or $k \in \mathscr{C}_{\to 0}$ are similar. □

[113] Steinwart and Christmann, *Support Vector Machines*, 2008, Corollary 4.36

C.1.3 Proof of Theorem 1.2.4

Proof. Equivalence between (i) & (ii). As KMEs are linear over $\mathcal{M}_f$, a kernel k is characteristic to $\mathcal{P}$ iff it is characteristic to $\mathcal{P} - P := \{\mu - P : \mu \in \mathcal{P}\}$, where P can be any fixed probability measure. This is equivalent to being characteristic to the linear span of $\mathcal{P} - P$. But the linear span of $\mathcal{P} - P$ is precisely $\mathcal{M}_f^0$, which concludes.

Equivalence of (ii) & (v): First of all, notice that, if (v), then k and k_0 define the same MMD on $\mathcal{M}_f^0$, because, for any $\mu \in \mathcal{M}_f^0$, $\mu(\mathbb{1}) = 0$, thus:

$$\begin{aligned}
\|\mu\|_{k_0}^2 &= \iint \langle \delta_x - \nu_0 \,,\, \delta_y - \nu_0 \rangle_k \, d\bar{\mu}(x) \, d\mu(y) \\
&= \iint k(x,y) \, d\bar{\mu}(x) \, d\mu(y) - \int \langle \delta_x \,,\, \nu_0 \rangle_k \, d\bar{\mu}(x) \int d\mu(y) \\
&\quad - \int d\bar{\mu}(x) \int \langle \nu_0 \,,\, \delta_y \rangle_k \, d\mu(y) - \|\nu_0\|_k^2 \iint d\bar{\mu}(x) \, d\mu(y) \\
&= \|\mu\|_k^2 \ ,
\end{aligned}$$

Thus k_0 is characteristic to $\mathcal{M}_f^0$ iff k is also. Thus (v) implies (ii). Conversely, if k_0 is characteristic to $\mathcal{M}_f^0$, then k_0 is either characteristic to $\mathcal{M}_f$, in which case choosing $k_0 = k$ and $\nu_0 = 0$ fulfills the requirements of (v); or there exists a non zero measure $\nu_0 \in \mathcal{M}_f$ such that $\Phi_{k_0}(\nu_0) = 0$. As Φ_{k_0} is linear, we can choose $\nu_0(\mathbb{1}) = 1$ without loss of generality. Supposing now that we are in the latter case, the proof proceeds as follows.

(a) Show that the constant function $\mathbb{1} \notin \mathcal{H}_{k_0}$.

(b) Construct a new Hilbert space of functions of the form $\mathcal{H}_k = \operatorname{span} \mathbb{1} \oplus \mathcal{H}_{k_0}$.
(c) Show that it has a reproducing kernel k.
(d) Show that k_0 and k fulfill the requirements of (v).

(a) Suppose that $\mathbb{1} \in \mathcal{H}_{k_0}$. Then $1 = \bar{\nu}_0(\mathbb{1}) = \int \langle \mathbb{1}, k_0(.,x)\rangle_{k_0} \, d\bar{\nu}_0(x) \overset{(*)}{=} \left\langle \mathbb{1}, \int k_0(.,x)\, d\nu_0(x)\right\rangle_{k_0} = \left\langle \mathbb{1}, \Phi_{k_0}(\nu_0)\right\rangle_{k_0} = 0$, where in $(*)$ we use the definition of KMEs (1.1) . Contradiction. Thus $\mathbb{1} \notin \mathcal{H}_{k_0}$.

(b) Define $\mathcal{H} := \operatorname{span} \mathbb{1} \oplus \mathcal{H}_{k_0}$ and equip it with the inner product $\langle .,. \rangle$ that extends the inner product of $\mathcal{H}_{k_0}$ so, that

$$\mathbb{1} \perp \mathcal{H}_{k_0} \quad \text{and} \quad \|\mathbb{1}\| = 1\,. \tag{C.3}$$

In other words, for any $f = c_f \mathbb{1} + f^\perp \in \mathcal{H}$ and any $g = c_g \mathbb{1} + g^\perp \in \mathcal{H}$:

$$\langle f, g\rangle := \left\langle f^\perp, g^\perp \right\rangle_{k_0} + c_f \bar{c_g}. \tag{C.4}$$

Obviously $\mathcal{H}$ is a Hilbert space of functions.

(c) We now construct k by first defining an injective embedding Φ and then showing that $k(x,y) := \langle \Phi(\delta_x), \Phi(\delta_y)\rangle$ is a reproducing kernel with KME Φ.
As $\mathcal{M}_f^0$ is a hyperplane in $\mathcal{M}_f$ and $\nu_0 \in \mathcal{M}_f \backslash \mathcal{M}_f^0$, each measure $\mu \in \mathcal{M}_f$ can be decomposed uniquely in a sum: $\mu = \mu^\perp + \mu(\mathbb{1})\nu_0$ where $\mu^\perp = \mu - \mu(\mathbb{1})\nu_0 \in \mathcal{M}_f^0$. We may thus define the following linear embedding $\Phi : \mathcal{M}_f \to \mathcal{H}$ by

$$\Phi(\mu) := \begin{cases} \Phi_{k_0}(\mu) & \text{if } \mu \in \mathcal{M}_f^0 \\ \mathbb{1} & \text{if } \mu = \nu_0 \end{cases} \quad \text{i.e.} \quad \begin{aligned} \Phi(\mu) &:= \Phi_{k_0}(\mu^\perp) + \mu(\mathbb{1})\mathbb{1} \\ &= \Phi_{k_0}(\mu) + \mu(\mathbb{1})\mathbb{1} \end{aligned}\,. \tag{C.5}$$

Noting that $\Phi(\mu)^\perp = \Phi(\mu^\perp) = \Phi_{k_0}(\mu^\perp) = \Phi_{k_0}(\mu)$ and using (C.4), we get

$$\begin{aligned} \forall f \in \mathcal{H},\ \forall x \in \mathcal{X},\quad \langle f, \Phi(\delta_x)\rangle &= \left\langle f^\perp, \Phi(\delta_x)^\perp \right\rangle_{k_0} + c_f \\ &= f^\perp(x) + c_f \mathbb{1}(x) = f(x)\,. \end{aligned} \tag{C.6}$$

So by defining $k(x,y) := \langle \Phi(\delta_y), \Phi(\delta_x)\rangle$ and applying (C.6) to $f = \Phi(\delta_y)$, we see that $\Phi(\delta_y) = k(.,y)$. Thus (C.6) may be rewritten as

$$\forall f \in \mathcal{H}, \forall x \in \mathcal{X}, \qquad \langle f, k(.,x)\rangle = f(x).$$

Thus $\mathcal{H}$ is an RKHS with reproducing kernel and Φ is its associated KME.

(d) As k_0 is characteristic to $\mathcal{M}_f^0$, Φ is injective over $\mathcal{M}_f^0$. And $\Phi(\nu_0) \in \mathcal{H}\backslash\Phi(\mathcal{M}_f^0)$. Thus Φ is injective over $\mathcal{M}_f$, so k is characteristic to $\mathcal{M}_f$. To conclude, (C.5) shows that

$$\begin{aligned}\langle \delta_y - \nu_0 \,,\, \delta_x - \nu_0 \rangle &= \big\langle \Phi_{k_0}(\delta_y) + (\delta_y - \nu_0)(\mathbb{1})\mathbb{1} \,,\, \Phi_{k_0}(\delta_x) + (\delta_x - \nu_0)(\mathbb{1})\mathbb{1} \big\rangle \\ &= \big\langle \Phi_{k_0}(\delta_y) + 0 \,,\, \Phi_{k_0}(\delta_x) + 0 \big\rangle \\ &= k_0(x, y) \,.\end{aligned}$$

Equivalence of (v) with (iii) & (iv): First, notice that the kernel k constructed in the proof of (v) $\Rightarrow$ (ii) verifies:

$$\begin{aligned}k(x, y) &= \langle \Phi(\delta_x) \,,\, \Phi(\delta_y) \rangle \\ &= \big\langle \Phi_{k_0}(\delta_x) + \delta_x(\mathbb{1})\mathbb{1} \,,\, \Phi_{k_0}(\delta_y) + \delta_y(\mathbb{1})\mathbb{1} \big\rangle \\ &= \big\langle \Phi_{k_0}(\delta_x) \,,\, \Phi_{k_0}(\delta_y) \big\rangle + \|\mathbb{1}\|^2 \\ &= k_0(x, y) + 1 \,,\end{aligned}$$

where we used (C.3), (C.5) and the fact that by construction $\langle .\,,\,. \rangle$ coincides with $\langle .\,,\,. \rangle_{k_0}$ on $\mathcal{M}_f^0$. Thus the proof of (v) $\Rightarrow$ (ii) shows that, if k_0 characteristic to $\mathcal{M}_f^0$, then the kernel $k_0(x, y) + 1$ is characteristic to $\mathcal{M}_f$, thus $\int$spd (Thm. 1.2.2). $k(x, y) := k_0(x, y) + 1$ is $\int$spd. More generally, if instead of fixing $\|\mathbb{1}\|_k = 1$ in (C.3) we fixed $\|\mathbb{1}\|_k = \epsilon$ for some real $\epsilon > 0$, then we would have ended up with an $\int$spd kernel k verifying $k(x, y) := k_0(x, y) + \epsilon^2$. Thus (ii) implies (iii) and (iv). Conversely, given any kernel k of the previous form, the inner products defined by k and k_0 coincide on $\mathcal{M}_f^0$. So if k is characteristic to $\mathcal{M}_f^0$, then so is k_0. Thus (iii) or (iv) implies (ii). □

C.1.4 Proof of Theorem 1.3.4 Continued

The proof of Theorem 1.3.4 used the following lemma.

Lemma C.1.2. *Let k be a continuous, $\int$spd kernel and let $(\mu_\alpha)_\alpha$ be bounded in $\mathcal{M}_+$ (meaning $\sup_\alpha \|\mu_\alpha\|_{TV} < \infty$). Then $\mu_\alpha \overset{w-k}{\longrightarrow} \mu \;\Rightarrow\; \mu_\alpha \overset{\sigma}{\longrightarrow} \mu$. Consequently: $\mu_\alpha \overset{\|.\|_k}{\longrightarrow} \mu \;\Rightarrow\; \mu_\alpha \overset{\sigma}{\longrightarrow} \mu$.*

Proof. We will show that $\mu_\alpha(f) \to \mu(f)$ for any $f \in \mathscr{C}_c$. As $\mathscr{C}_c$ is a dense subset of $\mathscr{C}_{\to 0}$ and μ_α is bounded, combining Prop. 32.5 and Thm. 33.2 of [119] then shows that $\mu_\alpha(f) \to \mu(f)$ for any $f \in \mathscr{C}_{\to 0}$ (weak-$*$ convergence), which implies weak-convergence, $\mu_\alpha \overset{\sigma}{\to} \mu$ [9], and thus concludes.

Let K be a compact subset of $\mathcal{X}$. First, we show that there exists a function $h \in \mathcal{H}_k$ such that $h(x) > 0$ for any $x \in K$. To do so, let $f \in \mathscr{C}_b$ such that $f \geqslant 1$ on K. k being $\int$spd and $\mathcal{M}_f$ being the dual of $(\mathscr{C}_b)_c$, $\mathcal{H}_k$ is dense in $(\mathscr{C}_b)_c$ (Thm. 1.2.2). So we can find a sequence of functions $f_n \in \mathcal{H}_k$ that converges to f for the topology of $(\mathscr{C}_b)_c$. By definition of the topology of $(\mathscr{C}_b)_c$, this implies in

[119] Treves, *Topological Vector Spaces, Distributions and Kernels*, 1967
[9] Berg et al., *Harmonic Analysis on Semigroups Theory of Positive Definite and Related Functions*, 1984, Chap. 2, Cor. 4.3

particular that the restrictions of f_n to K converge in infinity norm, meaning: $\sup_{x\in K} |f_n(x) - f(x)| \to 0$. Thus, for a sufficiently large n, $f_n > 0$ on K, so we can take $h = f_n$.

Now, let us define the measures $h.\mu_\alpha$ as $[h.\mu_\alpha](f) = \mu_\alpha(hf)$ for any $f \in \mathscr{C}_b$. Then $\|h.\mu_\alpha\|_{TV} \leqslant \|h\|_\infty \|\mu_\alpha\|_{TV}$, so the new net $(h.\mu_\alpha)_\alpha$ is bounded. But bounded sets are relatively compact for the weak-$*$ topology $w(\mathcal{M}_f, \mathscr{C}_{\to 0})$. ([119], or Banach-Alaoglu theorem). So we can extract a subnet $h.\mu_\beta$ of $h.\mu_\alpha$ that converges in weak-$*$ topology. Then $h.\mu_\beta$ is also a Cauchy-net for the weak-$*$ topology, meaning that for any $\epsilon > 0$ and any sufficiently large β, β':

[119] Treves, *Topological Vector Spaces, Distributions and Kernels*, 1967, Thm. 33.2

$$|\mu_\beta(hf) - \mu_{\beta'}(hf)| \leqslant \epsilon, \quad \forall f \in \mathscr{C}_{\to 0} \ .$$

This inequality holds in particular for functions f whose support is contained in K, which we denote $f \in \mathscr{C}_c(K)$. But the mapping $f \mapsto g := hf$ is a bijective map from $\mathscr{C}_c(K)$ to itself (because $h > 0$ on K), so we actually have $|\mu_\beta(g) - \mu_{\beta'}(g)| \leqslant \epsilon$ for any $g \in \mathscr{C}_c(K)$. But this holds for any compact subset K of $\mathcal{X}$. So the inequality also holds for any function $g \in \mathscr{C}_c(\mathcal{X})$, which shows that μ_β is a Cauchy-net for the topology of pointwise convergence in $\mathscr{C}_c(\mathcal{X})$, also known as the *vague* topology. But $\mathcal{M}_+$ is vaguely complete [14], so μ_β converges to a measure $\mu' \in \mathcal{M}_+$. But for any $f \in \mathscr{C}_c(\mathcal{X})$, $\mu'(f) = \lim_\beta \mu_\beta(f) = \lim_\alpha \mu_\alpha(f) = \mu(f)$, thus μ' and μ coincide on $\mathscr{C}_c(\mathcal{X})$, which is a dense subset of $\mathscr{C}_{\to 0}$. Thus $\mu' = \mu$, and $\mu_\alpha(f) \to \mu(f)$ for any $f \in \mathscr{C}_c$. □

[14] Bourbaki, *Intégration - Chapitres 1-4*, 1965, Chap III, §1, n.9, Prop 14

Note that if we additionally supposed that $\mathcal{H}_k \hookrightarrow \mathscr{C}_{\to 0}$ (meaning that k is c_0-universal), then Lemma C.1.2 is a simple consequence of Lemma 1.3.3 and the fact that weak-$*$ and weak convergence coincide on $\mathcal{P}$.

C.1.5 Proof of Theorem 1.4.5 Continued

Proof. We are left with proving (a) and (b). To do so, we will use the decomposition $D = \sum_{|p|\leqslant m} \partial^p \mu_p$ of Lemma 1.4.3. Indeed, k being in $\mathscr{C}_b^{(m,m)}$, by Corollary 1.1.4, $\partial^p \mu_p$ embeds into $\mathcal{H}_k$ for any $|p| \leqslant m$ and $\mu_p \in \mathcal{M}_f$. Thus

$$\begin{aligned}
\langle \partial^p \mu_p \, , \, \partial^q \mu_q \rangle_k &= \left\langle \Phi_{\partial^{(0,p)}k}(\mu_p) \, , \, \Phi_{\partial^{(0,q)}k}(\mu_q) \right\rangle_k \\
&= \iint \left\langle \partial^{(0,p)}k(.,y) \, , \, \partial^{(0,q)}k(.,x) \right\rangle_k \mathrm{d}\bar{\mu}_q(x)\, \mathrm{d}\mu_p(y) \\
&= \iint \partial^{(q,p)}k(x,y)\, \mathrm{d}\bar{\mu}_q(x)\, \mathrm{d}\mu_p(y) \\
&= \iiint \mathrm{i}^{|p+q|} \xi^{p+q} e^{\mathrm{i}(x-y)\xi} \, \mathrm{d}\Lambda(\xi)\, \mathrm{d}\bar{\mu}_q(x)\, \mathrm{d}\mu_p(x) \ ,
\end{aligned}$$

where the first line uses Proposition 1.4.2, the second line uses twice the definition of a weak integral (1.1), the third uses (C.2) from Lemma C.1.1 and the fourth line uses the fact that $\partial^{(q,p)}k(x,y) = (-1)^{|p|}\partial^{p+q}\psi(x-y)$ and $\mathscr{F}\partial^{p+q}\psi = i^{|p+q|}\xi^{p+q}\,\mathscr{F}\psi = i^{|p+q|}\xi^{p+q}\Lambda$.

Let us denote $\xi^p\Lambda$ the measure defined by $\xi^p\Lambda(A) := \int_A \xi^p\,d\Lambda(\xi)$. We will now show that $\xi^{p+q}\Lambda$ is finite, so that we can apply the usual Bochner theorem and permute the order of integration. To do so, notice that $\partial^{(p,p)}k(x,y) = (-1)^{|p|}\partial^{2p}\psi(x-y)$ is a continuous kernel, thus, by Bochner's theorem, its associated measure Λ_∂ is finite and verifies $\mathscr{F}\Lambda_\partial = \partial^{2p}\psi$. But the usual calculus rules with Fourier transforms show that $\partial^{2p}\psi = (-i)^{|2p|}\xi^{2p}\Lambda$. Thus $\Lambda_\partial = i^{|p|}\xi^{2p}\Lambda$, showing that $\tilde{\Lambda}$ is a finite measure. Noting now that $2|\xi^{p+q}| \leqslant \xi^{2p} + \xi^{2q}$, this also implies that $\xi^{p+q}\Lambda$ is a finite measure. Consequently:

$$\begin{aligned}
\langle \partial^p\mu_p\,,\,\partial^q\mu_q\rangle_k &= \iiint i^{|p+q|}e^{i(x-y)}\,d[\xi^{p+q}\Lambda](\xi)\,d\bar{\mu}_q(x)\,d\mu_p(x)\\
&= \iiint i^{|p+q|}e^{i(x-y)}\,d\bar{\mu}_q(x)\,d\mu_p(x)\,d\tilde{\Lambda}(\xi)\\
&= \int i^{|p+q|}\xi^{p+q}\,\mathscr{F}\mu_q(\xi)\overline{\mathscr{F}\mu_p(\xi)}\,d\Lambda(\xi)\\
&= \int [\mathscr{F}(\,\partial^p\mu_p)](\xi)\overline{[\mathscr{F}(\,\partial^q\mu_q)](\xi)}\,d\Lambda(\xi).
\end{aligned}$$

Thus, with the decomposition $D = \sum_{|p|\leqslant m}\partial^p\mu_p$, we get

$$\begin{aligned}
\|D\|_k^2 &= \left\|\sum_{|p|\leqslant m}\partial^p\mu_p\right\|_k^2\\
&= \int \sum_{|p|,|q|\leqslant m}[\mathscr{F}(\,\partial^p\mu_p)](\xi)\overline{[\mathscr{F}(\,\partial^q\mu_q)](\xi)}\,d\Lambda(\xi)\\
&= \int \Big|\sum_{|p|\leqslant m}[\mathscr{F}(\,\partial^p\mu_p)](\xi)\Big|^2\,d\Lambda(\xi)\\
&= \int |\,\mathscr{F}D(\xi)|^2\,d\Lambda(\xi)\,,
\end{aligned}$$

where we used the linearity of the Fourier operator on the last line. □

C.2 Chapter 2

C.2.1 Detailed Proof of Theorem 2.2.3

Notations, Reminders and Preliminaries

For any function $\psi \in L^1(\mathbb{R}^d)$ and any finite (signed or complex regular Borel) measure ν over $\mathbb{R}^d$, we define their convolution as:

$$\nu * \psi(x) := \int \psi(x - x') \, d\nu(x') \, .$$

We define the Fourier and inverse Fourier transforms of ψ and ν as

$$\mathscr{F}\psi(\omega) := (2\pi)^{-d/2} \int_{\mathbb{R}^d} e^{-i\langle \omega, x\rangle} \psi(x) \, dx \quad \text{and} \quad \mathscr{F}\nu(\omega) := (2\pi)^{-d/2} \int_{\mathbb{R}^d} e^{-i\langle \omega, x\rangle} \, d\nu(x) \, ,$$
$$\mathscr{F}^{-1}\psi(\omega) := (2\pi)^{-d/2} \int_{\mathbb{R}^d} e^{i\langle \omega, x\rangle} \psi(x) \, dx \quad \text{and} \quad \mathscr{F}^{-1}\nu(\omega) := (2\pi)^{-d/2} \int_{\mathbb{R}^d} e^{i\langle \omega, x\rangle} \, d\nu(x) \, .$$

Fourier transforms are of particular interest when working with translation-invariant kernel because of Bochner's theorem. Here we quote [125], but add a useful second sentence, which is immediate to show.

[125] Wendland, *Scattered Data Approximation*, 2004, Thm 6.6

Theorem C.2.1 (Bochner). *A continuous function $\psi : \mathbb{R}^d \to \mathbb{C}$ is positive definite if and only if it is the Fourier transform of a finite, nonnegative Borel measure ν over $\mathbb{R}^d$. Moreover, ψ is real-valued if and only if ν is symmetric.*

The next theorem, also quoted from [125], shows that the Fourier (inverse) transform may be seen as a unitary isomorphism from $L^2(\mathbb{R}^d)$ to $L^2(\mathbb{R}^d)$.

[125] Wendland, *Scattered Data Approximation*, 2004, Cor 5.25

Theorem C.2.2 (Plancherel). *There exists an isomorphic mapping $T : L^2(\mathbb{R}^d) \to L^2(\mathbb{R}^d)$ such that:*

(i) $\|Tf\|_{L^2(\mathbb{R}^d)} = \|f\|_{L^2(\mathbb{R}^d)}$ *for all* $f \in L^2(\mathbb{R}^d)$.
(ii) $Tf = \mathscr{F} f$ *for any* $f \in L^2(\mathbb{R}^d) \cap L^1(\mathbb{R}^d)$.
(iii) $T^{-1} g = \mathscr{F}^{-1} g$ *for all* $g \in L^2(\mathbb{R}^d) \cap L^1(\mathbb{R}^d)$.

The isomorphism is uniquely determined by these properties.

We will call T the Fourier transform over L^2 and note it $\mathscr{F}$.

Remark C.2.3. Combining Plancherel's and Bochner's theorems, we see that, if ψ is a continuous, positive definite (resp. and real-valued) function in $L^2(\mathbb{R}^d)$, then the measure ν from Bochner's theorem is absolutely continuous, and its density is $\mathscr{F}^{-1}\psi$. In particular, $\mathscr{F}^{-1}\psi$ is real-valued, nonnegative (resp. and symmetric).

Next, our proof of Theorem 2.2.3 will need the following result.

Lemma C.2.4. *Let $\mathcal{Z} = \mathbb{R}^{d'}$, $\psi \in L^2(\mathbb{R}^d)$ such that $\mathscr{F}\psi \in L^1(\mathbb{R}^d)$. Let k be the translation-invariant kernel $k(z, z') := \psi(z - z')$ and $h(z) := \mathscr{F}^{-1}\sqrt{\mathscr{F}\psi}(z)$. Let Z be any random variable on $\mathcal{Z}$, and $\hat{Z} := \{(z_i, w_i)\}_{i=1}^n$. Then:*

$$\left\|\hat{\mu}_Z^k - \mu_Z^k\right\|_k^2 = (2\pi)^{\frac{d'}{2}} \int_{z\in\mathcal{Z}} \left|\hat{\mu}_Z^h - \mu_Z^h\right|^2 dz \,. \tag{C.7}$$

Proof. (of Lemma C.2.4) For any finite (signed) measure ν over $\mathcal{Z} = \mathbb{R}^{d'}$, we define:

$$\mu_\nu^k := \int k(z, .)\, d\nu(z) \,.$$

Then we have:

$$\begin{aligned}
\left\|\mu_\nu^k\right\|_k^2 &= \int_{z\in\mathbb{R}^d}\int_{z'\in\mathbb{R}^d} \psi(z-z')\, d\nu(z)\, d\nu(z') \\
&= \int_{z\in\mathbb{R}^d}\int_{z'\in\mathbb{R}^d} \left((2\pi)^{-d'/2}\int_{\omega\in\mathbb{R}^d} e^{-i\langle\omega, z-z'\rangle}\, \mathscr{F}^{-1}\psi(\omega)\, d\omega\right) d\nu(z)\, d\nu(z') \\
&= \int_{\omega\in\mathbb{R}^d} (2\pi)^{-d'/2} \int_{z\in\mathbb{R}^d}\int_{z'\in\mathbb{R}^d} e^{-i\langle\omega, z-z'\rangle}\, d\nu(z)\, d\nu(z')\, \mathscr{F}^{-1}\psi(\omega)\, d\omega \\
&= \int_{\omega\in\mathbb{R}^d} (2\pi)^{d'/2}\, \mathscr{F}\nu(\omega)\, \mathscr{F}\nu(-\omega)\, \mathscr{F}^{-1}\psi(\omega)\, d\omega \\
&= (2\pi)^{d'/2} \int_{\omega\in\mathbb{R}^d} |\mathscr{F}\nu(\omega)|^2\, \mathscr{F}^{-1}\psi(\omega)\, d\omega
\end{aligned}$$

The second line uses the following: (i) ψ is continuous, because $\mathscr{F}\psi \in L^1(\mathbb{R}^d)$ (Riemann-Lebesgue lemma); (ii) Theorem C.2.1 (Bochner) and Remark C.2.3 from the Appendix. Third and fourth line use Fubini's theorem. Last line uses the fact that $\mathscr{F}\nu(-\omega)$ is the complex conjugate of $\mathscr{F}\nu$ because $\mathscr{F}\psi$ is positive (thus real-valued).

Applying this with $\nu = \hat{Q} - Q$, where Q is the distribution of Z and $\hat{Q} := \sum_i w_i \delta_{z_i}$, we get:

$$\begin{aligned}
\left\|\hat{\mu}_Z^k - \mu_Z^k\right\|_k^2 &= \left\|\mu_{\hat{Q}-Q}^k\right\|_k^2 \\
&= (2\pi)^{d'/2} \int_{\omega\in\mathbb{R}^d} \left|\mathscr{F}[\hat{Q} - Q](\omega)\right|^2 \mathscr{F}\psi(\omega)\, d\omega \\
&= (2\pi)^{d'/2} \int_{\omega\in\mathbb{R}^d} \left|\mathscr{F}[\hat{Q} - Q](\omega)\sqrt{\mathscr{F}\psi(\omega)}\right|^2 d\omega \\
&= (2\pi)^{d'/2} \int_{z\in\mathcal{Z}} \left|\mathscr{F}^{-1}\left[\mathscr{F}[\hat{Q} - Q]\sqrt{\mathscr{F}\psi}\right](z)\right|^2 dz \\
&= (2\pi)^{d'/2} \int_{z\in\mathcal{Z}} \left|[\hat{Q} - Q] * h(z)\right|^2 dz \\
&= (2\pi)^{d'/2} \int_{z\in\mathcal{Z}} \left|\sum_i w_i h(z - z_i) - \int h(z - z')\, dQ(z')\right|^2 dz \\
&= (2\pi)^{d'/2} \int_{z\in\mathcal{Z}} \left|\hat{\mu}_Z^h - \mu_Z^h(z)\right|^2 dz \,.
\end{aligned}$$

Third line uses the fact that $\mathscr{F}\psi$ is positive (see Appendix, Remark C.2.3). Fourth line uses Plancherel's theorem (see Appendix, Theorem C.2.2). Fifth line uses the fact that the Fourier (inverse) transform of a product equals the convolutional product of the (inverse) Fourier transforms [55, Theorem 1.4, and its generalisation to finite measures p.145]. □

We now state Theorem 1 from [54], which serves as basis to our proof. Slightly modifying[1] the notation of [1], for $0 < \theta < 1$ and $1 \leqslant q \leqslant \infty$ we will write $(E_0, E_1)_{\theta,q}$ to denote interpolation spaces, where E_0 and E_1 are Banach spaces that are continuously embedded into some topological Hausdorff vector space $\mathcal{E}$. Following [54], we also define $(E_0, E_1)_{1,2} := E_1$.

Theorem C.2.5 (Kanagawa et al.)**.** *Let X be a random variable with distribution P and let $\{(x_i, w_i)\}_{i=1}^n$ be random variables with joint distribution S satisfying Assumption 2.2.2 (with corresponding distribution Q). Let $\hat{\mu}_X := \sum_i w_i k(x_i, .)$ be an estimator of $\mu_X := \int k(x, .)\, dP(x)$ such that for some constants $b > 0$ and $0 < c \leqslant 1/2$:*

(i) $\mathbb{E}_S\left[\left\|\hat{\mu}_X - \mu_X\right\|_k\right] = O(n^{-b})$,
(ii) $\mathbb{E}_S\left[\sum_i w_i^2\right] = O(n^{-2c})$

as $n \to \infty$. Let θ be a constant such that $0 < \theta \leqslant 1$.

Then, for any function $g : \mathbb{R}^d \to \mathbb{R}$ in $\left(L^2(Q), \mathcal{H}_k\right)_{\theta,2}$, there exists a constant C, independent of n, such that:

$$\mathop{\mathbb{E}}_{S}\left[\left|\sum_i w_i g(x_i) - \mathop{\mathbb{E}}_{X\sim P}[g(X)]\right|\right] \leqslant C\, n^{-\theta b + (1/2-c)(1-\theta)} . \tag{C.8}$$

In the proof of our finite sample guarantee, we will need the following slightly modified version of this result, where we (a) slightly modify condition (ii) by asking that it holds almost surely, and (b) consider squared norms in Condition (i) and (C.8).

Theorem C.2.6 (Kanagawa et al.)**.** *Let X be a random variable with distribution P and let $\{(x_i, w_i)\}$ be random variables with joint distribution S satisfying Assumption 2.2.2 (with corresponding distribution Q). Let $\hat{\mu}_X := \sum_i w_i k(x_i, .)$ be an estimator of $\mu_X := \int k(x, .)\, dP(x)$ such that for some constants $b > 0$ and $0 < c \leqslant 1/2$:*

(i) $\mathbb{E}_S\left[\left\|\hat{\mu}_X - \mu_X\right\|_k^2\right] = O(n^{-2b})$,
(ii) $\sum_{i=1}^n w_i^2 = O(n^{-2c})$ *(with S-probability 1)* ,

as $n \to \infty$. Let θ be a constant such that $0 < \theta \leqslant 1$.

Then, for any function $g : \mathbb{R}^d \to \mathbb{R}$ in $\left(L^2(Q), \mathcal{H}_k\right)_{\theta,2}$, there exists a constant C, independent of n, such that:

$$\mathop{\mathbb{E}}_{S}\left[\left|\sum_i w_i g(x_i) - \mathop{\mathbb{E}}_{X\sim P}[g(X)]\right|\right] \leqslant C\, n^{-2(\theta b - (1/2-c)(1-\theta))} . \tag{C.9}$$

[54] Kanagawa et al., *Convergence Guarantees for Kernel-Based Quadrature Rules in Misspecified Settings*, 2016

[1] [1] introduce interpolation spaces using so-called *J*- and *K-methods*, resulting in two notations $(E_0, E_1)_{\theta,q;J}$ (Definition 7.12) and $(E_0, E_1)_{\theta,q;K}$ (Def 7.9) respectively. However, it follows from Theorem 7.16 that these two definitions are equivalent if $0 < \theta < 1$ and we simply drop the K and J subindices.

[1] Adams and Fournier, *Sobolev Spaces*, 2003, Chap 7

[54] Kanagawa et al., *Convergence Guarantees for Kernel-Based Quadrature Rules in Misspecified Settings*, 2016

Proof. The proof of this adapted version of Theorem 1 in [54] is almost a copy paste of the original proof, but with the appropriate squares to account for the modified condition (i), and with their f renamed to g here. The only slight non-trivial difference is in their Inequality (20). Replace their triangular inequality by Jensen's inequality to yield:

[54] Kanagawa et al., *Convergence Guarantees for Kernel-Based Quadrature Rules in Misspecified Settings*, 2016

$$\mathop{\mathbb{E}}_{S}\left[\left|\sum_{i=1}^{n} w_i g(x_i) - \mathop{\mathbb{E}}_{X\sim P}[g(X)]\right|^2\right] \leqslant \begin{cases} 3\,\mathbb{E}_S\left[\left|\sum_{i=1}^n w_i g(x_i) - \sum_{i=1}^n w_i g_{\lambda_n}(x_i)\right|^2\right] \\ +3\,\mathbb{E}_S\left[\left|\sum_{i=1}^n w_i g_{\lambda_n}(x_i) - \mathbb{E}_{X\sim P}[g_{\lambda_n}(X)]\right|^2\right] \\ +3\,\mathbb{E}_S\left[\left|\mathbb{E}_{X\sim P}[g_{\lambda_n}(X)] - \mathbb{E}_{X\sim P}[g(X)]\right|^2\right], \end{cases}$$

where g and g_{λ_n} are the functions that they call f and f_{λ_n}. □

We are now ready to prove 2.2.3.

Starting Proof of Theorem 2.2.3

Proof. This proof is self-contained: the sketch from the main part is not needed. Throughout the proof, C designates constants that depend neither on sample size n nor on radius R (to be introduced). But their value may change from line to line.

Let ψ be such that $k_z^t(z,z') = \psi(z-z')$. Then $\mathscr{F}\psi(\omega) = (1+\|\omega\|_2^2)^{-t}$ [125, Chapter 10]. Applying Lemma C.2.4 to the Matérn kernel k_z^t thus yields

$$\mathop{\mathbb{E}}_{S}\left[\left\|\hat{\mu}^{k_z^t}_{f(X)} - \mu^{k_z^t}_{f(X)}\right\|^2_{k_z^t}\right] = (2\pi)^{\frac{d'}{2}} \int_{\mathcal{Z}} \mathop{\mathbb{E}}_{S}\left[\left([\hat{\mu}^h_{f(X)} - \mu^h_{f(X)}](z)\right)^2\right] dz, \tag{C.10}$$

where $h = \mathscr{F}^{-1}\sqrt{\mathscr{F}\,k_z^t}$ is again a Matérn kernel, but with smoothness parameter $t/2 > d'/2$.

Step 1: Applying C.2.6

We now want to upper bound the integrand by using C.2.6. To do so, let $\mathcal{K}$ be the common compact support of P and marginals of $x_1,\dots,x_n$. Now, rewrite the integrand as:

$$\begin{aligned}
\mathop{\mathbb{E}}_{S}\left[\left([\hat{\mu}^h_{f(X)} - \mu^h_{f(X)}](z)\right)^2\right] &= \mathop{\mathbb{E}}_{S}\left[\left(\sum_i w_i h\big(f(x_i)-z\big) - \mathop{\mathbb{E}}_{X\sim P}\big[h\big(f(X)-z\big)\big]\right)^2\right] \\
&= \mathop{\mathbb{E}}_{S}\left[\left(\sum_i w_i h\big(f(x_i)-z\big)\varphi_{\mathcal{K}}(x_i) - \mathop{\mathbb{E}}_{X\sim P}\big[h\big(f(X)-z\big)\varphi_{\mathcal{K}}(X)\big]\right)^2\right] \\
&= \mathop{\mathbb{E}}_{S}\left[\left(\sum_i w_i g_z(x_i) - \mathop{\mathbb{E}}_{X\sim P}[g_z(X)]\right)^2\right],
\end{aligned} \tag{C.11}$$

where $\varphi_{\mathcal{K}}$ is any smooth function $\leqslant 1$, with compact support, that equals 1 on a neighborhood of $\mathcal{K}$ and where $g_z(x) := h(f(x) - z)\varphi_{\mathcal{K}}(x)$.

To apply C.2.6, we need to prove the existence of $0 < \theta \leqslant 1$ such that $g_z \in \left(L^2(Q), \mathcal{H}_{k_x^s}\right)_{\theta,2}$ for each $z \in \mathcal{Z}$. We will prove this fact in two steps: (a) first we show that $g_z \in \left(L^2(\mathbb{R}^d), \mathcal{H}_{k_x^s}\right)_{\theta,2}$ for each $z \in \mathcal{Z}$ and certain choice of θ and (b) we argue that $\left(L^2(\mathbb{R}^d), \mathcal{H}_{k_x^s}\right)_{\theta,2}$ is continuously embedded in $\left(L^2(Q), \mathcal{H}_{k_x^s}\right)_{\theta,2}$.

Step 1(a): Note that $g_z \in \mathcal{W}_2^{\min(\alpha,t/2)}(\mathbb{R}^d)$ because f is α-times differentiable, $h \in \mathcal{W}_2^{t/2}(\mathbb{R}^{d'})$ (thus g_z is $\min(\alpha, t/2)$-times differentiable in the distributional sense), and g_z has compact support (thus meets the integrability conditions of Sobolev spaces). As k_x^s is a Matérn kernel with smoothness parameter s, its associated RKHS $\mathcal{H}_{k_x^s}$ is the Sobolev space $\mathcal{W}_2^s(\mathbb{R}^d)$ [125, Chapter 10]. Now, if $s \leqslant \min(\alpha, t/2)$, then $g_z \in \mathcal{W}_2^s(\mathbb{R}^d) = \mathcal{H}_{k_x^s} = \left(L^2(\mathbb{R}^d), \mathcal{W}_2^s(\mathbb{R}^d)\right)_{1,2}$ and step (a) holds for $\theta = 1$. Thus for the rest of this step, we assume $s > \min(\alpha, t/2)$. It is known that $\mathcal{W}_2^s(\mathbb{R}^d) = B_{2,2}^s(\mathbb{R}^d)$ for $0 < s < \infty$ [1], where $B_{2,2}^s(\mathbb{R}^d)$ is the Besov space of smoothness s. It is also known that $B_{2,2}^s(\mathbb{R}^d) = \left(L^2(\mathbb{R}^d), \mathcal{W}_2^m(\mathbb{R}^d)\right)_{s/m,2}$ for any integer $m > s$ [1]. Applying this to $\mathcal{W}_2^{\min(\alpha,t/2)}(\mathbb{R}^d)$ and denoting $s' = \min(\alpha, t/2)$ we get

[1] Adams and Fournier, *Sobolev Spaces*, 2003, Page 255

[1] Adams and Fournier, *Sobolev Spaces*, 2003, Page 230

$$g_z \in \mathcal{W}_2^{s'}(\mathbb{R}^d) = \left(L^2(\mathbb{R}^d), \mathcal{W}_2^s(\mathbb{R}^d)\right)_{s'/s,2} = \left(L^2(\mathbb{R}^d), \mathcal{H}_{k_x^s}\right)_{s'/s,2}, \qquad \forall z \in \mathcal{Z}\,.$$

Thus, whatever s, step (a) is always satisfied with $\theta := \min(\frac{\alpha}{s}, \frac{t}{2s}, 1) \leqslant 1$.

Step 1(b): If $\theta = 1$, then $\left(L^2(\mathbb{R}^d), \mathcal{H}_{k_x^s}\right)_{1,2} = \mathcal{H}_{k_x^s} = \left(L^2(Q), \mathcal{H}_{k_x^s}\right)_{1,2}$. Now assume $\theta < 1$. Note that $L^2(\mathbb{R}^d)$ is continuously embedded in $L^2(Q)$, because we assumed that Q has a bounded density. Thus Theorem V.1.12 of [8] applies and gives the desired inclusion.

[8] Bennett and Sharpley, *Interpolation of Operators*, 1988

Now we apply C.2.6, which yields a constant C_z independent of n such that:

$$\mathop{\mathbb{E}}_{S}\left[\left([\hat{\mu}^h_{f(X)} - \mu^h_{f(X)}](z)\right)^2\right] \leqslant C_z n^{-2\nu}\,,$$

with $\nu := \theta b - (1/2 - c)(1 - \theta)$.

We now prove that the constants C_z are uniformly bounded. From Equations (18-19) of [54], it appears that $C_z = C\left\|T^{-\theta/2} g_z\right\|_{L^2(Q)}$, where C is a constant independent of z and $T^{-\theta/2}$ is defined as follows. Let T be the operator from $L^2(Q)$ to $L^2(Q)$ defined by

[54] Kanagawa et al., *Convergence Guarantees for Kernel-Based Quadrature Rules in Misspecified Settings*, 2016

$$Tf := \int k_x(x, .) f(x)\, dQ(x).$$

It is continuous, compact and self-adjoint. Denoting $(e_i)_i$ an orthonormal basis of eigenfunctions in $L^2(Q)$ with eigenvalues $\mu_1 \geqslant \mu_2 \geqslant \cdots \geqslant 0$, let $T^{\theta/2}$ be the operator from $L^2(Q)$ to $L^2(Q)$ defined by:

$$T^{\theta/2} f := \sum_{i=1}^{\infty} \mu_i^{\theta/2} \langle e_i , f \rangle_{L^2(Q)} e_i .$$

Using [98] together with [113] we conclude that $T^{\theta/2}$ is injective. Thus $\mu_i > 0$ for all i. Thus, if $\theta = 1$, Lemma 6.4 of [114] shows that the range of $T^{\theta/2}$ is $[\mathcal{H}_k]_\sim$, the image of the canonical embedding of $\mathcal{H}_k$ into $L^2(Q)$. And as Q has full support, we may identify $[\mathcal{H}_k]_\sim$ and $\mathcal{H}_k = \left(L^2(Q), \mathcal{H}_k\right)_{\theta,2}$. Now, if $\theta < 1$, Theorem 4.6 of [114] shows that the range of $T^{\theta/2}$ is $\left(L^2(Q), \mathcal{H}_k\right)_{\theta,2}$.

[98] Scovel et al., *Radial kernels and their RKHS*, 2010, Cor 4.9.i
[113] Steinwart and Christmann, *Support Vector Machines*, 2008, Thm 4.26.i
[114] Steinwart and Scovel, *Mercer's Theorem on General Domains*, 2012

Thus the inverse operator $T^{-\theta/2}$ is well-defined, goes from $\left(L^2(Q), \mathcal{H}_k\right)_{\theta,2}$ to $L^2(Q)$ and can be written in the following form:

$$T^{-\theta/2} f := \sum_{i=1}^{\infty} \mu_i^{-\theta/2} \langle e_i , f \rangle_{L^2(Q)} e_i . \tag{C.12}$$

Using this, we get:

$$\begin{aligned}
|C_z| &= C \left\| T^{-\theta/2} g_z \right\|_{L^2(Q)} \\
&= C \left\| \sum_{i=1}^{\infty} \mu_i^{-\theta/2} \langle e_i , h(f(\cdot) - z) \varphi_{\mathcal{K}}(\cdot) \rangle_{L^2(Q)} e_i \right\|_{L^2(Q)} \\
&\leqslant C \max_{z \in \mathcal{Z}} |h(z)| \left\| \sum_{i=1}^{\infty} \mu_i^{-\theta/2} \langle e_i , \varphi_{\mathcal{K}} \rangle_{L^2(Q)} e_i \right\|_{L^2(Q)} \\
&= C \max_{z \in \mathcal{Z}} |h(z)| \left\| T^{-\theta/2} \varphi_{\mathcal{K}} \right\|_{L^2(Q)} ,
\end{aligned}$$

which is a constant independent of z. Hereby, we used the fact that $\varphi_{\mathcal{K}} \in \left(L^2(Q), \mathcal{H}_k\right)_{\theta,2}$, because it is infinitely smooth and has compact support. Thus we just proved that

$$\mathop{\mathbb{E}}_{S} \left[\left([\hat{\mu}^h_{f(X)} - \mu^h_{f(X)}](z) \right)^2 \right] \leqslant C n^{-2\nu} . \tag{C.13}$$

Step 2: Splitting the integral in two parts

However, now that this upper bound does not depend on z anymore, we cannot integrate over all $\mathcal{Z}$ ($= \mathbb{R}^{d'}$). Thus we now decompose the integral in (C.10) as:

$$\int_{\mathcal{Z}} \mathop{\mathbb{E}}_{S} \left[\left([\hat{\mu}^h_{f(X)} - \mu^h_{f(X)}](z) \right)^2 \right] dz = \begin{cases} \displaystyle\int_{B_R} \mathop{\mathbb{E}}_{S} \left[\left([\hat{\mu}^h_{f(X)} - \mu^h_{f(X)}](z) \right)^2 \right] dz \\ \displaystyle + \int_{\mathcal{Z} \setminus B_R} \mathop{\mathbb{E}}_{S} \left[\left([\hat{\mu}^h_{f(X)} - \mu^h_{f(X)}](z) \right)^2 \right] dz , \end{cases} \tag{C.14}$$

where B_R denotes the ball of radius R, centered on the origin of $\mathcal{Z} = \mathbb{R}^{d'}$. We will upper bound each term by a function depending on R, and eventually make R depend on the sample size so as to balance both upper bounds.

On B_R we upper bound the integral by Rate (C.13) times the ball's volume (which grows like $R^{d'}$):

$$\int_{B_R} \mathop{\mathbb{E}}_{S} \left[\left([\hat{\mu}^h_{f(X)} - \mu^h_{f(X)}](z) \right)^2 \right] dz \leqslant C R^{d'} n^{-2\nu} \,. \tag{C.15}$$

On $\mathcal{Z} \backslash B_R$ we upper bound the integral by a value that decreases with R. The intuition is that, according to (C.11), the integrand is the expectation of sums of Matérn functions, which are all centered on a compact domain. Thus it should decay exponentially with z outside of a sufficiently large ball. Next we turn to the formal argument.

Let us define $\|f\|_{\mathcal{K}} := \max_{x \in \mathcal{X}} \|f(x)\varphi_{\mathcal{K}}(x)\|$, which is finite because $f\varphi_{\mathcal{K}}$ is an α-times differentiable (thus continuous) function with compact support. Now, Matérn kernels are radial kernels, meaning that there exists a function $\tilde{h}$ over $\mathbb{R}$ such that $h(x) = \tilde{h}(\|x\|)$ [118]. Moreover $\tilde{h}$ is strictly positive and decreasing. Using (C.11) we may write

[118] Tolstikhin et al., *Minimax Estimation of Kernel Mean Embeddings*, 2017, p.5

$$\begin{aligned}
\mathop{\mathbb{E}}_{S} \left[\left([\hat{\mu}^h_{f(X)} - \mu^h_{f(X)}](z) \right)^2 \right] &= \mathop{\mathbb{E}}_{S} \left[\left(\sum_i w_i h\big(f(x_i) - z\big)\varphi_{\mathcal{K}}(x_i) - \mathop{\mathbb{E}}_{X \sim P} \big[h\big(f(X) - z\big)\varphi_{\mathcal{K}}(X) \big] \right)^2 \right] \\
&\leqslant \mathop{\mathbb{E}}_{S} \left[\left(\sum_i w_i h\big(f(x_i) - z\big)\varphi_{\mathcal{K}}(x_i) \right)^2 + \left(\mathop{\mathbb{E}}_{X \sim P} \big[h\big(f(X) - z\big)\varphi_{\mathcal{K}}(X) \big] \right)^2 \right] \\
&\overset{(\dagger)}{\leqslant} \tilde{h}(\|z\| - \|f\|_{\mathcal{K}})^2 \mathop{\mathbb{E}}_{S} \left[\left(\left(\sum_i w_i \right)^2 + 1 \right) \right],
\end{aligned}$$

where we assumed $R > \|f\|_{\mathcal{K}}$ and used the fact that $\tilde{h}$ is a decreasing function in $(\dagger)$. Using Cauchy-Schwarz and applying hypothesis (ii), we get:

$$\begin{aligned}
\mathbb{E} \left[\left([\hat{\mu}^h_{f(X)} - \mu^h_{f(X)}](z) \right)^2 \right] &\leqslant \tilde{h}(\|z\| - \|f\|_{\mathcal{K}})^2 \mathop{\mathbb{E}}_{S} \left[\left(n \left(\sum_i w_i^2 \right) + 1 \right) \right] \\
&\leqslant C\, n^{1-2c}\, \tilde{h}(\|z\| - \|f\|_{\mathcal{K}})^2 \,.
\end{aligned}$$

Let $\mathcal{S}_{d'}$ be the surface area of the unit sphere in $\mathbb{R}^{d'}$. We have:

$$\begin{aligned}\int_{\mathcal{Z}\setminus B_R} \mathbb{E}\left[\left([\hat{\mu}^h_{f(X)} - \mu^h_{f(X)}](z)\right)^2\right] dz &\leqslant Cn^{1-2c}\int_{\mathcal{Z}\setminus B_R} \tilde{h}(\|z\| - \|f\|_{\mathcal{K}})^2\, dz \\ &\overset{(\dagger)}{=} Cn^{1-2c}\int_{r=R-\|f\|_{\mathcal{K}}}^{+\infty} \tilde{h}(r)^2 \mathcal{S}_{d'}(r+\|f\|_{\mathcal{K}})^{d'-1}\, dr \\ &\leqslant Cn^{1-2c}2^{d'-1}\int_{r=R-\|f\|_{\mathcal{K}}}^{+\infty} \tilde{h}(r)^2 \mathcal{S}_{d'} r^{d'-1}\, dr, \\ &\quad (\text{for } R \geqslant 2\|f\|_{\mathcal{K}}) \end{aligned} \tag{C.16}$$

where $(\dagger)$ switches to radial coordinates. From Lemma 5.13 of [125] we get, for any $r > 0$:

[125] Wendland, *Scattered Data Approximation*, 2004

$$|\tilde{h}(r)| \leqslant Cr^{t/2-d'/2}\sqrt{\frac{2\pi}{r}}e^{-r}e^{|d'/2-t/2|^2/(2r)}.$$

Recalling that $t > d'$ by assumption we have

$$\begin{aligned}&\int_{\mathcal{Z}\setminus B_R} \mathbb{E}\left[\left([\hat{\mu}^h_{f(X)} - \mu^h_{f(X)}](z)\right)^2\right] dz \\ &\quad\leqslant Cn^{1-2c}\int_{r=R-\|f\|_{\mathcal{K}}}^{+\infty} r^{t-2}e^{-2r}e^{(t-d')^2/(4r)}\, dr \\ &\quad\leqslant Cn^{1-2c}e^{\frac{(t-d')^2}{4(R-\|f\|_{\mathcal{K}})}}\int_{r=R-\|f\|_{\mathcal{K}}}^{+\infty} r^{t-2}e^{-2r}\, dr.\end{aligned}$$

Now, $t/2$ being by assumption a strictly positive integer, $t-2$ is an integer. Thus, using [40]

[40] Gradshteyn and Ryzhik, *Table of Integrals, Series, and Products*, 2007, 2.321.2

$$\int_{r=R-\|f\|_{\mathcal{K}}}^{+\infty} r^{t-2}e^{-2r}\, dr = e^{-2(R-\|f\|_{\mathcal{K}})}\left(\sum_{k=0}^{t-2}\frac{k!\binom{t-2}{k}}{2^{k+1}}(R-\|f\|_{\mathcal{K}})^{t-2-k}\right)$$

we continue by writing

$$\begin{aligned}&\int_{\mathcal{Z}\setminus B_R} \mathbb{E}\left[\left([\hat{\mu}^h_{f(X)} - \mu^h_{f(X)}](z)\right)^2\right] dz \\ &\quad\leqslant Cn^{1-2c}e^{\frac{(t-d')^2}{4(R-\|f\|_{\mathcal{K}})}}e^{-2(R-\|f\|_{\mathcal{K}})}(R-\|f\|_{\mathcal{K}})^{t-2} \\ &\qquad(\text{for } R \geqslant \|f\|_{\mathcal{K}} + 1) \qquad\qquad (C.17)\\ &\quad\leqslant Cn^{1-2c}(R-\|f\|_{\mathcal{K}})^{t-2}e^{-2(R-\|f\|_{\mathcal{K}})} \qquad (C.18)\\ &\qquad(\text{for } R \geqslant \|f\|_{\mathcal{K}} + \frac{(t-d')^2}{4}). \qquad (C.19)\end{aligned}$$

Step 3: Choosing R to balance the terms

Compiling (C.14), (C.15) and (C.18), we get:

$$\int_{\mathcal{Z}} \mathop{\mathbb{E}}_{S}\left[\left([\hat{\mu}^h_{f(X)} - \mu^h_{f(X)}](z)\right)^2\right] dz \leqslant \begin{cases} CR^{d'}n^{-2\nu} \\ +Cn^{1-2c}(R-\|f\|_{\mathcal{K}})^{t-2}e^{-2(R-\|f\|_{\mathcal{K}})}. \end{cases}$$

We now let R depend on the sample size n so that both rates be (almost) balanced. Ideally, defining $\gamma := \nu + 1/2 - c \geqslant 0$ and taking the log of these rates, we would thus solve

$$d' \log R = 2\gamma \log n + (t-2)\log(R - \|f\|_{\mathcal{K}}) - 2(R - \|f\|_{\mathcal{K}}) \,, \quad \text{(C.20)}$$

and stick the solution R_s back into either of the two rates. Instead, we will upper bound R_s and stick the upper bound into the first rate, $R^d n^{-2\nu}$, which is the one increasing with R. More precisely, we will now show that for large enough n we can upper bound R_s essentially with $2\gamma \log(n) + \|f\|_{\mathcal{K}}$, which also satisfies conditions (C.16), (C.17) and (C.19). This will complete the proof.

Note that $t > d' \geqslant 1$ and $t \in \mathbb{N}_+$. First assume $t = 2$. Then it is easy to check that (C.20) has a unique solution R_s satisfying $R_s \leqslant \gamma \log n + \|f\|_{\mathcal{K}}$ as long as $n \geqslant \exp\left(\frac{1-\|f\|_{\mathcal{K}}}{\gamma}\right)$.

Next, assume $t > 2$. Then for n large enough (C.20) has exactly 2 solutions and the larger of which will be denoted R_s. We now replace the right hand side of (C.20) with a lower bound $d' \log(R - \|f\|_{\mathcal{K}})$:

$$(d' - t + 2)\log(R - \|f\|_{\mathcal{K}}) = 2\gamma \log n - 2(R - \|f\|_{\mathcal{K}}) \,, \quad \text{(C.21)}$$

Clearly, (C.21) has one (if $d' - t + 2 \geqslant 0$) or two (if $d' - t + 2 < 0$) solutions, and in both cases the larger one, R_s^*, satisfies $R_s^* \geqslant R_s$. If $d' - t + 2 \geqslant 0$ then, for $n \geqslant e^{1/\gamma}$, $R_s^* \leqslant \|f\|_{\mathcal{K}} + \gamma \log n$, because

$$(d' - t + 2)\log(\gamma \log n) \geqslant 0$$

Finally, if $d' - t + 2 < 0$ then the smaller solution of (C.21) decreases to $\|f\|_{\mathcal{K}}$ and the larger one R_s^* tends to infinity with growing n. Evaluating both sides of (C.21) for $R = \|f\|_{\mathcal{K}} + 2\gamma \log n$ we notice that

$$(d' - t + 2)\log(2\gamma \log n) \geqslant -2\gamma \log n$$

for n large enough, as $\log n$ increases faster than $\log \log n$. This shows that $R_s^* \leqslant \|f\|_{\mathcal{K}} + 3\gamma \log n$. Thus there exists a constant C independent of n, such that, for any $n \geqslant 1$:

$$\begin{aligned} \mathbb{E}\left[\left\|[\hat{\mu}^{k_z}_{f(X)} - \mu^{k_z}_{f(X)}]\right\|^2_{k_z}\right] &\leqslant C(\log n)^{d'} n^{-2\nu} \\ &= O\left((\log n)^{d'} n^{-2\nu}\right) \,. \end{aligned}$$

□

C.2.2 Detailed Proof of 2.3.1

Proof.

$$
\begin{aligned}
&\left\|\hat{\mu}_{XY}^{k_{xy}} - \mu_{XY}^{k_{xy}}\right\|_{k_{xy}} = \left\|\hat{\mu}_{X}^{k_x} \otimes \hat{\mu}_{Y}^{k_y} - \mu_{X}^{k_x} \otimes \mu_{Y}^{k_y}\right\|_{k_{xy}} \\
&= \left\|\hat{\mu}_{X}^{k_x} \otimes \hat{\mu}_{Y}^{k_y} - \hat{\mu}_{X}^{k_x} \otimes \mu_{Y}^{k_y} + \hat{\mu}_{X}^{k_x} \otimes \mu_{Y}^{k_y} - \mu_{X}^{k_x} \otimes \mu_{Y}^{k_y}\right\|_{k_{xy}} \\
&= \left\|\hat{\mu}_{X}^{k_x} \otimes (\hat{\mu}_{Y}^{k_y} - \mu_{Y}^{k_y}) + (\hat{\mu}_{X}^{k_x} - \mu_{X}^{k_x}) \otimes \mu_{Y}^{k_y}\right\|_{k_{xy}} \\
&\leqslant \left\|\hat{\mu}_{X}^{k_x}\right\|_{k_x} \left\|\hat{\mu}_{Y}^{k_y} - \mu_{Y}^{k_y}\right\|_{k_y} + \left\|\mu_{Y}^{k_y}\right\|_{k_y} \left\|\hat{\mu}_{X}^{k_x} - \mu_{X}^{k_x}\right\|_{k_x} \\
&= \left\|\mu_{X}^{k_x} + \hat{\mu}_{X}^{k_x} - \mu_{X}^{k_x}\right\|_{k_x} \left\|\hat{\mu}_{Y}^{k_y} - \mu_{Y}^{k_y}\right\|_{k_y} + \left\|\mu_{Y}^{k_y}\right\|_{k_y} \left\|\hat{\mu}_{X}^{k_x} - \mu_{X}^{k_x}\right\|_{k_x} \\
&\leqslant \begin{cases} \left\|\mu_{X}^{k_x}\right\|_{k_x} \left\|\hat{\mu}_{Y}^{k_y} - \mu_{Y}^{k_y}\right\|_{k_y} + \left\|\mu_{Y}^{k_y}\right\|_{k_y} \left\|\hat{\mu}_{X}^{k_x} - \mu_{X}^{k_x}\right\|_{k_x} \\ + \left\|\hat{\mu}_{X}^{k_x} - \mu_{X}^{k_x}\right\|_{k_x} \left\|\hat{\mu}_{Y}^{k_y} - \mu_{Y}^{k_y}\right\|_{k_y} \end{cases} \\
&= O(s_n) + O(s_n) + O(s_n^2) = O(s_n + s_n^2).
\end{aligned}
$$

□

C.3 Chapter 3

C.3.1 Proof of Theorem 3.1.3

Proof. $\Leftarrow$: suppose that k is not characteristic to $P \in \mathcal{P}$. Then there exists $Q \in \mathcal{P}$ s.t. $Q \neq P$ but $\Phi_k(P) = \Phi_k(Q)$. Then the constant sequence $P_n = Q$ does not converge weakly to P, but converges to P in kernel metric.

$\Rightarrow$: Let k be continuous, characteristic to P and bounded. Let P_n be a sequence of probability measures. We will first show that if P_n is tight and converges to P in kernel metric, then $P_n \to_\sigma P$. Suppose that P_n did not converge weakly to P. Then there exists a subsequence $(P_l)_l$ of $(P_n)_n$ and a neighborhood of P which does not contain any P_l. $(P_l)_l$ is tight, thus by Prokhorov's theorem, $(P_l)_l$ is a precompact subset of $\mathcal{P}(\mathcal{X})$. As $\mathcal{P}(\mathcal{X})$ is complete, we may again extract a subsequence $(P_h)_h$ of $(P_l)_l$ which converges to a probability measure P', which cannot be P (because P_h is 'bounded away' from P). As k is continuous and bounded, any function $f \in \mathcal{H}_k$ is also in $\mathscr{C}_b$, thus $P_h(f) \to P'(f) = \langle f, \Phi_k(P')\rangle_k$ for any $f \in \mathcal{H}_k$. But we also have $\|P_h - P\|_k \to 0$, so by continuity of the inner product we have: $P_h(f) = \langle f, \Phi_k(P_h)\rangle_k \to \langle f, \Phi_k(P)\rangle_k = P(f)$. Thus, for any f in $\mathcal{H}_k$, we have: $\langle f, \Phi_k(P)\rangle_k = \langle f, \Phi_k(P')\rangle_k$, thus $\Phi_k(P) = \Phi_k(P')$, thus $P = P'$. Contradiction. Thus $P_n \to_\sigma P$.

Conversely, suppose that $P_n \to_\sigma P$. Then by Prokohrov's theorem, P_n is tight. Now let us show that $\mathrm{MMD}_k(P_n, P) \to 0$. As $P_n - P \to 0$ (in the space of finite signed measures), the tensor product $(P_n - P) \otimes (P_n - P)$ defined over $\mathcal{X} \times \mathcal{X}$ also converges weakly to 0 [9]. Thus:

[9] Berg et al., *Harmonic Analysis on Semigroups Theory of Positive Definite and Related Functions*, 1984, Chap 2, Prop 3.3

$$\begin{aligned}\mathrm{MMD}_k(P_n, P)^2 &= \iint k(x, y)\, d(P_n - P)(x)\, d(P_n - P)(y) \qquad \text{(C.22)}\\ &= [(P_n - P) \otimes (P_n - P)](k) \longrightarrow 0. \qquad \square\end{aligned}$$

C.3.2 Proof of Theorem 3.1.4

Proof. $\Rightarrow$: Let k be continuous, characteristic to P and $x \mapsto k(x, x)$ be $O(|x|^{2p})$. Let P_n be a tight sequence of probability measures with uniformly integrable α-moments that converges to P in kernel metric. Suppose for a moment that P_n did not weakly-α converge to P. Then we could find a weak-α neighborhood of P that does not contain any element of a subsequence P_l of P_n. By Proposition 7.1.5 of [2], the set $\{P_l\}_l$ would be precompact for the W_α norm. Thus, by Lemma 5.1.7, it would also be weak-α precompact, and we could extract a subsequence P_h of P_l that converges to a measure $P' \in \mathcal{P}_\alpha$. We would then get a contradiction, as in the preceding proof. Thus $P_n \to_\alpha P$.

[2] Ambrosio et al., *Gradient Flows*, 2005

Conversely, suppose that $P_n \to_\alpha P$, and define $\tilde{P}$ as the measure $\tilde{P}(A) := \int_A (1+|x|^\alpha)\,dP(x)$ and $\tilde{k}(x,y) := k(x,y)/((1+|x|^\alpha)(1+|y|^\alpha))$. Then $\tilde{P}_n \to \tilde{P}$, and using (C.22), we conclude that

$$\begin{aligned}\mathrm{MMD}_k(P_n,P)^2 &= \iint k(x,y)\,d(P_n-P)(x)\,d(P_n-P)(y)\\ &= \iint \frac{k(x,y)}{(1+|x|^\alpha)(1+|y|^\alpha)}\,d(\tilde{P}_n-\tilde{P})(x)\,d(\tilde{P}_n-\tilde{P})(y)\\ &= d_{\tilde{k}}(\tilde{P}_n,\tilde{P})^2 \longrightarrow 0\end{aligned}$$

$\Leftarrow$: Same as in the proof of Theorem 3.1.3. □

C.3.3 Proof of Proposition 3.2.5

Proof. Define the Lipschitz constant $M_1(b) = \sup_{x\neq y}|b(x)-b(y)|_2/|x-y|_2$. Since $k \in \mathscr{C}^{(1,1)}$ and b Lipschitz, $k_b \in \mathscr{C}^{(1,1)}$. Moreover, for each x, $k_b(x,\cdot) \in \mathscr{C}_{\to 0}$ and $\partial_x\partial_y k_b(x,\cdot) = \partial_x b(x)(\partial_x\partial_y k)(b(x),b(\cdot)))\partial_y b(\cdot)^T \in \mathscr{C}_{\to 0}$ since b is norm-coercive and Lipschitz and $k \in \mathscr{C}^{(1,1)}_{\to 0}$. In addition, if $|\cdot|_2$ designates the spectral norm for matrices, then $k \in \mathscr{C}^{(1,1)}_{\to 0}$ implies

$$\begin{aligned}&\sup_{x\in\mathcal{X}} \max(k_b(x,x), |(\partial_x\partial_y k_b)(x,x)|_2)\\ &\qquad\leqslant \sup_{y\in\mathcal{X}} \max(k(y,y), M_1(b)^2|(\partial_x\partial_y k)(y,y)|_2) < \infty,\end{aligned}$$

so $k_b \in \mathscr{C}^{(1,1)}_{\to 0}$. Now, select any $f \in \mathscr{C}^1_{\to 0}$ and any $\epsilon > 0$. Since $\mathscr{C}^1_c$ is dense in $\mathscr{C}^1_{\to 0}$, choose any $h \in \mathscr{C}^1_c$ with $\sup_{x\in\mathcal{X}}|f(x)-h(x)| \leqslant \epsilon$. Next, let $c = b^{-1}$ represent the inverse of b and $h_c(y) = h(c(y))$ denote the composition of h and c so that $h_c(b(x)) = h(x)$. By [19], $h_c \in \mathscr{C}^1_c \subset \mathscr{C}^1_{\to 0}$, and hence there exists $h_{c,\epsilon} \in \mathcal{H}_k$ such that

[19] Chen et al., *Stein Points*, 2018, Lem 8

$$\sup_{y\in\mathcal{X}} \max(|\partial_y(h_c - h_{c,\epsilon})(y)|_2, |(h_c - h_{c,\epsilon})(y)|_2) \leqslant \epsilon/\max(1,M_1(b)).$$

Now define $h_\epsilon(x) = h_{c,\epsilon}(b(x))$ so that $h_\epsilon \in \mathcal{H}_{k_b}$ by [19]. We have $\sup_{x\in\mathcal{X}}|h_\epsilon(x)-h(x)|_2 \leqslant \sup_{y\in\mathcal{X}}|h_{c,\epsilon}(y)-h_c(y)|_2 \leqslant \epsilon$, and

[19] Chen et al., *Stein Points*, 2018, Lem 9

$$\begin{aligned}\sup_{x\in\mathcal{X}}|\partial_x(h_\epsilon - h)(x)|_2 &= \sup_{x\in\mathcal{X}}|(\partial_x b(x))\partial_x(h_{c,\epsilon}-h_c)(b(x))|_2\\ &\leqslant M_1(b)\epsilon/\max(1,M_1(b)) \leqslant \epsilon,\end{aligned}$$

demonstrating that every $f \in \mathscr{C}^1_{\to 0}$ is 2ϵ-approximated by some $h_\epsilon \in \mathcal{H}_{k_b}$. Hence, k_b is c^1_0-universal. □

C.3.4 Proof of Theorem 3.2.3

Our proof mimics the proof of Theorem 2.2 in Chwialkowski et al. [21] (reproduced in Appendix B.1.1), but uses Schwartz-differentiation instead of the usual differentiation. More precisely, they work with expressions of the form $s_Q(x)\,dQ(x)$, which assumes that Q has a differentiable density, otherwise $s_Q(x) := \partial_x \log(q(x)) = \partial_x q(x)/q(x)$ is not defined. Instead, we replace those expressions by $\partial_x Q(x)\,dx$, which is a well defined order 1 Schwartz-distribution for any probability measure Q, and coincides with $s_Q(x)q(x)\,dx$ when Q has a differentiable density, because

[21] Chwialkowski et al., *A Kernel Test of Goodness of Fit*, 2016

$$s_Q(x)q(x)\,dx = \frac{\partial_x q(x)}{q(x)} q(x)\,dx = \partial_x q(x)\,dx = \partial_x Q(x)\,dx\,.$$

We first generalize Lemma 1 of Chwialkowski et al. [21], which essentially states that any κ-embeddable probability measure Q with differentiable density satisfies $\mathbb{E}_{x\sim Q}\left[\xi_Q(x,\cdot)\right] = 0$.

[21] Chwialkowski et al., *A Kernel Test of Goodness of Fit*, 2016

Lemma C.3.1. *Let k be a kernel in $\mathscr{C}_b^{(1,1)}$. Then for any probability measure Q and any $y \in \mathcal{X}$:*

$$\int_{\mathbb{R}^d} \sum_{1\leqslant i\leqslant d} k(x,y)\,\partial_{x_i} Q(x)\,dx + \partial_{x_i} k(x,y)\,Q(x)\,dx = 0 \tag{C.23}$$

Proof. (of Lemma C.3.1) Suppose that $k \in \mathscr{C}_b^{(1,1)}$. Then Corollary 1.1.4 shows that set of integrable distributions $\mathcal{D}^1_{L^1}$ embeds into $\mathcal{H}_k$. But $\partial_{x_i} Q \in \mathcal{D}^1_{L^1}$ for any i, thus $\partial_{x_i} Q$ embeds into $\mathcal{H}_k$. Thus Proposition 1.4.2 shows that, for any $y \in \mathcal{X}$ and $1 \leqslant i \leqslant d$:

$$\int k(x,y)\,\partial_{x_i} Q(x)\,dx = -\int \partial_{x_i} k(x,y)\,Q(x)\,dx\,. \qquad \square$$

We now finally proceed with the main proof.

Proof. (of Theorem 3.2.3) Combining the definitions of $\mathrm{KSD}_{k,P}(Q)$ and of characteristicness to $P \in \mathcal{P}_\alpha$ shows that statements (i) and (ii) are equivalent. Furthermore, applying Theorem 3.1.4 with κ and (i) proves the equivalence between $P_n \to_\alpha P$ and (a) and (c).

We now show the equivalence of (iii) and (ii). To do so, we use Lemma C.3.1 to rewrite ξ^i_P, and then conclude with Proposition 3.2.2. Indeed, because $k \in \mathscr{C}_b^{(1,1)}$, $\partial_x Q$ embeds into $\mathcal{H}_k$

(see proof of Lemma C.3.1). We may thus consider its embedding $\int k(x,\cdot)\partial_x Q(x)\,dx$ and add and subtract it from $\mathbb{E}_Q\left[\xi_P^i\right]$ as follows.

$$\begin{aligned}\mathop{\mathbb{E}}_{Q}\left[\xi_P^i\right] &= \int \Big(k(x,\cdot)s_P^i(x) + \partial_{x_i}k(x,\cdot)\Big)Q(x)\,dx \\ &= \int \Big(k(x,\cdot)\partial_{x_i}Q(x)\,dx + \partial_{x_i}k(x,\cdot)Q(x)\,dx\Big) \\ &\qquad + \int k(x,\cdot)\big(s_P^i(x)Q(x)\,dx - \partial_{x_i}Q(x)\,dx\big) \\ &= 0 + \int k(x,\cdot)\,D_i(x)\,dx\,.\end{aligned}$$

Here D_i denotes the Schwartz-distribution $s_P^i\,Q - \partial_{x_i}Q$, and the 0 comes from Lemma C.3.1. We conclude by noticing that, by Proposition 3.2.2, (ii) is equivalent to: for any $Q \in P_\alpha$, $Q = P$ iff for all $1 \leqslant i \leqslant d$, $\mathbb{E}_Q\left[\xi_P^i\right] = 0$, i.e. iff for all i, $D_i = 0$, which is condition (iii). Thus (ii) and (iii) are equivalent. □

C.4 Chapter 5

C.4.1 Proof of Lemma 5.1.1

For the first inequality, we use the fact that D_f is jointly convex. We write $P_X = (1-\beta)\frac{P_X - \beta R}{1-\beta} + \beta R$ which is a convex combination of two distributions when the assumptions are satisfied. The second inequality follows from using the triangle inequality for $\sqrt{D_f}$ and using convexity of D_f in its first argument.

C.4.2 Proof of Theorem 5.1.2

Before proving Theorem 5.1.2, we introduce two lemmas. The first one is about the determination of the constant λ, the second one is about comparing the divergences of mixtures.

Lemma C.4.1. *Let P and Q be two distributions, $\gamma \in [0,1]$ and $\lambda \in \mathbb{R}$. The function*

$$g(\lambda) := \int \left(\lambda - \gamma \frac{dQ}{dP}\right)_+ dP$$

is nonnegative, convex, nondecreasing, satisfies $g(\lambda) \leqslant \lambda$, and its right derivative is given by

$$g'_+(\lambda) = P\left(\lambda \cdot dP \geqslant \gamma \cdot dQ\right).$$

The equation $g(\lambda) = 1-\gamma$ has a solution λ^* (unique when $\gamma < 1$) with $\lambda^* \in [1-\gamma, 1]$. Finally, if $P(dQ = 0) \geqslant \delta$ for a strictly positive constant δ then $\lambda^* \leqslant (1-\gamma)\delta^{-1}$.

Proof. The convexity of g follows immediately from the convexity of $x \mapsto (x)_+$ and the linearity of the integral. Similarly, since $x \mapsto (x)_+$ is non-decreasing, g is non-decreasing.

We define the set $\mathcal{I}(\lambda)$ as follows:

$$\mathcal{I}(\lambda) := \{x \in \mathcal{X} : \lambda \cdot dP(x) \geqslant \gamma \cdot dQ(x)\}.$$

Now let us consider $g(\lambda+\epsilon) - g(\lambda)$ for some small $\epsilon > 0$. This can also be written:

$$\begin{aligned} g(\lambda+\epsilon) - g(\lambda) &= \int_{\mathcal{I}(\lambda)} \epsilon dP + \int_{\mathcal{I}(\lambda+\epsilon)\setminus\mathcal{I}(\lambda)} (\lambda+\epsilon) dP - \int_{\mathcal{I}(\lambda+\epsilon)\setminus\mathcal{I}(\lambda)} \gamma dQ \\ &= \epsilon P(\mathcal{I}(\lambda)) + \int_{\mathcal{I}(\lambda+\epsilon)\setminus\mathcal{I}(\lambda)} (\lambda+\epsilon) dP - \int_{\mathcal{I}(\lambda+\epsilon)\setminus\mathcal{I}(\lambda)} \gamma dQ. \end{aligned}$$

On the set $\mathcal{I}(\lambda+\epsilon)\setminus\mathcal{I}(\lambda)$, we have

$$(\lambda+\epsilon) dP - \gamma dQ \in [0, \epsilon].$$

So that

$$\epsilon P(\mathcal{I}(\gamma)) \leqslant g(\lambda+\epsilon)-g(\lambda) \leqslant \epsilon P(\mathcal{I}(\gamma)) + \epsilon P\big(\mathcal{I}(\lambda+\epsilon)\backslash\mathcal{I}(\lambda)\big) = \epsilon P(\mathcal{I}(\lambda+\epsilon))$$

and thus

$$\lim_{\epsilon\to 0^+} \frac{g(\lambda+\epsilon)-g(\lambda)}{\epsilon} = \lim_{\epsilon\to 0^+} P(\mathcal{I}(\lambda+\epsilon)) = P(\mathcal{I}(\lambda)).$$

This gives the expression of the right derivative of g. Moreover, notice that for $\lambda, \gamma > 0$

$$g'_+(\lambda) = P\left(\lambda \cdot dP \geqslant \gamma \cdot dQ\right) = P\left(\frac{dQ}{dP} \leqslant \frac{\lambda}{\gamma}\right) = 1 - P\left(\frac{dQ}{dP} > \frac{\lambda}{\gamma}\right) \geqslant 1-\gamma/\lambda$$

by Markov's inequality.

It is obvious that $g(0) = 0$. By Jensen's inequality applied to the convex function $x \mapsto (x)_+$, we have $g(\lambda) \geqslant (\lambda-\gamma)_+$. So $g(1) \geqslant 1-\gamma$. Also, $g = 0$ on $\mathbb{R}^-$ and $g \leqslant \lambda$. This means g is continuous on $\mathbb{R}$ and thus reaches the value $1-\gamma$ on the interval $(0,1]$ which shows the existence of $\lambda^* \in (0,1]$. To show that λ^* is unique we notice that since $g(x) = 0$ on $\mathbb{R}^-$, g is convex and non-decreasing, g cannot be constant on an interval not containing 0, and thus $g(x) = 1-\gamma$ has a unique solution for $\gamma < 1$.

Also by convexity of g,

$$g(0) - g(\lambda^*) \geqslant -\lambda^* g'_+(\lambda^*),$$

which gives $\lambda^* \geqslant (1-\gamma)/g'_+(\lambda^*) \geqslant 1-\gamma$ since $g'_+ \leqslant 1$. If $P(dQ = 0) \geqslant \delta > 0$ then also $g'_+(0) \geqslant \delta > 0$. Using the fact that g'_+ is increasing we conclude that $\lambda^* \leqslant (1-\gamma)\delta^{-1}$. □

Next we introduce some simple convenience lemma for comparing convex functions of random variables.

Lemma C.4.2. *Let f be a convex function, X, Y be real-valued random variables and $c \in \mathbb{R}$ be a constant such that*

$$\mathbb{E}[\max(c, Y)] = \mathbb{E}[X+Y].$$

Then we have the following bound:

$$\mathbb{E}[f(\max(c,Y))] \leqslant \mathbb{E}[f(X+Y)] - \mathbb{E}\big[X(f'(Y)-f'(c))_+\big] \leqslant \mathbb{E}[f(X+Y)]. \tag{C.24}$$

If in addition, $Y \leqslant M$ a.s. for $M \geqslant c$, then

$$\mathbb{E}[f(\max(c,Y))] \leqslant f(c) + \frac{f(M)-f(c)}{M-c}(\mathbb{E}[X+Y]-c). \tag{C.25}$$

Proof. We decompose the expectation with respect to the value of the max and use the convexity of f:

$$\begin{aligned}
&f(X+Y) - f(\max(c,Y)) \\
&\quad = \mathbb{1}_{[Y\leqslant c]}(f(X+Y) - f(c)) \\
&\qquad + \mathbb{1}_{[Y>c]}(f(X+Y) - f(Y)) \\
&\quad \geqslant \mathbb{1}_{[Y\leqslant c]} f'(c)(X+Y-c) + \mathbb{1}_{[Y>c]} X f'(Y) \\
&\quad = (1 - \mathbb{1}_{[Y>c]}) X f'(c) + f'(c)(Y - \max(c,Y)) \\
&\qquad + \mathbb{1}_{[Y>c]} X f'(Y) \\
&\quad = f'(c)(X+Y-\max(c,Y)) \\
&\qquad + \mathbb{1}_{[Y>c]} X(f'(Y) - f'(c)) \\
&\quad = f'(c)(X+Y-\max(c,Y)) + X(f'(Y) - f'(c))_+,
\end{aligned}$$

where we used that f' is non-decreasing in the last step. Taking the expectation gives the first inequality.

For the second inequality, we use the convexity of f on the interval $[c, M]$:

$$f(\max(c,Y)) \leqslant f(c) + \frac{f(M) - f(c)}{M - c}(\max(c,Y) - c).$$

Taking an expectation on both sides gives the second inequality. □

Theorem 5.1.2. We first apply Lemma C.4.1 with $\gamma = 1 - \beta$ and this proves the existence of λ^* in the interval $(\beta, 1]$, which shows that R^*_β is indeed well-defined as a distribution.

Then we use Inequality (C.24) of Lemma C.4.2 with $X = \beta \, dQ/dP_X$, $Y = (1-\beta) dP_Y/dP_X$, and $c = \lambda^*$. We easily verify that $X + Y = ((1-\beta) dP_Y + \beta \, dQ)/dP_X$ and $\max(c, Y) = ((1-\beta) dP_Y + \beta \, dQ^*_\beta)/dP_X$ and both have expectation 1 with respect to P_X. We thus obtain for any distribution Q,

$$D_f\left((1-\beta)P_Y + \beta Q^*_\beta \,\middle\|\, P_X\right) \leqslant D_f\left((1-\beta)P_Y + \beta Q \,\|\, P_X\right).$$

This proves the optimality of R^*_β. □

C.4.3 Proof of Theorem 5.1.3

Lemma C.4.3. *Let P and Q be two distributions, $\gamma \in (0,1)$, and $\lambda \geqslant 0$. The function*

$$h(\lambda) := \int \left(\frac{1}{\gamma} - \lambda \frac{dQ}{dP}\right)_+ dP$$

is convex, non-increasing, and its right derivative is given by $h'_+(\lambda) = -Q(1/\gamma \geqslant \lambda dQ(X)/dP(X))$. *Denote* $\Delta := P(dQ(X)/dP(X) = 0)$. *Then the equation*

$$h(\lambda) = \frac{1-\gamma}{\gamma}$$

has no solutions if $\Delta > 1-\gamma$, *has a single solution* $\lambda^\dagger \geqslant 1$ *if* $\Delta < 1-\gamma$, *and has infinitely many or no solutions when* $\Delta = 1-\gamma$.

Proof. The convexity of h follows immediately from the convexity of $x \mapsto (a-x)_+$ and the linearity of the integral. Similarly, since $x \mapsto (a-x)_+$ is non-increasing, h is non-increasing as well.

We define the set $\mathcal{J}(\lambda)$ as follows:

$$\mathcal{J}(\lambda) := \left\{ x \in \mathcal{X} \colon \frac{1}{\gamma} \geqslant \lambda \frac{dQ}{dP}(x) \right\}.$$

Now let us consider $h(\lambda) - h(\lambda+\epsilon)$ for any $\epsilon > 0$. Note that $\mathcal{J}(\lambda+\epsilon) \subseteq \mathcal{J}(\lambda)$. We can write:

$$\begin{aligned}
&h(\lambda) - h(\lambda+\epsilon) \\
&= \int_{\mathcal{J}(\lambda)} \left(\frac{1}{\gamma} - \lambda \frac{dQ}{dP} \right) dP - \int_{\mathcal{J}(\lambda+\epsilon)} \left(\frac{1}{\gamma} - (\lambda+\epsilon) \frac{dQ}{dP} \right) dP \\
&= \int_{\mathcal{J}(\lambda) \setminus \mathcal{J}(\lambda+\epsilon)} \left(\frac{1}{\gamma} - \lambda \frac{dQ}{dP} \right) dP + \int_{\mathcal{J}(\lambda+\epsilon)} \left(\epsilon \frac{dQ}{dP} \right) dP \\
&= \int_{\mathcal{J}(\lambda) \setminus \mathcal{J}(\lambda+\epsilon)} \left(\frac{1}{\gamma} - \lambda \frac{dQ}{dP} \right) dP + \epsilon \cdot Q(\mathcal{J}(\lambda+\epsilon)).
\end{aligned}$$

Note that for $x \in \mathcal{J}(\lambda) \setminus \mathcal{J}(\lambda+\epsilon)$ we have

$$0 \leqslant \frac{1}{\gamma} - \lambda \frac{dQ}{dP}(x) < \epsilon \frac{dQ}{dP}(x).$$

This gives the following:

$$\begin{aligned}
\epsilon \cdot Q(\mathcal{J}(\lambda+\epsilon)) &\leqslant h(\lambda) - h(\lambda+\epsilon) \\
&\leqslant \epsilon \cdot Q(\mathcal{J}(\lambda+\epsilon)) + \epsilon \cdot Q(\mathcal{J}(\lambda) \setminus \mathcal{J}(\lambda+\epsilon)) \\
&= \epsilon \cdot Q(\mathcal{J}(\lambda)),
\end{aligned}$$

which shows that h is continuous. Also

$$\begin{aligned}
\lim_{\epsilon \to 0^+} \frac{h(\lambda+\epsilon) - h(\lambda)}{\epsilon} &= \lim_{\epsilon \to 0^+} -Q(\mathcal{J}(\lambda+\epsilon)) \\
&= -Q(\mathcal{J}(\lambda)).
\end{aligned}$$

It is obvious that $h(0) = 1/\gamma$ and $h \leqslant \gamma^{-1}$ for $\lambda \geqslant 0$. By Jensen's inequality applied to the convex function $x \mapsto (a-x)_+$, we have

$h(\lambda) \geqslant \left(\gamma^{-1} - \lambda\right)_+$. So $h(1) \geqslant \gamma^{-1} - 1$. We conclude that h may reach the value $(1-\gamma)/\gamma = \gamma^{-1} - 1$ only on $[1, +\infty)$. Note that

$$h(\lambda) \to \frac{1}{\gamma} P\left(\frac{dQ}{dP}(X) = 0\right) = \frac{\Delta}{\gamma} \geqslant 0 \quad \text{as} \quad \lambda \to \infty.$$

Thus if $\Delta/\gamma > \gamma^{-1} - 1$ the equation $h(\lambda) = \gamma^{-1} - 1$ has no solutions, as h is non-increasing. If $\Delta/\gamma = \gamma^{-1} - 1$ then either $h(\lambda) > \gamma^{-1} - 1$ for all $\lambda \geqslant 0$ and we have no solutions or there is a finite $\lambda' \geqslant 1$ such that $h(\lambda') = \gamma^{-1} - 1$, which means that the equation is also satisfied by all $\lambda \geqslant \lambda'$, as h is continuous and non-increasing. Finally, if $\Delta/\gamma < \gamma^{-1} - 1$ then there is a unique $\lambda^\dagger$ such that $h(\lambda^\dagger) = \gamma^{-1} - 1$, which follows from the convexity of h. □

Next we introduce some simple convenience lemma for comparing convex functions of random variables.

Lemma C.4.4. *Let f be a convex function, X, Y be real-valued random variables such that $X \leqslant Y$ a.s., and $c \in \mathbb{R}$ be a constant such that*[2]

$$\mathbb{E}[\min(c, Y)] = \mathbb{E}[X].$$

Then we have the following lower bound:

$$\mathbb{E}[f(X) - f(\min(c, Y))] \geqslant 0.$$

[2] Generally it is not guaranteed that such a constant c always exists. In this result we assume this is the case.

Proof. We decompose the expectation with respect to the value of the min, and use the convexity of f:

$$\begin{aligned}
&f(X) - f(\min(c, Y)) \\
&\quad = \mathbb{1}_{[Y \leqslant c]}(f(X) - f(Y)) + \mathbb{1}_{[Y > c]}(f(X) - f(c)) \\
&\quad \geqslant \mathbb{1}_{[Y \leqslant c]} f'(Y)(X - Y) + \mathbb{1}_{[Y > c]}(X - c) f'(c) \\
&\quad \geqslant \mathbb{1}_{[Y \leqslant c]} f'(c)(X - Y) + \mathbb{1}_{[Y > c]}(X - c) f'(c) \\
&\quad = X f'(c) - \min(Y, c) f'(c),
\end{aligned}$$

where we used the fact that f' is non-decreasing in the previous to last step. Taking the expectation we get the result. □

Lemma C.4.5. *Let P_Y, P_X be two fixed distributions and $\beta \in (0, 1)$. Assume*

$$P_X\left(\frac{dP_Y}{dP_X} = 0\right) < \beta.$$

Let $\mathcal{M}(P_X, \beta)$ be the set of all probability distributions T such that $(1 - \beta)dT \leqslant dP_X$. Then the following minimization problem:

$$\min_{T \in \mathcal{M}(P_X, \beta)} D_f(T \| P_Y)$$

has the solution T^* *with density*

$$dT^* := \min(dP_X/(1-\beta), \lambda^\dagger dP_Y),$$

where $\lambda^\dagger$ *is the unique value in* $[1,\infty)$ *such that* $\int dT^* = 1$.

Proof. We will use Lemma C.4.4 with $X = dT(Z)/dP_Y(Z)$, $Y = dP_X(Z)/\big((1-\beta)dP_Y(Z)\big)$, and $c = \lambda^*$, $Z \sim P_Y$. We need to verify that assumptions of Lemma C.4.4 are satisfied. Obviously, $Y \geqslant X$. We need to show that there is a constant c such that

$$\int \min\left(c, \frac{dP_X}{(1-\beta)dP_Y}\right) dP_Y = 1.$$

Rewriting this equation we get the following equivalent one:

$$\begin{aligned}\beta &= \int (dP_X - \min(c(1-\beta)P_Y, dP_X)) \\ &= (1-\beta)\int \left(\frac{1}{1-\beta} - c\frac{dP_Y}{dP_X}\right)_+ dP_X.\end{aligned} \tag{C.26}$$

Using the fact that

$$P_X\left(\frac{dP_Y}{dP_X} = 0\right) < \beta$$

we may apply Lemma C.4.3 and conclude that there is a unique $c \in [1,\infty)$ satisfying (C.26), which we denote $\lambda^\dagger$. □

To conclude the proof of Theorem 5.1.3, observe that from Lemma C.4.5, by making the change of variable $T = (P_X - \beta Q)/(1-\beta)$ we can rewrite the minimization problem as follows:

$$\min_{Q:\, \beta dQ \leqslant dP_X} D_{f^\circ}\left(P_Y \middle\| \frac{P_X - \beta Q}{1-\beta}\right)$$

and we verify that the solution has the form $dQ^\dagger_\beta = \frac{1}{\beta}\big(dP_X - \lambda^\dagger(1-\beta)dP_Y\big)_+$. Since this solution does not depend on f, the fact that we optimized D_{f° is irrelevant and we get the same solution for D_f.

C.4.4 Proof of Lemma 5.1.4

The first inequality follows from the optimality of R^*_β (hence the value of the objective at R^*_β is smaller than at P_X), and the fact that D_f is convex in its first argument. The second inequality follows from the optimality of $R^\dagger_\beta$ (hence the objective at $R^\dagger_\beta$ is smaller than its value at P_X which itself satisfies the condition $\beta dP_X \leqslant dP_X$). For the third inequality, we combine the second inequality with the first inequality of Lemma 5.1.1 (with $Q = R = R^\dagger_\beta$).

C.4.5 Proof of Corollaries 5.1.5 and 5.1.6

For Corollary 5.1.5, combine Lemma 5.1.1, Theorem 5.1.2, and Lemma 5.1.4. Corollary 5.1.6 immediately follows from Lemma 5.1.1, Theorem 5.1.3, and Lemma 5.1.4. It is easy to verify that for $\gamma < \beta/4$, the coefficient is less than $(\beta/2 + \sqrt{1-\beta})^2 < 1$ (for $\beta > 0$).

C.5 Chapter 6

C.5.1 Proof of Proposition 6.1.3

Proof. Let $\epsilon\,\delta$ be an adversarial perturbation with $\|\delta\| = 1$ that locally maximizes the loss increase at point x, meaning that $\delta = \arg\max_{\|\delta'\|\leqslant 1} \partial_x \mathcal{L} \cdot \delta'$. Then, by definition of the dual norm of $\partial_x \mathcal{L}$ we have: $\partial_x \mathcal{L} \cdot (\epsilon\delta) = \epsilon \, |||\partial_x \mathcal{L}|||$. Thus

$$\tilde{\mathcal{L}}_{\epsilon,\|\cdot\|}(x,c) = \frac{1}{2}(\mathcal{L}(x,c) + \mathcal{L}(x+\epsilon\,\delta,c)) = \frac{1}{2}(2\mathcal{L}(x,c) + \epsilon\,|\partial_x\mathcal{L}\cdot\delta| + o(\|\delta\|)) =$$
$$= \mathcal{L}(x,c) + \frac{\epsilon}{2}\,|||\partial_x\mathcal{L}||| + o(\epsilon) = \mathcal{L}_{\epsilon,|||\cdot|||}(x,c) + o(\epsilon)\,. \quad (\square)$$

C.5.2 Proof of Theorem 6.2.1

Proof. Let x designate a generic coordinate of x. To evaluate the size of $\|\partial_x \mathcal{L}\|_q$, we will evaluate the size of the coordinates $\partial_x \mathcal{L}$ of $\partial_x \mathcal{L}$ by decomposing them into

$$\partial_x \mathcal{L} = \sum_{k=1}^{K} \frac{\partial \mathcal{L}}{\partial f_k} \frac{\partial f_k}{\partial x} =: \sum_{k=1}^{K} \partial_k \mathcal{L}\, \partial_x f_k,$$

where $f_k(x)$ denotes the logit-probability of x belonging to class k. We now investigate the statistical properties of the logit gradients $\partial_x f_k$, and then see how they shape $\partial_x \mathcal{L}$.

Step 1: Statistical properties of $\partial_x f_k$. Let $\mathcal{P}(x,k)$ be the set of paths p from input neuron x to output-logit k. Let $p-1$ and p be two successive neurons on path p, and $\tilde{p}$ be the same path p but without its input neuron. Let w_p designate the weight from $p-1$ to p and ω_p be the *path-product* $\omega_p := \prod_{p\in\tilde{p}} w_p$. Finally, let σ_p (resp. σ_p) be equal to 1 if the ReLU of node p (resp. if path p) is active for input x, and 0 otherwise.

As previously noticed by [5] using the chain rule, we see that $\partial_x f_k$ is the sum of all ω_p whose path is active, i.e. $\partial_x f_k(x) = \sum_{p\in\mathcal{P}(x,k)} \omega_p \sigma_p$. Consequently:

[5] Balduzzi et al., *Neural Taylor Approximations*, 2017

$$\mathop{\mathbb{E}}_{W,\sigma}\left[\partial_x f_k(x)^2\right] = \sum_{p\in\mathcal{P}(x,k)} \prod_{p\in\tilde{p}} \mathop{\mathbb{E}}_{W}\left[w_p^2\right] \mathop{\mathbb{E}}_{\sigma}\left[\sigma_p^2\right]$$
$$= |\mathcal{P}(x,k)| \prod_{p\in\tilde{p}} \frac{2}{d_{p-1}} \frac{1}{2} = \prod_{p\in\tilde{p}} d_p \cdot \prod_{p\in\tilde{p}} \frac{1}{d_{p-1}} = \frac{1}{d}\,. \quad (C.27)$$

The first equality uses H1 to decouple the expectations over weights and ReLUs, and then applies Lemma C.5.4 of Appendix C.5.3, which uses H3-H5 to kill all cross-terms and take the expectation over weights inside the product. The second equality uses H3 and the

fact that the resulting product is the same for all active paths. The third equality counts the number of paths from x to k and we conclude by noting that all terms cancel out, except d_{p-1} from the input layer which is d. Equation C.27 shows that $|\partial_x f_k| \propto 1/\sqrt{d}$.

Step 2: Statistical properties of $\partial_k \mathcal{L}$ and $\partial_x \mathcal{L}$. Defining $q_k(\boldsymbol{x}) := \frac{e^{f_k(\boldsymbol{x})}}{\sum_{h=1}^{K} e^{f_h(\boldsymbol{x})}}$ (the probability of image x belonging to class k according to the network), we have, by definition of the cross-entropy loss, $\mathcal{L}(\boldsymbol{x}, c) := -\log q_c(\boldsymbol{x})$, where c is the label of the target class. Thus:

$$\partial_k \mathcal{L}(\boldsymbol{x}) = \begin{cases} -q_k(\boldsymbol{x}) & \text{if } k \neq c \\ 1 - q_c(\boldsymbol{x}) & \text{otherwise,} \end{cases} \quad \text{and}$$

$$\partial_x \mathcal{L}(\boldsymbol{x}) = (1 - q_c)\, \partial_x f_c(\boldsymbol{x}) + \sum_{k \neq c} q_k \, (-\partial_x f_k(\boldsymbol{x})). \tag{C.28}$$

Using again Lemma C.5.4, we see that the $\partial_x f_k(\boldsymbol{x})$ are K centered and uncorrelated variables. So $\partial_x \mathcal{L}(\boldsymbol{x})$ is approximately the sum of K uncorrelated variables with zero-mean, and its total variance is given by $\big((1-q_c)^2 + \sum_{k \neq c} q_k^2\big)/d$. Hence the magnitude of $\partial_x \mathcal{L}(\boldsymbol{x})$ is $1/\sqrt{d}$ for all $\boldsymbol{x}$, so the ℓ_q-norm of the full input gradient is $d^{1/q-1/2}$. (6.6) concludes. □

Remark C.5.1. Equation C.28 can be rewritten as

$$\partial_x \mathcal{L}(\boldsymbol{x}) = \sum_{k=1}^{K} q_k(\boldsymbol{x}) \left(\partial_x f_c(\boldsymbol{x}) - \partial_x f_k(\boldsymbol{x})\right) . \tag{C.29}$$

As the term $k = c$ disappears, the norm of the gradients $\partial_x \mathcal{L}(\boldsymbol{x})$ appears to be controlled by the total error probability. This suggests that, even without regularization, trying to decrease the ordinary classification error is still a valid strategy against adversarial examples. It reflects the fact that when increasing the classification margin, larger gradients of the classifier's logits are needed to push images from one side of the classification boundary to the other. This is confirmed by Theorem 2.1 of [48]. See also (B.3) in Appendix B.3.2.

[48] Hein and Andriushchenko, *Formal Guarantees on the Robustness of a Classifier against Adversarial Manipulation*, 2017

C.5.3 Proof of Theorem 6.2.2

The proof of Theorem 6.2.2 is very similar to the one of Theorem 6.2.1, but we will need to first generalize the equalities appearing in (C.27). To do so, we identify the computational graph of a neural network to an abstract Directed Acyclic Graph (DAG) which we use to prove the needed algebraic equalities. We then concentrate on the statistical weight-interactions implied by assumption $(\mathcal{H})$, and finally throw

these results together to prove the theorem. In all the proof, o will designate one of the output-logits $f_k(\mathbf{x})$.

Lemma C.5.2. *Let $\mathbf{x}$ be the vector of inputs to a given DAG, o be any leaf-node of the DAG, x a generic coordinate of $\mathbf{x}$. Let $\mathbf{p}$ be a path from the set of paths $\mathcal{P}(x,o)$ from x to o, $\tilde{\mathbf{p}}$ the same path without node x, p a generic node in $\tilde{\mathbf{p}}$, and d_p be its input-degree. Then:*

$$\sum_{x\in\mathbf{x}} \sum_{\tilde{\mathbf{p}}\in\mathcal{P}(x,o)} \prod_{p\in\tilde{\mathbf{p}}} \frac{1}{d_p} = 1 \tag{C.30}$$

Proof. We will reason on a random walk starting at o and going up the DAG by choosing any incoming node with equal probability. The DAG being finite, this walk will end up at an input-node x with probability 1. Each path $\mathbf{p}$ is taken with probability $\prod_{p\in\tilde{\mathbf{p}}} \frac{1}{d_p}$. And the probability to end up at an input-node is the sum of all these probabilities, i.e. $\sum_{x\in\mathbf{x}} \sum_{\mathbf{p}\in\mathcal{P}(x,o)} \prod_{p\in\mathbf{p}} d_p^{-1}$, which concludes. □

The sum over all inputs x in (C.30) being 1, on average it is $1/d$ for each x, where d is the total number of inputs (i.e. the length of $\mathbf{x}$). It becomes an equality under assumption (𝒮):

Lemma C.5.3. *Under the symmetry assumption (𝒮), and with the previous notations, for any input $x \in \mathbf{x}$:*

$$\sum_{\mathbf{p}\in\mathcal{P}(x,o)} \prod_{p\in\tilde{\mathbf{p}}} \frac{1}{d_p} = \frac{1}{d}\,. \tag{C.31}$$

Proof. Let us denote $\mathcal{D}(x,o) := \{d_{\mathbf{p}}\}_{x\in\mathcal{P}(x,o)}$. Each path $\mathbf{p}$ in $\mathcal{P}(x,o)$ corresponds to exactly one element $d_{\mathbf{p}}$ in $\mathcal{D}(x,o)$ and vice-versa. And the elements d_p of $d_{\mathbf{p}}$ completely determine the product $\prod_{p\in\tilde{\mathbf{p}}} d_p^{-1}$. By using (C.30) and the fact that, by (𝒮), the multiset $\mathcal{D}(x,o)$ is independent of x, we hence conclude

$$\begin{aligned}\sum_{x\in\mathbf{x}} \sum_{\mathbf{p}\in\mathcal{P}(x,o)} \prod_{p\in\tilde{\mathbf{p}}} \frac{1}{d_p} &= \sum_{x\in\mathbf{x}} \sum_{d_{\mathbf{p}}\in\mathcal{D}(x,o)} \prod_{d_p\in d_{\mathbf{p}}} \frac{1}{d_p} \\ &= d \sum_{d_{\mathbf{p}}\in\mathcal{D}(x,o)} \prod_{d_p\in d_{\mathbf{p}}} \frac{1}{d_p} = 1\,.\end{aligned}$$ □

Now, let us relate these considerations on graphs to gradients and use assumptions (ℋ). We remind that path-product $\omega_{\mathbf{p}}$ is the product $\prod_{p\in\tilde{\mathbf{p}}} w_p$.

Lemma C.5.4. *Under assumptions (ℋ), the path-products $\omega_{\mathbf{p}}, \omega_{\mathbf{p}'}$ of two distinct paths $\mathbf{p}$ and $\mathbf{p}'$ starting from a same input node x, satisfy:*

$$\mathop{\mathbb{E}}_{W}\left[\omega_{\mathbf{p}}\,\omega_{\mathbf{p}'}\right] = 0 \quad \textit{and} \quad \mathop{\mathbb{E}}_{W}\left[\omega_{\mathbf{p}}^2\right] = \prod_{p\in\tilde{\mathbf{p}}} \mathop{\mathbb{E}}_{W}\left[w_p^2\right].$$

Furthermore, if there is at least one non-average-pooling weight on path $\mathbf{p}$*, then* $\mathbb{E}_W[\omega_{\mathbf{p}}] = 0$.

Proof. Hypothesis H4 yields

$$\mathop{\mathbb{E}}_{W}\left[\omega_{\mathbf{p}}^2\right] = \mathop{\mathbb{E}}_{W}\left[\prod_{p\in\tilde{\mathbf{p}}} w_p^2\right] = \prod_{p\in\tilde{\mathbf{p}}} \mathop{\mathbb{E}}_{W}\left[w_p^2\right].$$

Now, take two different paths $\mathbf{p}$ and $\mathbf{p}'$ that start at a same node x. Starting from x, consider the first node after which $\mathbf{p}$ and $\mathbf{p}'$ part and call p and p' the next nodes on $\mathbf{p}$ and $\mathbf{p}'$ respectively. Then the weights w_p and $w_{p'}$ are two weights of a same node. Applying H4 and H5 hence gives

$$\mathop{\mathbb{E}}_{W}\left[\omega_{\mathbf{p}}\,\omega_{\mathbf{p}'}\right] = \mathop{\mathbb{E}}_{W}\left[\omega_{\mathbf{p}\setminus p}\,\omega_{\mathbf{p}'\setminus p'}\right]\mathop{\mathbb{E}}_{W}\left[w_p\,w_{p'}\right] = 0\,.$$

Finally, if $\mathbf{p}$ has at least one non-average-pooling node p, then successively applying H4 and H3 yields: $\mathbb{E}_W[\omega_{\mathbf{p}}] = \mathbb{E}_W\left[\omega_{\mathbf{p}\setminus p}\right]\mathbb{E}_W[w_p] = 0$. □

We now have all elements to prove Theorem 6.2.2.

Proof. (**of Theorem 6.2.2**) For a given neuron p in $\tilde{\mathbf{p}}$, let $p-1$ designate the previous node in $\mathbf{p}$ of p. Let σ_p (resp. $\sigma_{\mathbf{p}}$) be a variable equal to 0 if neuron p gets killed by its ReLU (resp. path $\mathbf{p}$ is inactive), and 1 otherwise. Then:

$$\partial_x o \;=\; \sum_{\mathbf{p}\in\mathcal{P}(x,o)}\;\prod_{p\in\tilde{\mathbf{p}}} \partial_{p-1}\,p \;=\; \sum_{\mathbf{p}\in\mathcal{P}(x,o)} \omega_{\mathbf{p}}\,\sigma_{\mathbf{p}}$$

Consequently:

$$\begin{aligned}\mathop{\mathbb{E}}_{W,\sigma}\left[(\partial_x o)^2\right] &= \sum_{\mathbf{p},\mathbf{p}'\in\mathcal{P}(x,o)} \mathop{\mathbb{E}}_{W}\left[\omega_{\mathbf{p}}\,\omega_{\mathbf{p}'}\right]\mathop{\mathbb{E}}_{\sigma}\left[\sigma_{\mathbf{p}}\,\sigma_{\mathbf{p}'}\right]\\ &= \sum_{\mathbf{p}\in\mathcal{P}(x,o)}\;\prod_{p\in\tilde{\mathbf{p}}} \mathop{\mathbb{E}}_{W}\left[\omega_p^2\right]\mathop{\mathbb{E}}_{\sigma}\left[\sigma_p^2\right] \qquad \text{(C.32)}\\ &= \sum_{\mathbf{p}\in\mathcal{P}(x,o)}\;\prod_{p\in\tilde{\mathbf{p}}} \frac{2}{d_p}\,\frac{1}{2} = \frac{1}{d}\,,\end{aligned}$$

where the firs line uses the independence between the ReLU killings and the weights (H1), the second uses Lemma C.5.4 and the last uses Lemma C.5.3. The gradient $\boldsymbol{\partial}_{\mathbf{x}} o$ thus has coordinates whose squared expectations scale like $1/d$. Thus each coordinate scales like $1/\sqrt{d}$ and $\|\boldsymbol{\partial}_{\mathbf{x}} o\|_q$ like $d^{1/2-1/q}$. Conclude on $\|\boldsymbol{\partial}_x \mathcal{L}\|_q$ and $\epsilon_p\,\|\boldsymbol{\partial}_x \mathcal{L}\|_q$ by using Step 2 of the proof of Theorem 6.2.1.

Finally, note that, even without the symmetry assumption (S), using Lemma C.5.2 shows that

$$\begin{aligned}\mathop{\mathbb{E}}_{W}\left[\|\partial_x o\|_2^2\right] &= \sum_{x\in\mathbf{x}} \mathop{\mathbb{E}}_{W}\left[(\partial_x o)^2\right] \\ &= \sum_{x\in\mathbf{x}} \sum_{\mathbf{p}\in\mathcal{P}(x,o)} \prod_{p\in\tilde{\mathbf{p}}} \frac{2}{d_p}\frac{1}{2} = 1\,.\end{aligned}$$

Thus, with or without (S), $\|\partial_x o\|_2$ is independent of the input-dimension d. □

C.5.4 Proof of Theorem B.3.1

To prove Theorem B.3.1, we will actually prove the following more general theorem, which generalizes Theorem 6.2.2. Theorem B.3.1 is a straightforward corollary of it.

Theorem C.5.5. *Consider any feed-forward network with linear connections and ReLU activation functions that outputs logits $f_k(\mathbf{x})$ and satisfies assumptions (H). Suppose that there is a fixed multiset of integers $\{a_1,\dots,a_n\}$ such that each path from input to output traverses exactly n average pooling nodes with degrees $\{a_1,\dots,a_n\}$. Then:*

$$\|\partial_x f_k\|_2 \propto \frac{1}{\prod_{i=1}^n \sqrt{a_i}}\,. \tag{C.33}$$

Furthermore, if the net satisfies the symmetry assumption (S), then: $|\partial_x f_k| \propto \frac{1}{\sqrt{d\prod_{i=1}^n a_i}}$.

Two remarks. First, in all this proof, "weight" encompasses both the standard random weights, and the constant (deterministic) weights equal to $1/(\text{in-degree})$ of the average-poolings. Second, assumption H5 implies that the average-pooling nodes have disjoint input nodes: otherwise, there would be two non-zero deterministic weights w, w' from a same neuron that would hence satisfy: $\mathbb{E}_W[w\,w'] \neq 0$.

Proof. As previously, let o designate any fixed output-logit $f_k(\mathbf{x})$. For any path $\mathbf{p}$, let $\mathbf{a}$ be the set of average-pooling nodes of $\mathbf{p}$ and let $\mathbf{q}$ be the set of remaining nodes. Each path-product $\omega_{\mathbf{p}}$ satisfies: $\omega_{\mathbf{p}} = \omega_{\mathbf{q}}\omega_{\mathbf{a}}$, where $\omega_{\mathbf{a}}$ is a same fixed constant. For two distinct paths $\mathbf{p}, \mathbf{p}'$, Lemma C.5.4 therefore yields: $\mathbb{E}_W[\omega_{\mathbf{p}}^2] = \omega_{\mathbf{a}}^2\,\mathbb{E}_W[\omega_{\mathbf{q}}^2]$

and $\mathbb{E}_W[\omega_{\mathbf{p}}\omega_{\mathbf{p}'}] = 0$. Combining this with Lemma C.5.3 and under assumption (8), we get similarly to (C.32):

$$\begin{aligned}
\mathop{\mathbb{E}}_{W,\sigma}\left[(\partial_x o)^2\right] &= \sum_{\mathbf{p},\mathbf{p}'\in\mathcal{P}(x,o)} \omega_{\mathbf{a}}\omega_{\mathbf{a}'} \mathop{\mathbb{E}}_{W}[\omega_{\mathbf{q}}\,\omega_{\mathbf{q}'}] \mathop{\mathbb{E}}_{\sigma}[\sigma_{\mathbf{q}}\sigma_{\mathbf{q}'}] \\
&= \sum_{\mathbf{p}\in\mathcal{P}(x,o)} \prod_{i=1}^{n} \frac{1}{a_i^2} \prod_{q\in\tilde{\mathbf{q}}} \mathop{\mathbb{E}}_{W}\left[\omega_q^2\right] \mathop{\mathbb{E}}_{\sigma}\left[\sigma_q^2\right] \\
&= \underbrace{\prod_{i=1}^{n} \frac{1}{a_i}}_{\substack{\text{same value}\\ \text{for all } \mathbf{p}}} \underbrace{\sum_{\mathbf{p}\in\mathcal{P}(x,o)} \underbrace{\prod_{i=1}^{n} \frac{1}{a_i} \prod_{q\in\tilde{\mathbf{q}}} \frac{2}{d_q}\frac{1}{2}}_{\prod_{p\in\tilde{\mathbf{p}}}\frac{1}{d_p}}}_{=\frac{1}{d} \text{ (Lemma C.5.3)}} \\
&= \frac{1}{d}\prod_{i=1}^{n}\frac{1}{a_i}.
\end{aligned} \tag{C.34}$$

Therefore, $|\partial_x o| = |\partial_x f_k| \propto 1/\sqrt{d\prod_{i=1}^{n} a_i}$. Again, note that, even without assumption (8), using (C.34) and Lemma C.5.2 shows that

$$\begin{aligned}
\mathop{\mathbb{E}}_{W}\left[\|\partial_{\mathbf{x}} o\|_2^2\right] &= \sum_{x\in\mathbf{x}} \mathop{\mathbb{E}}_{W,\sigma}\left[(\partial_x o)^2\right] \\
&\overset{(C.34)}{=} \sum_{x\in\mathbf{x}} \prod_{i=1}^{n}\frac{1}{a_i} \sum_{\mathbf{p}\in\mathcal{P}(x,o)} \prod_{i=1}^{n}\frac{1}{a_i} \prod_{p\in\tilde{\mathbf{p}}}\frac{2}{d_p}\frac{1}{2} \\
&= \prod_{i=1}^{n}\frac{1}{a_i} \underbrace{\sum_{x\in\mathbf{x}} \sum_{\mathbf{p}\in\mathcal{P}(x,o)} \prod_{p\in\tilde{\mathbf{p}}}\frac{1}{d_p}}_{=1 \text{ (Lemma C.5.2)}} = \prod_{i=1}^{n}\frac{1}{a_i},
\end{aligned}$$

which proves (C.33). □

Bibliography

[1] R. A. Adams and J. J. F. Fournier. *Sobolev Spaces*. Academic Press, 2003.

[2] L. Ambrosio, N. Gigli, and G. Savaré. *Gradient Flows - In Metric Spaces and in the Space of Probability Measures*. Birkhäuser Verlag, Springer, 2005.

[3] L. Amsaleg, J. E. Bailey, D. Barbe, S. Erfani, M. E. Houle, V. Nguyen, and M. Radovanovic. *The Vulnerability of Learning to Adversarial Perturbation Increases with Intrinsic Dimensionality*. In: *IEEE Workshop on Information Forensics and Security*. 2017.

[4] M. Arjovsky, S. Chintala, and L. Bottou. *Wasserstein Generative Adversarial Networks*. In: *ICML*. 2017.

[5] D. Balduzzi, B. McWilliams, and T. Butler-Yeoman. *Neural Taylor Approximations: Convergence and Exploration in Rectifier Networks*. In: (2017).

[6] M. Balog, I. Tolstikhin, and B. Schölkopf. *Differentially Private Database Release via Kernel Mean Embeddings*. In: *ICML*. 2018.

[7] Y. Bengio, A. Courville, and P. Vincent. *Representation Learning: A Review and New Perspectives*. In: *Pattern Analysis and Machine Intelligence* 35 (8 2013), pp. 1798–1828.

[8] C. Bennett and R. C. Sharpley. *Interpolation of Operators*. Pure and Applied Mathematics. Elsevier Science, 1988.

[9] C. Berg, J. P. R. Christensen, and P. Ressel. *Harmonic Analysis on Semigroups Theory of Positive Definite and Related Functions*. Springer, 1984.

[10] A. Berlinet and C. Thomas-Agnan. *Reproducing Kernel Hilbert Spaces in Probability and Statistics*. Springer, 2004.

[11] M. Bińkowski, D. J. Sutherland, M. Arbel, and A. Gretton. *Demystifying MMD GANs*. In: *ICLR*. 2018.

[12] C. M. Bishop. *Training with noise is equivalent to Tikhonov regularization*. In: *Neural computation* 7.1 (1995), pp. 108–116.

[13] J. M. Borwein and A. S. Lewis. *Convex Analysis and Nonlinear Optimization: Theory and Examples*. Springer, 2006.

[14] N. Bourbaki. *Intégration - Chapitres 1-4*. Springer 2007 re-edition. Hermann, 1965.

[15] O. Bousquet, S. Gelly, I. Tolstikhin, C.-J. Simon-Gabriel, and B. Schölkopf. *From optimal transport to generative modeling: the VEGAN cookbook*. 2017. arXiv: 1705.07642.

[16] A. Caponnetto, C. A. Micchelli, M. Pontil, and Y. Ying. *Universal Multi-Task Kernels*. In: *JMLR* 9.7 (2008), pp. 1615–1646.

[17] C. Carmeli, E. De Vito, and A. Toigo. *Vector Valued Reproducing Kernel Hilbert Spaces of Integrable Functions and Mercer Theorem*. In: *Analysis and Applications* 4.4 (2006), pp. 377–408.

[18] T. Che, Y. Li, A. P. Jacob, Y. Bengio, and W. Li. *Mode Regularized Generative Adversarial Networks*. In: *ICLR*. 2017.

[19] W. Y. Chen, L. Mackey, J. Gorham, F.-X. Briol, and C. J. Oates. *Stein Points*. In: *ICML*. 2018.

[20] Y. Chen, M. Welling, and A. Smola. *Super-Samples from Kernel Herding*. In: *UAI*. 2010.

[21] K. Chwialkowski, H. Strathmann, and A. Gretton. *A Kernel Test of Goodness of Fit*. In: *NIPS*. 2016.

[22] M. Cisse, P. Bojanowski, E. Grave, Y. Dauphin, and N. Usunier. *Parseval Networks: Improving Robustness to Adversarial Examples*. In: *ICML*. 2017.

[23] DeepL. *DeepL Translator*. 2018. URL: https://www.deepl.com/translator.

[24] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. *ImageNet: A Large-Scale Hierarchical Image Database*. In: *CVPR*. 2009.

[25] H. Drucker and Y. LeCun. *Double Backpropagation Increasing Generalization Performance*. In: *International Joint Conference on Neural Networks*. 1991.

[26] M. Duc-Jacquet. *Approximation des Fonctionnelles Linéaires sur les Espaces Hilbertiens Autoreproduisants*. PhD thesis. Université Joseph-Fourier - Grenoble I, 1973.

[27] G. K. Dziugaite, D. M. Roy, and Z. Ghahramani. *Training Generative Neural Networks via Maximum Mean Discrepancy Optimization*. In: *UAI*. 2015.

[28] D. H. Fremlin, D. J. H. Garling, and R. G. Haydon. *Bounded Measures on Topological Spaces*. In: *Proceedings of the London Mathematical Society* s3-25.1 (1972), pp. 115–136.

[29] Y. Freund and R. E. Schapire. *A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting*. In: *Journal of Computer and System Sciences* 55.1 (1997), pp. 119–139.

[30] B. Fuglede and F. Topsoe. *Jensen-Shannon Divergence and Hilbert Space Embedding*. In: *IEEE International Symposium on Information Theory*. 2004.

[31] K. Fukumizu, F. Bach, and M. Jordan. *Kernel Dimensionality Reduction for Supervised Learning*. In: *JMLR* 5.12 (2004), pp. 73–99.

[32] K. Fukumizu, A. Gretton, G. R. Lanckriet, B. Schölkopf, and B. K. Sriperumbudur. *Kernel Choice and Classifiability for RKHS Embeddings of Probability Distributions*. In: *NIPS*. 2009.

[33] K. Fukumizu, A. Gretton, B. Schölkopf, and B. K. Sriperumbudur. *Characteristic Kernels on Groups and Semigroups*. In: *NIPS*. 2009.

[34] K. Fukumizu, A. Gretton, X. Sun, and B. Schölkopf. *Kernel Measures of Conditional Dependence*. In: *NIPS*. 2008.

[35] J. Gilmer, L. Metz, F. Faghri, S. S. Schoenholz, M. Raghu, M. Wattenberg, and I. Goodfellow. *Adversarial Spheres*. In: *ICLR Workshop*. 2018. arXiv: 1801.02774.

[36] I. J. Goodfellow, J. Shlens, and C. Szegedy. *Explaining and Harnessing Adversarial Examples*. In: *ICLR*. 2015.

[37] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. *Generative Adversarial Nets*. In: *NIPS*. 2014.

[38] A. D. Gordon, T. A. Henzinger, A. V. Nori, and S. K. Rajamani. *Probabilistic Programming*. In: *FOSE*. New York, NY, USA: ACM, 2014.

[39] J. Gorham and L. Mackey. *Measuring Sample Quality with Kernels*. In: *ICML*. 2017.

[40] I. S. Gradshteyn and I. M. Ryzhik. *Table of Integrals, Series, and Products*. Elsevier Academic Press, 2007.

[41] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola. *A Kernel Two-Sample Test*. In: *JMLR* 13 (2012), pp. 723–773.

[42] A. Gretton, K. M. Borgwardt, M. Rasch, B. Schölkopf, and A. J. Smola. *A Kernel Method for the Two-Sample-Problem*. In: *NIPS*. 2007.

[43] A. Gretton, K. Fukumizu, C. H. Teo, L. Song, B. Schölkopf, and A. J. Smola. *A Kernel Statistical Test of Independence*. In: *NIPS*. 2008.

[44] A. Gretton and L. Györfi. *Consistent Nonparametric Tests of Independence*. In: *JMLR* 11 (2010), pp. 1391–1423.

[45] A. Grover and S. Ermon. *Boosted Generative Models*. In: *AAAI*. 2018.

[46] C. Guilbart. *Etude des Produits Scalaires sur l'Espace des Mesures: Estimation par Projections*. PhD thesis. Université des Sciences et Techniques de Lille, 1978.

[47] K. He, X. Zhang, S. Ren, and J. Sun. *Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification*. In: *ICCV*. 2015.

[48] M. Hein and M. Andriushchenko. *Formal Guarantees on the Robustness of a Classifier against Adversarial Manipulation*. In: *NIPS*. 2017. arXiv: 1705.08475.

[49] M. Hein and O. Bousquet. *Hilbertian Metrics and Positive Definite Kernels on Probability Measures*. In: *AISTATS*. 2005.

[50] S. Hochreiter and J. Schmidhuber. *Simplifying Neural Nets by Discovering Flat Minima*. In: *NIPS*. 1995.

[51] H. Huang, G. M. Peloso, D. Howrigan, B. Rakitsch, C. J. Simon-Gabriel, J. I. Goldstein, M. J. Daly, K. Borgwardt, and B. M. Neale. *Bootstrat: Population Informed Bootstrapping for Rare Variant Tests*. In: (2016). bioRxiv: 10.1101/068999.

[52] J. Huang. *Statistics of Natural Images and Models*. PhD thesis. Providence, RI: Brown University, 2000.

[53] M. H. Kalos and P. A. Whitlock. *Monte Carlo Methods*. Wiley, 2008.

[54] M. Kanagawa, B. K. Sriperumbudur, and K. Fukumizu. *Convergence Guarantees for Kernel-Based Quadrature Rules in Misspecified Settings*. In: *NIPS*. 2016.

[55] Y. Katznelson. *An Introduction to Harmonic Analysis*. Cambridge University Press, 2004.

[56] D. P. Kingma and M. Welling. *Auto-Encoding Variational Bayes*. In: *ICLR*. 2014.

[57] A. Kitaoka. *Tilt Illusions after Oyama (1960): A Review*. In: *Japanese Psychological Research* 49.1 (2007), pp. 7–19.

[58] A. Kitaoka. *Akiyoshi's Illusion Page: Illusion of Frindged Edges*. 2018. URL: http://www.psy.ritsumei.ac.jp/~akitaoka/fringede.html.

[59] M. Korzeń and S. Jaroszewicz. *PaCAL: A Python Package for Arithmetic Computations with Random Variables*. In: *Journal of Statistical Software, Articles* 57.10 (2014), pp. 1–34.

[60] A. Krizhevsky. *Learning Multiple Layers of Features from Tiny Images*. Tech. rep. 2009.

[61] S. Lacoste-Julien, F. Lindsten, and F. Bach. *Sequential Kernel Herding : Frank-Wolfe Optimization for Particle Filtering*. In: *Artificial Intelligence and Statistics*. 2015.

[62] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. *Gradient-Based Learning Applied to Document Recognition*. In: *Proceedings of the IEEE*. Vol. 86 (11). 1998, pp. 2278–2324.

[63] O. Lehtö. *Some Remarks on the Kernel Function in Hilbert Function Space*. In: *Annales Academiae Scientiarum Fennicae*. A.I. Math.-Phy. 109 (1952), p. 6.

[64] C.-L. Li, W.-C. Chang, Y. Cheng, Y. Yang, and B. Poczos. *MMD GAN: Towards Deeper Understanding of Moment Matching Network*. In: *NIPS*. 2017.

[65] J. Q. Li and A. R. Barron. *Mixture Density Estimation*. In: *Biometrics* 53 (1997), pp. 603–618.

[66] Y. Li, K. Swersky, and R. Zemel. *Generative Moment Matching Networks*. In: *ICML*. 2015.

[67] F. Liese and I. Vajda. *On Divergences and Informations in Statistics and Information Theory*. In: *IEEE Transactions on Information Theory* 52.10 (2006), pp. 4394–4412.

[68] Q. Liu, J. Lee, and M. Jordan. *A Kernelized Stein Discrepancy for Goodness-of-fit Tests*. In: *ICML*. 2016.

[69] Z. Liu, P. Luo, X. Wang, and X. Tang. *Deep Learning Face Attributes in the Wild*. In: *ICCV*. 2015.

[70] C. Lyu, K. Huang, and H.-N. Liang. *A Unified Gradient Regularization Family for Adversarial Examples*. In: *ICDM*. 2015.

[71] A. Makhzani, J. Shlens, N. Jaitly, and I. Goodfellow. *Adversarial Autoencoders*. In: *ICLR*. 2016.

[72] S. Mallat. *Understanding Deep Convolutional Networks*. In: *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* 374.2065 (2016).

[73] K. S. McKinley. *Programming the World of Uncertain Things (Keynote)*. In: *ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*. 2016.

[74] L. Mescheder, S. Nowozin, and A. Geiger. *Adversarial Variational Bayes: Unifying Variational Autoencoders and Generative Adversarial Networks*. In: *ICML*. 2017.

[75] L. Metz, B. Poole, D. Pfau, and J. Sohl-Dickstein. *Unrolled Generative Adversarial Networks*. In: *ICLR*. 2017.

[76] C. A. Micchelli, Y. Xu, and H. Zhang. *Universal Kernels*. In: *JMLR* 7.12 (2006), pp. 2651–2667.

[77] D. Milios. *Probability Distributions as Program Variables*. PhD thesis. University of Edinburgh, 2009, p. 87.

[78] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard. *DeepFool: A Simple and Accurate Method to Fool Deep Neural Networks*. In: *CVPR*. 2016.

[79] K. Muandet, K. Fukumizu, B. Sriperumbudur, and B. Schölkopf. *Kernel Mean Embedding of Distributions: A Review and Beyond*. In: *Foundations and Trends in Machine Learning* 10.1-2 (2017), pp. 1–141.

[80] R. M. Neal. *Bayesian Learning for Neural Networks*. Vol. 118. Lecture Notes in Statistics. Springer, 1996.

[81] R. M. Neal. *Annealed Importance Sampling*. In: *Statistics and Computing* 11.2 (2001), pp. 125–139.

[82] S. Nowozin, B. Cseke, and R. Tomioka. *f-GAN: Training Generative Neural Samplers using Variational Divergence Minimization*. In: *NIPS*. 2016.

[83] C. J. Oates, J. Cockayne, F.-X. Briol, and M. Girolami. *Convergence Rates for a Class of Estimators Based on Stein's Method*. In: *Bernoulli - to appear* (2018). arXiv: 1603.03220.

[84] Y. Ollivier. *Auto-Encoders: Reconstruction versus Compression*. 2014. arXiv: 1403.7752.

[85] G. Peyré and M. Cuturi. *Computational Optimal Transport*. 2018. arXiv: 1803.00567.

[86] A. Radford, L. Metz, and S. Chintala. *Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks*. In: *ICLR*. 2016.

[87] J. Rauber, W. Brendel, and M. Bethge. *Foolbox v0.8.0: A Python toolbox to benchmark the robustness of machine learning models*. 2017. arXiv: 1707.04131.

[88] S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio. *Contractive Auto-Encoders: Explicit Invariance During Feature Extraction*. In: *ICML*. 2011.

[89] A. S. Ross and F. Doshi-Velez. *Improving the Adversarial Robustness and Interpretability of Deep Neural Networks by Regularizing Their Input Gradients*. In: *AAAI*. 2018.

[90] S. Rosset and E. Segal. *Boosting Density Estimation*. In: *NIPS*. 2002.

[91] B. Schölkopf, D. W. Hogg, D. Wang, D. Foreman-Mackey, D. Janzing, C.-J. Simon-Gabriel, and J. Peters. *Removing Systematic Errors for Exoplanet Search via Latent Causes*. In: *ICML*. 2015. arXiv: 1505.03036.

[92] B. Schölkopf, D. W. Hogg, D. Wang, D. Foreman-Mackey, D. Janzing, C.-J. Simon-Gabriel, and J. Peters. *Modeling Confounding by Half-Sibling Regression*. In: *PNAS* 113.27 (2016), pp. 7391–7398.

[93] B. Schölkopf, K. Muandet, K. Fukumizu, S. Harmeling, and J. Peters. *Computing Functions of Random Variables via Reproducing Kernel Hilbert Space Representations*. In: *Statistics and Computing* 25.4 (2015), pp. 755–766.

[94] B. Schölkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, 2001.

[95] Š. Schwabik. *Topics in Banach Space Integration*. Series in Real Analysis 10. World Scientific, 2005.

[96] L. Schwartz. *Espaces de fonctions différentiables à valeurs vectorielles*. In: *Journal d'Analyse Mathématique* 4.1 (1954), pp. 88–148.

[97] L. Schwartz. *Théorie des Distributions*. Hermann, 1978.

[98] C. Scovel, D. Hush, I. Steinwart, and J. Theiler. *Radial kernels and their reproducing kernel Hilbert spaces*. In: *Journal of Complexity* 26 (6 2010), pp. 641–660.

[99] C.-J. Simon-Gabriel and B. Schölkopf. *Kernel Distribution Embeddings: Universal Kernels, Characteristic Kernels and Kernel Metrics on Distributions*. 2016. arXiv: 1604.05251.

[100] C.-J. Simon-Gabriel and B. Schölkopf. *Kernel Distribution Embeddings: Universal Kernels, Characteristic Kernels and Kernel Metrics on Distributions*. In: *JMLR* (2018). to appear.

[101] C.-J. Simon-Gabriel and L. Mackey. *Targeted Convergence Characteristics of Maximum Mean Discrepancies and Kernel Stein Discrepancies*. In: *preprint* (2018).

[102] C.-J. Simon-Gabriel, Y. Ollivier, B. Schölkopf, L. Bottou, and D. Lopez-Paz. *Adversarial Vulnerability of Neural Networks Increases With Input Dimension*. 2018. arXiv: 1802.01421.

[103] C.-J. Simon-Gabriel, A. Scibior, I. O. Tolstikhin, and B. Schölkopf. *Consistent Kernel Mean Estimation for Functions of Random Variables*. In: *NIPS*. 2016, pp. 1732–1740.

[104] A. Sinha, H. Namkoong, and J. Duchi. *Certifiable Distributional Robustness with Principled Adversarial Training*. In: *ICLR*. 2018.

[105] A. Smola, A. Gretton, L. Song, and B. Schölkopf. *A Hilbert Space Embedding for Distributions*. In: *ALT*. 2007.

[106] M. D. Springer. *The Algebra of Random Variables*. Wiley, 1979, p. 470.

[107] B. K. Sriperumbudur. *On the Optimal Estimation of Probability Measures in Weak and Strong Topologies*. In: *Bernoulli* 22.3 (2016), pp. 1839–1893.

[108] B. K. Sriperumbudur, K. Fukumizu, and G. Lanckriet. *On the Relation between Universality, Characteristic Kernels and RKHS Embedding of Measures*. In: *AISTATS*. 2010.

[109] B. K. Sriperumbudur, K. Fukumizu, and G. R. Lanckriet. *Universality, Characteristic Kernels and RKHS Embedding of Measures*. In: *JMLR* 12 (2011), pp. 2389–2410.

[110] B. K. Sriperumbudur, A. Gretton, K. Fukumizu, G. R. Lanckriet, and B. Schölkopf. *Injective Hilbert Space Embeddings of Probability Measures*. In: *COLT*. 2008.

[111] B. K. Sriperumbudur, A. Gretton, K. Fukumizu, B. Schölkopf, and G. R. G. Lanckriet. *Hilbert Space Embeddings and Metrics on Probability Measures*. In: *JMLR* 11 (2010), pp. 1517–1561.

[112] I. Steinwart. *On the Influence of the Kernel on the Consistency of Support Vector Machines*. In: *JMLR* 2.12 (2001), pp. 67–93.

[113] I. Steinwart and A. Christmann. *Support Vector Machines*. Information Science and Statistics. Springer, 2008.

[114] I. Steinwart and C. Scovel. *Mercer's Theorem on General Domains: On the Interaction between Measures, Kernels, and RKHSs*. In: *Constructive Approximation* 35.3 (2012), pp. 363–417.

[115] I. O. Tolstikhin, S. Gelly, O. Bousquet, C.-J. Simon-Gabriel, and B. Schölkopf. *AdaGAN: Boosting Generative Models*. In: *NIPS*. 2017, pp. 5424–5433.

[116] I. Tolstikhin, O. Bousquet, S. Gelly, and B. Schoelkopf. *Wasserstein Auto-Encoders*. In: *ICLR*. 2018.

[117] I. Tolstikhin, S. Gelly, O. Bousquet, C.-J. Simon-Gabriel, and B. Schölkopf. *AdaGAN: Boosting Generative Models*. 2017. arXiv: 1701.02386.

[118] I. Tolstikhin, B. K. Sriperumbudur, and K. Muandet. *Minimax Estimation of Kernel Mean Embeddings*. In: *JMLR* 18.86 (2017), pp. 1–47.

[119] F. Treves. *Topological Vector Spaces, Distributions and Kernels*. Academic Press, 1967.

[120] Z. Tu. *Learning Generative Models via Discriminative Approaches*. In: *CVPR*. 2007.

[121] C. Villani. *Optimal Transport: Old and New*. Springer, 2009.

[122] A. Wald. *Statistical Decision Functions Which Minimize the Maximum Risk*. In: *Annals of Mathematics* 46.2 (1945), pp. 265–280.

[123] Y. Wang, L. Zhang, and J. van de Weijer. *Ensembles of Generative Adversarial Networks*. In: *Workshop on Adversarial Training, NIPS*. 2016. arXiv: 1612.00991.

[124] M. Welling, R. S. Zemel, and G. E. Hinton. *Self Supervised Boosting*. In: *NIPS*. 2002.

[125] H. Wendland. *Scattered Data Approximation*. Cambridge University Press, 2004.

[126] R. C. Williamson. *Probabilistic Arithmetic*. PhD thesis. University of Queensland, 1989, p. 316.

[127] H. Xu, C. Caramanis, and S. Mannor. *Robustness and Regularization of Support Vector Machines*. In: *JMLR* 10 (2009), pp. 1485–1510.

[128] X. Yuan, P. He, Q. Zhu, R. R. Bhat, and X. Li. *Adversarial Examples: Attacks and Defenses for Deep Learning*. 2017. arXiv: 1712.07107.

List of Figures

List of Tables