

Inherent Limitations of Dimensions for Characterizing Learnability

by

Tosca Lechner

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Computer Science

Waterloo, Ontario, Canada, 2024

© Tosca Lechner 2024

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner: Avrim Blum
Professor and Chief Academic Officer,
Toyota Technological Institute at Chicago (TTIC)

Supervisor: Shai Ben-David
Professor,
Cheriton School of Computer Science, University of Waterloo

Internal Members: Gautam Kamath
Assistant Professor,
Cheriton School of Computer Science, University of Waterloo

Yaoliang Yu
Associate Professor,
Cheriton School of Computer Science, University of Waterloo

Internal-External Member: Aukosh Jagannath
Associate Professor,
Department of Statistics and Actuarial Science, University of Waterloo

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners. I understand that my thesis may be made electronically available to the public.

Statement of Contributions

This work is based on the following published papers.

- ”On Learnability with Computable Learners” (joint work with Sushant Agarwal, Nivasini Anathakrishnan, Shai Ben-David and Ruth Uerner; published at ALT 2020): All authors have contributed to this work. The authors are in alphabetical order. Chapter 10 is based on this work.
- ”Learning Losses for Strategic Classification” (joint work with Ruth Uerner; published at AAAI 2022): All authors have contributed to this work. The authors are in alphabetical order. Parts of Chapter 8 are based on this work.
- ”Strategic Classification with Unknown User Manipulations” (joint work with Shai Ben-David and Ruth Uerner; published at ICML 2023): I am the primary author of this work. Parts of Chapter 8 are based on this work.
- ”Adversarially Robust Learning with Uncertain Perturbation sets” (joint work with Vinayak Pathak and Ruth Uerner; published at NeurIPS 2023): All authors have contributed to this work. The authors are in alphabetical order. Chapter 9 is based on this work.
- ”Distribution Learnability and Robustness”(joint work with Shai Ben-David, Alex Bie and Gautam Kamath, published at NeurIPS 2023): All authors have contributed to this work. The authors are in alphabetical order. Chapter 6 and Chapter 7 is based on this work. I only included parts that I had a direct contribution to.
- ”Inherent limitations of dimensions for characterizing learnability of distribution classes.” (joint work with Shai Ben-David; published at COLT 2024): I am the primary contributor of this work. Section 3.1, Chapter 5 and Chapter 12 are based on this work.
- ”On the Computability of robust PAC learning.” (joint work with Pascale Gourdeau and Ruth Uerner; published at COLT 2024): All authors have contributed to this work. The authors are in alphabetical order. Chapter 11 is based on this work.

Abstract

The fundamental theorem of statistical learning establishes the equivalence between various notions of both agnostic and realizable Probably Approximately Correct (PAC) learnability for binary classification with the finiteness of the VC dimension. Motivated by these foundational results, this work explores whether similar characterizations of learnability and sample complexity can be defined through dimensions in other learning models. We introduce the concept of a scale-invariant dimension and demonstrate that for a range of learning tasks, including distribution learning, any such dimension fails to capture learnability. Additionally, we define Computable PAC (CPAC) learning and show that not all classes are properly computably PAC learnable, highlighting a significant limitation in classical PAC frameworks. Our analysis further reveals that, unlike binary classification, realizable learning does not always imply agnostic learning in settings such as distribution learning and CPAC learning. Finally, we address learning under conditions of data corruption, whether from adversaries or self-manipulating agents, and show that the extent of prior knowledge about manipulation capabilities can significantly affect learnability. We address various ways in overcoming uncertainty in manipulation capabilities by either learning manipulation capabilities from distribution shifts or further oracle access or by allowing abstentions. We also show that this uncertainty in manipulation capabilities can sometimes be overcome without additional oracle access in the realizable case while the agnostic case requires such resources. These findings further underscore that in order to characterize agnostic learnability it is not always sufficient to understand the realizable case.

Acknowledgements

I would like to thank Avrim Blum, Aukosh Jagannath, Yaoliang Yu and Gautam Kamath for agreeing to serve on my committee. Special thanks to Yaoliang and Gautam for making time for all my PhD seminars in the past months and providing helpful feedback!

There are many great people who have helped and supported me throughout my PhD!

I would like to thank Shai Ben-David for accepting me as his PhD student and being a great supervisor throughout. I was lucky to work with someone with so much insight into the field so closely. I thank you for always being supportive, encouraging me to voice and develop my own ideas and helping me build confidence as a researcher. I also want to thank you for being patient, offering support and advice outside of research and making me rediscover my love for swimming!

Thanks to Ruth Urner, for first making me discover my love for learning theory and giving me a chance at a time where that maybe wasn't obvious. You made me believe in myself and inspired me to want to become a researcher! I'm happy that we were able to continue doing research throughout my PhD and I kept learning from you. Most importantly I'm grateful that you turned from a mentor into a good friend and am grateful for all your countless and generous support, advice and good times over the years. Also thank for making me get into exercise (that really helped keep me sane in the pandemic)!

Thank you to my many collaborators! I'm happy that I can call many of you my friends. In particular I want to thank the co-authors of the papers this thesis is based on: Sushant Agarwal, Nivasini Ananthakrishnan, Shai Ben-David, Alex Bie, Pascale Gourdeau, Gautam Kamath, Vinayak Pathak and Ruth Urner! I also want to thank Anmol Kabra and Serena Wang for inviting me to a their project and to them and Mina Karzand, and Nati Srebro for their work and finishing the paper.

I want to thank Shai's research group for always being full of smart, fun and nice people. I learned from every one of you! Nivasini and Sushant, it was so much fun to start research together, discuss ideas, suffer through marking, live together, write our first papers together and walk home together at 7am in the morning, watch research talks while drinking wine and support each other in the first year. Sushant, thanks for being one of my first friends here and making it easy to belong. Nivasini, thanks for being inspiring, all the research discussions we had while walking in the cold and our later fun times in Berkeley. Niki, Matt and Alex, thanks for making Shai's group great again, after everyone had left! I learned a lot from our research group meetings and enjoyed our hangouts. Alireza and Farnam, I enjoyed our discussions in the brief overlap we had and I am excited to see the research you will be doing in the future.

I made a lot of amazing friends in my time in Waterloo. Thanks to Sushant, Nivasini and Shuchita for the amazing months living in University Avenue! I want to thank Amine Mhedhbi for being a patient roommate and fun, caring friend and an inspiring person, who pushes the people around him to do their best. Thanks for making me apply to things and making me more ambitious and put more effort into my PhD. Being stuck in lockdown with you might have been one of the best things to happen to me, professionally speaking. ;) Thanks to Alex and Mercedes for being part of my pandemic bubble and celebrating holidays together. Thanks to Melica, Andreas, Shubhankar, Aseem, Kritika, and all the friends mentioned above and everyone from the international grad student community who were there in those weird few years being stuck far from where each of us are from and making the best of this new temporary home. I enjoyed our fun time and am happy I got to know each of you!

Thanks to Shai Ben-David, Bin Yu, Zack Lipton and Ruth Urner for organizing the Interpretability Cluster at the Simon's Institute in Berkely! And thanks to the organizers and participants (Awa Dieng, Sebastian Bordt, Anmol Kabra, Gyorgy Turan, Rich Caruana, Michal Moshkovitz, Simina Brânzei, Sina Fazelpour) for making those six weeks one of the greatest experiences in my PhD.

Thanks to Apple and the selection committee for giving me the Apple Waterloo PhD Fellowship for Machine Learning and Data Science. It was an honor and I appreciated the support. Thank you to Alireza Mousavi and Yunyao Li for mentoring and hosting my internship at Apple in Seattle. You were always kind and supportive and let me explore theory while providing a more applied perspective.

Thanks to the Vector Institute for continued support through the Vector Research Grants, as well as through organization with events, travel support and generally helpful staff.

I also want to thank all of my awesome friends abroad! While I wasn't always the best with staying in touch, I knew I could count on you and you offered support, inspiration, kindness and grounded me whenever we did end up meeting, talking and connecting. Thanks for hosting me, for insights, good memories, for your practical and emotional support, visiting me, spontaneously letting me crash at your place, letting me keep you up at night until it's time for me to go to bed in my time zone, reminding me of my good qualities, making me laugh, making me think, for glorious food and letting me know that I can always come back even if I'm stuck on finally finishing that last e-mail! This is aimed at many people, but in particular at David Lorch, Jan Leike, Renata Barreto, Alex Surdina, Marc und Ulrike Endesfelder, Agnes Köhler, Melanie Kirch, Henrike Stockel, Oliver Caspari, Patrick Lahr and Reimar Leike.

Lastly I want to thank my family for their unconditional support. I want to thank my brother Milan for being true to himself and being a voice of reason. Also thank you for visiting me in Canada and thanks to you and Lauren for hosting me in Munich! Thanks to my father for always encouraging me to think for myself, question things, discussing ideas and giving me fun riddles as a child. Probably no one has shaped my thinking as much as you. Thanks to my mother for being kind, caring and accepting. Your capacity of seeing good and stay positive is inspiring.

Table of Contents

Examining Committee	ii
Author's Declaration	iii
Statement of Contributions	iv
Abstract	v
Acknowledgements	vi
I PAC Learning and the Fundamental Theorem	1
1 Introduction	2
1.1 Overview of Contributions	5
2 Preliminaries	9
2.1 PAC Learning for binary classification	9
2.2 General PAC Learning model	12
3 When does the Fundamental Theorem Generalize to other Learning Tasks?	15
3.1 Characterizing Dimensions	16

3.1.1	Notions of characterization of learning tasks	16
3.1.2	Notions of quantitative characterization of statistical learning	17
3.1.3	Notions of qualitative characterization of statistical learning	18
3.1.4	Comparing the different notions of characterization	19
3.1.5	Sufficient conditions for Uncharacterizability by Scale-Invariant Dimensions	20
3.1.6	A condition implying no sample complexity dimension	21
3.1.7	A condition implying no finitary characterization	23
3.1.8	Discussion	24
3.2	Agnostic vs Realizable Learning	25
II	Distribution Learning	29
4	Distribution Learning Preliminaries	30
4.1	Introduction	30
4.1.1	Definitions of Learnability	33
4.2	Additional Preliminaries: Methods for Distribution Learning	35
4.2.1	Yatracos learner	35
4.2.2	Sample Compression	36
4.2.3	Useful Inequalities	37
4.3	Related Work	38
5	Impossibility of Characterizing Distribution Learning	40
5.1	Preliminaries and Construction	41
5.2	Impossibility of Characterizing the Sample Complexity	42
5.3	Distributions Learning cannot be characterized by a finitary characterization	43
5.4	Polynomial complexity distribution learning	43
5.5	Proofs: Uncharacterizability of Distribution Learning	44

6	Realizable vs Agnostic Learning	47
6.1	Overview of Results	47
6.2	Realizable Learnability does not Imply Agnostic Learnability	48
6.3	Proofsketch of Theorem 6.2.1	49
6.4	Proof of remaining Claims for Theorem 6.2.1	50
6.4.1	Proof of Claim 6.3.1	50
6.5	The Existence of a Realizable Compression Scheme does not imply the Existence of a Robust Compression Scheme	53
7	Robust Distribution Learnability	56
7.1	Definitions Robust Learnability	56
7.1.1	Overview of Results	57
7.2	Learnability Implies Additive Robust Learnability	60
7.3	Subtractive Robust Learnability Implies Robust Learnability	61
7.4	Realizable Learnability Does Not Imply Subtractively Robust Learnability	61
7.5	Proofs	62
7.5.1	Proof of Theorem 7.2.1	62
7.5.2	Proof of Theorem 7.3.1	66
7.6	Known contamination levels	70
7.6.1	Proof of Theorem 7.6.3	74
III	Robust Learning with Uncertain Manipulation Capabilities	76
8	Strategic Classification	80
8.1	Introduction	80
8.1.1	Related Work	84
8.1.2	Overview on contributions	86

8.2	The strategic loss	88
8.2.1	Basic Learning Theoretic Notions for Classification	88
8.2.2	Strategic Classification	88
8.2.3	Comparison with Alternative Formalisms for Strategic Classification	91
8.3	Learnability with the Strategic Loss	93
8.3.1	Warm up: Loss Classes and Learnability	93
8.3.2	Sufficient Conditions for Strategic Loss Learnability	95
8.4	Strategic loss with respect to an approximate manipulation graph	98
8.5	Learning a manipulation graph from neighborhoods of sample points	99
8.6	Learning Manipulation Capabilities from Distribution Shifts	101
8.7	Classes of manipulation graphs that allow realizable learnability	104
8.7.1	Note on the maximally robust ERM paradigm	107
8.8	Batch-Learning for strategic classification	110
8.8.1	Class of Thresholds	112
8.8.2	General VC-Classes	114
8.9	Conclusion	117
8.10	Proofs	118
8.10.1	Realizable Learning	121
8.10.2	Batch Learning	124
9	Adversarially Robust Learning	131
9.1	Introduction	131
9.1.1	Related work	133
9.1.2	Overview of results	134
9.2	Setup	135
9.3	Learning with a totally ordered perturbation class	140
9.4	Beyond totally ordered perturbation classes	141
9.4.1	Learning with the standard adversarial loss	142

9.4.2	Learning with the adversarial abstention loss	143
9.5	Conclusion	146
9.6	Maximally Robust Empirical Risk Minimization	147
9.7	Agnostic learning for finite unions of total orders	149
9.8	ϵ -cover Learning	150
9.8.1	Abstention learning for regions with different radii	150
9.9	Proofs	151
9.9.1	Proof of Theorem 1	151
9.9.2	Proof of Observation 1	152
9.9.3	Proof of Observation 2	152
9.9.4	Proof of Observation 4	153
9.9.5	Proof of Theorem 4	153
9.9.6	Proof of Theorem 5	154
9.9.7	Proof of Theorem 6	155

IV Computable PAC Learning 156

10 On Learnability With Computable Learners 157

10.1	Overview of Contributions	157
10.2	Defining Computable PAC (CPAC) Learning	158
10.2.1	General background	158
10.2.2	CPAC learnability	160
10.3	Summary of Results	162
10.4	Proper CPAC Learning	163
10.4.1	Computable (approximations to) empirical risk minimization	166
10.5	Computable non-uniform Learning	168
10.6	Improper CPAC Learning	170
10.7	Conclusion and Overview over Related Work	174
10.7.1	Computable dimensions	175

11 On the Computability of Robust PAC Learning	176
11.1 Related Work	177
11.1.1 Robust CPAC Learning: Definition	178
11.2 Simple Sufficient Conditions for Robust CPAC Learnability	181
11.3 Relating CPAC and Robust CPAC Learning	182
11.3.1 Robust CPAC learnability is not implied by Robust PAC + CPAC learnability	182
11.3.2 Robust CPAC learnability and its relationship to the computability of the robust loss and RERM	186
11.4 Necessary Conditions for Robust CPAC Learning	188
11.4.1 No-Free-Lunch Theorems	188
11.4.2 Relationship between the effective VC dimension and effective shattering dimension	192
11.5 Conclusions for Robust CPAC Learning	192
11.6 Remaining Proofs and Remarks from Section 11.2	193
11.6.1 Proof of Theorem 11.2.3	193
11.6.2 Decidable Representation of Perturbation Types	194
11.7 Proofs and Algorithms from Section 11.3.1	195
11.7.1 Proof of Theorem 11.3.1	195
11.7.2 Algorithm for the Decidable Representation in Theorem 11.3.2	199
11.7.3 Useful Lemmas for Theorem 11.3.2	199
11.8 Proofs from Section 11.3.2	202
11.8.1 Proof of Theorem 11.3.3	202
11.8.2 Proof of Theorem 11.3.4	204
11.8.3 Proof of Proposition 11.3.5	206
11.9 Proofs from Section 11.4	207
11.9.1 Proof of Lemma 11.4.4	207
11.9.2 Proof of Theorem 11.4.6	208
11.10 Remarks on the robust shatterability in the impossibility results from Section 11.3	208

V	Other Learning Tasks	210
12	Impossibility of Characterizing Other Learning Tasks	211
12.1	Classification Learning with respect to a restricted class of distributions . .	211
12.2	Learning Real-Valued Functions with Real-Valued Losses	213
12.3	Proofs	215
	References	219

Part I

PAC Learning and the Fundamental Theorem

Chapter 1

Introduction

Machine Learning applications have become more ubiquitous in many domains in recent years. While these models often seem impressive in real life, a fundamental understanding of their capabilities is still missing. One avenue for attempting to gain such understanding is statistical learning theory. This theory aims to provide mathematical guarantees about a model’s accuracy given the model’s flexibility and the amount of training data it receives. Of particular interest is the notion of Probably Approximately Correct (PAC) Learning by Valiant [1984], which gives uniform and distribution-free sample size bounds that guarantee low expected error with high probability.

The celebrated ‘fundamental theorem of statistical learning’ provides a clean characterization of PAC learnability of binary classification in terms of the combinatorial Vapnik Chervonenkis dimension (VC-dimension) [Blumer et al., 1989]. Furthermore, the learning rates for any class H of binary valued functions are fully determined (up to constants) by the VC-dimension of that class.

This fundamental result sparked a quest for notions of dimension that similarly characterize the learnability of other learning tasks. For some tasks, such as online learning of binary classifiers such dimensions have indeed been established. For other statistical learning tasks, some parameters have been proposed but not proven to provide the required characterizations.

In contrast, the results of [Ben-David et al., 2017b] showed for the first time that for some type of problems, such as EMX learnability¹ no such characterization can be proved

¹EMX, or General Statistical Learning, is the problem in which given a class of random variables and an i.i.d. sample generated by some unknown probability distribution P , the learner aims to find a member of that class that has close to maximal expectation w.r.t. P

to exist by the common axioms of mathematics (the ZFC set theory).

In this work, we investigate the existence of characterizing dimensions for several statistical learning problems for which that question remained open (most notably the learnability of classes of discrete probability distributions). We show that, quite surprisingly, no such characterizations exist. More concretely, we show that learnability of these problem cannot be characterized in a *scale-invariant* way (Like the VC-dimension characterization of binary classification). Our results do not rule out the existence of scale-sensitive characterization that depend on fixing the accuracy parameter.

Our results answer some long-standing open questions; The survey paper by [Diakonikolas, 2016] asks (Open Problem 1.5.1): *“Is there a “complexity measure” of a distribution class C that characterizes the sample complexity of learning C ?”*

[Hopkins et al., 2022a] state *“Unlike the standard model, very little is known about distribution-family learnability. While a number of works have made some progress on this front, a characterization of learnability remains elusive despite some 30 years of effort”* (end of Section 4 there).

[Benedek and Itai, 1991] ask about the characterization of PAC learnability of binary-valued classifiers w.r.t. a given class of probability distributions. They conjectured a characterization that was refuted by [Dudley et al., 1994]. The latter repeats the question of finding a characterization for that task. Similar open questions are later stated by [Kulkarni and Vidyasagar, 1997] and [Vidyasagar et al., 2001]. It is worth noting that the earlier proposed characterizations were *scale sensitive*. Our results show that there can be no scale invariant characterization of such learnability.

Of course, results showing the non-existence of some characterization, rely on a precise definition of such “characterizations”. The notions we discuss in this paper are generalizations of a family of well studied learnability characterizations that refer to learnability to an arbitrarily small inaccuracy parameter. These include the fundamental Vapnik-Chervonenkis dimension, the Littlestone dimension, and the notions discussed for EMX learning ([Ben-David et al., 2017b]). However, our definitions do not capture the so called ‘scale sensitive’ dimensions such as γ - fat shattering (Alon et al. [1997]). We discuss this further in subsection 3.1.4 below.

The equivalence of PAC learnability and the VC-dimension in the fundamental theorem is shown via the equivalence of the “uniform convergence” property and a hypothesis class having finite VC-dimension. The fundamental theorem further states that classes are learnable if and only if they are learnable by an empirical risk minimizing (ERM) learner, i.e. a learner that outputs a classifier from the class with minimal sample error. Learners fulfilling the requirement of always outputting a hypothesis from the class are also

known as *proper learners*. The fundamental theorem thus states an equivalence between proper and improper learnability for binary classification. Furthermore, since the equivalence of learnability and uniform convergence holds in both the agnostic and the realizable case, the fundamental theorem also states an equivalence between agnostic and realizable learnability. A theme throughout this thesis will be to analyse different settings for these equivalences.

Going beyond binary classification with 0/1-loss there are many known examples of settings where learnability and uniform convergence are not equivalent [Daniely et al., 2015]. Moreover, the equivalence between proper and improper learnability can break in such settings. Known examples of this phenomenon include multiclass learnability with infinite label spaces [Daniely et al., 2014], adversarially robust learning [Montasser et al., 2019] and settings in which learners are required to satisfy computational efficiency criteria. A commonality to these settings is the fact that one can encode information that is either computationally or information-theoretically hard to obtain, but is irrelevant to successful generalization if outputs outside of the hypothesis class are permissible.

Furthermore, the equivalence of realizable and agnostic learnability has been shown in many settings, such as multi-class learnability, adversarially robust learning, classification learning with restricted marginals. In settings where uniform convergence does not hold such equivalences are often obtained via sample-compression or creating a finite set of candidate hypotheses from which it is easy to select an optimal candidate. The ubiquity of such results has lead researchers to proclaim that: "Realizable Learning is All You Need" [Hopkins et al., 2022a]. In contrast, we will present several natural learning settings in which such equivalences do not hold. We will show that in the case of distribution learning with respect to total variation distance also known as *density estimation* there are classes which are learnable in the realizable case but not agnostically learnable. These classes consist of distributions which have small mass on a support element which is unique to them, thus allowing identification of the ground truth distributions from the class once such an element is observed in the sample.

Another consideration we are interested in is the potential for corrupted data. We consider different kinds and times of corruption in different settings. In particular, we consider training time corruptions for distribution learning and at test time for classification learning. We will see how different knowledge in the manipulation capabilities can yield different outcomes for learnability. In particular for classification learning, we are interested in two settings: *Adversarially robust learning*, in which an adversary manipulates authentic sample points in a humanly imperceptible way in order to fool the learned classifier, and *strategic classification*, where instances represent self-motivated individuals who can change their feature vectors in order to obtain a preferred classification outcome.

We show that both settings can be modeled via a similar loss function if the manipulation capabilities are known and then provide a similar analysis of learnability for strategic classification as was given by [Montasser et al. \[2019\]](#) for adversarial robustness. We also argue that in both settings the common assumption that manipulation capabilities are perfectly known are not always met in real-world settings. We propose a setting in which the prior knowledge of manipulation capabilities is modeled via a class of candidate manipulation capabilities. We then show a separation between realizable and agnostic learnability in some of those setting and discuss various ways of overcoming the uncertainty of manipulation capabilities — and thus true pointwise loss. The methods we discuss include access to perfect-attack-oracles and allowing abstention in the adversarially robust setting. For strategic classification we propose a batch-learning setting, modelling scenarios in which there are regular pools of applicants. We show that for particular classes of hypotheses and manipulation capabilities, the distribution shift between batches can be utilized to estimate the manipulation capabilities.

Lastly, we also consider learning with computability requirements. We note that the common formulation of PAC learning only considers classes of functions and that learners are functions from samples to hypotheses. We propose a novel computable PAC (CPAC) learning framework, where we require both learners and outputs to be computable functions. We show that proper CPAC learning is not characterized by the VC-dimension, i.e. there are classes of finite VC dimension for which there is no proper computable learner. Further, we show a separation for the proper agnostic and the proper realizable learnability. Our setting later inspired others to extend the result of non-characterizability to the improper case, showing that there are decidable representable classes of finite VC dimension which can not be improperly learned in the agnostic case [[Sterkenburg, 2022](#)]. Furthermore, it was shown that while our CPAC setting cannot be characterized by the VC dimension, there is a computable version of the VC-dimension, i.e., the *effective VC-dimension* which characterizes improper agnostic CPAC learnability.

Our results extend what was known about the learnability in a variety of settings and provides some insight into the requirements for characterizing learnability. In particular we show the limitations of dimensions for characterizing learnability.

1.1 Overview of Contributions

Characterizing Dimensions The motivation for this research was to address the question of characterizing learnability, hopefully via some appropriate parameter of a

class (a.k.a. dimension) of various tasks for which no such characterization have been shown.

- The main finding of our work is that for the tasks listed, there can be no characterization (or learnability indicating dimension) that meets some natural requirements.
- As a secondary contribution, we propose formal requirements on what learnability characterizations should satisfy. All the notions that are known to characterize PAC learnability (rather than some form of weak learnability, or learnability up to some fixed accuracy) meet our proposed requirements.
- We also present two novel techniques for proving such non-existence results that we hope would find further applications.

The notions of characterizing dimension will be introduced in Section 3.1. This section will also include a discussion of the general techniques to show that a task cannot be characterized by a scale-invariant dimension. We will show that distribution learning cannot be characterized by a scale-invariant dimension in Chapter 5. We will also show such uncharacterizability for classification learning with restricted marginals and regression learning in Chapter 12.

Distribution Learning One central topic of this thesis is distribution learning. In Chapter 4 we will introduce the setting of distribution learning. This will be followed by the chapter discussing the uncharacterizability of distribution learning by a scale-invariant dimension (Chapter 5). In Chapter 6 we will show that agnostic and realizable distribution learning are not equivalent, i.e., there are distribution classes that are learnable in the realizable case, but not in the agnostic one. Furthermore, we will discuss more general notions of robust distribution learning in Chapter 7. We show that in contrast to the general robust case, realizable learnability of a distribution class does imply that it is also learnable in the additively robust case, also known as Huber contamination [Huber, 1964].

Strategic Classification We introduce a novel binary loss function for the setting of Strategic Classification introduced by Hardt et al. [2016], incorporating both the requirement for accuracy and the requirement for the avoidance of social burden [Milli et al., 2019], i.e., ensuring that individuals do not have to change their feature vector in order to be classified correctly. We analyse PAC learnability with respect to this loss function for known manipulation structures and show that all finite VC classes are learnable, but sometimes only improperly. We further discuss the problem of

learning when the manipulation structure, given by a so-called manipulation-graph, is not perfectly known. We first introduce a similarity measure between manipulation graphs and show that misestimating the structure within a small distance does not greatly deteriorate the quality of the classification outcome. We then propose different ways of learning the manipulation structure. We first show sufficient requirements for learning from finitely many sample points for which their full manipulation capabilities are known. We then propose a batch learning setting for strategic classification, in which we can use the distribution shift between batches to estimate the manipulation graph. We also introduce the novel maximally robust empirical risk minimizing (MRERM) paradigm for classes of candidate manipulation graphs which are totally ordered by inclusion. In the realizable case we can then show that with the use of this paradigm combined with the compression learning approach from [Montasser et al. \[2019\]](#) all finite VC classes are learnable in the realizable case, despite the uncertainty in manipulation structure. We also show that additional information is needed for the agnostic case. Our results for strategic classification are discussed in Chapter 8.

Adversarial Robustness with Uncertain Perturbation Sets Similar to the strategic setting we also investigate adversarially robust learning with uncertain perturbation sets. We show that finite VC classes are learnable with additional access to a perfect-attack-oracle if the class of candidate perturbation types can be decomposed into a finite number of sets, each of which is totally ordered by inclusion. This includes the case in which the perturbation types are balls with a fixed but unknown radius in a fixed but unknown norm from a set of finitely many candidate norms. Our results overcome lower bounds in terms of the Littlestone dimension for completely unknown perturbation types [[Montasser et al., 2021](#)]. We further show that there are pairs of hypothesis classes and perturbation type classes which cannot be learned even with access to a perfect-attack-oracle, but can be learned when allowing abstention on corrupted points. Our results on adversarially robust learning with uncertain perturbation types can be found in Chapter 9.

Computable Learning We introduce the notion of computable PAC (CPAC) learning in Chapter 10. We introduce some general requirements on the representation of hypothesis classes and show that even classes that meet such requirements and have finite VC dimension can be impossible to learn by a proper CPAC learner. We also introduce the notion of robust CPAC learning, which requires learners to be both adversarially robust as well as computable. We show that there are classes that are robustly learnable and CPAC learnable but not robustly CPAC learnable, thus learn-

ability in this setting cannot be explained by separately investigating computable and robust learnability separately. We also show that the computability of the robust loss is neither necessary nor sufficient for CPAC learnable classes to be robustly CPAC learnable. Lastly, we show a lower bound for robust CPAC learnability via a computable version of the robust shattering dimension. The introduction and discussion of robust CPAC learning can be found in Chapter 11.

This thesis is based on the following published conference papers:

- "On Learnability with Computable Learners" (joint work with Sushant Agarwal, Nivasini Anathakrishnan, Shai Ben-David and Ruth Urner; published at ALT 2020)
- "Learning Losses for Strategic Classification" (joint work with Ruth Urner; published at AAAI 2022)
- "Strategic Classification with Unknown User Manipulations" (joint work with Shai Ben-David and Ruth Urner; published at ICML 2023)
- "Adversarially Robust Learning with Uncertain Perturbation sets" (joint work with Vinayak Pathak and Ruth Urner; published at NeurIPS 2023)
- "Distribution Learnability and Robustness" (joint work with Shai Ben-David, Alex Bie and Gautam Kamath, published at NeurIPS 2023)
- "Inherent limitations of dimensions for characterizing learnability of distribution classes." (joint work with Shai Ben-David; COLT 2024)
- "On the Computability of robust PAC learning." (joint work with Pascale Gourdeau and Ruth Urner; COLT 2024)

Chapter 2

Preliminaries

In this section we will first introduce PAC learning for binary classification and some basic theory, including the fundamental theorem of statistical learning theory. We will then introduce a general PAC learning setup and briefly introduce several settings discussed in this thesis.

2.1 PAC Learning for binary classification

In this subsection I will give a brief overview of the basic definitions and theory for PAC of binary. This section summarizes parts of [Shalev-Shwartz and Ben-David, 2014a].

Let $\mathcal{X}$ be the input space and $\mathcal{Y}$ the label space. In binary classification we assume $\mathcal{Y} = \{0, 1\}$. We let $\mathcal{H} \subseteq \{0, 1\}^{\mathcal{X}}$ denote a *hypothesis class* on $\mathcal{X}$. Given a joint distribution D on $\mathcal{X} \times \mathcal{Y}$ and a hypothesis $h \in \mathcal{H}$, the *risk* (or error) of h with respect to D is defined as

$$L_D(h) = \mathbb{P}_{(x,y) \sim D} (h(x) \neq y) \quad .$$

We will also denote by $\ell^{0/1} : \mathcal{H} \times \mathcal{X} \times \{0, 1\} \rightarrow \{0, 1\}$ the 0-1 loss function $\ell^{0/1}(h, x, y) = \mathbf{1}[h(x) \neq y]$, and we use the notation $L_S(h) = \frac{1}{m} \sum_{i=1}^m \ell^{0/1}(h, x_i, y_i)$ to denote the *empirical risk* of h on a sample $S = \{(x_i, y_i)\}_{i=1}^m \in (\mathcal{X} \times \mathcal{Y})^m$. The PAC-learning framework of [Valiant, 1984] is now given in the following definition.

Definition 2.1.1 (Agnostic PAC Learnability). *A hypothesis class $\mathcal{H}$ is PAC learnable in the agnostic setting if there exists a learner $\mathcal{A}$ and function $m(\cdot, \cdot)$ such that for all*

$\varepsilon, \delta \in (0, 1)$ and for any distribution D , if the input to $\mathcal{A}$ is an i.i.d. sample S from D of size at least $m(\varepsilon, \delta)$, then, with probability at least $(1 - \delta)$ over the samples, the learner outputs a hypothesis $\mathcal{A}(S)$ with $L_D(\mathcal{A}(S)) \leq \inf_{h \in \mathcal{H}} L_D(h) + \varepsilon$. The class is said to be PAC learnable in the realizable setting if the above holds under the condition that $\inf_{h \in \mathcal{H}} L_D(h) = 0$.

It is well known that a (binary) hypothesis class is PAC learnable if and only if it has finite VC dimension [Vapnik and Chervonenkis, 1971b], which is defined below.

Proper learnability and empirical risk minimization. For all the above notions of learnability, we call the class $\mathcal{H}$ if there exists a learner that satisfies the corresponding definition of learnability and always outputs functions from $\mathcal{H}$. A learner $\mathcal{A}$ is said to *perform empirical risk minimization (ERM)*, or *implement an $\text{ERM}_{\mathcal{H}}$ oracle*, if it always outputs a predictor from the hypothesis class of minimal empirical risk, that is for all $m \in \mathbb{N}$ and samples $S \in (\mathcal{X} \times \mathcal{Y})^m$,

$$\mathcal{A}(S) \in \arg \min_{h \in \mathcal{H}} L_S(h) .$$

One common way of showing that a class is PAC learnable, is by showing that it satisfies the uniform convergence property.

Definition 2.1.2 (ε -approximation). A sample set S is called an ε -approximation with respect to domain $\mathcal{X} \times \mathcal{Y}$, hypothesis class $\mathcal{H}$ and distribution D , if

$$\forall h \in \mathcal{H} : |L_S(h) - L_P(h)| \leq \varepsilon$$

Definition 2.1.3 (Uniform Convergence). We say a class $\mathcal{H}$ has the uniform convergence property (with respect to a domain $\mathcal{X} \times \mathcal{Y}$ and loss ℓ if there exists a function $m_{\mathcal{H}}^{\text{UC}} : (0, 1) \rightarrow \mathbb{N}$ such that for every $\varepsilon, \delta \in (0, 1)$ and for every probability distribution D over $\mathcal{X} \times \mathcal{Y}$, if S is a sample of size $m \geq m_{\mathcal{H}}^{\text{UC}}(\varepsilon, \delta)$ drawn i.i.d. from D , then with probability at least $1 - \delta$, S is ε -representative (w.r.t. $\mathcal{X} \times \mathcal{Y}$, $\mathcal{H}$ and D).

It is immediate, that a hypothesis satisfying the uniform convergence property is also PAC-learnable. Furthermore, in [Vapnik and Chervonenkis, 1971b] showed that a class having uniform convergence property is equivalent to a class having finite VC-dimension. We will now state the definition of VC dimension.

Definition 2.1.4 (Shattering and VC dimension [Vapnik and Chervonenkis, 1971b]). Given a class of functions $\mathcal{H}$ from $\mathcal{X}$ to $\{0, 1\}$, we say that a set $S \subseteq \mathcal{X}$ is shattered by $\mathcal{H}$ if the restriction of $\mathcal{H}$ to S is the set of all function from S to $\{0, 1\}$. The VC dimension of a hypothesis class $\mathcal{H}$, denoted $\text{VC}(\mathcal{H})$, is the size d of the largest set that can be shattered by $\mathcal{H}$. If no such d exists then $\text{VC}(\mathcal{H}) = \infty$.

We can now state the fundamental theorem of statistical learning theory, which shows the equivalence of several properties.

Theorem 2.1.5 (Fundamental Theorem of Statistical Learning [Shalev-Shwartz and Ben-David, 2014a]). Let $\mathcal{H}$ be a hypothesis class of functions from a domain $\mathcal{X}$ to $\{0, 1\}$ and let the loss function be the 0-1-loss. Then, the following are equivalent:

1. $\mathcal{H}$ has the uniform convergence property.
2. Any ERM rule is a successful agnostic PAC learner for $\mathcal{H}$
3. $\mathcal{H}$ is agnostically PAC learnable
4. $\mathcal{H}$ is realizably PAC learnable
5. Any ERM rule is a successful realizable PAC learner
6. $\mathcal{H}$ has finite VC dimension

Furthermore the VC dimension also characterizes the sample complexity of PAC learning.

Theorem 2.1.6 (Fundamental Theorem of Statistical Learning (Quantitative Version) [Shalev-Shwartz and Ben-David, 2014a]). Let $\mathcal{H}$ be a hypothesis class of functions from $\mathcal{X}$ to $\{0, 1\}$ and let the loss function be the 0-1 loss. Assume that $\text{VC}(\mathcal{H}) = d < \infty$. Then there are absolute constants C_1, C_2 such that:

1. $\mathcal{H}$ has the uniform convergence property with sample complexity

$$C_1 \frac{d + \log(1/\delta)}{\varepsilon^2} \leq m_{\mathcal{H}}^{\text{UC}} \leq C_2 \frac{d + \log(1/\delta)}{\varepsilon^2}$$

2. $\mathcal{H}$ is agnostic PAC learnable with sample complexity

$$C_1 \frac{d + \log(1/\delta)}{\varepsilon^2} \leq m_{\mathcal{H}} \leq C_2 \frac{d + \log(1/\delta)}{\varepsilon^2}$$

3. $\mathcal{H}$ is realizably PAC learnable with sample complexity

$$C_1 \frac{d + \log(1/\delta)}{\varepsilon} \leq m_{\mathcal{H}} \leq C_2 \frac{d \log(1/\varepsilon) + \log(1/\delta)}{\varepsilon}$$

2.2 General PAC Learning model

We consider a general notion of statistical learning tasks. These consist of the following elements

- a domain $\mathcal{Z}$ from which the input-instances/training-instances are sampled
- A class of benchmark models H (in some cases we denote it by $\mathcal{Q}$).
- A class of permissible data generating distributions $\mathcal{P} \subset \Delta(\mathcal{Z})$, where $\Delta(\mathcal{Z})$ denotes all distributions over the domain $\mathcal{Z}$.
- A set of possible outputs of a learner $\mathcal{F}$. Usually, $H \subseteq \mathcal{F}$.
- A loss/ approximation measure $L : \mathcal{F} \times \Delta(\mathcal{Z}) \rightarrow \mathbb{R}_0^+$ (where $\mathbb{R}_0^+$ denotes the set of non-negative real numbers).

The approximation error of a class H w.r.t. some data generating distribution P is defined as $\text{opt}(H, P) = \inf_{h \in H} L(h, P)$.

Definition 2.2.1 (PAC learnability). *The setup that we address here is more general than the common PAC learning setup in that we consider a class of permissible data generating distributions on top of the commonly discussed class of models. In the definition below we refer to a fixed set of learner outputs $\mathcal{F}$ and a fixed loss function $L : \mathcal{F} \times \Delta\mathcal{Z} \rightarrow \mathbb{R}_0^+$.*

- A pair of classes $H \times \mathcal{P}$ is α -agnostic PAC learnable w.r.t. to $L : \mathcal{F} \times \Delta\mathcal{Z} \rightarrow \mathbb{R}_0^+$, if there is a learner $\mathcal{A} : \bigcup_{m \in \mathbb{N}} \mathcal{Z}^m \rightarrow \mathcal{F}$ and a sample complexity function $m_H^\alpha : (0, 1)^2 \rightarrow \mathbb{N}$, such that for every $\varepsilon, \delta > 0$, every $P \in \mathcal{P}$ and every $m \geq m_H^\alpha(\varepsilon, \delta)$, we have

$$L(\mathcal{A}(S), P) \leq \alpha \cdot \text{opt}(H, P) + \varepsilon$$

with probability $1 - \delta$ over $S \sim P^m$.

We omit the parameter α in the definition when we refer to $\alpha = 1$.

- We denote by $m_H(\varepsilon, \delta)$ the minimum number m that satisfies the above requirement for $\alpha = 1$.
- We say a class of models H is learnable if $H \times \Delta(\mathcal{Z})$ is learnable.

- We say H is PAC-learnable in the realizable case w.r.t. L if $H \times \{P \in \Delta(\mathcal{Z}) : \text{opt}(H, P) = 0\}$ is PAC learnable with respect to L . We will sometimes refer to the sample complexity of realizable learning by m^{re} to distinguish it from the sample complexity of agnostic learning.
- We say a class of distributions $\mathcal{P}$ is PAC-learnable with respect to $L : \mathcal{F} \times \Delta(\mathcal{Z}) \rightarrow \mathbb{R}_0^+$ if $\mathcal{F} \times \mathcal{P}$ is PAC learnable with respect to L .

This definition captures many different PAC learning tasks that appear in the literature. In this thesis I will address the following settings:

Binary classification with additional requirements While we have seen that PAC learning of binary classifiers is beautifully characterized by the VC-dimension, there are additional requirements of interest that change this characterization. In particular we consider the following settings:

Robust Learning We consider cases in which data can be corrupted at test-time. The most prominent case is the setting of *adversarial robustness*, in which an adversary adds humanly imperceptible perturbations to an authentic test point in order to fool the learned classifier. This setting can be modeled via a loss function and has been widely discussed in the literature [Montasser et al., 2019]. Another setting we discuss and model by a loss function is *strategic classification*, where instances represent self-motivated individuals who change their feature vector in order to obtain their preferred classification outcome. In both robust learning settings we will consider the case where manipulation capabilities are not perfectly known to the learner.

Computable PAC Learning While our definitions so far have modeled learners as functions, in practice learners need to be implemented by algorithms. We propose the requirement that learners and hypotheses are computable. The details of this definition will be given in Part IV.

Distribution Learning. On important setting I am interested in is PAC learning of distributions with respect to total variation distance. The standard definition PAC learning of a distribution class $\mathcal{P}$ with respect to total variation(TV)-distance is captured by the above definition. In this case, we have $H = \mathcal{P}$, $\mathcal{F} = \Delta(\mathcal{Z})$ and $L = d_{\text{TV}} : \Delta(\mathcal{Z}) \times \Delta(\mathcal{Z}) \rightarrow \mathbb{R}_0^+$. A more detailed introduction into this setting is given in Chapter 4.

Classification Learning with distributions restricted to a fixed marginal. The standard PAC learning definitions for classification learning are ‘distribution free’. Namely, learnability requires success with respect to any data-generating probability distributions. [Benedek and Itai, 1991] proposed a variant of that notion that requires success only w.r.t. some fixed probability distribution. They then raise the idea of extending the definitions by considering a *class* of distributions. This idea is then picked up in [Dudley et al., 1994] with the following definition:

Definition 2.2.2. *A hypothesis class H is PAC-learnable with respect to a class of marginal distributions $\mathcal{D}$, if there is a learner $\mathcal{A}$ and a sample complexity function $m_{H,\mathcal{D}} : (0, 1)^2 \rightarrow \mathbb{N}$, if for every $\varepsilon, \delta > 0$, every $D \in \mathcal{D}$ and every labeling rule $h \in H$ and every $m \geq m_{H,\mathcal{D}}^{0/1}(\varepsilon, \delta)$, with probability $1 - \delta$ over $S \sim (D, h)^m$, we have*

$$L_P^{0/1}(\mathcal{A}(S)) < \varepsilon.$$

(where $L_P^{0/1}$ is the usual 0/1 loss).

We will say a class $\mathcal{P} = \{(D, h) : D \in \mathcal{D}, h \in H\}$ of distribution classes is *classification learnable*, if the class of labelling rules H is learnable with respect to $\mathcal{D}$.

Real valued functions with real valued losses. We will consider the case where $\mathcal{Y} = \mathbb{R}$, $\mathcal{Z} = \mathcal{X} \times \mathbb{R}$ (for an arbitrary metric space $\mathcal{X}$), $\mathcal{F} = \mathcal{Y}^{\mathcal{X}}$, $\mathcal{P} = \Delta(\mathcal{Z})$ and $L : \mathcal{F} \times \Delta(\mathcal{Z}) \rightarrow \mathbb{R}_0^+$ with $L(h, P) = \mathbb{E}_{(x,y) \sim P}[\ell(h, x, y)]$, where $\ell(h, x, y) = g(|h(x) - y|)$, where g is some continuous non-constant monotonous function with $g(0) = 0$.

Chapter 3

When does the Fundamental Theorem Generalize to other Learning Tasks?

The central theme of this thesis is to understand settings in which other learning tasks allow for a characterization similar to the fundamental theorem of statistical learning. In particular, we address the following questions:

- **Do we have a characterization through a dimension?** In particular, we address the question of whether we can characterize learnability by either
 - a qualitative characterizing dimension, which distinguishes learnable from non-learnable classes.
 - a quantitative dimension, which characterizes the sample-complexity of a dimension.

We see that the fundamental theorem of statistical learning theorem offers both of those characterizations. On the other hand the work of [Ben-David et al., 2019] showed that there are some learning problems whose learnability is independent of ZFC and thus are not characterizable by a combinatorial dimension. The question of finding such a characterization for different learning settings as been an active line of research. In Section 3.1.3 I discuss our formalization of combinatorial characterization – extending concepts from [Ben-David et al., 2019] and show some general criteria to obtain hardness results.

- **Are agnostic and realizable learnability equivalent?** The question whether learnability under realizability assumptions extends to non-realizable setting has a long history. For PAC learning of binary classification the fundamental theorem of statistical learning theory tells us that the two notions are equivalent. Going beyond binary classification, recent work [Hopkins et al., 2022a] shows that the equivalence of realizable and agnostic learnability extends across a wide variety of settings. These include models with no known characterization of learnability such as supervised learning with arbitrary distributional assumptions and more general loss functions, as well as a host of other popular settings such as robust learning, partial learning, fair learning, and the statistical query model. In contrast, my work in [Ben-David et al., 2023], [Agarwal et al., 2020], [Gourdeau et al., 2024] shows that this is not the case for all learning models.
- **Are all learnable classes learnable through an ERM/properly learnable?**

Another question that will come into play in several parts of this thesis is how corruption changes the landscape of learnability. Those corruptions can either come from adversaries or from instances representing self-motivated agents. In different parts of this thesis we will consider either manipulations that happen at test-time or manipulations that happen at training-time. We will see that different prior knowledge in the manipulation capabilities changes the learnability of a class. In the next section, we will discuss notions qualitative and quantitative characterizing dimensions and show some sufficient criteria that imply impossibility of characterizing a learning task by such dimensions. The following section will discuss when realizable and agnostic learnability are equivalent.

3.1 Characterizing Dimensions

3.1.1 Notions of characterization of learning tasks

Towards showing the “characterization” of some learning tasks is impossible, we need clear definitions of what such characterizations are.

We consider two common types of characterization of learning:

1. Quantitative notions that reflect the sample complexity of the learning task (the way the fundamental theorem of statistical learning shows that the *Vapnik-Chervonenkis dimension* characterizes the learning rates of leaning w.r.t. a given hypothesis class).

2. Qualitative notions that distinguish between learnable and non-learnable classes of models.

Below, we propose formal requirements for both types of characterizations.

All the notions that are known to characterize PAC learnability (rather than some form of weak learnability, or learnability up to some fixed accuracy) meet our proposed requirements. These include VC-dimension, Littlestone dimension, Natarajan dimension and Graph dimension.

Our main results show that no such characterization is possible for a variety of learning tasks, including the task of learning discrete distributions ([Kearns et al., 1994, Devroye and Lugosi, 2001, Silverman, 1986], learning of binary classifications w.r.t. a restricted sets of marginal distributions (Benedek and Itai [1991], Dudley et al. [1994]), and learnability of classes of real-valued functions with continuous losses.

We now elaborate our definitions of such notions.

3.1.2 Notions of quantitative characterization of statistical learning

Let $\mathcal{C}$ denote the family of all learnable classes w.r.t. some learning task.

Definition 3.1.1. A strong sample complexity dimension is a mapping from $d : \mathcal{C} \rightarrow \mathbb{N} \cup \{\infty\}$, such that a class $\mathcal{Q}$ of models is PAC learnable if and only if $d(\mathcal{Q}) \neq \infty$ and there are functions $f : \mathbb{N} \rightarrow \mathbb{N}$ and $g : (0, 1)^2 \rightarrow \mathbb{N}$ such that for every PAC learnable class of distributions $\mathcal{Q}$, $m_{\mathcal{Q}}(\varepsilon, \delta) \leq f(d(\mathcal{Q}))g(\varepsilon, \delta)$ for all $(\varepsilon, \delta) \in (0, 1)$. In other words, there is a sample complexity upper bound function that factorizes into a factor depending only on the dimension of a class and a factor depending only on the accuracy and confidence parameters.

Note that the fundamental theorem of statistical learning [Blumer et al., 1989] shows that the VC-dimension is a strong sample complexity dimension for binary classification.

Definition 3.1.2. A weak sample complexity dimension is a mapping from $d : \mathcal{C} \rightarrow \mathbb{N} \cup \{\infty\}$, such that a class $\mathcal{Q}$ of models is PAC learnable if and only if $d(\mathcal{Q}) \neq \infty$ and there are functions $f : \mathbb{N} \times (0, 1)^2 \rightarrow \mathbb{N}$ such that for every PAC learnable class of distributions $\mathcal{Q}$, and every $(\varepsilon, \delta) \in (0, 1)$, $m_{\mathcal{Q}}(\varepsilon, \delta) \leq f(d(\mathcal{Q}), \varepsilon, \delta)$. In other words, all the information needed about a class of distributions $\mathcal{Q}$ to determine (or upper bound) its sample complexity function $m_{\mathcal{Q}}(\varepsilon, \delta)$ is captured in its dimension $d(\mathcal{Q})$.

We will also sometimes refer to monotone real mappings $d : \mathcal{C} \rightarrow \mathbb{R} \cup \{\infty\}$ as "weak sample complexity dimensions" if there exists a corresponding f satisfying the condition in Definition 3.1.2.

Clearly, every strong sample complexity dimension is also a weak one. We also note that, for a satisfying characterization, one might also require a lower bound of the sample complexity in terms of f and d . However, as we are only presenting impossibility results, it suffices to show that even this less ambitious goal is not achievable.

3.1.3 Notions of qualitative characterization of statistical learning

The notions of dimension that we have discussed above were *quantitative* - aiming to capture the sample complexity functions of learning classes. We showed that such dimensions do not exist for problems like distribution learning.

Lacking quantitative notions one can still seek *qualitative* characterizations of learnability. Namely, conditions that distinguish learnable classes from non-learnable ones. In the case of binary classification tasks, the distinction between finite and infinite VC-dimension serves as such a characterization.

Definition 3.1.3. A finitary characterization of learnability for a learning task is a countable set of formulas¹ W such that:

1. A class H is not learnable if and only if it satisfies all the formulas in W .
2. For every $\alpha \in W$ and every H that satisfies α there is a finite subset $H_\alpha \subseteq H$ such that for every H' , if $H_\alpha \subseteq H'$ then H' satisfies α .

We say that a finitary characterization W is uniformly bounded if for every $\alpha \in W$ there is a finite number n_α such that for every H satisfying α there is a subset $H_\alpha \subseteq H$ as above of size at most n_α .

Note that most of the known characterizations of learnability of learning tasks are finitary.

¹formally, these are properties of the learning problem expressed as first order formulas in many-sorted logic that has types/sorts for elements of the class of models $\mathcal{F}$, generating distributions (members of $\mathcal{P}$), domain elements and rational numbers for values of the loss function L . For brevity we keep it be clarified by the examples below.

Examples:

1. The characterization of binary classification learning by VC-dimension; The characterizing W can be any set that contains the formulas stating " H shatters a set of size d " for infinitely many d 's.
2. The characterization of online learnability by the Littlestone dimension; The characterizing W can be any set that contains the formulas stating " H L -shatters a tree of depth d " for infinitely many d 's. dimension characterizing robust learning
3. The characterization of multi-class learnability with finite label sets by the finiteness of the Natarajan dimension, of by the finiteness of the graph dimension.
4. The characterizations by a combinatorial dimension based on the one-inclusion graph. In these characterizations, the complexity of a problem is demonstrated by a finite graph (the 1-inclusion graph). However, there is no a-priori bound to the sizes of these graphs. The resulting complexity parameters are reflected by either the maximum degree or maximum out-degree of some orientation of that graph. Such a characterizations were shown for multi-class learning [Daniely and Shalev-Shwartz, 2014], [Brukhim et al., 2022], partial concept classes [Alon et al., 2021], and adversarially robust learning [Montasser et al., 2022]. In general the graphs used in these definitions can be infinite and thus do not a priori fulfill the finite evidence property. However in some cases the dimensions are formulated with respect to finite subgraphs that serve as evidence sets [Montasser et al., 2022], while in others the existence of finite evidence can be shown through compactness arguments [Brukhim et al., 2022]. However in both of these cases the evidence sets would not be uniform.

There are also several conjectured characterizations that fall into this category. For example, characterizing the learnability of a class of probability discrete distributions by the finiteness of the VC-dimension of the Yatracos sets induced by that class. For a class F of functions from X to $\mathbb{R}$, their Yatracos class is the family of subsets of X defined as

$$Y(F) := \{ \{x \in X : f_1(x) \geq f_2(x)\} \text{ for some } f_1, f_2 \in F \}$$

3.1.4 Comparing the different notions of characterization

The two types of characterizations introduced above are related but none of them implies the other. Sample complexity dimensions do not restrict the format (or syntax) of a

characterization - the dimension function d can be *any* function (from classes to reals or natural numbers). In this respect, the notion of finitary characterization is more restrictive - it restricts the format of the characterization.

On the other hand, finitary characterizations are weaker, in the sense that they do not provide any information about learning rates. They are only required to distinguish learnable from non-learnable classes.

In many cases, however, there are characterizations that meet both definitions. Every notion of sample complexity dimension where the statements of the form $d(H) \geq k$ have finite size ‘evidences’ (like a set of members of H shattering a domain subset). In such case the finitary characterization set W is just the set of statements $\{d(H) \geq k : k \in \mathbb{N}\}$.

The notion of *finite character dimension* of Ben-David et al. [2017b] has two definitions there. The first one requires that there is a class parameter, $D(F)$ such that a class F is PAC learnable if and only if $D(F)$ is finite. They also require *finite character*. Namely, that the parameter (or dimension) $D(F)$ has a “finite” character if for every $d \in \mathbb{N}$ and a class F , the statement $D(F) \geq d$ can be demonstrated by a finite set of domain points and a finite set of members of F . This definition is clearly equivalent (for classes of functions) to our notion of finitary characterization.

The second definition there, requiring that the statements $D(F) \geq d$ can be expressed by certain first-order formulas, is more syntactically restrictive than our definition, but on the other hand, more lenient as it does not require the existence of finite evidence sets.

Finally, it is worth mentioning again that all the types of characterizations that we discuss in the paper refer to PAC Learnability. Namely, learnability to arbitrarily small accuracy parameter. In setups where no such a characterization is known to exist, there is also a discussion of *scale-sensitive* learnability dimensions (e.g., Alon et al. [1997]). Scale-sensitive characterizations allow setting a separate condition for each level of accuracy. Our results do not rule out scale-sensitive characterizations. One should note that in setups satisfying Boosting a scale-sensitive characterization is also a PAC learnability characterization (and therefore ruled out by our results).

3.1.5 Sufficient conditions for Uncharacterizability by Scale-Invariant Dimensions

Our main tool for showing the impossibility of having sample complexity dimensions that provide a quantitative characterization for learning tasks is the basic notion of cofinality

of an ordered set. We start by recalling its definition and some basic properties:

Some notions of ordered sets

Definition 3.1.4 (Cofinality). *Let $(X, \leq)$ be an ordered set.*

- *For subsets $A, B \subseteq X$, we say that A is cofinal in B if for every $b \in B$ there exists some $a \in A$ such that $b \leq a$.*
- *The cofinality of an ordered set $(X, \leq)$ is the minimal cardinality of a subset A that is cofinal in X .*

Note that for subsets A, B, C of X , if A is cofinal in B and B is cofinal in C then A is cofinal in C .

Definition 3.1.5 (Dominance ordering of functions). *Let $(X, \leq_X)$, $(Y, \leq_Y)$ be linearly ordered sets where X has no maximal element. For functions $f, g : X \rightarrow Y$, we say that f eventually dominates g if there exists some $x \in X$ such that for every $x' \in X$, if $x \leq_X x'$ then $g(x') \leq_Y f(x')$. We denote this relation by $g \leq_{ed} f$.*

Claim 3.1.6. Consider $\mathbb{N}^{\mathbb{N}}$ (the set of all functions from the natural numbers to natural numbers). The cofinality of $(\mathbb{N}^{\mathbb{N}}, \leq_{ed})$ is uncountable.

Proof. Consider any countable $A \subseteq \mathbb{N}^{\mathbb{N}}$, let $\{g_n : n \in \mathbb{N}\}$ be an enumeration of the members of A . Define $f : \mathbb{N} \rightarrow \mathbb{N}$ by $f(n) = \max\{g_i(n) : i \leq n\} + 1$. Clearly f dominates every member of A (and no member of A dominates f) showing that A is not cofinal in $(\mathbb{N}^{\mathbb{N}}, \leq_{ed})$. $\square$

3.1.6 A condition implying no sample complexity dimension

The next lemma applies this notion to the existence of characterizing sample complexity (quantitative) dimensions.

Lemma 3.1.7. *Let $\mathcal{C}$ denote the family of all learnable classes w.r.t. some learning task. For any given learnable class $\mathcal{Q}$ consider the function $m_{\mathcal{Q}}(\frac{1}{\cdot}, \frac{1}{\cdot}) : \mathbb{N} \rightarrow \mathbb{N}$ that maps a natural number k to $m_{\mathcal{Q}}(\frac{1}{k}, \frac{1}{\cdot})$.*

If the set $\{m_{\mathcal{Q}}(\frac{1}{\cdot}, \frac{1}{\cdot}) : \mathcal{Q} \in \mathcal{C}\}$ is cofinal in $\mathbb{N}^{\mathbb{N}}$ (under the eventual dominance ordering) then there exists no weak sample complexity dimension for that task.

Proof. Let $d : \mathcal{C} \rightarrow \mathbb{N}$ be a weak sample complexity dimension. For any $d \in \mathbb{N}$, let $f'_d(k) = f(d, 1/k, 1/7)$ for all k . $\{f'_d : \mathcal{Q} \in \mathcal{C}\}$ is cofinal in the set of sample complexity functions $\{m_{\mathcal{Q}}(1/\cdot, 1/7) \in \mathbb{N}^{\mathbb{N}} : \mathcal{Q} \in \mathcal{C}\}$ (under the $\leq_{ed}$ ordering of functions). Thus the cofinality of $(\{m_{\mathcal{Q}}(1/\cdot, 1/7) : \mathbb{N} \rightarrow \mathbb{N} : \mathcal{Q} \in \mathcal{C}\}, \leq_{ed})$ is at most countable. Since we assume that the set $\{m_{\mathcal{Q}}(\frac{1}{\cdot}, \frac{1}{7}) : \mathcal{Q} \in \mathcal{C}\}$ is cofinal in $\mathbb{N}^{\mathbb{N}}$, we get a contradiction to the uncountable cofinality of $\mathbb{N}^{\mathbb{N}}$. □

Definition 3.1.8. For an ordered set $(\mathcal{X}, \leq)$, we say that a notion of dimension $d : \mathcal{C} \rightarrow \mathcal{X}$ is monotonic if for every pair of classes $\mathcal{Q}_1, \mathcal{Q}_2$, the implied sample complexity functions $f(d(\mathcal{Q}), 1/k, 1/7)$ are monotonically increasing. Namely,

$$d(\mathcal{Q}_1) \geq d(\mathcal{Q}_2) \text{ implies that } f(d(\mathcal{Q}_1), 1/k, 1/7) \geq_{ed} f(d(\mathcal{Q}_2), 1/k, 1/7).$$

Theorem 3.1.9. If the set $\{m_{\mathcal{Q}}(\frac{1}{\cdot}, \frac{1}{7}) : \mathcal{Q} \in \mathcal{C}\}$ is cofinal in $\mathbb{N}^{\mathbb{N}}$ (under the eventual dominance ordering) then there exists no monotonic real-valued function that is a weak sample complexity dimension for that task.

Proof. Noting that the real numbers have countable cofinality, the proof of the natural-valued dimension applies to the monotonic real-valued dimension as well. □

These two results have implications for several learning settings. In particular we show the following results.

Theorem 3.1.10. There is no weak sample complexity dimension for distribution learning (neither in the realizable nor in the 3-agnostic case of distribution learning).

Corollary 3.1.11. There exist no monotonic real-valued function that is a weak sample complexity dimension for distribution learning.

Theorem 3.1.12. There is no weak sample complexity dimension for classification learning of distribution classes.

Theorem 3.1.13. There is no weak sample complexity dimension for PAC learning real-valued function classes with respect to L^g (in neither the realizable nor the 1-agnostic case).

3.1.7 A condition implying no finitary characterization

In order to state our sufficient condition for the impossibility of a finitary characterization, we first need to introduce a general version of ε -approximation.

Definition 3.1.14. *For a given learning task, we say that a class of outputs $H' \subset \mathcal{F}$ is an ε -approximation for $H \times \mathcal{P}$ w.r.t. to L , if for every $(h, p) \in H \times \mathcal{P}$, there is a $h' \in H'$ such that $L(h', p) \leq L(h, p) + \varepsilon$. Using the same slight abuse of notation as in the definition of PAC learning, we will say that:*

- H' is an ε -approximation for H if it is an ε -approximation for $H \times \Delta(\mathcal{Z})$
- H' is an ε -approximation for $\mathcal{P}$ if it is an ε -approximation for $\mathcal{F} \times \mathcal{P}$

We now state and prove our main general result concerning our notion of qualitative characterizations of learnability.

Theorem 3.1.15. *The learnability of any learning task that satisfies the following two properties cannot be characterized by a finitary characterization.*

1. *Every finite union of learnable classes of hypotheses is learnable.*
2. *There exists a learnable class H_0 and non-learnable classes $\{H_k : k \in \mathbb{N}\}$ such that for every $k \in \mathbb{N}$, H_0 is an ε_k approximation of H_k and $\lim_{k \rightarrow \infty} \varepsilon_k = 0$.*

Proof. Assume, b.w.o.c. that W is a finitary characterization of the learning task. Let $W = \{\alpha_k : k \in \mathbb{N}\}$ be any enumeration of W . For each H_k let $\hat{H}_k$ be a finite subset of H_k be such that every $H \supset \hat{H}_k$ satisfies α_k and let $\hat{H} = \bigcup_{k \in \mathbb{N}} \hat{H}_k$. On one hand, since for every k , $\hat{H}_k \subseteq \hat{H}$, $\hat{H}$ satisfies every α_k and is therefore not learnable (by the first requirement from a characterizing W). Towards a contradiction, let us show that $\hat{H}$ is learnable. This holds because given any $\varepsilon > 0$, the set $K_\varepsilon = \{k : \varepsilon_k \geq \varepsilon/2\}$ is a finite set. Therefore by our assumptions on the learning task, the class $H_0 \cup \bigcup_{k \in K_\varepsilon} \hat{H}_k$ is learnable. Therefore, for any given δ there is a learner $\mathcal{A}$ and some $m(\varepsilon/2, \delta)$ so that training samples of larger size guarantee $(\varepsilon/2, \delta)$ success for $\mathcal{A}$ on such samples. Since H_0 is an $\varepsilon/2$ approximation to each H_k for which $k \notin K_\varepsilon$ (w.r.t. the given learning instance) so being $\varepsilon/2$ off the minimum loss minimizer on H_0 implies being within ε of the loss minimizer in $\hat{H}$. $\square$

This result has implications for several learning tasks, which can be shown to be inherently scale sensitive. In particular we get the following theorems as corollaries of Theorem 3.1.15.

Theorem 3.1.16. *The learnability of classes of discrete distributions cannot be characterized by a finitary characterization. This statement holds both for realizable PAC learnability and for 3-agnostic PAC learnability.*

Theorem 3.1.17. *There is no finitary characterization of classification learning for distribution classes.*

Theorem 3.1.18. *There is no finitary characterization of PAC learning real-valued function classes with respect to L^q (in neither the realizable nor the 1-agnostic case).*

3.1.8 Discussion

We discuss both the quantitative and qualitative characterizations and proposed some general properties of learning tasks that imply that their learnability is not captured by such characterizations. Our work inspired by the work of Ben-David et al. [2017b] which was the first to show the existence of a learning task that cannot be characterized. While their work laid the groundwork and gave a first formal definition of general dimensions for statistical learnability, we extended those definitions and also proposed a definition for quantitative characterizability.

Our work expands the understanding of uncharacterizability of learning problems in crucial ways. The results from Ben-David et al. [2017b] applied to a newly defined learning task - Expectation Maximization - and relied on the existence of classes whose learnability is undecidable in ZFC. In contrast, our results apply to several natural learning tasks whose characterizability have thus far eluded the community (Diakonikolas [2016]) and are ‘absolute’ in not referring to notions of provability.

Another distinction to the EMX learning task in Ben-David et al. [2017b] is the fact that the definition of EMX learning requires learning to be *proper*, i.e. the output of a successful learner needs to be element of the class that is being learned. Without this requirement the EMX setting becomes trivial, as any class can be learned by the constant learner that outputs the whole domain set for any input. In contrast, our results address the more general case of learning (and can also be easily extended to the proper case as well).

We note that all of our results rely on the construction of a sequence ε -weakly learnable classes for decreasing ε , which are not fully PAC learnable. For most tasks with known characterizations, there is an equivalence between weak learning and PAC learning. It

might be interesting to further explore the connection between that equivalence and the characterizability of a learning problem.

We believe that we have not exhausted the implications of our approach and that our definitions of characterizations and our techniques are also applicable to more learning tasks.

3.2 Agnostic vs Realizable Learning

In many settings realizable and agnostic learnability are equivalent. In the case of binary classification, this equivalence can be easily explained by the equivalence between uniform convergence and realizable PAC learning. Uniform convergence gives a guarantee that any ERM would be a successful learner in both the realizable and the agnostic case, as the sample error or any hypothesis of the class - including the one minimizing the sample risk - is close to its true risk. However uniform convergence is not a necessary criterion for the equivalence between agnostic and realizable learning. For example in multi-class learning with infinite label sets and in adversarial robustness uniform convergence does not hold [Daniely et al., 2015, Montasser et al., 2019]. However in both settings an equivalence of agnostic and realizable learning can be shown via compression schemes [David et al., 2016, Montasser et al., 2019]. The insight is based on the fact that given an arbitrary sample S a compression based on a maximal realizable sub-sample sample $S' \subset S$ also serves as an agnostic compression for S . More generally, Hopkins et al. [2022a] suggested that "Realizable Learning is all you need" and suggested general strategies for showing how to reduce agnostic learning to realizable learning. We will now briefly present one of their proposed strategy and then discuss how some of the requirements for this strategy are not fulfilled in some of the models we discuss in this work.

Algorithm 1: Agnostic to realizable reduction [Hopkins et al., 2022a]

Input: realizable learner $\mathcal{A}$, unlabelled sample oracle O_U , labelled sample oracle O_L
 Draw an unlabelled sample $S_U \sim O_U$ and a labelled sample $S_L \sim O_L$
 Run $\mathcal{A}$ on all possible labellings of S_U to get $C(S_U) = \{\mathcal{A}(S_U, h(S_U)) | h \in H|_{S_Y}\}$
Return $h \in C(S_U)$ with lowest error on S_L

We note that in the case where only a labelled sample is given it can be randomly split into S_U and S_L , where labels from S_U are ignored.

Hopkins et al. [2022a] show that with this kind of algorithm agnostic learnability can be reduced to realizable learnability in the classification learning setting with a restricted

family of marginal distributions [Benedek and Itai, 1991]. There are other variations of this learning strategy, all based on the idea that the realizable learner is used on different candidate samples — either different labellings of the same unlabelled sample or different subsamples of the sample — one of which is guaranteed to yield a similar optimal hypothesis as the agnostic sample. The realizable learner now guarantees that a close to optimal hypothesis is indeed found and part of the candidate hypothesis class. In a second step the optimal candidate hypothesis among a finite set of hypotheses is now provided. Since in most learning settings we consider finite hypothesis classes satisfy uniform convergence, finding the optimal class within the candidate set is easy despite the fact that this is an agnostic learning problem. While this approach works in some settings in this work we present several scenarios in which such an approach provably has to fail and indeed agnostic and realizable learnability are distinct. In particular we show the following examples for non-equivalence between realizable and agnostic learnability:

Distribution Learning In Chapter 6 we show that there are classes of distributions, which are learnable in the realizable case, but not in the agnostic case. We construct a class of distributions that does not satisfy uniform convergence in such a way that for every distribution in the class there is a in a small support element that is unique to the distribution. In the realizable case we can now identify a distribution if the sample contains an element of this unique support bit. Thus the class is realizable learnable. In the agnostic setting this element does not necessarily have mass in the ground-truth distribution, therefore not allowing a learner to exploit this identification. The reason why an approach like the one described by Hopkins et al. [2022a] does not work is that distribution learning is unsupervised. Looking at similar approaches, we also see that there is no clear way of “generating a set of (sub)samples such that one of the samples is guaranteed to be realizable” which could then be used to create a finite candidate set, as there is no clear notion of “realizable sample” in distribution learning. A compression approach also does not generalize from the realizable to the agnostic setting, as the points necessary for compression might have high probability of occurring in a sample from a distribution of the class, while distributions with small total variation distance to the class might not have any points suitable for compression in their support. In contrast, a different picture emerges if we consider the additive robust setting also known as Huber contamination [Huber, 1964], where ground truth distributions consist of a mixture of two distributions, with a high weight component that is element of the distribution class and a low weight component that can be an arbitrary distribution. In this case with high probability any such distribution contains a large sub-sample which can be viewed as a sample from a distribution of the class which is close to the ground-truth distribution. Thus the guarantee of a

realizable learner for the class would produce a good candidate distribution for the additive robust setting. Similar to the approach discussed a learner can now iterate over all sub-samples to produce a finite candidate set. The best hypothesis from the class can then be learned via uniform convergence over the Yatacos sets [Yatacos, 1985]. Similarly, if the class can be learned via a realizable compression scheme the same compression scheme can be used in the additive robust case.

Robust Learning with Unknown Manipulation Capabilities While both strategic classification and adversarially robust classification learning *with known manipulation capabilities* do not satisfy uniform convergence, in both settings realizable and agnostic learnability are equivalent. This equivalence comes from the equivalence of learnability to learning via sample compression. A realizable compression scheme can be made into an agnostic compression scheme by using the compression on the largest realizable sub-sample. However, in adversarially or strategically robust learning settings with unknown manipulation capabilities this equivalence breaks down and we can show that there are classes that are learnable without additional oracle access in the realizable case but not in the agnostic case. The reason for this is that the true loss function is not known to the learner, if the manipulation capabilities are not known. Thus when given a training sample in the agnostic case, it is impossible to determine its largest realizable sub-sample — a step necessary for generalizing from realizable to agnostic compression. In particular, if the class of possible candidates for manipulation capabilities is ordered by inclusion, the guarantee of realizability is sufficient to overcome the uncertainty in manipulation capabilities. This works via a novel maximally robust empirical risk minimizing (MRERM) paradigm, which chooses a hypothesis with respect to the “maximum manipulation capability” that still allows for a realizable labelling of the sample. However, such a paradigm can only be used in the realizable case. These results are discussed in Part III.

Computable PAC Learning We show that if we require learners to be computable we do not have an equivalence between agnostic and realizable learnability. While every hypothesis class with finite VC dimension that can be recursively enumerably represented is properly learnable in the realizable case, we show that there are recursively enumerably representable classes of VC dimension 1 which can not be properly agnostically CPAC learned. This result was later strengthened to also hold for the improper case [Sterkenburg, 2022]. The separation here comes from the fact that the guarantee of realizability for the distribution, also guarantees that any sample of that distribution is realizable. This in turn gives a stopping criterion for an ERM procedure that iterates through all members of the class. On the other hand in the

agnostic setting such a stopping criterion does not necessarily exist. Furthermore, there are recursively representable hypothesis classes of finite VC dimension for which there is no computable realizability oracle — ,i.e., no algorithm, that given a sample decides whether the sample is realizable. However, any reduction from agnostic to realizable learning that creates a set of candidates needs to either run on potentially non-realizable samples or use a realizability oracle. Thus it is impossible to have such a general reduction with a guarantee to always halt.

Part II

Distribution Learning

Chapter 4

Distribution Learning Preliminaries

4.1 Introduction

Distribution learning (sometimes called *density estimation*) refers to the following statistical task: given i.i.d. samples from some (unknown) distribution p , produce an estimate of p . This is one of the most fundamental and well-studied questions in both statistics Devroye and Lugosi [2001] and computer science Diakonikolas [2016], sometimes equivalent to classic problems of parameter estimation in parametric settings (e.g., Gaussian mean estimation).

It is easy to see that no learner can meaningfully approximate any given p without having some prior knowledge. In the realizable setting, the problem then becomes: assuming the sample generating distribution p belongs to a given class of distributions $\mathcal{Q}$, and given parameters $\varepsilon, \delta \in (0, 1)$, output some distribution $\hat{p}$ such that with probability at least $1 - \delta$, (over the generated input samples) the statistical distance between p and $\hat{p}$ is at most ε . Specifically, we employ *total variation distance*, the most studied metric in density estimation Devroye and Lugosi [2001], Diakonikolas [2016], using $d_{\text{TV}}(p, q)$ to denote the total variation distance between distributions p and q . This case, when $p \in \mathcal{Q}$, is often called the *realizable* setting. If, for some particular class $\mathcal{Q}$, this is doable with a finite number of samples $m(\varepsilon, \delta)$ (independent of the choice of p), then we say the distribution class is (ε, δ) -*learnable*.¹ A class is *learnable* if it is (ε, δ) -*learnable* for every $(\varepsilon, \delta) \in (0, 1)^2$. A significant amount of work has focused on proving bounds on $m(\varepsilon, \delta)$ for a number of classes $\mathcal{Q}$ of interest – for example, one can consider the class $\mathcal{Q}$ of all Gaussian distributions $\mathcal{N}(\mu, \Sigma)$ in some Euclidean space $\mathbb{R}^d$.

¹For the sake of exposition, we defer formal definitions of our learnability notions to Section 4.1.1.

However, this framework is restrictive in the sense that it requires the unknown distribution to be *exactly* a member of the class $\mathcal{Q}$ of interest. This may not be the case for a variety of possible reasons, including some innocuous and some malicious. As one example, while it is a common modelling assumption to posit that data comes from a Gaussian distribution, Nature rarely samples exactly from Gaussians, we consider this only to be a convenient *approximation*. More generally, the class $\mathcal{Q}$ that is part of the learner’s input can be thought of as reflecting some prior knowledge about the task at hand. Such prior knowledge is almost always only an approximation of reality. Alternatively, we may be in an adversarial setting, where a malicious actor has the ability to modify an otherwise well-behaved distribution, say by injecting datapoints that are not i.i.d. samples from p , (known in the machine learning literature as data poisoning attacks [Biggio et al. \[2012\]](#), [Steinhardt et al. \[2017\]](#), [Diakonikolas et al. \[2019\]](#), [Goldblum et al. \[2020\]](#), [Geiping et al. \[2021\]](#), [Lu et al. \[2023\]](#)).

More formally, the classic problem of *agnostic* learnability is generally described as follows: given a (known) class of distributions $\mathcal{Q}$, and a (finite) set of samples drawn i.i.d. from some (unknown) distribution p , find a distribution $\hat{p}$ whose statistical distance from p is not much more than that of the closest member of $\mathcal{Q}$. It is not hard to see that this is equivalent to a notion of *robust* learnability, where the distribution p is not viewed as arbitrary, but instead an adversarial corruption of some distribution within $\mathcal{Q}$.² Given their equivalence, throughout this work, we will use agnostic and robust learnability interchangeably.

The difference between a robust setting and the previous realizable one is that now, instead of assuming $p \in \mathcal{Q}$ and asking for an arbitrarily good approximation of p , we make no prior assumption about the data-generating distribution and only ask to approximate as well as (or close to) what the best member of some “benchmark” class $\mathcal{Q}$ can do.

The question of characterizing distribution learning has long remained open for both the realizable and the agnostic/robust setting [Diakonikolas \[2016\]](#). While there are many guarantees for particular distribution classes, such as the class of mixtures of k Gaussians [Ashtiani et al. \[2018\]](#) there is no known parameter or dimension characterizing learnability for distribution learning, neither in terms of characterizing learnability nor in terms of characterizing the sample complexity.

In this work we address this question, showing that any such characterization is necessarily scale-sensitive, i.e., dependent on the error parameter ε . We show this necessity for

²A related notion of robust learnability instead imagines the adversary modifies the *samples* from a distribution in $\mathcal{Q}$, rather than the distribution itself. This *adaptive* model is discussed further in Section [7.1.1](#).

both the realizable and the agnostic case.

Furthermore, we address the following question: Assuming a class of distributions is learnable, under which notions of robustness is it guaranteed to be robustly learnable? We will show that realizable learnability does not imply agnostic/general robust learnability. This means that any (necessarily scale-sensitive) characterization of distribution learnability needs to distinguish between realizable and agnostic learnability. We focus entirely on information-theoretic learnability, and eschew concerns of computational efficiency. Indeed, our question of interest is so broad that computationally efficient algorithms may be too much to hope for.

We shall consider a few variants of robust learnability. Specifically, we will impose requirements on the nature of the difference between the data-generating distribution p and members of the class $\mathcal{Q}$. Obviously, such requirements can only make the task of robust learning easier.

One such model that we consider is *additive robustness*. The underlying sample-generating distribution is restricted to be a mixture of a distribution p from $\mathcal{Q}$, and some arbitrary ‘contaminating’ distribution q . In the Statistics community, this model is known as *Huber’s contamination model* [Huber \[1964\]](#).³ Analogously, one also consider *subtractive robustness*. It includes the case where the starting point is a distribution in the class $\mathcal{Q}$, but a fraction of the probability mass is removed and samples are drawn from the resulting distribution (after rescaling). These two models are related to adversaries who can add or remove points from a sampled dataset, see discussion at the end of Section 7.1.1.

A significant line of work focuses on understanding the sample complexity of agnostic distribution learning (see examples and discussion in Section 4.3). Most study restricted classes of distributions, with analyses that are only applicable in certain classes. Some works have found quantitative separations between the different robustness models. For instance, for the specific class of Gaussian probability distributions [Diakonikolas et al. \[2016\]](#), [Lai et al. \[2016\]](#), [Diakonikolas et al. \[2017b, 2018\]](#) give strong evidence that efficient algorithms for mean estimations can achieve better error if they are only required to be additively robust, rather than robust in general. However, such findings are again restricted to specific cases, and say little about the overall relationship between learnability in general and these various notions of robust learnability.

Current results leave open a more comprehensive treatment of robustness in distribution

³We note that our additive robustness definition differs slightly from Huber’s contamination model. However the two models are equivalent up to a small factor. We discuss this in more detail when we define additive robust learning and give an alternative definition called Huber additive robustness. Our results hold for both additive robustness and Huber additive robustness.

learning. Specifically, what is the relative power of these different robustness models, and what is their impact on which types of distributions are learnable? Are there more generic ways to design robust learning algorithms?

Our main contributions are the following:

- There is no scale-invariant dimension characterizing the learnability of a distribution class (Chapter 5)
- There is no scale-invariant dimension characterizing the sample complexity of distribution learning (Chapter 5)
- We show that there are distribution classes that are realizably learnable but not agnostically learnable (Chapter 6)
- We give a generic algorithm which converts a learner for a distribution class into a learner for that class which is robust to additive corruptions. (Chapter 7)
- We show that there exist distribution classes which are learnable, but are no longer learnable after subtractive corruption.(Chapter 7)

Other results explore implications related to compression schemes.

This part of the thesis is based on two papers:

- "Inherent limitations of dimensions for characterizing learnability of distribution classes." (joint work with Shai Ben-David; published at COLT 2024): Chapter 5
- "Distribution Learnability and Robustness"(joint work with Shai Ben-David, Alex Bie and Gautam Kamath; published at NeurIPS 2023): Chapter 6 and Chapter 7

The writing of the current chapter is based in part on both of these works. Our work on robust learnability also has implications for private learning which are further explored in the published paper [Ben-David et al., 2023].

4.1.1 Definitions of Learnability

In order to more precisely describe our results, we define various notions of learnability. We start with the standard notion of PAC learnability for a distribution class. We get samples from a distribution p belonging to a distribution class $\mathcal{Q}$, and the goal is to output a distribution similar to p . Before defining distribution learning, we will first define the distance measure between distributions we are interested in.

Definition 4.1.1 (Total Variation Distance). *The total variation distance $d_{\text{TV}} : \Delta(\mathcal{X}) \times \Delta(\mathcal{X}) \rightarrow [0, 1]$ between two distributions p and q is defined as*

$$\begin{aligned} d_{\text{TV}}(p, q) &= \sup_{A \subset \mathcal{X}: A \text{ measurable}} |p(A) - q(A)| \\ &= \int \frac{1}{2} |p(x) - q(x)| dx \end{aligned}$$

With a slight abuse of notation, we use $p(A)$ to denote the probability mass of a set $A \subset \mathcal{X}$ according to p and use $p(x)$ for the density of a point $x \in \mathcal{X}$ according to p . We will primarily discuss the case of discrete distributions, i.e. $\mathcal{X} = \mathbb{N}$. Thus for the most part of our discussion these notions are equivalent.

Definition 4.1.2 (Realizable Learnability of distribution classes). *We say that a class $\mathcal{Q}$ of probability distributions is learnable (or, realizably learnable) if there exists a learner A and a function $m_{\mathcal{Q}}^{\text{re}} : (0, 1)^2 \rightarrow \mathbb{N}$, such that for every probability distribution $p \in \mathcal{Q}$, and every $(\varepsilon, \delta) \in (0, 1)^2$, for $m \geq m_{\mathcal{Q}}^{\text{re}}(\varepsilon, \delta)$ the probability over samples S of that size drawn i.i.d. from the distribution p that $d_{\text{TV}}(A(S), p) \leq \varepsilon$ is at least $1 - \delta$.*

We next introduce the more challenging setting of robust, or agnostic, learning. In this setting, the sampled distribution is within bounded distance to the distribution class $\mathcal{Q}$, rather than being in $\mathcal{Q}$ itself.

We will introduce a notion (more commonly considered in the agnostic learning literature) which does not assume the distance from the sampling distribution to be fixed, but which gives a bound that depends on that distance.

Definition 4.1.3 (Agnostic/Robust learnability of distribution classes). *For $\alpha > 0$, we say that a class $\mathcal{Q}$ of probability distributions is α -robustly learnable (also referred to as α -agnostically learnable) if there exists a learner A and a function $m_{\mathcal{Q}}^{\alpha} : (0, 1)^2 \rightarrow \mathbb{N}$, such that for every probability distribution p , and $(\varepsilon, \delta) \in (0, 1)^2$, for $m \geq m_{\mathcal{Q}}^{\alpha}(\varepsilon, \delta)$ the probability over samples S of that size drawn i.i.d. from the distribution p that $d_{\text{TV}}(A(S), p) \leq \alpha \cdot \min\{d_{\text{TV}}(p, p') : p' \in \mathcal{Q}\} + \varepsilon$ is at least $1 - \delta$.*

We say a class $\mathcal{Q}$ is robustly (or agnostically) learnable if there is $\alpha \in \mathbb{R}$ such that $\mathcal{Q}$ is α -robustly learnable.⁴

⁴In other works robust or agnostic learning can also refer to the case where $\alpha = 1$. In those cases what we call robust or agnostic learnability is referred to as semi-agnostic learning.

We note, that the definitions of agnostic and realizable learnability of distribution classes is an instance of the General PAC-learning framework of Definition 2.2.1.

In Chapter 7 we will further explore different definitions of robust learnability.

We note that these definitions are special cases of the PAC learning definition in the Setup Section (Definition 2.2.1), where $\mathcal{Z} = \mathcal{X} = \mathbb{N}$, $\mathcal{F} = \Delta(\mathbb{N})$, $\mathcal{Q} = H$ and $L = TV : \Delta(\mathbb{N}) \rightarrow \Delta(\mathbb{N}) \rightarrow \mathbb{R}_0^+$.

4.2 Additional Preliminaries: Methods for Distribution Learning

4.2.1 Yatracos learner

We recall a classic theorem for the problem of hypothesis selection. Given a set of candidate hypothesis distributions, the algorithm selects one which is close to an unknown distribution (to which we have sample access). The requisite number of samples from the unknown distribution is logarithmic in the size of the set of candidates.

Theorem 4.2.1 (Yatracos’ 3-robust learner for finite classes (Theorem 4.4 of Ashtiani et al. [2018], based on Theorem 1 of Yatracos [1985])). *Let $\mathcal{C}$ be a finite class of distributions over a domain $\mathcal{X}$. There exists an algorithm $\mathcal{A}$ such that for any $(\varepsilon, \delta) \in (0, 1)^2$ and for any distribution p over $\mathcal{X}$, given a sample S of size*

$$m = O\left(\frac{\log |\mathcal{C}| + \log(1/\delta)}{\varepsilon^2}\right)$$

drawn i.i.d. from p , we have that with probability $\geq 1 - \delta$,

$$d_{TV}(p, \mathcal{A}(S)) \leq 3 \cdot \min\{d_{TV}(p, p') : p' \in \mathcal{C}\} + \varepsilon.$$

The learner is based on the so-called *Yatracos sets*.⁵ For every $q_i, q_j \in \mathcal{C}$, define the Yatracos set between q_i and q_j to be $A_{i,j} = \{x : q_i(x) \geq q_j(x)\}$.⁶ Furthermore, let

$$\mathcal{Y}(\mathcal{C}) = \{A_{i,j} \subset \mathcal{X} : q_i, q_j \in \mathcal{C}\}$$

⁵These sets are also sometimes called Scheffé sets in the literature.

⁶Note that this definition is asymmetric: $A_{i,j} \neq A_{j,i}$.

denote the set of all pairwise Yatracos sets between distributions in the set $\mathcal{C}$.

The Yatracos learner now considers the A -distance [Kifer et al. \[2004\]](#) between two distributions with respect to the Yatracos sets, i.e., it considers

$$d_{\mathcal{Y}(\mathcal{C})}(p', q') = \sup_{B \in \mathcal{Y}(\hat{\mathcal{H}})} |p'(B) - q'(B)|.$$

This distance looks at the supremum of the discrepancy between the distributions across the Yatracos sets. Consequently, for any two distributions p', q' we have $d_{\text{TV}}(p', q') \geq d_{\mathcal{Y}(\mathcal{C})}(p', q')$, since total variation distance is the supremum of the discrepancy across *all* possible sets. Furthermore, if $q', p' \in \hat{\mathcal{H}}$, then $d_{\text{TV}}(p', q') = d_{\mathcal{Y}(\mathcal{C})}(p', q')$, since either of the Yatracos sets between the two distributions serves as a set that realizes the total variation distance between them. The finiteness of the class of Yatracos sets can then be used to show that only finitely many such comparisons need to find a close-to-optimal candidate. Furthermore, if the set of Yatracos sets for an infinite set of distributions has finite VC-dimension, we can use the uniform convergence with respect to the Yatracos sets to show learnability with the same learning strategy.

4.2.2 Sample Compression

Another method for learning distribution proposed in [Ashtiani et al. \[2018\]](#) is through sample compression schemes. We will let $\mathcal{Q}$ be a class of distribution over some domain $\mathcal{X}$. A compression scheme for $\mathcal{Q}$ involves two parties: an encoder and a decoder.

- The encoder has some distribution $q \in \mathcal{Q}$ and receives n samples from q . They send a succinct message (dependent on q) to the decoder, which will allow the decoder to output a distribution close to q . This message consists of a subset of size τ of the m samples, as well as t additional bits.
- The decoder receives the τ samples and t bits and outputs a distribution which is close to q .

Since this process inherently involves randomness (of the samples drawn from q), we require that this interaction succeeds at outputting a distribution close to q with only constant probability.

More formally, we have the following definitions for a decoder and a (robust) compression scheme.

Definition 4.2.2 (decoder, Definition 4.1 of [Ashtiani et al. \[2018\]](#)). A decoder for $\mathcal{Q}$ is a deterministic function $\mathcal{J} : \bigcup_{m=0}^{\infty} \mathcal{X}^m \times \bigcup_{m=0}^{\infty} \{0, 1\}^m \rightarrow \mathcal{Q}$, which takes a finite sequence of elements of $\mathcal{X}$ and a finite sequence of bits, and outputs a member of $\mathcal{Q}$.

The formal definition of a compression scheme follows.

Definition 4.2.3 (robust compression schemes, Definition 4.2 of [Ashtiani et al. \[2018\]](#)). Let $\tau, t, m : (0, 1) \rightarrow \mathbb{Z}_{\geq 0}$ be functions, and let $r \geq 0$. We say $\mathcal{Q}$ admits (τ, t, m) r -robust compression if there exists a decoder $\mathcal{J}$ for $\mathcal{Q}$ such that for any distribution $q \in \mathcal{Q}$ and any distribution p on $\mathcal{X}$ with $d_{\text{TV}}(p, q) \leq r$, the following holds:

For any $\varepsilon \in (0, 1)$, if a sample S is drawn from $p^{m(\varepsilon)}$, then, with probability at least $2/3$, there exists a sequence L of at most $\tau(\varepsilon)$ elements of S , and a sequence B of at most $t(\varepsilon)$ bits, such that $d_{\text{TV}}(\mathcal{J}(L, B), \mathcal{Q}) \leq r + \varepsilon$.

Note that S and L are sequences rather than sets, and can potentially contain repetitions.

Theorem 4.2.4 (Compression implies learning, Theorem 4.5 of [Ashtiani et al. \[2018\]](#)). Suppose $\mathcal{Q}$ admits (τ, t, m) r -robust compression. Let $\tau'(\varepsilon) := \tau(\varepsilon) + t(\varepsilon)$. Then $\mathcal{Q}$ can be $\max\{3, 2/r\}$ -learned in the agnostic setting using

$$O\left(m\left(\frac{\varepsilon}{6}\right) \log\left(\frac{1}{\delta}\right) + \frac{\tau'(\varepsilon/6) \log(m(\varepsilon/6) \log_3(1/\delta)) + \log(1/\delta)}{\varepsilon^2}\right) = \tilde{O}\left(m\left(\frac{\varepsilon}{6}\right) + \frac{\tau'(\varepsilon/6) \log m(\varepsilon/6)}{\varepsilon^2}\right)$$

samples. If $\mathcal{Q}$ admits (τ, t, m) non-robust compression, then $\mathcal{Q}$ can be learned in the realizable setting using the same number of samples.

4.2.3 Useful Inequalities

We also use the following form of the Chernoff bound.

Proposition 4.2.5 (Chernoff bounds). Let $X = \sum_{i=1}^m X_i$ where $X_1 \dots X_m$ are independent draws from $\text{Bernoulli}(p)$. For any $t \in (0, 1)$, $\mathbb{P}[X/m \leq (1 - t)p] \leq \exp(-t^2 mp/2)$.

4.3 Related Work

The survey paper by [Diakonikolas, 2016] asks (Open Problem 1.5.1): *“Is there a ‘complexity measure’ of a distribution class C that characterizes the sample complexity of learning C ?”*. Our work addresses this question.

Our work fits more into the Computer Science literature on distribution estimation, initiated by the work of [Kearns et al., 1994], which was in turn inspired by Valiant’s PAC learning model [Valiant, 1984]. Since then, several works have focused on algorithms for learning specific classes of distributions, see, e.g., [Chan et al., 2013, 2014a,b, Li and Schmidt, 2017, Ashtiani et al., 2018]. A recent line of work, initiated by [Diakonikolas et al., 2016, Lai et al., 2016], focuses on computationally-efficient robust estimation of multivariate distributions, see, e.g., [Diakonikolas et al., 2017a, Steinhardt et al., 2018, Diakonikolas et al., 2018, Kothari et al., 2018, Hopkins and Li, 2018, Diakonikolas et al., 2019, Liu and Moitra, 2021, 2022, Bakshi et al., 2022, Jia et al., 2023] and [Diakonikolas and Kane, 2022] for a reference. In contrast to all of these works, we focus on broad and generic connections between learnability and robust learnability, rather than studying robust learnability of a particular class of distributions.

Some of our algorithmic results employ tools from hypothesis selection, a problem which focuses on agnostic learning with respect to a specified finite set of distributions. The most popular approaches are based on ideas introduced by Yatracos [Yatracos, 1985] and subsequently refined by Devroye and Lugosi [Devroye and Lugosi, 1996, 1997, 2001]. Several others have studied hypothesis selection with an eye for several considerations, including running time, approximation factor, robustness, privacy, parallelization, and more [Mahalanabis and Stefankovic, 2008, Daskalakis et al., 2012, Daskalakis and Kamath, 2014, Suresh et al., 2014, Diakonikolas et al., 2016, Ashtiani et al., 2018, Acharya et al., 2018, Bun et al., 2019, Bousquet et al., 2019, Gopi et al., 2020, Bousquet et al., 2022, Aden-Ali et al., 2021]. Robust estimation is a classic subfield of Statistics, see, for example, the classic works [Tukey, 1960, Huber, 1964]. We will later discuss the connection between robust and agnostic learning.

Another active line of research in distribution learning is distribution learning with constraints on differential privacy [Dwork et al., 2006], e.g., [Karwa and Vadhan, 2018, Kamath et al., 2019, Bun and Steinke, 2019, Acharya et al., 2021, Cai et al., 2021, Kamath et al., 2022b,a], and [Kamath and Ullman, 2020] for a survey. A number of these works have focused on connections between robustness and privacy [Dwork and Lei, 2009, Bun et al., 2019, Kamath et al., 2020, Avella-Medina, 2020, Brown et al., 2021, Liu et al., 2021, 2022, Kothari et al., 2022, Ramsay et al., 2022, Slavkovic and Molinari, 2022, Hopkins

et al., 2022b, Georgiev and Hopkins, 2022, Hopkins et al., 2023, Alabi et al., 2023]. Again, these results either focus on specific classes of distributions, or give implications that require additional technical conditions, whereas we aim to give characterizations of robust learnability under minimal assumptions.

Chapter 5

Impossibility of Characterizing Distribution Learning

In this chapter, we will address the question of whether the learning task of distribution learning can be characterized by a scale-invariant dimension in the sense of Definition 3.1.3 and Definition 3.1.2. We show that there can not be either of these characterizations for distribution learning.

Our results answer some long-standing open questions; The survey paper by [Diakonikolas, 2016] asks (Open Problem 1.5.1): *”Is there a “complexity measure” of a distribution class C that characterizes the sample complexity of learning C ?”*

Theorem 3.1.10. *There is no weak sample complexity dimension for distribution learning (neither in the realizable nor in the 3-agnostic case of distribution learning).*

Corollary 3.1.11. *There exist no monotonic real-valued function that is a weak sample complexity dimension for distribution learning.*

Corollary 3.1.11 follows directly from Theorem 3.1.10 and Theorem 3.1.9.

Theorem 3.1.16. *The learnability of classes of discrete distributions cannot be characterized by a finitary characterization. This statement holds both for realizable PAC learnability and for 3-agnostic PAC learnability.*

We note, that while we only consider constructions of discrete distributions in this section, the corresponding results on uncharacterizability for *general distribution learning* follow directly from these results.

We will show these theorems using Lemma 3.1.7 and Theorem 3.1.15 respectively. In order to do so, we need to construct classes of distributions that meet the requirements of these more general results. We will first describe a construction that can be used for both theorems. We then show some properties of this construction which will be needed for both theorems, namely an upper bound (Lemma 5.1.2) and a lower bound (Lemma 5.1.3) on its sample complexity. We will then state the proofs of Theorem 3.1.10 and Theorem 3.1.16. Lastly, we will end the section with discussing an extension of the qualitative impossibility result to classes of distributions with polynomial sample complexity.

5.1 Preliminaries and Construction

Throughout this section we will also need the fact that finite classes are learnable. We state Theorem 3.4 from Ashtiani et al. [2018], which is a slight rephrasing of Theorem 6.3 from Devroye and Lugosi [2001].

Theorem 5.1.1. *Ashtiani et al. [2018], Devroye and Lugosi [2001] For any finite class of distributions $\mathcal{Q} = \{q_1, \dots, q_m\}$, there exists a deterministic algorithm which 3-agnostic PAC learns Q with sample complexity $m_{\mathcal{Q}}(4\varepsilon, \delta) \leq \frac{\log(3m^2) + \log(1/\delta)}{2\varepsilon^2}$.*

We will now describe our construction. For a natural number $n \in \mathbb{N}$ and a (usually small) mixture parameter $\gamma \in (0, 1)$, we define the finite class,

$$P_{\gamma, n} = \{(1 - \gamma)\delta_0 + \gamma U_A : A \subset \{1, \dots, n\}\},$$

where δ_0 denotes the distribution with all its mass on point 0 and U_A denotes a uniform distribution over the set A . Intuitively, this class thus consists of a heavy non-flexible part $((1 - \gamma)\delta_0)$ and a highly flexible part with low weight (γU_A) . For any distribution p the TV-distance to an element of $P_{\gamma, n}$ only depends in a small part on the low-weight component. However, the low-weight flexible part, will make this class hard to learn for small ε . We now take union over these classes $P_{\gamma, n}$ for different combinations of γ and n , which allows us to control the behaviour of the sample complexity and fulfill the requirements for both. For sequences $\bar{\gamma} : \mathbb{N} \rightarrow [0, 1]$ and $\bar{n} : \mathbb{N} \rightarrow \mathbb{N}$, we define

$$\mathcal{Q}_{\bar{\gamma}, \bar{n}} = \bigcup_{i=1}^{\infty} P_{\bar{\gamma}(i), \bar{n}(i)}. \quad (5.1)$$

We furthermore define $\bar{\gamma}^{-1}(\varepsilon) = \arg \min \{i \in \mathbb{N} : \text{for every } j \geq i, \bar{\gamma}(j) \leq \varepsilon\}$ and $n_{\max}(i) = \max_{j \in \{1, \dots, i\}} \bar{n}(j)$. We will now show that infinite classes of this kind can be learnable, even

as n grows to infinity (and thus making the class in some sense "infinitely flexible"), by controlling the mixture parameter γ .

Lemma 5.1.2. *Let $\mathcal{Q} = \mathcal{Q}_{\bar{\gamma}, \bar{n}}$ with $\lim_{i \rightarrow \infty} \bar{\gamma}(i) = 0$. Then $\mathcal{Q}$ is 3-agnostic PAC learnable with sample complexity $m_{\mathcal{Q}}(\varepsilon, \delta) \leq (128(\log(3(\bar{\gamma}^{-1}(\frac{\varepsilon}{4})n_{\max})\bar{\gamma}^{-1}(\frac{\varepsilon}{4}) + 1)^n) + \log(\frac{1}{\delta}))/(\varepsilon^2)$.*

Proof. Assume $\lim_{i \rightarrow \infty} \bar{\gamma}(i) = 0$. Let $\varepsilon > 0$. Then for every $\frac{\varepsilon}{4} > 0$ there is an N , such that for every $N' \geq N$, $\bar{\gamma}(N') < \frac{\varepsilon}{4}$. We can now focus on 3-agnostic learning the finite class $\mathcal{Q}' = \{\delta_0\} \cup (\bigcup_{i=0}^N P_{\bar{\gamma}(i), \bar{n}(i)})$, as learning as for any $q \in \mathcal{Q}_{\bar{\gamma}, \bar{n}}$, there is $p \in \mathcal{Q}'$ with $TV(p, q) < \frac{\varepsilon}{4}$. Thus by triangle inequality for any $q' \in \Delta(\mathcal{Z})$, we get $\inf_{p \in \mathcal{Q}'} TV(p, q') \leq \inf_{q \in \mathcal{Q}} TV(q, q') + \frac{\varepsilon}{4}$. Thus if we have a 3-agnostic PAC learner $\mathcal{A}$ for $\mathcal{Q}'$ with sample complexity $m_{\mathcal{Q}'}(\frac{\varepsilon}{4}, \delta)$, we can use it as a PAC learner for $\mathcal{Q}$ with sample complexity $m_{\mathcal{Q}}(\varepsilon, \delta)$. Now using Theorem 5.1.1, we can conclude that $\mathcal{Q}$ is learnable with sample complexity $m_{\mathcal{Q}}(\varepsilon, \delta) \leq 128 \frac{\log(3(\bar{\gamma}^{-1}(\frac{\varepsilon}{4})n_{\max}(\bar{\gamma}^{-1}(\frac{\varepsilon}{4})))^2 + \log(1/\delta)}{\varepsilon^2}$. □

However, we can also lower bound the sample complexity of these kinds of classes in the following way.

Lemma 5.1.3. *For $\mathcal{Q} = P_{\gamma, 4n}$, we have $m_{\mathcal{Q}}(\frac{\gamma}{8}, \frac{1}{7}) \geq m_{\mathcal{Q}}^{re}(\frac{\gamma}{8}, \frac{1}{7}) \geq n$.*

The construction and argument follow from a common no-free lunch style argument. For details we refer the reader to Section 5.5 in the Appendices (in the supplementary file).

5.2 Impossibility of Characterizing the Sample Complexity

We can now use this construction and bounds to prove Theorem 3.1.10.

Proof of Theorem 3.1.10. Based on the confinality considerations described above, it suffices to show that the set $\{m_{\mathcal{Q}}(1/\cdot, 1/7) \in \mathbb{N}^{\mathbb{N}} : \mathcal{Q} \in \mathcal{C}\}$ is cofinal in $\mathbb{N}^{\mathbb{N}}$. Let $g \in \mathbb{N}^{\mathbb{N}}$ be arbitrary. Now consider the class $\mathcal{Q} = \mathcal{Q}_{\bar{\gamma}, \bar{n}}$ as constructed in the previous section, where $\bar{\gamma}(k) = \frac{8}{k}$ and $\bar{n}(k) = 8(g(k) + 1)$. Then according to Lemma 5.1.2, $\mathcal{Q}$ is learnable, as $\lim_{k \rightarrow \infty} \bar{\gamma}(k) = 0$. Furthermore, we know that for every $k \in \mathbb{N}$ we have $P_{\frac{8}{k}, \bar{n}(k)} \subset \mathcal{Q}$. Thus, by Lemma 5.1.3, for every $k \in \mathbb{N}$: $g(k) < \frac{\bar{n}(k)}{4} \leq m_{\mathcal{Q}}(\frac{1}{k}, \frac{1}{7})$. Thus, $g \leq_{ed} m_{\mathcal{Q}}(\cdot, \frac{1}{7})$.

Therefore $\{m_{\mathcal{Q}}(1/\cdot, 1/7) : \mathcal{Q} \in m_{\mathcal{C}}\}$ is indeed cofinal in $\mathbb{N}^{\mathbb{N}}$. As the bounds of Lemma 5.1.2 and Lemma 5.1.3 both hold for the realizable distribution learning, we can prove that $\{m_{\mathcal{Q}}^{re}(1/\cdot, 1/7) : \mathcal{Q} \in m_{\mathcal{C}}\}$ is cofinal in $\mathbb{N}^{\mathbb{N}}$ by the same construction and argument. Thus, we have proved our claim. □

5.3 Distributions Learning cannot be characterized by a finitary characterization

We can now focus our attention the impossibility of qualitative characterization of distribution learning and finally prove Theorem 3.1.16.

Proof of Theorem 3.1.16. We only need to show that the two conditions from Theorem 3.1.15 are fulfilled by the problem of distribution learning. Condition 1 holds, as according to Theorem 5.1.1 every finite class of distributions is learnable. This means we can define a learner for any finite union of learnable sets, by running the learners for each of the learnable sets on the input to create a finite set of candidates and then learn the candidate set. Condition 2 holds by the following construction: $H_0 = \{\delta_0\}$ and $H_k = \mathcal{Q}_{\bar{\gamma}_k, \bar{n}}$, as defined in equation (1), where $\bar{\gamma}_k(i) = 1/k$ and $\bar{n}(i) = i$. It is clear that, H_0 is an ε_k -approximation of H_k for $\varepsilon_k = 1/k$ as all elements of H_k have $\frac{k-1}{k}$ mass on the point 0. Furthermore we have $\lim_{k \rightarrow \infty} \varepsilon_k = 0$. Lastly, we have to argue that every H_k is not learnable. We now note that for every $n \in \mathbb{N}$ the class $P_{\varepsilon_k, 4n} \subset H_k$ (as defined in Section 3). Thus we can apply Lemma 5.1.3 to obtain, for every $n \in \mathbb{N}$, $m_{H_k}(\frac{\varepsilon_k}{8}, \frac{1}{7}) \geq n$. We note that any instance of the word "learnable" in this proof can either mean "realizable PAC learnable" or "3-agnostic PAC learnable". The proof is correct in both cases. □

5.4 Polynomial complexity distribution learning

Another, more restricted definition of learning, is one that requires specific bounds on the sample complexity.

Definition 5.4.1. We say a class H is polynomially PAC learnable, if $m_H(\varepsilon, \delta) \in \text{poly}(1/\varepsilon, 1/\delta)$.

We note, that for many learning tasks, like binary classification, classes are polynomially PAC learnable, if and only if they are PAC learnable. However, we have seen in this

section that for the task of distribution learning, there are PAC learnable classes which are not polynomially PAC learnable. Arguably, in many scenarios one is more interested in polynomially learnable classes. We therefore pose the question, whether it is possible to give a *qualitative* characterization of *polynomially learnable* classes. In the case of distribution learning, we can give a partial answer, showing that there is no *uniformly-bounded* finitary characterization of polynomial distribution learning.

Theorem 5.4.2. *There is no uniformly-bounded finitary characterization of polynomial distribution learning (with respect to TV-distance). This result holds for both the realizable and the 3-agnostic case of distribution learning.*

5.5 Proofs: Uncharacterizability of Distribution Learning

Lemma 5.1.3. *For $\mathcal{Q} = P_{\gamma, 4n}$, we have $m_{\mathcal{Q}}(\frac{\gamma}{8}, \frac{1}{7}) \geq m_{\mathcal{Q}}^{re}(\frac{\gamma}{8}, \frac{1}{7}) \geq n$.*

Proof of Lemma 5.1.3. Our proof follows a typical no-free-lunch-style argument. Consider $\mathcal{Q}' = \{(1 - \gamma)\delta_0 + \gamma U_A : A \subset \{1, \dots, 4n\} \text{ and } |A| = n\}$. We will show a lower bound of learning this class of distributions and then conclude that this lower bound also holds for $\mathcal{Q}$, as $\mathcal{Q}' \subset \mathcal{Q}$. Now let $\mathcal{A}$ be any learner. Furthermore, let $S_1, \dots, S_k$ be the set of all sequences of size n with elements in the set $\{0, \dots, 4n\}$. We have

$$\mathbb{E}_{S \sim q_i} [TV(q_i, \mathcal{A}(S))] = \sum_{j=1}^k q_i^n(S_j) TV(q_i, \mathcal{A}(S_j))$$

Now for every S_j and every $q_{i_1}, q_{i_2} \in \mathcal{Q}'$, we have that if $S_j \in \text{supp}(q_{i_1}^n)$, then

$$q_{i_2}^n(S_j) = \begin{cases} q_{i_1}^n(S_j) & \text{if } S_j \subset \text{supp}(q_{i_2}^n) \\ 0 & \text{otherwise} \end{cases}$$

Let us denote $C_j = \{x \in \{1, \dots, 4n\} : x \in S_j\}$. Furthermore for a set $A = \{x_1, \dots, x_p\}$ with $C_j \subset A \subset \{1, \dots, 4n\}$ and $|A| \leq 2n$, let $\bar{A} = \{x'_1, \dots, x'_{p'}\}$, be such that $x_{l_1} < x_{l_2}$ and $x'_{l_1} < x'_{l_2}$ for $l_1 < l_2$. Now let us define $g_j(A) = C_j \cup \{x'_l \in \bar{A} : x_{p-l} \in A \setminus C_j\}$. We note that $|g_j(A)| = |A|$, $A \cap g_j(A) = C_j$ and $g_j(g_j(A)) = A$.

Now for any $q_i \in \mathcal{Q}'$, let us denote $A_i = \text{supp}(q_i) \setminus \{0\}$. Now let us define

$$f_j(q_i) = \begin{cases} (1 - \gamma)\delta_0 + \gamma U_{g_j(A_i)} & \text{if } C_j \in A_i \\ \delta_{4n+1} & \text{otherwise.} \end{cases}$$

We note that if $q_i^n(S_j) > 0$, then $C_j \in A_i$ and $f_j(f_j(q_i)) = q_i$ and $f_j(q_i) \in \mathcal{Q}'$. In this case we furthermore have $TV(q_i, f_j(q_i)) \geq \frac{\gamma}{2}$. Furthermore if $q_i^n(S_j) = 0$, then $f_j(q_i)^n(S_j) = 0$. Taking both of these cases together we have for all i and for all j : $q_i^n(S_j) = f_j(q_i)^n(S_j)$.

Now we can put everything together into a no-free-lunch style argument.

$$\begin{aligned} \max_{q_i \in \mathcal{Q}'} \mathbb{E}_{S \sim q_i} [TV(q_i, \mathcal{A}(S))] &= \max_{q_i \in \mathcal{Q}'} \sum_{j=1}^k q_i^n(S_j) TV(q_i, \mathcal{A}(S_j)) \\ &\geq \frac{1}{T} \sum_{i=1}^T \sum_{j=1}^k q_i^n(S_j) TV(q_i, \mathcal{A}(S_j)) \\ &= \frac{1}{2T} \sum_{i=1}^T \sum_{j=1}^k q_i^n(S_j) TV(q_i, \mathcal{A}(S_j)) + \frac{1}{2T} \sum_{i=1}^T \sum_{j=1}^k f_j(q_i)^n(S_j) TV(f_j(q_i), \mathcal{A}(S_j)) \\ &= \frac{1}{2T} \sum_{i=1}^T \sum_{j=1}^k q_i^n(S_j) TV(q_i, \mathcal{A}(S_j)) + f_j(q_i)^n(S_j) TV(f_j(q_i), \mathcal{A}(S_j)) \\ &= \frac{1}{2T} \sum_{i=1}^T \sum_{j=1}^k q_i^n(S_j) (TV(q_i, \mathcal{A}(S_j)) + TV(f_j(q_i), \mathcal{A}(S_j))) \\ &\geq \frac{1}{2T} \sum_{i=1}^T \sum_{j=1}^k q_i^n(S_j) TV(q_i, f_j(q_i)) \\ &\geq \frac{1}{2T} \sum_{i=1}^T \sum_{j=1}^k q_i^n(S_j) \frac{\gamma}{2} = \frac{\gamma}{4}. \end{aligned}$$

Now, by Lemma B.1 of [Shalev-Shwartz and Ben-David \[2014a\]](#), we get

$$\begin{aligned} \max_{q_i \in \mathcal{Q}'} \mathbb{P}_{S \sim q_i^n} [TV(q_i, \mathcal{A}(S)) \geq \frac{\gamma}{8}] &= \max_{q_i \in \mathcal{Q}'} \mathbb{P}_{S \sim q_i^n} [TV(q_i, \mathcal{A}(S)) \geq 1 - \frac{7\gamma}{8}] \\ &\geq \max_{q_i \in \mathcal{Q}'} \frac{\mathbb{E}_{S \sim q_i^n} [TV(q_i, \mathcal{A}(S))] - \frac{1}{8}}{\frac{7}{8}} \geq \frac{1}{7}. \end{aligned}$$

Thus, $m_{\mathcal{Q}'}(\frac{\gamma}{8}, \frac{1}{7}) \geq n$. Therefore $m_{\mathcal{Q}}^{re}(\frac{\gamma}{8}, \frac{1}{7}) \geq n$. $\square$

Theorem 5.4.2. *There is no uniformly-bounded finitary characterization of polynomial distribution learning (with respect to TV-distance). This result holds for both the realizable and the 3-agnostic case of distribution learning.*

Proof of Theorem 5.4.2. Assume, b.w.o.c. that W is a uniformly bounded finitary characterization of polynomial distribution learning. From W being uniformly bounded, we know that for every α there is n , such that for every class H , there is a subset H_α with $|H_\alpha| \leq n$ and such that for every H' if $H' \subset H_\alpha$, then H' satisfies α . Let $W = \{\alpha_k : k \in \mathbb{N}\}$ an enumeration of W , that is ordered by the size of the corresponding n_k , i.e. such that for every $k \leq k'$, we have $n_k \leq n_{k'}$. We define $f(k) = k \cdot n_k$. Consider $H_0 = \{\delta_0\}$ and $H_k = \mathcal{Q}_{\bar{\gamma}_k, \bar{n}}$ with $\bar{n}(i) = i$ and $\bar{\gamma}_k(i) = \max\{1/f(i), 1/f(k)\}$. Now H_k is not learnable as for every $m \in \mathbb{N}$, $P_{1/f(k), 4m} \subset H_k$, meaning that by Lemma 5.1.3 for every m , $m_{H_k}(1/(8f(k)), 1/7) \geq m$. From the uniformly bounded finitary characterization, we know that there is $\hat{H}_k$, with $|\hat{H}_k| = n_k$ and every H' if $H' \subset \hat{H}_k$ then H' satisfies α_k . Let $\hat{H} = H_0 \cup (\bigcup_{k \in \mathbb{N}} \hat{H}_k)$. By construction, we have that $\hat{H}$ satisfies W . Furthermore, when aiming for (ε, δ) -success, it is sufficient to restrict our attention to learning the $\varepsilon/4$ approximation $H_{\varepsilon/4} = H_0 \cup (\bigcup_{k=1}^{\bar{\gamma}^{-1}(\varepsilon/4)} \hat{H}_k)$ of $\hat{H}$, where $\bar{\gamma}^{-1}(\varepsilon) = \min\{i : \text{for all } j > i, \bar{\gamma}(j) < \varepsilon\} = \min\{k : kn_k \geq \frac{1}{\varepsilon}\}$. Thus, $|H_{\varepsilon/4}| \in \text{poly}(\frac{1}{\varepsilon})$. From Theorem 5.1.1 we thus get that $H_{\varepsilon/4}$ is polynomially learnable, which implies that $\hat{H}$ is polynomially PAC learnable w.r.t. to TV distance. Learning here can either mean 3-agnostic or realizable learnability. The result holds for both cases. $\square$

Chapter 6

Realizable vs Agnostic Learning

6.1 Overview of Results

In this chapter we explore the relation between realizable and agnostic distribution learning. Our main result shows that a distribution class being learnable does *not* imply that it is robustly or agnostically learnable.

Theorem 6.1.1. *[Informal] There exists a class that is realizably learnable, but not agnostically/robustly learnable.*

Our proof of this theorem proceeds by constructing a class of distributions that is learnable, but classes obtained by subtracting light-weight parts of these distributions are not α -robustly learnable with respect to the original learnable class. More concretely, our construction works as follows. We start by considering a distribution class that, by itself, is not learnable with any finite number of samples. We map each distribution in that class to a new distribution, which additionally features a point with non-trivial mass that “encodes” the identity of the distribution, thus creating a new class of distributions which *is* learnable. Subtractive contamination is then able to “erase” this point mass, leaving a learner with sample access only to the original (unlearnable) class. Our construction is inspired by the construction of the previous chapter, showing that the learnability of classes of probability distributions cannot be characterized by any notion of combinatorial dimension.

Adjacent to distribution learning is the notion of *sample compression schemes*. Recent work by Ashtiani, Ben-David, Harvey, Liaw, Mehrabian, and Plan [Ashtiani et al. \[2018\]](#)

expanded notions of sample compression schemes to apply to the task of learning probability distributions. They showed that the existence of such sample compression schemes for a class of distributions imply the learnability of that class. While the existence of sample compression schemes for classification tasks imply the existence of such schemes for robust learning, the question if similar implication hold for distribution learning schemes was not answered. We strongly refute this statement. We use a construction similar to that of Theorem 6.2.1, see Section 6.5 for more details.

Theorem 6.1.2 (Informal). *The existence of compression schemes for a class of probability distributions does not imply the existence of robust compression schemes for that class.*

6.2 Realizable Learnability does not Imply Agnostic Learnability

In this chapter we show that there are classes of distributions which are realizable learnable, but not agnostically learnable.

Theorem 6.2.1. *There are classes of distributions $\mathcal{Q}$, such that $\mathcal{Q}$ is realizable learnable, but for every $\alpha \in \mathbb{R}$, $\mathcal{Q}$ is not robustly or agnostically learnable. Moreover, the sample complexity of learning $\mathcal{Q}$ can be arbitrarily close to (but larger than) linear. Namely, for any super-linear function g , there is a class $\mathcal{Q}_g$, with*

- $\mathcal{Q}_g$ is realizable learnable with sample complexity $m_{\mathcal{Q}_g}^{re}(\varepsilon, \delta) \leq \log(1/\delta)g(1/\varepsilon)$;
- for every $\alpha \in \mathbb{R}$, $\mathcal{Q}_g$ is not α -robustly learnable.

The key idea to the proof for all of these theorems is to construct a class which is easy to learn in the realizable case, by having each distribution of the class have a unique support element that is not shared by any other distributions in the class. Distributions on which this “indicator element” has sufficient mass will be easily identified, independent of how rich the class is on other domain elements. That richness makes the class hard to learn from samples that miss those indicators. Furthermore, we construct the class in a way that its members are close in total variation distance to distributions that place no weight on those indicator elements.

This is done by making the mass on these indicator elements small, so that the members of a class of distributions that results from deleting these indicator bits are close to the

initially constructed class, $\mathcal{Q}_g$. In order to make this work for every target accuracy and sample complexity, we need to have a union of such classes with decreasingly small mass on the indicator bits. In order for this to not interfere with the realizable learnability, we let the distributions with small mass on the indicator bits have most of their mass on one point $(0, 0)$ that is the same for all distributions in the class. This ensures that distributions for which the indicator bit will likely not be observed because their mass is smaller than some η are still easily ε -approximated by a constant distribution $(\delta_{(0,0)})$. Lastly we ensure the impossibility of agnostic learnability, by controlling the rate at which η approaches zero to be faster than the rate at which ε approaches zero. With this intuition in mind, we will now describe the construction and proof of this theorem.

6.3 Proofs sketch of Theorem 6.2.1

We first define the distributions in $\mathcal{Q}_g$. Let $\{A_i \subset \mathbb{N} : i \in \mathbb{N}\}$ be an enumeration of all finite subsets of $\mathbb{N}$. Define distributions over $\mathbb{N} \times \mathbb{N}$ as follows:

$$q_{i,j,k} = \left(1 - \frac{1}{j}\right) \delta_{(0,0)} + \left(\frac{1}{j} - \frac{1}{k}\right) U_{A_i \times \{2j+1\}} + \frac{1}{k} \delta_{(i,2j+2)}, \quad (6.1)$$

where, for every finite set W , U_W denotes the uniform distribution over W . For a monotone, super-linear function $g : \mathbb{N} \rightarrow \mathbb{N}$, we now let $\mathcal{Q}_g = \{q_{i,j,g(j)} : i, j \in \mathbb{N}\}$. The first bullet point of the theorem (the class is learnable) follows from Claim 6.3.1 and the second bullet point (the class is not agnostic learnable) follows from Claim 6.3.2 below.

Claim 6.3.1. For a monotone function $g : \mathbb{N} \rightarrow \mathbb{N}$, let $\mathcal{Q}_g = \{q_{i,j,g(j)} : i, j \in \mathbb{N}\}$. Then, the sample complexity of $\mathcal{Q}_g$ in the realizable case is upper bounded by

$$m_{\mathcal{Q}_g}^{re}(\varepsilon, \delta) \leq \log(1/\delta)g(1/\varepsilon).$$

This claim can be proved by showing that the following learner defined by

$$\mathcal{A}(S) = \begin{cases} q_{i,j,g(j)} & \text{if } (i, 2j+2) \in S \\ \delta_{(0,0)} & \text{otherwise} \end{cases}$$

is a successful learner in the realizable case. Intuitively, this learner is successful for distributions $q_{i,j,g(j)}$ for which j is large (i.e., $j > \frac{1}{\varepsilon}$), since this will mean that $d_{TV}(q_{i,j,g(j)}, \delta_{(0,0)})$ is small. Furthermore, it is successful for distributions $q_{i,j,g(j)}$ for which j is small (i.e., upper

bounded by some constant dependent on ε), because this will lower bound the probability $1/g(j)$ of observing the indicator bit on $(i, 2j + 2)$. Once the indicator bit is observed the distribution will be uniquely identified.

Claim 6.3.2. For every function $g \in \omega(n)$ the class $\mathcal{Q}_g$ is not α -agnostically learnable for any $\alpha > 0$.

This claim can be proven by showing that for every α , there is η , such that the class of distributions Q' for which the following holds:

- Q' is not $\alpha\eta$ -weakly learnable¹
- for every $q' \in Q'$ there is $q \in \mathcal{Q}_g$ and arbitrary r with $q' = (1 - \eta)q + \eta r$ which is not $\alpha\eta$ -weakly learnable.

We construct this class and show that it is not learnable by using the construction and Lemmas from Chapter 5.

6.4 Proof of remaining Claims for Theorem 6.2.1

We will now show the proofs of the main result of this section. We start with proving the upper bound of Claim 6.3.1 and the lower bound of Claim 6.3.2 before proving Theorem 6.2.1.

6.4.1 Proof of Claim 6.3.1

Proof. Let the realizable learner $\mathcal{A}$ be

$$\mathcal{A}(S) = \begin{cases} q_{i,j,g(j)} & \text{if } (i, 2j + 2) \in S \\ \delta_{(0,0)} & \text{otherwise} \end{cases}$$

Note that for all $\mathcal{Q}_g$ -realizable samples this learner is well-defined. Furthermore, we note that in the realizable case, whenever $\mathcal{A}$ outputs a distribution different from $\delta_{(0,0)}$,

¹We provide a definition for ε -weak learnability as Definition 6.4.1 We note that the definition we provide is what would usually be referred to as $(1/2 - \varepsilon)$ -weak learnability in the supervised learning literature. For simplicity, because ε is our parameter of interest, we reparameterized the definition to be more intuitive.

then $\mathcal{A}(S)$ outputs the ground-truth distribution, i.e., the output has TV-distance 0 to the true distribution. Lastly, we note, that for an i.i.d. sample $S \sim q_{i,j,g(j)}^m$, we have the following upper bound for the learner identifying the correct distribution:

$$\mathbb{P}_{S \sim q_{i,j,g(j)}^m}[\mathcal{A}(S) = q_{(j)}] = \mathbb{P}_{S \sim q_{i,j,g(j)}^m}[(i, 2j+2) \in S] = 1 - (1 - 1/g(j))^m.$$

We note, that since g is a monotone function, if $\varepsilon \leq \frac{1}{j}$, then $g(j) \leq g(\frac{1}{\varepsilon})$ and therefore,

$$(1 - 1/g(j))^m \leq (1 - 1/g(1/\varepsilon))^m.$$

Furthermore for $q_{i,j,g(j)}$, we have that $d_{\text{TV}}(\delta_{(0,0)}, q_{i,j,g(j)}) = \frac{1}{j}$.

Putting these two observations together, we get

$$\mathbb{P}_{S \sim q_{i,j,g(j)}^m}[d_{\text{TV}}(\mathbb{A}(S), q_{i,j,g(j)}) \geq \varepsilon] \leq \begin{cases} (1 - 1/g(1/\varepsilon))^m & \text{if } \frac{1}{j} \leq \varepsilon \\ 0 & \text{if } \frac{1}{j} > \varepsilon \end{cases}.$$

Thus, for every $q \in \mathcal{Q}_g$,

$$\mathbb{P}_{S \sim q^m}[d_{\text{TV}}(\mathbb{A}(S), q) \geq \varepsilon] \leq (1 - 1/g(1/\varepsilon))^m \leq \exp\left(-\frac{m}{g(1/\varepsilon)}\right).$$

Letting the left-hand side equal the failure probability δ and solving for m , we get,

$$\begin{aligned} \log \delta &\geq \frac{-m}{g(1/\varepsilon)} \\ \log(\delta)g(1/\varepsilon) &\geq -m \\ m &\geq -\log(\delta)g(1/\varepsilon) = \log(1/\delta)g(1/\varepsilon). \end{aligned}$$

Thus, we have a sample complexity bound of

$$m_{\mathcal{Q}_g}(\varepsilon, \delta) \leq \log(1/\delta)g(1/\varepsilon).$$

□

Proof of Claim 6.3.2

Now, we show a lower bound, that our class $\mathcal{Q}_g$ is *not* robustly learnable. Before we do that, we require a few more preliminaries. For a distribution class $\mathcal{Q}$ and a distribution p , let their total variation distance be defined by

$$d_{\text{TV}}(p, \mathcal{Q}) = \inf_{q \in \mathcal{Q}} d_{\text{TV}}(p, q).$$

We also use Lemma 5.1.3 from the previous section.

Lemma 5.1.3. *For $\mathcal{Q} = P_{\gamma, 4n}$, we have $m_{\mathcal{Q}}(\frac{\gamma}{8}, \frac{1}{7}) \geq m_{\mathcal{Q}}^{re}(\frac{\gamma}{8}, \frac{1}{7}) \geq n$.*

Finally, we recall the definition of weak learnability, which says that a distribution class is learnable only for some particular value of the accuracy parameter.

Definition 6.4.1. *A class $\mathcal{Q}$ is ε -weakly learnable, if there is a learner $\mathcal{A}$ and a sample complexity function $m : (0, 1) \rightarrow \mathbb{N}$, such that for ever $\delta \in (0, 1)$ and every $p \in \mathcal{Q}$ and every $m \geq m(\delta)$,*

$$\mathbb{P}_{S \sim p^m}[d_{\text{TV}}(\mathcal{A}(S), p) \leq \varepsilon] < \delta.$$

Learnability clearly implies ε -weak learnability for every $\varepsilon \in (0, 1)$. While in some learning models (e.g., binary classification) learnability and weak learnability are equivalent, the same is not true for distribution learning as we have seen by the constructions in the previous section.

We are now ready to prove that $\mathcal{Q}_g$ is not agnostically learnable.

Claim 6.3.2. For every function $g \in \omega(n)$ the class $\mathcal{Q}_g$ is not α -agnostically learnable for any $\alpha > 0$.

Proof. Consider

$$q'_{i,j} = \frac{1}{1 - \frac{1}{g(j)}} \left(\left(1 - \frac{1}{j}\right) \delta_{(0,0)} + \left(\frac{1}{j} - \frac{1}{g(j)}\right) U_{A_i \times \{2j+1\}} \right)$$

Note that for every $q'_{i,j}$ and every $q_{i,j,g(j)}$ there is some distribution r , such that $q_{i,j,g(j)} = (1 - \frac{1}{g(j)})q'_{i,j} + \frac{1}{g(j)}r$. Therefore, in order to show that $\mathcal{Q}_g$ is not α -agnostically learnable, it is sufficient to show that there are j and ε , such that the class $\mathcal{Q}'_j = \{q'_{i,j} : i \in \mathbb{N}\}$ is not $(\frac{\alpha}{g(j)} + \varepsilon)$ -weakly learnable. Let us denote $\gamma(j) = \frac{\frac{1}{j} - \frac{1}{g(j)}}{8(1 - \frac{1}{g(j)})}$. We will now show that the class $\mathcal{Q}'_j = \{q'_{i,j} : i \in \mathbb{N}\}$ is not γ -weakly learnable. Recalling notation from Lemma 5.1.3, we note that for every $n \in \mathbb{N}$ the class $P_{\gamma(j), 4n} \subset \mathcal{Q}'_j$. By monotonicity of the sample complexity and Lemma 5.1.3, we have $m_{\mathcal{Q}'_j}(\gamma(j), \frac{1}{7}) \geq m_{P_{\gamma(j), 4n}}(\gamma(j), \frac{1}{7}) \geq n$, proving that this class is not $\gamma(j)$ -weakly learnable.

Lastly, we need to show that for every α , there are ε and j , such that $\gamma(j) \geq (\frac{\alpha}{g(j)} + \varepsilon)$. Let $\varepsilon = \frac{1}{g(j)}$. Then we have

$$\begin{aligned}
& \frac{\alpha}{g(j)} + \varepsilon &< \gamma(j) \\
& \Leftrightarrow \frac{(\alpha + 1)}{g(j)} &< \frac{\frac{1}{j} - \frac{1}{g(j)}}{8(1 - \frac{1}{g(j)})} \\
& \Leftrightarrow \frac{8(\alpha + 1)}{g(j)} \left(1 - \frac{1}{g(j)}\right) &< \frac{1}{j} - \frac{1}{g(j)} \\
& \Leftrightarrow \frac{8(\alpha + 2)}{g(j)} - \frac{8(\alpha + 1)}{g(j)^2} &< \frac{1}{j}.
\end{aligned}$$

Thus it is sufficient to show that there is j , such that $8(\alpha + 2)j < g(j)$. Note that g is a superlinear function, i.e., for every $c \in \mathbb{R}$, there is $t_c \in \mathbb{N}$, such that for every $t \geq t_c$, $g(t) \geq ct$. This implies that for every $\alpha \in \mathbb{R}$ there are indeed ε and j , such that $\gamma(j) > \frac{\alpha}{g(j)} + \varepsilon$. Thus for any super-linear function g and any $\alpha \in \mathbb{R}$, the class $\mathcal{Q}_g$ is not α -agnostically learnable. □

The result of Theorem 6.2.1 follows directly from the construction of class $\mathcal{Q}_g$ for Theorem 6.2.1, the Claim 6.3.1.

6.5 The Existence of a Realizable Compression Scheme does not imply the Existence of a Robust Compression Scheme

Sample compression schemes are combinatorial properties of classes that imply their learnability. For a variety of learning tasks, such as binary classification or expectation maximization a class has a sample compression scheme if and only if it is learnable [Moran and Yehudayoff \[2016\]](#), [Ben-David et al. \[2017a\]](#). For classification tasks, sample compression for realizable samples implies agnostic sample compression. [Ashtiani et al. \[2018\]](#) used compression schemes to show learnability of classes of distributions in the realizable case, but

left open the question if for learning probability distributions, the existence of realizable sample compression schemes implies the existence of similar schemes for the non-realizable (agnostic, or robust) settings. We provide a negative answer to this question.

More concretely, let $\mathcal{Q}$ be a class of distributions over some domain X . A compression scheme for $\mathcal{Q}$ involves two agents: an encoder and a decoder.

- The encoder knows a distribution q and receives a sample S generated by this distribution. The encoder picks a bounded size sub-sample and sends it, possibly with a few additional bits to the decoder.
- The decoder receives the message and uses an agreed upon decoding rule (that may depend on $\mathcal{Q}$ but not on q or S) to constructs a distribution p that is close to q .

Of course, there is some probability that the samples are not representative of the distribution q , in which case the compression scheme will fail. Thus, we only require that the decoding succeed with constant probability.

We say that a class $\mathcal{Q}$ has a sample compression scheme (realizable or robust) if for every accuracy parameter $\varepsilon > 0$, the minimal required size of the sample S , and upper bounds on the size of the sub-sample and number of additional bits in the encoder's message depend only of $\mathcal{Q}$ and ε (and are independent of the sample generating q and on the sample S).

A realizable compression scheme is required to handle only q 's in $\mathcal{Q}$ and output p such that $d_{\text{TV}}(p, q) \leq \varepsilon$, while a robust compression scheme should handle any q but the decoder's output p is only required to be $\min_{q \in \mathcal{Q}} \{d_{\text{TV}}(p, q)\} + \varepsilon$ close to q .

Theorem 6.5.1 (Formal version of Theorem 6.1.2). *For every $\alpha \in \mathbb{R}$, the existence of a realizable compression scheme, does not imply the existence of an α -robust compression scheme. That is, there is a class $\mathcal{Q}$ that has a realizable compression scheme, but for every $\alpha \in \mathbb{R}$, $\mathcal{Q}$ does not have an α -robust compression scheme.*

Proof. Consider the class $\mathcal{Q} = \mathcal{Q}_g$ from Theorem 6.2.1. We note that this class has a compression scheme of size 1. However, from Ashtiani et al. [2018], we know that having a α -robust compression scheme implies α -agnostic learnability. We showed in Theorem 6.2.1 that for every α and for every superlinear function g , the class $\mathcal{Q}_g$ is not α -agnostically learnable. It follows that $\mathcal{Q}_g$ does not have an α -robust compression scheme. $\square$

Theorem 6.5.2. *The class $\mathcal{Q} = \mathcal{Q}_g$ from Theorem 6.2.1 has a compression scheme of message size 1 (i.e., using just a single sample point).*

Proof. Let $m(\varepsilon)$ be $10/g(\varepsilon)$ and, given a sample S of at least that size, let the encoder pick a subset $L(S) \subseteq S$ with

$$L(S) = \begin{cases} \{(i, 2j + 2)\} & \text{if } (i, 2j + 2) \in S \\ \{(0, 0)\} & \text{otherwise} \end{cases}$$

Let the decoder output

$$\mathcal{J}(L) = \begin{cases} q_{i,j,g(j)} & \text{if } (i, 2j + 2) \in L \\ \delta_{(0,0)} & \text{otherwise} \end{cases}$$

With this construction established, the analysis follows very similarly to the analysis in the proof of Claim 6.3.1. $\square$

We note that the claim of Theorem 6.1.2 (and Theorem 6.5.1) follows directly.

Chapter 7

Robust Distribution Learnability

7.1 Definitions Robust Learnability

We first note that the definition of agnostic learnability given in Section 4.1.1 also satisfies an interpretation of robust learning, where an adversary can corrupt a distribution by an arbitrary small change in TV-distance. The results from the previous sections thus imply that realizable learnability of distributions does not imply generally robust learnability. In this section we will now investigate what happens if the adversary is restricted to only certain kinds of manipulations. In particular, we consider *additive* and *subtractive* adversaries, who can either only add or only subtract a small mixture component to/of the distribution respectively.

Definition 7.1.1 (Additively robust learnability). *We say that a class $\mathcal{Q}$ of probability distributions is additive α -robustly learnable if there exists a learner A and a function $m_{\mathcal{Q}} : (0, 1)^2 \rightarrow \mathbb{N}$, such that for every $\eta \in (0, 1)$, every probability distribution q , every $p \in \mathcal{Q}$, and $(\varepsilon, \delta) \in (0, 1)^2$, for $m \geq m_{\mathcal{Q}}(\varepsilon, \delta)$ the probability over samples S of that size drawn i.i.d. from the distribution $\eta q + (1 - \eta)p$, that $d_{\text{TV}}(A(S), \eta q + (1 - \eta)p) \leq \alpha\eta + \varepsilon$ is at least $1 - \delta$.*

We say a class $\mathcal{Q}$ of probability distributions is additively robust learnable, if there is $\alpha \in \mathbb{R}$, such that $\mathcal{Q}$ is additively α -robustly learnable.

If the goal is to recover a distribution from the class $\mathcal{Q}$ that was additively corrupted, the goal of learning is $d_{\text{TV}}(A(S), p) \leq \alpha\eta + \varepsilon$, instead of $d_{\text{TV}}(A(S), \eta q + (1 - \eta)p) \leq \alpha\eta + \varepsilon$. We refer to learning with respect to this learning alternative goal as *Huber α -robust learnability*.

It is clear that those notions only differ by η , i.e., α -robust learnability implies Huber $(\alpha+1)$ -robust learnability and Huber α -robust learnability implies $(\alpha + 1)$ -robust learnability.

Definition 7.1.2 (Subtractive robust learnability). *We say that a class $\mathcal{Q}$ of probability distributions is subtractive α -robustly learnable if there exists a learner A and a function $m_{\mathcal{Q}} : (0, 1)^2 \rightarrow \mathbb{N}$, such that for every corruption level $\eta \in (0, 1)$, every probability distribution p for which there exists a probability distribution q such that $\eta q + (1 - \eta)p \in \mathcal{Q}$, and for every $(\varepsilon, \delta) \in (0, 1)^2$, for $m \geq m_{\mathcal{Q}}(\varepsilon, \delta)$ the probability over samples S of that size drawn i.i.d. from the distribution p , that $d_{\text{TV}}(A(S), p) \leq \alpha\eta + \varepsilon$ is at least $1 - \delta$. We say a class of distributions $\mathcal{Q}$ is subtractive robustly learnable, if there exists $\alpha \in \mathbb{R}$, such that $\mathcal{Q}$ is subtractive α -robustly learnable.*

7.1.1 Overview of Results

We explore how different robustness models affect learnability of distributions, showing strong separations between them.

As we have seen in the previous section, a distribution class being learnable does *not* imply that it is .

Theorem 6.1.1. *[Informal] There exists a class that is realizably learnable, but not agnostically/robustly learnable.*

This result can also be shown as an immediate corollary of a slightly stronger result that shows that learnability does *not* imply subtractive robust learnability.

Theorem 7.1.3 (Informal). *There exists a class that is realizably learnable, but not subtractive robustly learnable.*

The stronger result can be shown with the same construction and proof as Theorem 6.1.1. However, in this presentation of the work, we will not go into the details of this proof, but note that the stronger result also follows as an immediate corollary of a combination of the other theorems in this chapter.

Our main result of this chapter shows in contrast to the general and subtractive case, if we restrict the adversary to only additive corruptions, realizabile learnability is indeed sufficient to guarantee robustness. That is realizability learnability indeed implies additively robust learnability.

Theorem 7.1.4 (Informal). *Any class of probability distributions $\mathcal{Q}$ which is realizable learnable, is also additively robustly learnable.*

Note that, since additively robust learnability trivially implies learnability, this shows an *equivalence* between learnability and additively robust learnability.

Our algorithm enumerates over all subsets of the dataset of an appropriate size, such that at least one subset contains no samples from the contaminating distribution. A realizable learner is applied to each subset, and techniques from hypothesis selection [Yatracos \[1985\]](#), [Devroye and Lugosi \[1996, 1997, 2001\]](#) are used to pick the best of the learned distributions. Further details appear in Section 7.2.

We also note that since our robust learning algorithm enumerates all large subsets of the training dataset, it is *not* computationally efficient. Indeed, for such a broad characterization, this would be too much to ask. Efficient algorithms for robust learnability are an exciting and active field of study, but outside the scope of this work. For further discussion see Section 4.3.

Thus far, we have shown that additive and realizable learnability are equivalent and that both subtractive and general robust learnability are both not implied by realizable learnability. It remains to settle the question whether subtractive and general robust learnability are fundamentally different. General robustness, where probability mass can be both added *and* removed, is more powerful than either model individually. However, if a class is additive robustly learnable *and* subtractive robustly learnable, is it robustly learnable? Though this is intuitively true, we are not aware of an immediate proof. Using a similar argument as Theorem 7.1.4, we derive a stronger statement: that subtractively robust learnability implies robust learnability.

Theorem 7.1.5 (Informal). *If a class $\mathcal{Q}$ is subtractively robustly learnable, then it is also robustly or agnostically learnable.*

Quantitative versions of these statements can be found in their respective theorems.

We also note, that by definition, it directly follows that for every α :

- If a class of probability distribution $\mathcal{Q}$ is α -robustly learnable, then it is also additive α -robustly learnable and subtractive α -robustly learnable
- If a class of probability distribution $\mathcal{Q}$ is *additively* α -robustly learnable, then it is also learnable in the realizable case.

- If a class of probability distribution $\mathcal{Q}$ is *subtractive* α -robustly learnable, then it is learnable in the realizable case).

To summarize the qualitative versions of our findings:

- Learnability and additively robust learnability are equivalent (Theorem 7.1.4, formal version: Theorem 7.2.1);
- Learnability does not imply subtractively robust learnability (Theorem 7.1.3, formal version: Theorem 7.4.1);
- Subtractively robust learnability implies robust learnability (Theorem 7.1.5, formal version: Theorem 7.3.1);

Quantitative versions of these statements can be found in their respective theorems.

Known vs unknown level of corruption So far, we only looked at learners that were unaware of the exact level of contamination, but had to give a guarantee with respect to the true but unknown contamination level. This is the standard formulation when considering agnostic learnability. An alternative version can then be defined dependent on a parameter η , which describes a fixed level of contamination that is known by the learner. Our robust learning notions trivially imply those η -dependent notions. Furthermore, one can show that robust algorithms designed with the knowledge of η can be modified into versions that can do without that knowledge Jain et al. [2022]. We give the details on the definition and the relation between these learning models in Section 7.6. We further show in that section that for learning with known contamination models, the same picture emerges: realizable learning does not imply robustness with respect to subtractive contaminations (Theorem 7.6.3), but realizable and additively robust learning are equivalent. Similarly, subtractively robust learnability implies general robust learnability.

Adaptive Adversaries. Our definition of robustness allows an adversary to make changes to the underlying distribution. Equivalently, it corresponds to an adversary who can add or remove points from a dataset, but must commit to these modifications *before* the dataset is actually sampled. A stronger¹ adversary would be able to choose which points to add or remove *after* seeing the sampled dataset. Such an adversary is referred to as *adaptive*. Since adaptive adversaries are stronger than the ones we consider, any impossibility result

¹And in the case of removals, much more natural

that we show also holds in this settings (e.g., learnability does not imply subtractive robust learnability when the adversary is adaptive). It is an interesting open question to understand whether our algorithms can be strengthened to work in this setting. Some positive evidence in this direction is due to Blanc, Lange, Malik, and Tan [Blanc et al. \[2022\]](#), who show that adaptive and non-adaptive adversaries have qualitatively similar power in many settings.

7.2 Learnability Implies Additive Robust Learnability

We recall Theorem [7.2.1](#), which shows that any class that is realizably learnable is also additive robustly learnable.

Theorem 7.2.1. *[Formal version of Theorem [7.1.4](#)] Any class of probability distributions $\mathcal{Q}$ which is realizably learnable is also additively 3-robustly learnable. Furthermore any class $\mathcal{Q}$ which is learnable in the realizable case is Huber additively 2-robustly learnable.*

We prove this theorem by providing an algorithm based on classical tools for hypothesis selection [Yatracos \[1985\]](#), [Devroye and Lugosi \[1996, 1997, 2001\]](#). These methods take as input a set of samples from an unknown distribution and a collection of hypotheses distributions. If the unknown distribution is close to one of the hypotheses, then, given enough samples, the algorithm will output a close hypothesis. Roughly speaking, our algorithm looks at all large subsets of the dataset, such that at least one will correspond to an uncontaminated set of samples. A learner for the realizable setting (whose existence we assumed) is applied to each to generate a set of hypotheses, and we then use hypothesis selection to pick one with sufficient accuracy. The proof of Theorem [7.3.1](#) (showing that subtractively robust learnability implies robust learnability) follows almost the exact same recipe, except the realizable learner is replaced with a learner robust to subtractive contaminations. We recall some preliminaries in Section [4.2](#). We then prove Theorem [7.2.1](#) in Section [7.5.1](#), and we formalize and prove a version of Theorem [7.3.1](#) in Section [7.3](#).

We note that $\alpha = 2$ and $\alpha = 3$ are often the optimal factors to expect in distribution learning settings, even for the case of finite distribution classes. For example, for proper agnostic learning the factor $\alpha = 3$ is known to be optimal for finite collections of distributions, which holds for classes with only 2 distributions [Bousquet et al. \[2019\]](#). Similarly the factor of $\alpha = 2$ is optimal if the notion of learning is relaxed to improper learners [Bousquet](#)

et al. [2019], Chan et al. [2014b]. While we are not aware of lower bounds for the additive setting, a small constant factor such as 2 is within expectations for these problems.

For the proof of Theorem 7.2.1, we refer the reader to Section 7.5.1.

7.3 Subtractive Robust Learnability Implies Robust Learnability

Similarly, we can show that robustness with respect to a subtractive adversary implies robustness with respect to a general adversary. We note that this theorem requires a change in constants from α to $(2\alpha + 4)$.

Theorem 7.3.1. *[Formal version of Theorem 7.1.5] If a class $\mathcal{Q}$ is subtractive α -robustly learnable, then it is also $(2\alpha + 4)$ -robustly learnable.*

The proof follows a similar argument as the proof of Theorem 7.2.1 and can be found in Section 7.5.2 in the appendix. We also note that with the same proof we get a similar result for learning with fixed corruption levels.

7.4 Realizable Learnability Does Not Imply Subtractively Robust Learnability

We now can state a slightly stronger version of Theorem 6.2.1, which says that realizable learning does not imply robust learning even if the corruptions are just limited to subtractive corruptions.

Corollary 7.4.1. *[Formal version of Theorem 7.1.3] There are classes of distributions $\mathcal{Q}$, such that $\mathcal{Q}$ is realizable learnable, but for every $\alpha \in \mathbb{R}$, $\mathcal{Q}$ is not subtractively α -robustly learnable. Moreover, the sample complexity of learning $\mathcal{Q}$ can be arbitrarily close to (but larger than) linear. Namely, for any super-linear function g , there is a class $\mathcal{Q}_g$, with*

- $\mathcal{Q}_g$ is realizable learnable with sample complexity $m_{\mathcal{Q}_g}^{\text{re}}(\varepsilon, \delta) \leq \log(1/\delta)g(1/\varepsilon)$;
- for every $\alpha \in \mathbb{R}$, $\mathcal{Q}_g$ is not subtractively α -robustly learnable.

This result immediately follows from the facts that "realizable learnability does not imply robust/agnostic learnability" [6.2.1](#) and [7.3.1](#). Alternatively, one can also proof this result separately using the same construction used for Theorem [6.2.1](#). Note that the proof of Theorem [6.2.1](#) is written in such a way that one can easily replace "α-agnostic" with "subtractively α-robust" in the proof to obtain the result that realizable learnability does not imply subtractively α-robust learnability.

Furthermore, for the weaker robustness criterion of robust learning with respect to known corruption levels, we get a similar negative result, i.e., we show that realizable learnability does not imply subtractive learnability with known corruption level. This statement is formalized as Theorem [7.6.3](#).

7.5 Proofs

7.5.1 Proof of Theorem [7.2.1](#)

Proof. Recall that $\mathcal{Q}$ is a class of probability distributions which is realizable learnable. We let $\mathcal{A}_{\mathcal{Q}}^{re}$ be a realizable learner for $\mathcal{Q}$, with sample complexity $n_{\mathcal{Q}}^{re}$. Let accuracy parameters $\varepsilon, \delta > 0$ be arbitrary. We will define $m_1 \geq \max \left\{ 2m_{\mathcal{Q}}^{re} \left(\frac{\varepsilon}{9}, \frac{\delta}{5} \right), \frac{162(1+\log(\frac{5}{\delta}))}{\varepsilon^2} \right\}$ and $m_2 \geq \frac{162(2m_1 + \log(\frac{5}{\delta}))}{\varepsilon^2} \geq \frac{162(1+\log(\frac{5}{\delta}))}{\varepsilon^2}$, and $m = m_1 + m_2$ be their sum. Let η be the true corruption level.² Our additive robust learner will receive a sample $S \sim (\eta r + (1 - \eta)p)^m$ of size m where r is some arbitrary distribution. We can view a subset $S' \subset S$ as the "clean" part, being i.i.d. generated by p . The size of this clean part $|S'| = m'$ is distributed according to a binomial distribution $\text{Bin}(m, 1 - \eta)$. By a Chernoff bound ([Proposition 4.2.5](#)), we get

$$\begin{aligned} \mathbb{P} \left[m' \leq \left(1 - \frac{\varepsilon}{9} - \eta \right) m \right] &\leq \mathbb{P} \left[m' \leq \left(1 - \frac{\varepsilon}{9} - \eta + \frac{\eta\varepsilon}{9} \right) m \right] = \mathbb{P} \left[m' \leq \left(1 - \frac{\varepsilon}{9} \right) (1 - \eta) m \right] \\ &\leq \exp \left(- \left(\frac{\varepsilon}{9} \right)^2 m (1 - \eta) / 2 \right) = \exp \left(- \frac{\varepsilon^2}{162} m (1 - \eta) \right) \leq \exp \left(- \frac{\varepsilon^2}{162} m \left(1 - \frac{3}{4} \right) \right) \end{aligned}$$

Thus, given that $m = m_1 + m_2 \geq 2 \left(\frac{162(1+\log(\frac{5}{\delta}))}{\varepsilon^2} \right)$ with probability at least $1 - \frac{\delta}{5}$, we have $m' \geq m \left(1 - \eta - \frac{\varepsilon}{9} \right)$. For the rest of the argument we will now assume that we have indeed

²Note, that while our analysis frequently uses η , we construct the learner in a way that is uninformed by η . Thus our guarantees hold for all η simultaneously

$m' \geq m(1 - \eta - \frac{\varepsilon}{9})$. The learner now randomly partitions the sample S into S_1 and S_2 of sizes m_1 and m_2 , respectively. Now let $S'_1 = S_1 \cap S'$ and $S'_2 = S_2 \cap S'$ be the intersections of these sets with the clean set S' , and m'_1 and m'_2 be their respective sizes. We note that $m'_1 \sim \text{Hypergeometric}(m, m', m_1)$ and $m'_2 \sim \text{Hypergeometric}(m, m', m_2)$.³ Thus, assuming $m' \geq m(1 - \eta - \frac{\varepsilon}{9})$ using Proposition 4.2.5, we have that

$$\mathbb{P} \left[|S'_1| \leq \left(1 - \eta - \frac{2\varepsilon}{9}\right) m_1 \right] \leq e^{-\frac{2}{81}\varepsilon^2 m_1}$$

and

$$\mathbb{P} \left[|S'_2| \leq \left(1 - \eta - \frac{2\varepsilon}{9}\right) m_2 \right] \leq e^{-\frac{2}{81}\varepsilon^2 m_2},$$

where the probability is over the random partition of S . We note that by our choices of m_1 and m_2 , with probability $1 - \frac{2\delta}{5}$, the clean fractions of S_1 and S_2 (namely $\frac{|S'_1|}{|S_1|}$ and $\frac{|S'_2|}{|S_2|}$) are each at least $(1 - \eta - \frac{2\varepsilon}{9})$.

Let

$$\hat{\mathcal{H}} = \{\mathcal{A}_Q^{re}(S'') : S'' \subset S_1\}$$

be the set of distributions output by the realizable learning algorithm $\mathcal{A}_Q^{re}$ when given as input all possible subsets of S_1 . We know that this set of distributions $|\hat{\mathcal{H}}|$ is of size 2^{m_1} . By the guarantee of the realizable learning algorithm $\mathcal{A}_Q^{re}$, if there exists a “clean” subset $S'_1 \subset S_1$ where $|S'_1| \geq m_1(1 - \eta - \frac{2\varepsilon}{9})$ (i.e., $S'_1 \sim p^{m_1(1 - \eta - \frac{2\varepsilon}{9})}$), then with probability $1 - \frac{\delta}{5}$ there exists a candidate distribution $q^* \in \hat{\mathcal{H}}$ with $d_{TV}(p, q^*) = \frac{\varepsilon}{9}$.

We now define and consider the *Yatracos sets*.⁴ For every $q_i, q_j \in \hat{\mathcal{H}}$, define the Yatracos set between q_i and q_j to be $A_{i,j} = \{x : q_i(x) \geq q_j(x)\}$.⁵ We let

$$\mathcal{Y}(\hat{\mathcal{H}}) = \{A_{i,j} \subset \mathcal{X} : q_i, q_j \in \hat{\mathcal{H}}\}$$

denote the set of all pairwise Yatracos sets between distributions in the set $\hat{\mathcal{H}}$.

We now consider the A -distance Kifer et al. [2004] between two distributions with respect to the Yatracos sets, i.e., we consider

$$d_{\mathcal{Y}(\hat{\mathcal{H}})}(p', q') = \sup_{B \in \mathcal{Y}(\hat{\mathcal{H}})} |p'(B) - q'(B)|.$$

³Recall that $\text{Hypergeometric}(N, K, n)$ is the random variable of the number of “successes” when n draws are made without replacement from a set of size N , where K elements of the set are considered to be successes.

⁴These sets are also sometimes called Scheffé sets in the literature.

⁵Note that this definition is asymmetric: $A_{i,j} \neq A_{j,i}$.

This distance looks at the supremum of the discrepancy between the distributions across the Yatracos sets. Consequently, for any two distributions p', q' we have $d_{\text{TV}}(p', q') \geq d_{\mathcal{Y}(\hat{\mathcal{H}})}(p', q')$, since total variation distance is the supremum of the discrepancy across *all* possible sets. Furthermore, if $q', p' \in \hat{\mathcal{H}}$, then $d_{\text{TV}}(p', q') = d_{\mathcal{Y}(\hat{\mathcal{H}})}(p', q')$, since either of the Yatracos sets between the two distributions serves as a set that realizes the total variation distance between them.

We will now show Huber additive 2-robust learnability and then derive 3-robust learnability. Thus we need to show that the learner approximates the “clean” part of the distribution p .

Suppose there is some $q^* \in \hat{\mathcal{H}}$ with $d_{\text{TV}}(p, q^*) \leq \frac{\varepsilon}{9}$. Then for every $q \in \hat{\mathcal{H}}$:

$$\begin{aligned} d_{\text{TV}}(p, q) &\leq d_{\text{TV}}(p, q^*) + d_{\text{TV}}(q^*, q) \\ &\leq \frac{\varepsilon}{9} + d_{\mathcal{Y}(\hat{\mathcal{H}})}(q^*, q) \\ &\leq \frac{\varepsilon}{9} + d_{\mathcal{Y}(\hat{\mathcal{H}})}(q^*, p) + d_{\mathcal{Y}(\hat{\mathcal{H}})}(p, q) \\ &\leq \frac{\varepsilon}{9} + d_{\text{TV}}(q^*, p) + d_{\mathcal{Y}(\hat{\mathcal{H}})}(p, q) \\ &\leq \frac{2\varepsilon}{9} + d_{\mathcal{Y}(\hat{\mathcal{H}})}(p, q). \end{aligned}$$

Lastly, we will argue that we can empirically approximate $d_{\mathcal{Y}(\hat{\mathcal{H}})}(p, q)$, which we can then use to select a hypothesis. We note that, since $\mathcal{Y}(\hat{\mathcal{H}})$ is a finite set of size $\leq \binom{m_1}{(1 - \eta - \frac{2\varepsilon}{9})m_1}^2 = \binom{m_1}{(\eta + \frac{2\varepsilon}{9})m_1}^2 \leq ((m_1^{(\eta + \frac{2\varepsilon}{9})})^2 = m_1^{2(\eta + \frac{2\varepsilon}{9})m_1}$, we have uniform convergence⁶ with respect to $\mathcal{Y}(\hat{\mathcal{H}})$. Recall that, we assumed that there is “clean” subsample $S_2'' \subset S_2$, which is i.i.d. distributed according to p and of size $(1 - \frac{2\varepsilon}{9} - \eta)m_1$. We also note that the clean samples S_1 and S_2 are drawn independently from each other. Thus with probability $1 - \frac{\delta}{5}$, S_2'' is $\frac{\varepsilon}{9}$ -representative of p with respect to $\mathcal{Y}(\hat{\mathcal{H}})$. For a sample S_0 and a set $B \subset \mathcal{X}$, let us denote $S_0(B) = \frac{|S_0 \cap B|}{|S_0|}$. Because of the $\frac{\varepsilon}{9}$ -representativeness of S_2'' , we

⁶A collection of sets $\mathcal{W}$ has the *uniform convergence property* if for every $(\varepsilon, \delta) \in (0, 1)^2$ there is a number $m_{\mathcal{W}}(\varepsilon, \delta)$ such that for every probability distribution p , with probability $\geq (1 - \delta)$ over samples S of size $m > m_{\mathcal{W}}(\varepsilon, \delta)$ generated i.i.d. by p , a sample S is ε -representative for $\mathcal{W}$ with respect to p . Namely, for every $A \in \mathcal{W}$, $\left| \frac{|A \cap S|}{|S|} - p(A) \right| \leq \varepsilon$. If $\mathcal{W}$ is finite then $m_{\mathcal{W}}(\varepsilon, \delta) \leq \frac{\log(|\mathcal{W}|) + \log(1/\delta)}{\varepsilon^2}$. For more details see Chapter 4 in [Shalev-Shwartz and Ben-David \[2014b\]](#).

have for every $B \in \mathcal{Y}(\hat{\mathcal{H}})$:

$$|p(B) - S_2''(B)| \leq \frac{\varepsilon}{9}.$$

Thus,

$$\begin{aligned}
& |p(B) - S_2(B)| \\
&= \left| p(B) - \frac{|S_2 \cap B|}{|S_2|} \right| \\
&\leq \max \left\{ \left| p(B) - \frac{|S_2'' \cap B|}{|S_2|} \right|, \left| p(B) - \frac{|S_2'' \cap B| + (\eta + \frac{2\varepsilon}{9})m_2}{|S_2|} \right| \right\} \\
&\leq \max \left\{ \left| p(B) - \frac{S_2''(B)|S_2''|}{m_2} \right|, \left| p(B) - \frac{S_2''(B)|S_2''| + (\eta + \frac{2\varepsilon}{9})m_2}{m_2} \right| \right\} \\
&\leq \max \left\{ \left| p(B) - \left(1 - \eta - \frac{2\varepsilon}{9}\right) S_2''(B) \right|, \left| p(B) - \frac{S_2''(B) \left(1 - \eta - \frac{2\varepsilon}{9}\right) m_2 + \left(\eta + \frac{2\varepsilon}{9}\right) m_2}{m_2} \right| \right\} \\
&\leq \max \{ |p(B) - S_2''(B)| + \left| S_2''(B) - \left(1 - \eta - \frac{2\varepsilon}{9}\right) S_2''(B) \right|, \\
&\quad \left| p(B) - \left(S_2''(B) \left(1 - \eta - \frac{2\varepsilon}{9}\right) + \left(\eta + \frac{2\varepsilon}{9}\right) \right) \right| \} \\
&\leq \max \left\{ \frac{3\varepsilon}{9} + \eta, \left| p(B) - S_2''(B) \left(1 - \eta - \frac{2\varepsilon}{9}\right) - \left(\eta + \frac{2\varepsilon}{9}\right) \right| \right\} \\
&\leq \max \left\{ \frac{3\varepsilon}{9} + \eta, \left| p(B) - S_2''(B) + S_2''(B) \left(\eta + \frac{2\varepsilon}{9}\right) - \left(\eta + \frac{2\varepsilon}{9}\right) \right| \right\} \\
&\leq \max \left\{ \frac{3\varepsilon}{9} + \eta, |p(B) - S_2''(B)| + \left| S_2''(B) \left(\eta + \frac{2\varepsilon}{9}\right) - \left(\eta + \frac{2\varepsilon}{9}\right) \right| \right\} \\
&\leq \max \left\{ \frac{3\varepsilon}{9} + \eta, \frac{\varepsilon}{9} + |S_2''(B) - 1| \left(\eta + \frac{2\varepsilon}{9}\right) \right\} \\
&\leq \max \left\{ \frac{3\varepsilon}{9} + \eta, \frac{\varepsilon}{9} + \left(\eta + \frac{2\varepsilon}{9}\right) \right\} \\
&\leq \frac{3\varepsilon}{9} + \eta
\end{aligned}$$

Let the empirical A -distance with respect to the Yatracos sets be defined by

$$d_{\mathcal{Y}(\hat{\mathcal{H}})}(q, S) = \sup_{B \in \mathcal{Y}(\hat{\mathcal{H}})} |q(B) - S(B)|.$$

Now if the learner outputs $\hat{q} \in \arg \min_{q \in \mathcal{H}} d_{\mathcal{Y}(\hat{\mathcal{H}})}(q, S_2)$, then putting all of our guarantees together, we get that with probability $1 - \delta$

$$\begin{aligned}
d_{\text{TV}}(\hat{q}, p) &\leq \frac{2\varepsilon}{9} + d_{\mathcal{Y}(\hat{\mathcal{H}})}(\hat{q}, p) \\
&\leq \frac{5\varepsilon}{9} + \eta + d_{\mathcal{Y}(\hat{\mathcal{H}})}(\hat{q}, S_2) \\
&\leq \frac{5\varepsilon}{9} + \eta + d_{\mathcal{Y}(\hat{\mathcal{H}})}(q^*, S_2) \\
&\leq \frac{8\varepsilon}{9} + 2\eta + d_{\mathcal{Y}(\hat{\mathcal{H}})}(q^*, p) \\
&\leq \frac{8\varepsilon}{9} + 2\eta + \frac{\varepsilon}{9} \leq 2\eta + \varepsilon.
\end{aligned}$$

Thus we get Huber 2-additive robust learning success. This automatically implies 3-additive robust learning success. $\square$

7.5.2 Proof of Theorem 7.3.1

In this section we will provide the proof for Theorem 7.3.1.

Theorem 7.3.1. *[Formal version of Theorem 7.1.5] If a class $\mathcal{Q}$ is subtractive α -robustly learnable, then it is also $(2\alpha + 4)$ -robustly learnable.*

Proof. Let $\mathcal{Q}$ be a concept class that is subtractively α -robust learnable. Then there exists a successful subtractive α -robust learner $\mathcal{A}_{\mathcal{Q}}^{\text{sub}}$ with sample complexity $m_{\mathcal{Q}}^{\text{sub}}$ for the class $\mathcal{Q}$. Let ε and δ and be arbitrary. Let $m_1 \geq \max \left\{ 4m_{\mathcal{Q}}^{\text{sub}}(\frac{\varepsilon}{9}, \frac{\delta}{5}), \frac{162(1+\log(\frac{5}{\delta}))}{\varepsilon^2} \right\}$ and $m_2 \geq \frac{(2n_1 + \log(\frac{5}{\delta}))}{\varepsilon^2}$. Lastly let $m = m_1 + m_2$.

Let $p \in \mathcal{Q}$ be arbitrary. The α -robust learner receives a sample $S \sim p^m$ such that there is $q \in \mathcal{Q}$ such that $d_{\text{TV}}(p, q) = \eta$. Thus there exists a distributions q_1, q_2, q_3 , such that $(1 - \eta)q_1 + \eta q_2 = p$ and $(1 - \eta)q_1 + q_3 = q$. We now use the same learning strategy as in Theorem 1.5: We split the sample randomly into two subsamples S_1 and S_2 , where we use S_1 to learn candidate sets and then use S_2 to select the hypothesis from the candidate set. The goal in both settings is to find as close an approximation to q_1 as possible. The candidate based on S_1 is created by feeding subsamples of S_1 into the subtractively robust learner in such a way that with high probability one of the subsamples is guaranteed to

be i.i.d. generated by q_1 and thus (with high probability) yield a good hypothesis. More precisely, the learner randomly splits the sample S into S_1 and S_2 with $|S_1| = m_1$ and $|S_2| = m_2$. We now define the “clean” part of $S' \subset S$, i.e. the part of S' that is i.i.d. distributed according to q_1 . We note that the size of this “clean” sample $|S'| = m'$ is a random variable and distributed according to the binomial distributions $\text{Binom}(m, 1 - \eta)$. Now applying Chernoff bound, with the same argument as in the proof of Theorem 7.2.1, we get that with probability $1 - \frac{\delta}{5}$, we have $m' \geq m(1 - \eta - \frac{\varepsilon}{9})$. Now let $S'_1 = S_1 \cap S'$ and $S'_2 = S_2 \cap S'$ be the “clean parts” of the subsamples S_1 and S_2 respectively. The sizes $|S'_1| = m'_1$ and $|S'_2| = m'_2$. We note that $m'_1 \sim \text{Hypergeometric}(m, m(1 - \eta), m_1)$ and $m'_2 \sim \text{Hypergeometric}(m, m(1 - \eta), m_2)$. Thus,

$$\Pr_{\text{random split}} \left[|S'_1| \leq \left(1 - \eta - \frac{2\varepsilon}{9}\right) m_1 \right] \leq e^{-\frac{2}{81}\varepsilon^2 m_1}$$

and

$$\Pr_{\text{random split}} \left[|S'_2| \leq \left(1 - \eta - \frac{2\varepsilon}{9}\right) m_2 \right] \leq e^{-\frac{2}{81}\varepsilon^2 m_2}.$$

Taking together the guarantees on our random splits and the size of m' , we note that by our choices of m_1 and m_2 with probability $1 - \frac{3\delta}{5}$, the fractions of the parts that are i.i.d. generated by q_1 (namely $\frac{|S'_1|}{|S'_1|}$ and $\frac{|S'_2|}{|S'_2|}$) are at least $(1 - \eta - \frac{2\varepsilon}{9})$. Going forward we will assume that this is indeed the case.

Let

$$\hat{\mathcal{H}} = \left\{ \mathcal{A}_{\mathcal{Q}}^{\text{sub}}(\tilde{S}) : \tilde{S} \subset S'_1 \right\}.$$

Using our assumption that $|S'_1| \geq (1 - \eta - \frac{2\varepsilon}{9})m_1$, we know that there is $S''_1 \subset S'_1$ with $\mathcal{A}_{\mathcal{Q}}^{\text{sub}}(S''_1) \in \mathcal{H}'$. As $S''_1 \sim q_1^{(1 - \eta - \frac{2\varepsilon}{9})m_1}$, by the learning guarantee of $\mathcal{A}_{\mathcal{Q}}^{\text{sub}}$ with probability $1 - \frac{\delta}{5}$, there is a candidate distribution $q^* \in \hat{\mathcal{H}}$ with $d_{\text{TV}}(p, q^*) \leq d_{\text{TV}}(p, q_1) + d_{\text{TV}}(q_1, q^*) = \eta + (\alpha\eta + \frac{\varepsilon}{9}) = (\alpha + 1)\eta + \frac{\varepsilon}{9}$.

We now consider the Yatracos sets. For every $q_i, q_j \in \hat{\mathcal{H}}$, let $A_{i,j} = \{x : q_i(x) \geq q_j(x)\}$ and let

$$\mathcal{Y}(\hat{\mathcal{H}}) = \left\{ A_{ij} \subset \mathcal{X} : q_i, q_j \in \hat{\mathcal{H}} \right\}.$$

We now consider the A -distance [Kifer et al. \[2004\]](#) between two distributions with respect to the Yatracos sets, i.e., we consider

$$d_{\mathcal{Y}(\hat{\mathcal{H}})}(p', q') = \sup_{B \in \mathcal{Y}(\hat{\mathcal{H}})} |p'(B) - q'(B)|.$$

We note, that for any two distributions p', q' we have $d_{\text{TV}}(p', q') \geq d_{\mathcal{Y}(\hat{\mathcal{H}})}(p', q')$. Furthermore, if $q', p' \in \hat{\mathcal{H}}$, then $d_{\text{TV}}(p', q') = d_{\mathcal{Y}(\hat{\mathcal{H}})}(p', q')$. Assume there is $q^* \in \hat{\mathcal{H}}$ with $d_{\text{TV}}(p, q^*) \leq (\alpha + 1)\eta + \frac{\varepsilon}{9}$, then for every $q \in \hat{\mathcal{H}}$:

$$\begin{aligned}
d_{\text{TV}}(p, q) &\leq d_{\text{TV}}(p, q^*) + d_{\text{TV}}(q^*, q) \\
&\leq (\alpha + 1)\eta + \frac{\varepsilon}{9} + d_{\mathcal{Y}(\hat{\mathcal{H}})}(q^*, q) \\
&\leq (\alpha + 1)\eta + \frac{\varepsilon}{9} + d_{\mathcal{Y}(\hat{\mathcal{H}})}(q^*, p) + d_{\mathcal{Y}(\hat{\mathcal{H}})}(p, q) \\
&\leq (\alpha + 1)\eta + \frac{\varepsilon}{9} + d_{\text{TV}}(q^*, p) + d_{\mathcal{Y}(\hat{\mathcal{H}})}(p, q) \\
&\leq 2(\alpha + 1)\eta + \frac{2\varepsilon}{9} + d_{\mathcal{Y}(\hat{\mathcal{H}})}(p, q).
\end{aligned}$$

Lastly, we will argue that we can empirically approximate $d_{\mathcal{Y}(\hat{\mathcal{H}})}(p, q)$, which we can then use to select a hypothesis. We note that, since $\mathcal{Y}(\hat{\mathcal{H}})$ is a finite set of size $|\mathcal{Y}(\hat{\mathcal{H}})| = (2^{m_1})^2 = 2^{2m_1}$, we have uniform convergence with respect to $\mathcal{Y}(\hat{\mathcal{H}})$. Recall that $S'_2 \sim q_1^{m'_2}$ and by our previous assumption $m'_2 \leq m_2 \left(1 - \eta - \frac{2\varepsilon}{9}\right)$. Thus by our choice of m_2 , with probability $1 - \frac{\delta}{5}$, there is $S''_2 \subset S'_2 \subset S_2$ with $|S''_2| = \left(1 - \eta - \frac{2\varepsilon}{9}\right) m_2$ such that S''_2 is $\frac{\varepsilon}{9}$ -representative of q_1 with respect to $\mathcal{Y}(\hat{\mathcal{H}})$. For a sample S_0 and a set $B \subset \mathcal{X}$, let us denote $S_0(B) = \frac{|S_0 \cap B|}{|S_0|}$. Because of the $\frac{\varepsilon}{9}$ -representativeness of S'_2 , we have for every $B \in \mathcal{Y}(\hat{\mathcal{H}})$:

$$|q_1(B) - S''_2(B)| \leq \frac{\varepsilon}{9}$$

Thus,

$$\begin{aligned}
|q_1(B) - S_2(B)| &\leq |q_1(B) - S_2''(B)| + |S_2''(B) - S_2(B)| \\
&\leq \frac{\varepsilon}{9} + \left| \frac{|S_2 \cap B|}{|S_2|} - \frac{|S_2'' \cap B|}{|S_2''|} \right| \\
&\leq \frac{\varepsilon}{9} + \left| \frac{|S_2 \cap B|}{m_2} - \frac{|S_2'' \cap B|}{m_2(1 - \eta - \frac{2\varepsilon}{9})} \right| \\
&= \frac{\varepsilon}{9} + \left| \frac{|S_2 \cap B|(1 - \eta - \frac{2\varepsilon}{9}) - |S_2'' \cap B|}{(1 - \eta - \frac{2\varepsilon}{9})m_2} \right| \\
&\leq \frac{\varepsilon}{9} + \max \left\{ \frac{(|S_2'' \cap B| + (\eta + \frac{2\varepsilon}{9})m_2)(1 - \eta - \frac{2\varepsilon}{9}) - |S_2'' \cap B|}{(1 - \eta - \frac{2\varepsilon}{9})m_2}, \right. \\
&\quad \left. \frac{||S_2'' \cap B|(1 - \eta - \frac{2\varepsilon}{9}) - |S_2'' \cap B||}{(1 - \eta - \frac{2\varepsilon}{9})m_2} \right\} \\
&\leq \frac{\varepsilon}{9} + \max \left\{ \frac{|(\eta + \frac{2\varepsilon}{9})m_2(1 - \eta - \frac{2\varepsilon}{9}) - |S_2'' \cap B|(\eta + \frac{2\varepsilon}{9})|}{(1 - \eta - \frac{2\varepsilon}{9})m_2}, \frac{(\eta + \frac{2\varepsilon}{9})}{(1 - \eta - \frac{2\varepsilon}{9})} \right\} \\
&\leq \frac{\varepsilon}{9} + \max \left\{ \frac{|(\eta + \frac{2\varepsilon}{9})(m_2(1 - \eta - \frac{2\varepsilon}{9}))|}{(1 - \eta - \frac{2\varepsilon}{9})m_2}, \eta + \frac{2\varepsilon}{9} \right\} \\
&\leq \frac{\varepsilon}{9} + \eta + \frac{2\varepsilon}{9} \leq \frac{3\varepsilon}{9} + \eta
\end{aligned}$$

Let us remember that the empirical A -distance with respect to the Yatracos is defined by

$$d_{\mathcal{Y}(\hat{\mathcal{H}})}(q, S) = \sup_{B \in \mathcal{Y}} |q(B) - S(B)|.$$

Now if the learner outputs $\hat{q} \in \arg \min_{q \in \hat{\mathcal{H}}} d_{\mathcal{Y}(\hat{\mathcal{H}})}(q, S_2)$, then putting all of our guarantees

together, with probability $1 - \delta$ we get

$$\begin{aligned}
d_{\text{TV}}(\hat{q}, p) &\leq 2(\alpha + 1)\eta + \frac{2\varepsilon}{9} + d_{\mathcal{Y}}(\hat{q}, p) \\
&\leq 2(\alpha + 1)\eta + \frac{2\varepsilon}{9} + d_{\mathcal{Y}}(\hat{q}, q_1) + d_{\mathcal{Y}}(q_1, p) \\
&\leq 2(\alpha + 1)\eta + \frac{2\varepsilon}{9} + \eta + \left(\eta + \frac{3\eta}{9}\right) + d_{\mathcal{Y}}(\hat{q}, S_2) \\
&\leq 2(\alpha + 2)\eta + \frac{5\varepsilon}{9} + d_{\mathcal{Y}}(\hat{q}, S_2) \\
&\leq 2(\alpha + 2)\eta + \frac{5\varepsilon}{9} + d_{\mathcal{Y}}(q^*, S_2) \\
&\leq 2(\alpha + 2)\eta + \frac{5\varepsilon}{9} + \left(\eta + \frac{3\varepsilon}{9}\right) + d_{\mathcal{Y}}(q^*, q_1) \\
&\leq (2\alpha + 3)\eta + \frac{8\varepsilon}{9} + \eta + d_{\mathcal{Y}}(q^*, p) \\
&\leq (2\alpha + 4)\eta + \varepsilon + d_{\text{TV}}(q^*, p).
\end{aligned}$$

□

7.6 Known contamination levels

So far, we only looked at learners that were unaware of the exact level of corruption, but had to give a guarantee with respect to the true but unknown corruption level. An easier model assumes that the level of corruption η is actually known.

Definition 7.6.1 (Learnability with respect to known corruption).

- For $0 \leq \eta \leq 1$ and $\alpha > 0$, we say that a class $\mathcal{Q}$ of probability distributions is η - α -robustly learnable if there exists a learner A and a function $m_{\mathcal{Q}} : (0, 1)^2 \rightarrow \mathbb{N}$, such that for every probability distribution p such that $\min\{d_{\text{TV}}(p, p') : p' \in \mathcal{Q}\} \leq \eta$ and $(\varepsilon, \delta) \in (0, 1)^2$, for $m \geq m_{\mathcal{Q}}(\varepsilon, \delta)$ the probability over samples S of that size drawn i.i.d. from the distribution p that $d_{\text{TV}}(p, A(S)) \leq \alpha\eta + \varepsilon$ is at least $1 - \delta$.
- Given parameters $0 \leq \eta \leq 1$ and $\alpha > 0$, we say that a class $\mathcal{Q}$ of probability distributions is η -additive α -robustly learnable if there exists a learner A and a function $m_{\mathcal{Q}} : (0, 1)^2 \rightarrow \mathbb{N}$, such that for every probability distribution q , every $p \in \mathcal{Q}$, and $(\varepsilon, \delta) \in (0, 1)^2$, for $m \geq m_{\mathcal{Q}}(\varepsilon, \delta)$ the probability over samples S of that size drawn

i.i.d. from the distribution $\eta q + (1 - \eta)p$, that $d_{\text{TV}}(A(S), \eta q + (1 - \eta)p) \leq \alpha\eta + \varepsilon$ is at least $1 - \delta$.

- *Given parameters $0 \leq \eta \leq 1$ and $\alpha > 0$, we say that a class $\mathcal{Q}$ of probability distributions is η -subtractive α -robustly learnable if there exists a learner A and a function $m_{\mathcal{Q}} : (0, 1)^2 \rightarrow \mathbb{N}$, such that for every probability distribution p for which there exists a probability distribution q such that $\eta q + (1 - \eta)p \in \mathcal{C}$, and for every $(\varepsilon, \delta) \in (0, 1)^2$, for $m \geq m_{\mathcal{Q}}(\varepsilon, \delta)$ the probability over samples S of that size drawn i.i.d. from the distribution p , that $d_{\text{TV}}(A(S), p) \leq \alpha\eta + \varepsilon$ is at least $1 - \delta$.*

We note that it is easy to see that for every $\eta \in (0, 1)$

- α -robust learnability implies (η, α) -robust learnability,
- additive α -robust learnability implies η -additive α -robust learnability,
- and subtractive α -robust learnability implies η -subtractive α -robust learnability.

As a consequence, all of our positive learning results for α -robust, additive α -robust and subtractive α -robust learning, yield positive results for (η, α) -robust learning, η -additive α -robust learning and η -subtractive α -robust learning, respectively. In particular, this means that Theorem 7.2.1 also hold if we replace “additive α -robust” with “ η -additive α -robust” in those statements.

More surprisingly, it has been shown that in many cases, robust algorithms designed with knowledge of the distance η to $\mathcal{Q}$ can be modified to succeed without that knowledge Jain et al. [2022]. In particular, we have the following equivalence between the robust learning notion given in Section 7.1 and the robustness notions in Definition 7.6.1

Theorem 7.6.2 (Adaptation from Jain et al. [2022]). *Let $\alpha \geq 1$.*

- *If a class $\mathcal{Q}$ is (η, α) -robustly learnable for every $\eta \in (0, 1)$, then $\mathcal{Q}$ is 3α -robustly learnable.*
- *If a class $\mathcal{Q}$ is η -additive α -robustly learnable for every $\eta \in (0, 1)$, then $\mathcal{Q}$ is additive 3α -robustly learnable.*
- *If a class $\mathcal{Q}$ is η -subtractive, α -robustly learnable for every $\eta \in (0, 1)$, then $\mathcal{Q}$ is subtractive 3α -robustly learnable.*

- Proof.* • Consider a class $\mathcal{Q}$ that is (η, α) -robustly learnable. Now consider any $\varepsilon, \delta \in (0, 1)$. We know that for every $i \in 0, \dots, \lceil \frac{8\alpha}{\varepsilon} \rceil$, there is a learner A_i and sample-complexity function $m_{\mathcal{Q},i}$ such that A is a successful $(\frac{i\varepsilon}{8\alpha}, \alpha)$ -robust learner for $\mathcal{Q}$ with sample-complexity $m_{\mathcal{Q},i}$. Now let $m = m_{\mathcal{Q}}(\varepsilon, \delta) = O\left(\frac{\frac{\varepsilon}{4} + \log(\frac{2}{\delta})}{(\varepsilon/4)^2} + \sum_{i=0}^{\lceil 8\alpha/\varepsilon \rceil} m_{\mathcal{Q},i}(\frac{\varepsilon}{8\alpha}, \frac{\delta}{2})\right)$. Now let $S \sim p^m$. Now separate S into subsamples $S_0, S_1, \dots, S_{\lceil \frac{8\alpha}{\varepsilon} \rceil}$ with $|S_0| = O\left(\frac{\frac{\varepsilon}{4} + \log(\frac{2}{\delta})}{(\varepsilon/4)^2}\right)$ and for every $1 \leq i \leq \lceil \frac{8\alpha}{\varepsilon} \rceil$, $|S_i| = m_{\mathcal{Q},i}(\frac{\varepsilon}{8\alpha}, \frac{\delta}{2})$. Now for every i , let $q_i = A_i(S_i)$. Let η be the true corruption level, i.e., $\min_{q \in \mathcal{C}} d_{\text{TV}}(p, q) = \eta$. Then for $j = \lceil \frac{\eta}{8\alpha} \rceil$, we have by the learning guarantee of A_j , that with probability $1 - \frac{\delta}{2}$, $d_{\text{TV}}(q_j, p) = \alpha \lceil \frac{j\varepsilon}{8\alpha} \rceil + \frac{\varepsilon}{8\alpha} \leq \alpha \left(\eta + \frac{\varepsilon}{8\alpha}\right) \frac{\varepsilon}{8\alpha} \leq \alpha\eta + \frac{\varepsilon}{4}$. Now let $\mathcal{H} = \{q_i : i \in [\lceil \frac{4\alpha}{\varepsilon} \rceil]\}$. From Theorem 4.2.1, we know that $\mathcal{H}$ can be 3-robustly learned with sample complexity $|S_0|$. Thus if we run the Yatracos learner on S_0 , combined with the previous learning guarantee, we get that with probability $1 - \delta$, the learner outputs a hypothesis q with $d_{\text{TV}}(p, q) \leq 3(\alpha\eta + \frac{\varepsilon}{4}) + \frac{\varepsilon}{4} = 3\alpha\eta + \varepsilon$.
- Consider a class $\mathcal{Q}$ that is η -additively α -robustly learnable. Now consider any $\varepsilon, \delta \in (0, 1)$. We know that for every $i \in 0, \dots, \lceil \frac{8\alpha}{\varepsilon} \rceil$, there is a learner A_i and sample-complexity function $m_{\mathcal{Q},i}$ such that A is a successful $\frac{i\varepsilon}{8\alpha}$ -additive α -robust learner for $\mathcal{Q}$ with sample-complexity $m_{\mathcal{Q},i}$. Now let $m = m_{\mathcal{Q}}(\varepsilon, \delta) = O\left(\frac{\frac{\varepsilon}{4} + \log(\frac{2}{\delta})}{(\varepsilon/4)^2} + \sum_{i=0}^{\lceil 8\alpha/\varepsilon \rceil} m_{\mathcal{Q},i}(\frac{\varepsilon}{8\alpha}, \frac{\delta}{2})\right)$. Let $u = (1 - \eta)p + \eta q$. Now let $S \sim u^m$. Now separate S into subsamples $S_0, S_1, \dots, S_{\lceil \frac{8\alpha}{\varepsilon} \rceil}$ with $|S_0| = O\left(\frac{\frac{\varepsilon}{4} + \log(\frac{2}{\delta})}{(\varepsilon/4)^2}\right)$ and for every $1 \leq i \leq \lceil \frac{8\alpha}{\varepsilon} \rceil$, $|S_i| = m_{\mathcal{Q},i}(\frac{\varepsilon}{8\alpha}, \frac{\delta}{2})$. Now for every i , let $q_i = A_i(S_i)$. Let η be the true corruption level, i.e., $\eta = \min\{\eta' : \exists p' \in \mathcal{Q} \text{ distribution } r \text{ with } u = (1 - \eta')p' + \eta r\}$. Now let $j = \lceil \frac{\eta}{8\alpha} \rceil$. We note, that since $\eta < \frac{8j\alpha}{\varepsilon}$ there is $p' \in \mathcal{Q}$ and distributions r, r' , such $u = (1 - \eta)p' + \eta r = (1 - \frac{8j\alpha}{\varepsilon})p' + \frac{8j\alpha}{\varepsilon}r'$. Thus, by the learning guarantee of A_j , with probability $1 - \frac{\delta}{2}$, $d_{\text{TV}}(q_j, u) = \alpha \lceil \frac{j\varepsilon}{8\alpha} \rceil + \frac{\varepsilon}{8\alpha} \leq \alpha \left(\eta + \frac{\varepsilon}{8\alpha}\right) \frac{\varepsilon}{8\alpha} \leq \alpha\eta + \frac{\varepsilon}{4}$. Now let $\mathcal{H} = \{q_i : i \in [\lceil \frac{4\alpha}{\varepsilon} \rceil]\}$. From Theorem 4.2.1, we know that $\mathcal{H}$ can be 3-robustly learned with sample complexity $|S_0|$. Thus if we run the Yatracos learner on S_0 , combined with the previous learning guarantee, we get that with probability $1 - \delta$, the learner outputs a hypothesis u' with $d_{\text{TV}}(u, u') \leq 3(\alpha\eta + \frac{\varepsilon}{4}) + \frac{\varepsilon}{4} = 3\alpha\eta + \varepsilon$.
 - Consider a class $\mathcal{Q}$ that is η -subtractive α -robustly learnable. Now consider any $\varepsilon, \delta \in (0, 1)$. We know that for every $i \in 0, \dots, \lceil \frac{8\alpha}{\varepsilon} \rceil$, there is a learner A_i and sample-complexity function $m_{\mathcal{Q},i}$ such that A is a successful $\frac{i\varepsilon}{8\alpha}$ -subtractive α -robust

learner for $\mathcal{Q}$ with sample-complexity $m_{\mathcal{Q},i}$. Furthermore, let $m = m_{\mathcal{Q}}(\varepsilon, \delta) = O\left(\frac{\frac{\varepsilon}{8\alpha} + \log(\frac{2}{\delta})}{(\varepsilon/4)^2} + \sum_{i=0}^{\lceil 8\alpha/\varepsilon \rceil} m_{\mathcal{Q},i}\left(\frac{\varepsilon}{8\alpha}, \frac{\delta}{2}\right)\right)$. Now let $S \sim p^m$. Now separate S into subsamples $S_0, S_1, \dots, S_{\lceil \frac{8\alpha}{\varepsilon} \rceil}$ with $|S_0| = O\left(\frac{\frac{\varepsilon}{4} + \log(\frac{2}{\delta})}{(\varepsilon/4)^2}\right)$ and for every $1 \leq i \leq \lceil \frac{8\alpha}{\varepsilon} \rceil$, $|S_i| = m_{\mathcal{Q},i}(\frac{\varepsilon}{8\alpha}, \frac{\delta}{2})$. Now for every i , let $q_i = A_i(S_i)$. Let η be the true corruption level, i.e., $\eta = \min\{\eta' : \exists q \in \mathcal{Q} \text{ distribution } r \text{ with } q = (1 - \eta')p + \eta r\}$. Now let $j = \lceil \frac{\eta}{8\alpha} \rceil$. We note, that since $\eta < \frac{8j\alpha}{\varepsilon}$ there is $q' \in \mathcal{Q}$ and distributions r, r' , such $q' = (1 - \eta)p + \eta r = (1 - \frac{8j\alpha}{\varepsilon})p + \frac{8j\alpha}{\varepsilon}r'$. Thus, by the learning guarantee of A_j , with probability $1 - \frac{\delta}{2}$, $d_{\text{TV}}(q_j, p) = \alpha \lceil \frac{j\varepsilon}{8\alpha} \rceil + \frac{\varepsilon}{8\alpha} \leq \alpha\left(\eta + \frac{\varepsilon}{8\alpha}\right) \frac{\varepsilon}{8\alpha} \leq \alpha\eta + \frac{\varepsilon}{4}$. Now let $\mathcal{H} = \{q_i : i \in [\lceil \frac{4\alpha}{\varepsilon} \rceil]\}$. From Theorem 4.2.1, we know that $\mathcal{H}$ can be 3-robustly learned with sample complexity $|S_0|$. Thus if we run the Yatracos learner on S_0 , combined with the previous learning guarantee, we get that with probability $1 - \delta$, the learner outputs a hypothesis q with $d_{\text{TV}}(p, q) \leq 3(\alpha\eta + \frac{\varepsilon}{4}) + \frac{\varepsilon}{4} = 3\alpha\eta + \varepsilon$. $\square$

We will now discuss some more specific results for models with known contamination. We again consider the relation between subtractive and realizable learning. Similar to our previous result, we can show that realizable learning does not imply η -subtractive α -robust learning.

Theorem 7.6.3. *For every $\alpha > 0$, there exists a class that is learnable, but not η -subtractively α -robustly learnable for any $0 \leq \eta \leq \frac{1}{16\alpha}$.*

We note, that Theorem 7.6.3 is stronger than Corollary 7.4.1, in the sense that η -subtractive α -robust learning is the weaker learning requirement. However, Corollary 7.4.1 is stronger in the sense that we provide a single class that is not α -robustly learnable for every α , whereas in the proof of Theorem 7.6.3 we give a different class for each α (though the two constructions are similar). The proof of this theorem can be found in Section 7.6.1. Lastly we will consider the relationship between subtractively robust and robust learning. We can again show that subtractive robust learning implies robust learning in the following sense

Corollary 7.6.4. *If a class $\mathcal{Q}$ is η -subtractively α -robustly learnable, then it is also η -($2\alpha + 4$)-robustly learnable.*

While this statement does not follow directly from the statement of Theorem 7.3.1, the proof for Theorem 7.3.1 works as a proof for Corollary 7.6.4. The proof can be found in

Section 7.5.2. We have thus shown, that if we shift our attention to models with known contamination levels, the same picture emerges:

- realizable and additively robust learning are equivalent;
- subtractively robust and generally robust learning are equivalent;
- and realizable learnability does not imply generally robust learnability.

7.6.1 Proof of Theorem 7.6.3

The result of Theorem 7.6.3 follows directly from the construction of class $\mathcal{Q}_g$ for Theorem 6.2.1, the Claim 6.3.1 that shows this class is realizable learnable, and an adapted version for Claim 6.3.2, which states the following:

Claim 7.6.5. For every α , there is $g(t) \in O(t^2)$, such that for every $0 \leq \eta \leq \frac{1}{16\alpha}$ the class $\mathcal{Q}_g$ is not η -subtractive α -robustly learnable.

Proof. Let $\alpha > 1$ be arbitrary. Let $g : \mathbb{N} \rightarrow \mathbb{N}$ be defined by $g(t) = 32\alpha t^2$ for all $t \in \mathbb{N}$. Now for every $0 \leq \eta \leq \frac{1}{16\alpha}$, there exists some j , such that $\frac{j}{g(j)} \leq \eta \leq \frac{1}{16\alpha j}$

For such j , we consider the distributions

$$q'_{i,j} = \left(1 - \frac{1}{j}\right) \delta_{(0,0)} + \frac{1}{j} U_{A_i \times \{2j+1\}}$$

as in the proof of Claim 6.3.1. Recall that the element of $\mathcal{Q}_g$ are of the form

$$q_{i,j,g(j)} = \left(1 - \frac{1}{j}\right) \delta_{(0,0)} + \left(\frac{1}{j} - \frac{1}{g(j)}\right) U_{A_i \times \{2j+1\}} + \frac{1}{g(j)} \delta_{(i,2j+2)}$$

Then we have,

$$\begin{aligned} q_{i,j,g(j)} &= \left(1 - \frac{1}{j}\right) \delta_{(0,0)} + \left(\frac{1}{j} - \frac{1}{g(j)}\right) U_{A_i \times \{2j+1\}} + \frac{1}{g(j)} \delta_{(i,2j+2)} = \\ &= \left(1 - \frac{1}{j}\right) \delta_{(0,0)} + \left(\frac{g(j) - j}{jg(j)}\right) U_{A_i \times \{2j+1\}} + \frac{j}{jg(j)} \delta_{(i,2j+2)} = \\ &= \left(\frac{g(j) - j}{g(j)}\right) \left(\left(1 - \frac{1}{j}\right) \delta_{(0,0)} + \frac{1}{j} U_{A_i \times \{2j+1\}}\right) + \frac{j}{g(j)} \left(\left(1 - \frac{1}{j}\right) \delta_{(0,0)} + \frac{1}{j} \delta_{(i,2j+2)}\right) \\ &= \left(1 - \frac{j}{g(j)}\right) q'_{i,j} + \frac{j}{g(j)} \left(\left(1 - \frac{1}{j}\right) \delta_{(0,0)} + \frac{1}{j} \delta_{(i,2j+2)}\right). \end{aligned}$$

Thus for every element $q'_{i,j}$ of the class $\mathcal{Q}'_j = \{q'_{i,j} : i \in \mathbb{N}\}$, there is a distribution p , such that $(1 - \eta)q'_{i,j} + \eta p \in \mathcal{Q}_g$. That is, every element of $\mathcal{Q}'_j$ results from the η -subtractive contamination of some element in $\mathcal{Q}_g$. Thus, for showing that $\mathcal{Q}_g$ is not η -subtractive α -robustly learnable, it is sufficient to show, that $\mathcal{Q}'_j$ is not $(\alpha\eta + \varepsilon)$ -weakly learnable for $\varepsilon = \frac{1}{16j}$. As we have seen in the proof of Claim 6.3.2, we can use Lemma 5.1.3 to show that for every n , we have $n_{\mathcal{Q}'_j}(\frac{1}{8j}, \frac{1}{7}) \geq n$. Lastly, we need that $\frac{1}{8j} \geq \alpha\eta + \varepsilon$, or after replacing ε , we need $\frac{1}{16j} \geq \alpha\eta$. This follows directly from the choice of g .

□

Part III

Robust Learning with Uncertain Manipulation Capabilities

In this part, I will investigate binary classification that is subject to manipulation and how the prior knowledge of manipulation capabilities and behaviours changes the landscape of learnability. In particular I will discuss how a lack of certainty in manipulation capabilities can make learning impossible, while also showing that if the class of potential candidates for manipulation capabilities satisfies certain structural niceness conditions, learning is possible despite the uncertainty. I will discuss two major topics with emerged from motivations for learning subject to manipulation:

Chapter 8: Strategic Classification Instances in this setting represent self-motivated agents, that have some amount of control over their feature vectors. In particular, they can pay some cost in order to change their feature vectors. This chapter is based on two works:

- Learning Losses for Strategic Classification (joint work with Ruth Uerner, AAAI 2022): Sections [8.2-8.5](#)
- Strategic Classification with Unknown User Manipulations (joint work with Ruth Uerner and Shai Ben-David, ICML 2023): Sections [8.6-8.8.2](#)

Chapter 9: Adversarial Robustness There is an adversary with the goal to fool the classifier. The adversary can corrupt data through "humanly imperceptible" manipulations. This field has emerged from the existence of adversarial examples in image classification, i.e., images that looked indistinguishable from training data, but was classified incorrectly by the trained model. This chapter is based on one paper:

- Adversarially Robust Learning with Uncertain Perturbation Sets (joint work with Vinayak Pathak and Ruth Uerner, NeurIPS 2023)

In both lines of work I investigate the role of the prior knowledge of the manipulation structure in learning. In particular I look at the implications if the partial order induced by the loss on the manipulation structure is simple and give learning guarantees in both cases. Moreover, we can see that in both settings the realizable and the agnostic case are different for unknown manipulation structures. This is due to the fact that in the robustly realizable case, the fact of realizability can be exploited to find a hypothesis in the class that is realizable with respect to the maximum manipulation structure. We establish this notion of maximally robust empirical risk minimization criterion and exploit different use-cases. While both settings share similarities, I also want to highlight a few differences between the two settings, before introducing both chapters.

- in the strategic settings we consider there is a universally preferred label outcome for all instances, e.g. representing the decision "accept", while in adversarially robust setting it is assumed that the goal of the adversary is primarily to fool the classifier, i.e. cause incorrect predictions independent of the outcome label
- The manipulation structures, i.e. manipulation graphs in the strategic setting and perturbation regions in the adversarial setting represent different things: The edges in the manipulation graphs represent actions that have a cost that is lower than the benefit of being classified with label 1, the perturbation regions represent "regions of imperceptible change". We note that feasible actions to an individual in the strategic setting might indeed be perceivable.
- If the labels in the adversarial setting represent humanly perceptible categories, as in image classification, when the question is whether a human would see a particular animal on a picture, then humanly imperceptible changes should not make a difference to the Bayes classifier. Thus in the adversarial setting the assumption that the Bayes classifier is robust is plausible. In contrast, in many cases where strategic actions are allowed we can not assume that the Bayes classifier is robust.
- In the strategic setting we can assume that every instance is self-motivated. Thus any strategic possibility is likely to be exploited. We can use this fact to observe distribution shifts in the population after the deployment of classifiers to learn the manipulation capabilities. This does not hold in the adversarial case in which the adversary is supposed to represent an exceptional case. While our assumption that every possible strategic possibility will be used might be too strong to model reality precisely, a discounted version of this distribution shift is likely to happen in reality, i.e. it might be the case that a constant fraction of the population will manipulate or there are other plausible and simple ways to model the distribution shift.
- Many adversarial examples arise from the fact that the data lives in high dimension. The adversarial manipulations often occur in a region that is outside of the support of the distribution and are therefore not plausible. Furthermore it is plausible that for an adversarially generated example, abstaining from prediction is a reasonable option, as there might not be an incentive to provide a service to an adversary. In contrast, in many strategic settings, e.g. if one has to decide whether to accept or deny applicants, the possibility of abstention does not exist.
- In strategic classification there are more potential criteria for optimization than accuracy, such as the avoidance of social burden. That is one might not want to force individuals to incur the cost of manipulation in order to be classified correctly.

I will now list some differences to the adversarial setting which might come into play in some strategic scenarios, but which we do not model in our work:

- In strategic classification it makes sense to model individuals as incentive-driven agents. While a best-response model might not always be plausible, the awareness of the cost and incentive structures of individuals might be plausibly exploited. Our discussion is limited to the discretized case of manipulation graphs. This gives rise to a worst-case analysis that does not require individuals to act according to a best response model, but it might also be less optimal than more complex modellings of plausible behaviours that arise from a given cost landscape
- In the strategic setting, changes in the feature vector might also be due to changes in qualifications of the individual. For example an individual might study for a test to improve their score and might improve their knowledge and skills in a field along the way. While there have been works in the literature discussing these effects, our work does not consider changes due to improvement. We will leave this direction for future work.

Chapter 8

Strategic Classification

8.1 Introduction

In many scenarios where a decision rule is learned from data, the publication of this decision rule has an effect on the distribution of the underlying population that may harm the quality of the rule. Consider for instance the following scenario: a college or university has large amounts of records of students who at some point applied to the school, got admitted and then either succeeded or failed at obtaining a degree. Based on these records, the university sets (and publishes) admission criteria with the intent to admit students that are likely to successfully graduate. It then receives a set of applications for admission for the next year, some of which will lead to admission. In the next year (and upcoming years), the university will need to set and publish admission criteria again, its aim still being to attract and admit students that are likely to succeed. This scenario differs from a classic (statistical) decision-making setup in several ways: first, the entities that decisions are to be made for are human beings, and as such may actually adapt their application materials (as best as they can) to fit the published admission criteria. The decision maker may not know in advance in what ways the applicants can modify their credentials to get accepted. In addition, when it is time to publish a decision rule for the next round, the decision maker does not have feedback on the quality of last year's admission yet (since students usually take several years before they graduate or leave school without a degree). Thus the only information about the results from the last published decision rule was the set of (potentially strategically modified) applications.

Many human-centric real-life applications of machine learning, such as decisions on loan applications or bail recommendations, share characteristics with the above sketched

scenario: there is a need for transparent classification and therefore a need (or maybe even a legal requirement) of publishing the decision rule to be used. This requirement for transparency, while in most scenarios well justified, has the effect that individuals might use this knowledge to adapt to or game the rules, i.e. they might change their feature vectors strategically in order to receive a desired outcome from the published classifier. However, this change of features often does not correspond to a change in their ground-truth label. Such feature manipulations then yield a loss in accuracy of the learned classifier after its publication. Moreover, often by the time the next round of decision-making is due, the outcomes from the previous rounds are not known yet. That is, in addition to some labelled data to start with, a learner has access only to unlabelled data that potentially contains manipulated features in subsequent rounds.

The effects and incentives resulting from strategic behavior in classification scenarios have received substantial attention from both machine learning and game-theoretic perspectives in recent years [Hardt et al., 2016, Milli et al., 2019, Haghtalab et al., 2020, Tsirtsis and Rodriguez, 2020, Zhang and Conitzer, 2021], with Hardt et al [Hardt et al., 2016] first proposing the field of strategic classification as a study of the phenomenon of learning classifiers which are robust to strategic manipulations. Most works study this as a two-player game between an institution that publishes a decision rule and a population of best responding agents to be classified. Given the classifier, these agents may change their feature representations in order to obtain a more favorable classification outcome. To prevent the induced additional classification error the institution will publish a modified predictor, not transparently reflecting the underlying intent and potentially causing additional harm to sub-populations that may be less equipped to perform the required changes to their representations [Hu et al., 2019].

In our work, we propose a learning theoretic take on this scenario. In machine learning, it is common to model desiderata for a learning outcome in form of a loss function. The goal of the learning process is then to identify a predictor that minimizes this loss in expectation over a data-generating distribution. Thus, we here define a novel loss function for learning under strategic manipulations. The aim of this loss is to induce a combination of two (potentially competing) requirements: achieving low classification error taking into account that individuals being classified may manipulate their features, and discouraging such feature manipulations overall. Prior work has shown that these may be conflicting requirements [Zhang and Conitzer, 2021]. Our proposed loss function thus aims to induce a balanced combination of these requirements rather than strictly enforcing one, and then only observing the effect on the other (as is implied by frameworks that aim to minimize classification error under best-responding agents [Hardt et al., 2016, Milli et al., 2019] or enforcing incentive-compatibility [Zhang and Conitzer, 2021]).

To define our *strategic loss* we employ an abstraction of the plausible feature manipulations in form of a *manipulation graph* [Zhang and Conitzer, 2021]. An edge $\mathbf{x} \rightarrow \mathbf{x}'$ in this graph indicates that an individual with feature vector $\mathbf{x}$ may change their features to present as $\mathbf{x}'$ if this leads to a positive classification, for example since the utility of this change in classification exceeds the cost of the change between these vectors. We define our strategic loss in dependence of this graph and carefully motivate the proposed loss in terms of requirements and effects from previous literature. In particular, our loss models the goal of design a decision rule which is accurate as well as designed to withstand feature manipulations. There are two main motivations for discouraging such feature changes: either manipulated instances will be misclassified after the manipulation (resulting in false positives) or true positive instances are forced to mis-represent themselves in order to be classified correctly. This second consideration is also known as “social burden” [Milli et al., 2019] as individuals typically face a cost for this.

We then analyze the sample complexity of learning with this loss function. We identify sufficient conditions for proper learnability that take into account the interplay between a hypothesis class and an underlying manipulation graph. Moreover, we show that every class that has finite VC-dimension is learnable with respect to this loss by drawing a connection to results in the context of learning under adversarial perturbations [Montasser et al., 2019]. This effect may be surprising, since it presents a contrast to learning VC-classes with the sole requirement of minimizing classification error under strategic feature manipulations, which has been shown can lead to some VC-classes not being learnable [Zhang and Conitzer, 2021]. Thus, our analysis shows that balancing classification error with disincentivizing feature manipulations can reduce the complexity of the learning problem.

This first analysis, as well as many other works in the literature commonly assume, that the manipulation structure is known in terms of either a cost-of-manipulations function or as in the case of our loss, by the manipulation graph. However such knowledge of the feature manipulation capabilities is not always available, in particular not with exact accuracy. Different sub-populations may have differences in their manipulation graphs (different capabilities to manipulate their features) or a manipulation graph may be inferred from data and therefore exhibit statistical errors. We introduce a novel distance measure between manipulation graphs by drawing connections to learning bounds in transfer learning [Ben-David et al., 2010, Mansour et al., 2009] and show that the strategic loss of a learned predictor when employing a different manipulation graph can be bounded in terms of this distance measure. We show that the quality of learning outcomes under our loss function is robust to inaccuracies in the manipulation graph within this similarity measure. Such a robustness property helps in cases, in which the assumed graph might not exactly reflect agents’ responses, but is a good approximation.

We then address the problem of how the manipulation capabilities can be inferred from data. We first discuss with respect to our similarity measure from finite data that is labelled by their set of manipulation capabilities. We then discuss the more ambitious goal of learning manipulation capabilities not from explicit knowledge of capabilities, but only from the implied distribution shift, assuming beneficial manipulations will be made.

We note that in most decision-making setups where a decision rule (classifier) is learned from available data, the decision-making is not a one-time event, but rather the learned decision rule is to be repeatedly employed over a long time, and ideally updated to adapt to potential changes in the data generation. This allows us to utilize the knowledge of manipulation capabilities we can infer from distribution shifts. In this work, we focus on the scenario where such changes are (only) the result of strategic feature adaptations to the published decision rules.

We propose a novel formalization of this batch-learning setting for strategic classification. In our framework, learning and data-generation proceed in rounds. Initially, the learner receives a labeled sample S_0 from some underlying (unmanipulated) distribution. Given this sample of labeled data, the learner decides on (and publishes) a classifier h^0 . From then on, in each round t , the learner receives an unlabeled sample from the same data distribution, with the caveat that the features were strategically manipulated in response to h^{t-1} . The learner then decides on classifier h^t , based on labeled data sample S_0 and unlabeled samples $S_1, S_2, \dots, S_{t-1}$. While the learner does not have access to the underlying feature manipulation capabilities of the instances, we assume that the true manipulation structure is a member of a class of possible such structures (graphs). We show that by exploiting the observed distribution shift in this batch-learning setting it is possible to learn the optimal classifier in terms of the strategic loss even without knowing the underlying manipulation capabilities. For a wide range of combinations of *hypothesis classes* $\mathcal{H}$ and *manipulation graph classes* $\mathcal{G}$, we provide first positive results and analysis in this learning setup. More specifically, we derive bounds on sufficient sample sizes as well as the number of rounds for the learner to produce a classification rule with low loss. We focus in particular on graph classes which are totally ordered by inclusion, which captures the case in which it is unknown how manipulation costs compare to the value of being classified with the desired label. We show that in these cases batch-learning is possible if the VC-dimension of the loss-class of $(\mathcal{G} \times \mathcal{H})$ is finite. Roughly, the finiteness of the VC-dimension of the loss class, makes it possible to successfully estimate the distribution shift caused by a deployed hypothesis. The total order on the manipulation graphs allows to use this information to do a binary search on a discretized version of the hypothesis class.

We also show that for totally ordered $\mathcal{G}$ and $\mathcal{H}$ with finite VC-dimension it is possible

to successfully improperly learn $\mathcal{H}$ with respect to $\mathcal{G}$ in the robustly realizable case with only access to unmanipulated data. In order to achieve this last result, we introduce a new paradigm, called maximally robust empirical risk minimization (MRERM) and use it to recreate the compression argument from Montasser et al. [2019]. MRERM picks a hypothesis that is robust with respect to the maximal graph that allows for robust realizability of the sample, in case such a maximal graph exists. However, such a maximal graph may not exist in some finite VC classes. We use the set-theoretic concept of ultrafilters to define an extension of the hypothesis class that is guaranteed to have an MRERM for every realizable sample and has the same VC dimension as our original class.

8.1.1 Related Work

The concern that learning outcomes might be compromised when agents adapt their feature vectors in response to published classification rules was first pointed out over a decade ago [Dalvi et al., 2004, Brückner and Scheffer, 2011]. The area has received substantial interest from the research community in recent years, both in the context of adversarial robustness [Feige et al., 2015, Cullina et al., 2018, Montasser et al., 2019, 2021] and robustness to strategic feature manipulations. Hardt et al. formally introduced the setup where agents aim to improve their decision outcomes and termed it “strategic classification” [Hardt et al., 2016]. In addition to the cost of induced misclassification, previous work has pointed out that changes to the decision boundary aiming to prevent false positives, may force true positive instances to manipulate their features for retaining their positive classification. This (also undesirable) effect has been summarized under the concept of “social burden” [Milli et al., 2019, Jagadeesan et al., 2021]. It has also been shown that the cost of social burden might be disproportionately paid by underrepresented or disadvantaged sub-groups of a population [Milli et al., 2019, Hu et al., 2019]. Recent work on strategic classification has pointed out that strategic feature modification can also be a positive effect, for example when applicants respond by studying better for tests and learning specific skills [Haghtalab et al., 2020], and addressed this phenomenon through a causality lens [Miller et al., 2020, Tsirtsis and Rodriguez, 2020, Shavit et al., 2020]. Some recent works have further explored this interplay between gaming and improvement [Chen et al., 2021] and aligned incentives [Levanon and Rosenfeld, 2022].

While many previous studies in this area have taken a game theoretic perspective, some recent work has analyzed strategic classification in a PAC learning framework [Zhang and Conitzer, 2021, Sundaram et al., 2021]. Similarly, our work follows a new trend of not requiring agents to be cost-minimizing agents [Jagadeesan et al., 2021, Chen et al., 2020], as there is a potential limit of the rationality of agents [Jagadeesan et al., 2021]. In the PAC

learning setting, the consideration of irrational agents is modelled by capturing the sets of possible manipulations in the *manipulation graph* [Zhang and Conitzer, 2021], which only distinguishes between feasible and infeasible manipulations. We proposed using this notion of manipulation graph the objectives of discouraging strategic manipulations for the sake of avoiding misclassification and avoiding contributions to social burden have been jointly modelled in a loss function, the *strategic loss*. Throughout our work we use this notion of loss and frame our learning goals in terms of the strategic loss function. Similar to our work there was recently another proposed version of loss based on the manipulation graph, which only optimizes for accuracy and does not consider social burden [Shao et al., 2023]. The PAC-learnability with this loss function is different, as it requires the strategic VC-dimension to be finite, while finite VC classes can be impossible to strategically PAC-learn, even improperly [Sundaram et al., 2021].

There has been some recent work on learning with respect to unknown manipulation structures in an online setting [Dong et al., 2018]. The first results in this line of research was given in form of regret bounds for linear classifiers under the assumption that only instances of one label would manipulate [Dong et al., 2018]. Similarly, Ahmadi et al introduce a version of the perceptron algorithm which takes possible feature manipulations into account [Ahmadi et al., 2021]. They show finite mistakes bounds for their algorithm for both known and unknown cost functions under a linear separability assumptions, which is akin to our strategically robust realizability assumption. Both of these works [Dong et al., 2018, Ahmadi et al., 2021] do not require any knowledge of unmanipulated data in their setting but assume immediate label feedback for each (possibly manipulated) classified instance. Thus their results are complementary to our results in the strategic realizable case where we achieve robustness without having access to manipulated data. Furthermore, both works assume that agents are cost-minimizing, i.e., best-response. We also note that the notion of loss in those settings is slightly different, as they do not incorporate the notion of social burden into their success criterion. Similarly, the work of Shao et al. [2023] analysed learnability with respect to unknown manipulation capabilities in both online and offline settings with respect to their social-burden-agnostic, accuracy-optimizing version of the strategic loss [Shao et al., 2023]. They give learning guarantees for finite hypothesis classes in both settings for various possibilities for manipulation capabilities. The work on strategic online learning was later generalized to settings in which manipulations were modelled through manipulation graphs [Ahmadi et al., 2023]. They show lower bounds in terms of the maximal degree of the manipulation graph and upper bounds for finite hypothesis classes and finite manipulation degrees. A follow-up work later showed bounds in terms of a strategic version of the Littlestone dimension [Ahmadi et al., 2024]. Furthermore, PAC-learnability guarantees been shown with respect to an unknown unknown

perturbation structure in adversarially robust classification [Montasser et al., 2021] with the assumption of an additional oracle, which we will discuss in more detail in the next section.

Our sample complexity analysis is then hinging on techniques developed in the context of learning under adversarial perturbations, a learning scenario which has received considerable research attention in recent years [Feige et al., 2015, Cullina et al., 2018, Montasser et al., 2019, 2021]. While the learning problems are not identical, we present how strategic behaviour can be modeled as a form of “one-sided adversarial perturbation” and inheritance of resulting learning guarantees.

Finally, our batch-learning framework bears some similarities with the setting of lifelong learning [Pentina and Lampert, 2014, Pentina and Ben-David, 2015, Balcan et al., 2015, Pentina and Uner, 2016, Balcan et al., 2020a]. In lifelong learning, a learning algorithm aims to perform well and adapt to a stream of related, but not identical learning tasks. Our setup distinguishes itself from standard lifelong learning goals in that the changes in input data are actually induced by the published decision rules from the previous round, while the actual target task remains the same.

8.1.2 Overview on contributions

In Section 8.2 we review our notation and then introduce our new notion of strategic loss and motivate it. Our main contributions can be summarized as follows:

Strategic manipulation loss We propose a novel loss function for learning in the presence of strategic feature manipulations. We carefully motivate this loss by relating it to concepts of social burden and incentive compatibility (and their potential trade-offs with accuracy) in prior literature (Section 8.2).

Sample complexity analysis of PAC learning We analyze (PAC type) learnability of VC-classes with the strategic loss. We provide sufficient conditions (and examples of when they are satisfied) for learnability with a proper learner. By drawing connections and adapting results from learning under adversarial perturbations to our setup, we also show that, while proper learnability can not always be guaranteed, every VC-class is learnable under the strategic loss with an improper learner (Section 3.1.2).

Robustness to inaccurate manipulation information We investigate the impact of using an approximate manipulation graph to yield a surrogate strategic loss function

for cases where the true manipulation graph is not known or not accessible. For this, we introduce a novel similarity measure on graphs and show that if graphs are similar with respect to our notion then they yield reasonable surrogate strategic losses for each other (Theorem 8.4.2, Section 8.4).

Learning the manipulation graph from neighborhood of sample We explore the question of whether it is possible to learn a manipulation graph that yields a good surrogate strategic loss. We identify a sufficient condition for a class of graphs $\mathcal{G}$ being learnable with respect to our previously defined similarity measure for graphs (Theorem 8.5.4, Section 8.5), which in turn guaranteed the learning of a reasonable surrogate loss.

Learning manipulation capabilities from distribution shift We further explore the possibility of learning the manipulation structure for a particular hypothesis from $\mathcal{H}$ from induced distribution shifts (Lemma 8.6.5 and Observation 8.6.6). We identify sufficient conditions for when learning from two unlabelled samples is possible, if one of them was manipulated by a hypothesis (Section 8.6).

Loss-induced order of graph class and maximally robust empirical risk minimization

We consider a setting in which the prior knowledge of the manipulation structure is given by a class of candidate manipulation graphs. We consider a partial order induced by the strategic loss. We show that for graph classes that are totally ordered with respect to this order, we can show learnability for finite VC-classes in the realizable case. These results are based on the novel learning paradigm of *maximally robust empirical risk minimization*.

Batch-learning for unknown manipulation We consider a novel strategic batch-learning problem in which the manipulation graph is learned alongside the classification rule in order to achieve optimal classification (Definition 8.8.2). Importantly, we only assume prior knowledge of a graph class $\mathcal{G}$ which contains the true manipulation graph, but not exact knowledge of the true manipulation graph. We propose a formal learning protocol (Definition 8.8.1) and success criterion (Definition 8.8.2) for this setup and show that learning in this setting is possible for a wide variety of hypothesis classes $\mathcal{H}$ and graph classes $\mathcal{G}$. In Section 8.8 we present possibility results for learning under the strategic batch learning protocol for totally ordered graph classes. These results are based on learning the distribution shift between batches.

The results and writing of this chapter are based on two papers:

- "Learning Losses for Strategic Classification" (Tosca Lechner, Ruth Urner, AAAI 2022 [Lechner and Urner, 2022]): Section 8.2, Section 8.4, Section 8.5
- "Strategic Classification with Unknown User Manipulations" (Tosca Lechner, Shai Ben-David, Ruth Urner, ICML 2023 [Lechner et al., 2023b])

All proofs can be found in the proof section (Section 8.10) at the end of this Chapter.

8.2 The strategic loss

8.2.1 Basic Learning Theoretic Notions for Classification

We employ the standard setup of PAC learning for binary classification introduced in the preliminaries of this thesis. We quickly recall some notations. We let $\mathcal{X} \subseteq \mathbb{R}^d$ denote the domain and $\mathcal{Y}$ (mostly $\mathcal{Y} = \{0, 1\}$) a (binary) label space. We denote data generating process as a distribution P over $\mathcal{X} \times \mathcal{Y}$ and let $P_{\mathcal{X}}$ denote the marginal of P over $\mathcal{X}$. We use the notation $(\mathbf{x}, y) \sim P$ to indicate that $(\mathbf{x}, y)$ is a sample from distribution P and $S \sim P^n$ to indicate that set S is a sequence (for example a training or test data set) of n i.i.d. samples from P . Further, we use notation $\eta_P(\mathbf{x}) = \mathbb{P}_{(\mathbf{x}, y) \sim P}[y = 1 \mid \mathbf{x}]$ to denote the *regression* or *conditional labeling function* of P . We say that the distribution has *deterministic labels* if $\eta_P(\mathbf{x}) \in \{0, 1\}$ for all $\mathbf{x} \in \mathcal{X}$.

A *classifier* or *hypothesis* is a function $h : \mathcal{X} \rightarrow \mathcal{Y}$. A classifier h can naturally be viewed a subset of $\mathcal{X} \times \mathcal{Y}$, namely $h = \{(\mathbf{x}, y) \in \mathcal{X} \times \mathcal{Y} \mid \mathbf{x} \in \mathcal{X}, y = h(\mathbf{x})\}$. We let $\mathcal{F}$ denote the set of all Borel measurable functions from $\mathcal{X}$ to $\mathcal{Y}$ (or all functions in case of a countable domain). A *hypothesis class* is a subset of $\mathcal{F}$, often denoted by $\mathcal{H} \subseteq \mathcal{F}$. For a loss function $\ell : \mathcal{F} \times \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$ we denote the expected loss for a distribution P as $\mathcal{L}_P$ and the empirical loss for a sample S as $\mathcal{L}_S$.

8.2.2 Strategic Classification

Learning objectives in prior work

The possibilities for strategic manipulations of a feature vector are often modeled in terms of a cost function $\text{cost} : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}_0^+$, so that $\text{cost}(\mathbf{x}, \mathbf{x}')$ indicates how expensive it is for an individual with feature vector $\mathbf{x}$ to present as $\mathbf{x}'$. A natural minimal assumption a

cost function should satisfy is $\text{cost}(\mathbf{x}, \mathbf{x}) = 0$ for all feature vectors $\mathbf{x}$. It is then typically assumed that instances best-respond to a published classifier, in that the individual $\mathbf{x}$ would choose to pay the cost of presenting as $\mathbf{x}'$ as long as the cost doesn't exceed the utility that would be gained from the difference in classification outcome. Assuming the benefit of individual $\mathbf{x}$ receiving classification 1 over classification 0 is γ , the manipulation would happen if $\text{cost}(\mathbf{x}, \mathbf{x}') \leq \gamma$ and $h(\mathbf{x}) = 0$ while $h(\mathbf{x}') = 1$ for a given classifier. That is, we can define the best response of an individual with feature vector $\mathbf{x}$ facing classifier h as

$$\text{br}(\mathbf{x}, h) = \arg \max_{\mathbf{x}' \in \mathcal{X}} [\gamma \cdot h(\mathbf{x}') - \text{cost}(\mathbf{x}, \mathbf{x}')],$$

with ties broken arbitrarily, and assuming that, if the original feature vector $\mathbf{x}$ is among those maximizing the above, then the individual would choose to maintain the original features. An often assumed learning goal is then *performative optimality* [Perdomo et al., 2020, Jagadeesan et al., 2021], which stipulates that a learner should aim to maximize accuracy on the distribution it induces via the agent responses. That is, this objective can be phrased as minimizing

$$\mathbb{E}_{(\mathbf{x}, y) \sim P} \mathbf{1}[h(\text{br}(\mathbf{x}, h)) \neq y]$$

An alternative view on this setup, if the agent responses are deterministic, is to view the above as minimizing the binary loss of the *effective hypothesis* $\hat{h} : \mathcal{X} \rightarrow \{0, 1\}$ that is induced by h and the agents' best responses $\text{br}(\cdot, \cdot)$ [Zhang and Conitzer, 2021], defined as

$$\hat{h}(\mathbf{x}) = h(\text{br}(\mathbf{x}, h)). \quad (8.1)$$

The goal of performative optimality has been combined with the notion of *social burden* that is induced by a classifier [Milli et al., 2019, Jagadeesan et al., 2021]. This notion reflects that it is undesirable for a (truly) positive instance to be forced to manipulate its features to obtain a (rightfully) positive classification. This is modeled by considering the *burden* on a positive individual to be the cost that is incurred by reaching for a positive classification and the *social burden* incurred by a classifier to be the expectation with respect to the data-generating process over these costs:

$$\text{brd}_P(h) = \mathbb{E}_{(\mathbf{x}, y) \sim P} \left[\min_{\mathbf{x}' \in \mathcal{X}} \{ \text{cost}(\mathbf{x}, \mathbf{x}') \mid h(\mathbf{x}') = 1 \} \mid y = 1 \right]$$

It has been shown that optimizing for performative optimality (under the assumption of deterministic best-responses) also incurs maximal social burden [Jagadeesan et al., 2021].

A new loss function for learning under strategic manipulations

Arguably, to seek performative optimality (or minimize the binary loss over the effective hypothesis class) the cost function as well as the value γ (or function $\gamma : \mathcal{X} \rightarrow \mathbb{R}$) of positive classification needs to be known (or at least approximately known). We assume that the value γ is the same for every individual and that it is positive. Thus we assume that $y = 1$ is the universally preferred label. To take best responses into account, a learner needs to know what these best responses may look like. In that case, we may ignore the details of the cost function and value γ , and simply represent the collection of *plausible manipulations* as a directed graph structure $g = (\mathcal{X}, E_g)$ over the feature space $\mathcal{X}$ [Zhang and Conitzer, 2021]. The edge-set E_g consists of all pairs $(\mathbf{x}, \mathbf{x}')$ with $\text{cost}(\mathbf{x}, \mathbf{x}') \leq \gamma$, and we will also use the notation $\mathbf{x} \rightarrow \mathbf{x}'$ for $(\mathbf{x}, \mathbf{x}') \in E_g$, and write $g = (\mathcal{X}, E_g)$. We note that this formalism is valid for both countable (discrete) and uncountable domains.

In short, we model the set of admissible feature manipulations as a *manipulation graph* $g = (\mathcal{X}, E_g)$, where a manipulation from x to x' is admissible if and only if the (directed) edge (x, x') exists in E_g . We will denote the neighborhood set of a point $x \in \mathcal{X}$ according to graph g by $B_g(x) = \{x' \in \mathcal{X} : (x, x') \in E_g\}$. We will denote the true manipulation graph by $g^\rightarrow$. In some parts of this chapter, we will not assume this graph to be known during the learning process. Rather, we assume the learner has prior knowledge of some graph class $\mathcal{G}$ such that $g^\rightarrow \in \mathcal{G}$. The class of all manipulation graphs will be denoted by $\mathcal{G}_{\text{all}}$.

Now for given a fixed *manipulation graph* $g = (\mathcal{X}, E_g)$ we design a loss function for classification in the presence of strategic manipulation that reflects both classification errors and the goal of disincentivizing manipulated features as much as possible. Our proposed loss function below models that it is undesirable for a classifier to assign $h(\mathbf{x}) = 0$ and $h(\mathbf{x}') = 1$ if feature vector $\mathbf{x}$ can present as $\mathbf{x}'$. This is independent of a true label y (e.g. if $(\mathbf{x}, y)$ is sampled from the data generating process). If the label $y = 0$ is not positive, the point gets misclassified when $\mathbf{x}$ presents as $\mathbf{x}'$. On the other hand, if the true label is 1, then either a true positive instance is forced to manipulate their features to obtain a rightly positive outcome (and this contributes to the social burden), or, if the choice is to not manipulate the features, the instance will be misclassified (prior work has also considered models where true positive instance are “honest” and will not manipulate their features [Dong et al., 2018]).

Here, we propose to incorporate both misclassification and contributions to social burden into a single loss function that a learner may aim to minimize.

Definition 8.2.1 (Strategic Loss). *For a given manipulation graph g , the strategic loss*

$\ell^g : \mathcal{F} \times \mathcal{X} \times \mathcal{Y} \rightarrow \{0, 1\}$ is defined by

$$\ell^g(h, x, y) = \begin{cases} 1 & \text{if } h(x) \neq y \\ 1 & \text{if } h(x) = 0 \text{ and} \\ & B_g(x) \cap h^{-1}(1) \neq \emptyset \\ 0 & \text{otherwise.} \end{cases}$$

Note that the first two cases are not mutually exclusive. The above loss function discretizes the social burden by assigning a loss of 1 whenever a positive individual is required to manipulate features. As for the standard classification loss, the above point-wise definition of a loss function allows to define the *true strategic loss* $\mathcal{L}_P^g(h)$ and *empirical strategic loss* $\mathcal{L}_S^g(h)$ of a classifier with respect to a distribution P or a data sample S .

The following loss captures the second condition only:

Definition 8.2.2 (Manipulation Loss). *The manipulation loss $\ell^{\text{mani}} : \mathcal{G}_{\text{all}} \times \mathcal{F} \times \mathcal{X} \rightarrow \{0, 1\}$ is defined by*

$$\ell^{\text{mani}}(g, h, x) = \begin{cases} 1 & \text{if } h(x) = 0 \text{ and} \\ & B_g(x) \cap h^{-1}(1) \neq \emptyset \\ 0 & \text{otherwise} \end{cases}$$

For a fixed manipulation graph g , we will also call the manipulation loss $\ell^{\text{mani}}(g, \cdot, \cdot)$ the "strategic component loss". Moreover, $\ell^g(h, x, y) \leq \ell^{\text{mani}}(g, h, x) + \ell^{0/1}(h, x, y)$.

Short notation for known manipulation graph In the following sections we will analyse learning only with respect to a fixed manipulation graph $g = (\mathcal{X}, E_g)$, assuming that this graph is indeed the true manipulation graph $g^\rightarrow$. In this cases we will sometimes abbreviate $g^\rightarrow$ as $\rightarrow$. As a result we will abbreviate ℓ^g , $\mathcal{L}_S^g$ and $\mathcal{L}_P^g$ as $\ell^\rightarrow$, $\mathcal{L}_S^\rightarrow$ and $\mathcal{L}_P^\rightarrow$ respectively. The resulting notation for the strategic component loss will be $\ell^{\rightarrow, \perp}$, $\mathcal{L}_S^{\rightarrow, \perp}$ and $\mathcal{L}_P^{\rightarrow, \perp}$. Furthermore, we will sometimes use the short notation $x \rightarrow x'$ to denote that $(x, x') \in E_{g^\rightarrow}$.

8.2.3 Comparison with Alternative Formalisms for Strategic Classification

To motivate our proposed loss, we here discuss several scenarios where, we'd argue, minimizing the strategic loss leads to a more desirable learning outcome than learning with a

binary loss, while taking strategic manipulations into account. As discussed above, a common approach to modeling classification in a setting where strategic manipulations may occur is to assume that all agents will best-respond to a published classifier. That is, if $h(\mathbf{x}) = 0$, $h(\mathbf{x}') = 1$ and $\mathbf{x} \rightarrow \mathbf{x}'$, then the agent with initial feature vector $\mathbf{x}$ will *effectively* receive classification 1. A natural modeling is then to consider the effective hypothesis $\hat{h}$ induced h (see Equation 8.1) and aim to minimize the classification error with the *effective class* $\hat{\mathcal{H}} = \{f \mid f = \hat{h} \text{ for some } h \in \mathcal{H}\}$ [Zhang and Conitzer, 2021]. However it has been shown, that the VC-dimension of $\hat{\mathcal{H}}$ may be arbitrarily larger than the VC-dimension of $\mathcal{H}$, and may even become infinite [Zhang and Conitzer, 2021]. When learning this effective class $\hat{\mathcal{H}}$ with respect to the binary loss (which corresponds to aiming for performative optimality), this will imply that the class is not learnable. By contrast, we will show below that any class of finite VC-dimension remains learnable with respect to the strategic loss (Theorem 8.3.8).

It has also been shown that the negative effects in terms of sample complexity of considering the effective hypothesis class can be avoided by considering only *incentive-compatible* hypotheses in $\mathcal{H}$, that is outputting only such hypotheses that will not induce any feature manipulations in response to the published classifier [Zhang and Conitzer, 2021]. While this avoids the growths in terms of VC-dimension, it may prohibitively increase the approximation error of the resulting (pruned) class as we show in the example below. We would argue that this illustrates how low sample complexity, in itself, is not a sufficient criterion for learning success.

Example 8.2.3. *Consider $\mathcal{X} = \mathbb{N}$ and a manipulation graph that includes edges $n \rightarrow n+1$ and $n \rightarrow n-1$ for all $n \in \mathbb{N}$. This is a reasonable structure, considering that the cost of moving the (one-dimensional) feature by 1 is worth a positive classification outcome. However, the only two hypotheses that are incentive compatible in this case are the two constant functions $h_0 : \mathcal{X} \rightarrow \{0\}$ and $h_1 : \mathcal{X} \rightarrow \{1\}$. Thus, requiring incentive compatibility forces the learner to assign all points in the space with the same label. This class, in fact, has low sample complexity. However, arguably, restricting the learning to such a degree (and suffering the resulting classification error, which will be close to 0.5 for distributions with balanced classes), is, in most cases not a reasonable price to pay for dis-incentivising feature manipulations.*

The following example illustrates how our loss function can be viewed as incorporating the notion of social burden directly into the loss.

Example 8.2.4. *Let's again consider a domain $\mathcal{X} = \mathbb{N}$ and a manipulation graph $\mathcal{M}$ with edges $n \rightarrow n+1$ for all $n \in \mathbb{N}$. We consider distributions that have support*

$\{(1, 0), (2, 0), (3, 1), (4, 1)\}$, thus only these four points have positive probability mass and a hypothesis class of thresholds $\mathcal{H} = \{h_a \mid a \in \mathbb{R}\}$, with $h_a(\mathbf{x}) = \mathbf{1}[\mathbf{x} \geq a]$. The true labeling on these distributions is $\eta(\mathbf{x}) = h_{2.5}(\mathbf{x})$. On all distributions, where all four points have positive mass the performatively optimal hypothesis (or effective hypothesis of minimal binary loss) however is $h_{3.5}$. The social burden incurred then is $\text{brd}_P(h_{3.5}) = P((3, 1)) \cdot \text{cost}(3, 4)$. It is important to note that the performative optimality of $h_{3.5}$ is independent of the distribution P over the points. A learner that minimizes the strategic loss, on the other hand, will take the distribution P into account and output $h_{2.5}$ if $P((2, 0)) < P((3, 1))$, while outputting $h_{3.5}$ if $P((2, 0)) > P((3, 1))$. If the difference in mass of these points (or the margin areas in a more general setting) is significant, then minimizing the strategic loss will opt for allowing a small amount of manipulation in turn for outputting a correct classification rule in case $P((2, 0)) \ll P((3, 1))$; and it will opt for changing the classification rule, accept a small amount of social burden in exchange for preventing a large amount of manipulations and resulting classification errors, in case $P((2, 0)) \gg P((3, 1))$. We would argue that this reflects a desirable learning outcome.

8.3 Learnability with the Strategic Loss

8.3.1 Warm up: Loss Classes and Learnability

It is well known that a class $\mathcal{H}$ is learnable (with respect to the set of all distributions) if the *loss class* induced by a 0/1-valued loss function ℓ has finite VC-dimension. In the case of the binary classification loss, this is in fact a characterization for learnability (and the VC-dimension of the loss class is identical to the VC-dimension of the hypothesis class $\mathcal{H}$). In general, bounded VC-dimension of the loss class is a sufficient condition for learnability (the VC-dimension provides an upper bound on the sample complexity), but it is not a necessary condition (it doesn't, in general, yield a lower bound on the sample complexity of learning a class $\mathcal{H}$ with respect to some loss ℓ). We start by reviewing these notions for the classification loss and then take a closer look at the loss class induced by the strategic loss.

Let ℓ be a loss function and h be a classifier. We define the *loss set* $h_\ell \subseteq \mathcal{X} \times \mathcal{Y}$ as the set of all labeled instances $(\mathbf{x}, y)$ on which h suffers loss 1. The *loss class* $\mathcal{H}_\ell$ is the collection of all loss sets (in the literature, the loss class is often described as the function class of indicator functions over these sets). In the case of binary classification loss $\ell^{0/1}$, the loss set of a classifier h is exactly the complement of h in $\mathcal{X} \times \mathcal{Y}$. That is, in this case the loss set of h is also a binary function over the domain $\mathcal{X}$ (namely the function

$\mathbf{x} \mapsto |h(\mathbf{x}) - 1|$). For the strategic loss on the other hand, the loss set of a classifier h is not a function, since it can contain both $(\mathbf{x}, 0)$ and $(\mathbf{x}, 1)$ for some points $\mathbf{x} \in \mathcal{X}$, namely if $h(\mathbf{x}) = 0$ and there exists an $\mathbf{x}'$ with $\mathbf{x} \rightarrow \mathbf{x}'$ and $h(\mathbf{x}') = 1$. For a class $\mathcal{H}$ we let $\mathcal{H}_{\ell_{0/1}}$ denote the loss class with respect to the binary loss and $\mathcal{H}_{\ell \rightarrow}$ the loss class with respect to the strategic loss.

Definition 8.3.1. Let $\mathcal{Z}$ be some set and $\mathcal{U} \subseteq 2^{\mathcal{Z}}$ be a collection of subsets of $\mathcal{Z}$. We say that a set $S \subseteq \mathcal{Z}$ is shattered by $\mathcal{U}$ if

$$\{U \cap S \mid U \in \mathcal{U}\} = 2^S,$$

that is, every subset of S can be obtained by intersecting S with some set U from the collection $\mathcal{U}$. The VC-dimension of $\mathcal{U}$ is the largest size of a set that is shattered by $\mathcal{U}$ (or ∞ if $\mathcal{U}$ can shatter arbitrarily large sets).

It is easy to verify that for the binary loss, the VC-dimension of $\mathcal{H}$ as a collection of subsets of $\mathcal{X} \times \mathcal{Y}$ is identical with the VC-dimension of $\mathcal{H}_{\ell_{0/1}}$ (and this also coincides with the VC-dimension of $\mathcal{H}$ as a binary function class [Shalev-Shwartz and Ben-David, 2014a]; VC-dimension is often defined for binary functions rather than for collection of subsets, however, this is limiting for cases where the loss class is not a class of functions).

We now show that the VC-dimension of a class $\mathcal{H}$ and its loss class with respect to the strategic loss can have an arbitrarily large difference. Similar results have been shown for the binary loss class of the effective class $\hat{\mathcal{H}}$ induced by a manipulation graph [Zhang and Conitzer, 2021]. However the binary loss class of $\hat{\mathcal{H}}$ is different from the strategic loss class of $\mathcal{H}$ and, as we will see, the implications for learnability are also different.

Observation 8.3.2. For any $d \in \mathbb{N} \cup \{\infty\}$ there exists a class $\mathcal{H}$ and a manipulation graph $g^{\rightarrow} = (\mathcal{X}, E_{g^{\rightarrow}})$ with $\text{VC}(\mathcal{H}) = 1$ and $\text{VC}(\mathcal{H}_{\ell \rightarrow}) \geq d$.

On the other hand, we prove that the VC-dimension of the strategic loss class $\mathcal{H}_{\ell \rightarrow}$ is always at least as large as the VC-dimension of the original class.

Observation 8.3.3. For any hypothesis class $\mathcal{H}$ and any manipulation graph $g^{\rightarrow} = (\mathcal{X}, E_{g^{\rightarrow}})$, we have $\text{VC}(\mathcal{H}) \leq \text{VC}(\mathcal{H}_{\ell \rightarrow})$.

Standard VC-theory tells us that, for the binary classification loss, the sample complexity of learning a hypothesis class is determined by the VC-dimension of the class (and thus the VC-dimension of the loss class since these are identical for the binary loss). Any learner that acts according to the ERM (Empirical Risk Minimization) principle is a successful learner with respect to the binary loss for classes of bounded VC-dimension d . We here

briefly recap the underpinnings of this result. For the realizable case, a sample $S \sim P^n$ of size at least $\tilde{O}(d \log(1/\delta)/\varepsilon)$ is an ε -net for the loss class (with probability at least $1 - \delta$) [Haussler and Welzl, 1987], thus, for every hypothesis that has loss at least ε , there would be a sample point indicating that. Choosing a hypothesis of zero empirical loss, then guarantees true loss bounded by ε . In the agnostic case, a sample of size $O(d \log(1/\delta)/\varepsilon^2)$ is an ε -approximation for the loss class, that is, we have $|\mathcal{L}_P(h) - \mathcal{L}_S(h)| \leq \varepsilon$ for every $h \in \mathcal{H}$ with high probability, which in turn implies that any ERM learner is a successful agnostic PAC learner. For the binary loss there are complementing lower bounds for the sample complexity of learning VC-classes. However, unlike for the binary loss, the VC-dimension of the strategic loss class does not imply a lower bound on the sample complexity (as we see in Theorem 8.3.4).

For further details we refer the reader to [Shalev-Shwartz and Ben-David, 2014a]. In the case of general loss classes with values in $\{0, 1\}$, the VC-dimension does not characterize learnability. In particular, we next show that the VC-dimension of the strategic loss class does not imply a lower bound on the sample complexity.

Theorem 8.3.4. *For every $d \in \mathbb{N} \cup \{\infty\}$, there exists a hypothesis class $\mathcal{H}$ with $\text{VC}(\mathcal{H}_{\ell \rightarrow}) = d$ that is learnable with sample complexity $O(\log(1/\delta)/\varepsilon)$ in the realizable case.*

8.3.2 Sufficient Conditions for Strategic Loss Learnability

In the previous section, we have seen that the loss class having a finite VC-dimension is a sufficient (but not necessary) condition for learnability with respect to the strategic loss. We have also seen that the VC-dimension of $\mathcal{H}_{\ell \rightarrow}$ can be arbitrarily larger than the VC-dimension of $\mathcal{H}$. To start exploring what determines learnability under the strategic loss, we provide a sufficient condition for a class to be properly learnable with respect to the strategic loss.

Note that for a hypothesis h , the strategic loss set $h_{\ell \rightarrow}$ can be decomposed into the loss set of h with respect to the binary loss and the component that comes from the strategic manipulations. Formally, we can define the *strategic component loss*.

Definition 8.3.5. *We let the strategic component loss with respect to manipulation graph $\rightarrow$ be defined as*

$$\ell^{\rightarrow, \perp}(h, \mathbf{x}) = \mathbf{1}[h(\mathbf{x}) = 0 \wedge \exists \mathbf{x}' : \mathbf{x} \rightarrow \mathbf{x}', h(\mathbf{x}') = 1]$$

We note that $\ell^{\rightarrow}(h, \mathbf{x}, y) \leq \ell^{0/1}(h, \mathbf{x}, y) + \ell^{\rightarrow, \perp}(h, \mathbf{x})$. We will denote the true strategic component loss with respect to marginal distribution $P_{\mathcal{X}}$ as $\mathcal{L}_{P_{\mathcal{X}}}^{\rightarrow, \perp}$.

For the loss sets, we then get

$$h_{\ell^{0/1}} = \{(\mathbf{x}, y) \in \mathcal{X} \times \mathcal{Y} \mid h(\mathbf{x}) \neq y\},$$

and

$$h_{\ell^{\rightarrow, \perp}} = \{(\mathbf{x}, y) \in \mathcal{X} \times \mathcal{Y} \mid h(\mathbf{x}) = 0 \\ \wedge \exists \mathbf{x}' : \mathbf{x} \rightarrow \mathbf{x}', h(\mathbf{x}') = 1\}.$$

This implies

$$h_{\ell^{\rightarrow}} = h_{\ell^{0/1}} \cup h_{\ell^{\rightarrow, \perp}}$$

for all classifiers $h \in \mathcal{F}$, and thereby

$$\mathcal{H}_{\ell^{\rightarrow}} = \{h_{\ell^{0/1}} \cup h_{\ell^{\rightarrow, \perp}} \mid h \in \mathcal{H}\}.$$

By standard counting arguments on the VC-dimension of such unions (see, for example Chapter 6 of [Shalev-Shwartz and Ben-David, 2014a] and exercises in that chapter), it can be shown this decomposition implies that $\text{VC}(\mathcal{H}_{\ell^{\rightarrow}}) \leq d \log d$ for $d = \text{VC}(\mathcal{H}_{\ell^{0/1}}) + \text{VC}(\mathcal{H}_{\ell^{\rightarrow, \perp}}) = \text{VC}(\mathcal{H}) + \text{VC}(\mathcal{H}_{\ell^{\rightarrow, \perp}})$. Thus, if both the class $\mathcal{H}$ itself and the class of strategic components have finite VC-dimension, then $\mathcal{H}$ is properly learnable by any learner that is an ERM for the strategic loss:

Theorem 8.3.6. *Let $\mathcal{H}$ be a hypothesis class with finite $\text{VC}(\mathcal{H}) + \text{VC}(\mathcal{H}_{\ell^{\rightarrow, \perp}}) = d < \infty$. Then $\mathcal{H}$ is properly PAC learnable with respect to the strategic loss (both in the realizable and the agnostic case).*

Whether the class of strategic components has finite VC-dimension intrinsically depends on the interplay between the hypothesis class $\mathcal{H}$ and the graph structure of the manipulation graph. In Observation 8.3.2, we have seen that the graph structure can yield the strategic component sets to have much larger complexity than the original class. To illustrate the conditions in Theorem 8.3.6, we here provide a few natural examples, where the VC-dimension of the strategic components can be bounded:

- Example 8.3.7.**
1. $\mathcal{M} = (\mathcal{X}, \rightarrow)$ being complete, implies $\text{VC}(\mathcal{H}_{\ell^{\rightarrow, \perp}}) = \text{VC}(\mathcal{H})$.
 2. If $\mathcal{M}$ corresponds to a partial order over $\mathcal{X}$ (that is, in particular $\mathcal{M}$ is acyclic) and the class $\mathcal{H}$ is a subset of complements of initial segments in $\mathcal{M}$ (that is if, for some $h \in \mathcal{H}$ and $\mathbf{x} \in \mathcal{X}$ we have $h(\mathbf{x}) = 0$, then we also have $h(\mathbf{x}') = 0$ for all $\mathbf{x}'$ that precede $\mathbf{x}$ in the partial order), then the VC-dimension of $\mathcal{H}_{\ell^{\rightarrow, \perp}}$ is bounded by the size of a largest antichain in $\mathcal{M}$.

3. If $\mathcal{X} = \mathbb{R}^d$, $\mathcal{H}$ consists of linear classifiers and the plausible manipulations $\mathbf{x} \rightarrow \mathbf{x}'$ are determined by $\|\mathbf{x} - \mathbf{x}'\|_p \leq r$ two points being close in some standard norm, then the VC-dimension of $\mathcal{H}_{\ell \rightarrow, \perp}$ is bounded by $2d + 2$.
4. If $\mathcal{X} = \mathbb{R}^d$, $\mathcal{H}$ consists of linear classifiers and we have plausible manipulations $\mathbf{x} \rightarrow \mathbf{x}'$ if and only if $\mathbf{x}$ and $\mathbf{x}'$ differ on only one coordinate. Then the VC-dimension of the resulting class of strategic components is $d + 1$ (since for every h , for every point $\mathbf{x}$ such that $h(\mathbf{x}) = 0$, there is a $\mathbf{x}'$ such that $\mathbf{x} \rightarrow \mathbf{x}'$ is an edge and $h(\mathbf{x}') = 1$. Thus, for every half-space h , sets of zeros of h that are connected to a 1 is exactly $\{\mathbf{x} : h(\mathbf{x}) = 0\}$, and therefore $\text{VC}(\mathcal{H}_{\ell \rightarrow, \perp}) = \text{VC}(\mathcal{H})$).

Theorem 8.3.6 provides a strong sufficient condition under which any empirical risk minimizer for the strategic loss is a successful agnostic learner for a class of finite VC-dimension. We believe, in many natural situations the conditions in that theorem will hold, and analyzing in more detail which graph structure, combinations of graphs structures and hypothesis classes or classes of cost function lead to the strategic component sets having finite VC-dimension is an intriguing direction for further research.

We close this section with two results, both stated in Theorem 8.3.8, on the learnability under the strategic loss in the general case where the VC-dimension of the strategic component sets may be infinite. First, there are classes and manipulation graphs for which no proper learner is (PAC-) successful, even in the realizable case. Second, for any class of finite VC-dimension and any manipulation graph, there exists an improper PAC learner. These results follow by drawing a connection from learning under the strategic loss to learning under an adversarial loss [Montasser et al., 2019]. In the general adversarial loss setup, every domain instance $\mathbf{x}$ is assigned a set of potential perturbation $\mathcal{U}(\mathbf{x})$, and the adversarial loss of a hypothesis h is then defined as

$$\ell^{\mathcal{U}}(h, \mathbf{x}, y) = \mathbf{1}[\exists \mathbf{x}' \in \mathcal{U}(\mathbf{x}) : h(\mathbf{x}') \neq y].$$

The strategic loss can be viewed as a one-sided version of the adversarial loss, where the perturbation sets differ conditional on the label of a point, and where $\mathcal{U}(\mathbf{x}, 1) = \{\mathbf{x}\}$, while $\mathcal{U}(\mathbf{x}, 0) = \{\mathbf{x}' \in \mathcal{X} \mid \mathbf{x} \rightarrow \mathbf{x}'\}$. The following results on learnability with the strategic loss then follow by slight modifications of the corresponding proofs for learning under adversarial loss.

Theorem 8.3.8. (Adaptation of Theorem 1 and Theorem 4 in [Montasser et al., 2019])

There exists a hypothesis class $\mathcal{H}$ with $\text{VC}(\mathcal{H}) = 1$ that is not learnable with respect to the strategic loss by any proper learner $\mathcal{A}$ for $\mathcal{H}$ even in the realizable case. On the other hand, every class $\mathcal{H}$ of finite VC-dimension is learnable (by some improper learner).

8.4 Strategic loss with respect to an approximate manipulation graph

In many situations one might not have direct access to the true manipulation graph $g^\rightarrow = (V, E_{g^\rightarrow})$, but only to some approximate graph $g' = (V, E_{g'})$. In this section, we will investigate how this change of manipulation graph impacts the corresponding loss function. We define a criterion for measuring the similarity of graphs with respect to hypothesis class $\mathcal{H}$ and show that similar graphs will yield similar strategic losses. That is, we show an upper bound on the true strategic loss of a hypothesis h (i.e., strategic loss with respect to the true manipulation graph) in terms of the graph similarity and the surrogate strategic loss of h (i.e., the strategic loss with respect to the approximate graph). We will use $\mathbf{x} \rightsquigarrow \mathbf{x}'$ to denote $(\mathbf{x}, \mathbf{x}') \in E_{g'}$. As the set of vertices V is always equal to $\mathcal{X}$ in our setting, the graphs $g^\rightarrow$ and g' are uniquely defined by $\rightarrow$ and $\rightsquigarrow$ respectively. We will therefore use $\rightarrow$ and $g^\rightarrow$, as well as $\rightsquigarrow$ and g' interchangeably.

We now define the distance between graphs with respect to a hypothesis class $\mathcal{H}$ by the impact a change of manipulation graph has on the strategic component loss of elements of $\mathcal{H}$. This definition and its later use are inspired by works on domain adaptation [Ben-David et al., 2010, Mansour et al., 2009].

Definition 8.4.1. *For two manipulation graphs, given by $\rightarrow$ and $\rightsquigarrow$, we let their $\mathcal{H}$ - $P_{\mathcal{X}}$ -distance be defined as*

$$d_{\mathcal{H}, P_{\mathcal{X}}}(\rightarrow, \rightsquigarrow) = \sup_{h \in \mathcal{H}} \mathbb{E}_{\mathbf{x} \sim P_{\mathcal{X}}} [|\ell^{\rightarrow, \perp}(h, \mathbf{x}) - \ell^{\rightsquigarrow, \perp}(h, \mathbf{x})|]$$

We will now bound the strategic manipulation loss $\mathcal{L}_P^g(h)$ with respect to the true graph $\rightarrow$ in terms of the strategic manipulation loss $\mathcal{L}_P^{\rightsquigarrow}(h)$ with respect to the approximate graph $\rightsquigarrow$ and the $\mathcal{H}$ - $P_{\mathcal{X}}$ -distance between $\rightarrow$ and $\rightsquigarrow$.

Theorem 8.4.2. *Let $\mathcal{H}$ be any hypothesis class and $\rightarrow, \rightsquigarrow$ two manipulation graphs. Then for any distribution P over $\mathcal{X} \times \mathcal{Y}$ and any $h \in \mathcal{H}$ we have*

$$\begin{aligned} \mathcal{L}_P^g(h) &\leq \mathcal{L}_P^{0/1}(h) + \mathcal{L}_P^{\rightsquigarrow, \perp}(h) + d_{\mathcal{H}, P_{\mathcal{X}}}(\rightarrow, \rightsquigarrow) \\ &\leq 2\mathcal{L}_P^{\rightsquigarrow}(h) + d_{\mathcal{H}, P_{\mathcal{X}}}(\rightarrow, \rightsquigarrow). \end{aligned}$$

Furthermore, by rearranging the result, we get

$$\frac{1}{2}\mathcal{L}_P^{\rightsquigarrow}(h) - d_{\mathcal{H}, P_{\mathcal{X}}}(\rightarrow, \rightsquigarrow) \leq \mathcal{L}_P^g(h).$$

We note that the expression $d_{H,P_{\mathcal{X}}}(\rightarrow, \rightsquigarrow)$ is independent of the labeling and can therefore be estimated using data without any label information. Furthermore, we have seen that small $d_{H,P_{\mathcal{X}}}(\rightarrow, \rightsquigarrow)$ tightens the upper as well as the lower bound on $\mathcal{L}_P^g(h)$. Therefore, $d_{H,P_{\mathcal{X}}}(\rightarrow, \rightsquigarrow)$ is a suitable distance measure for approximating the structure of the manipulation graph. In the following section, we will explore learning $\rightsquigarrow$ with low $d_{H,P_{\mathcal{X}}}(\rightarrow, \rightsquigarrow)$ from finite samples.

8.5 Learning a manipulation graph from neighborhoods of sample points

In the last section, we have assumed to be given an approximate manipulation graph which we can use to learn a classifier with low strategic loss. We now want to go one step further and pose the goal of learning a manipulation graph $\rightsquigarrow$ from a predefined class of graphs $\mathcal{G}$ such that $\ell^{\rightsquigarrow, \perp}$ serves as a good strategic surrogate loss for $\ell^{\rightarrow, \perp}$. From Theorem 8.4.2 we already know that $\ell^{\rightsquigarrow, \perp}$ is a good surrogate loss for $\ell^{\rightarrow, \perp}$ if $d_{H,P_{\mathcal{X}}}(\rightarrow, \rightsquigarrow)$ is small. This section will thus focus on learning an approximate manipulation graph $\rightsquigarrow \in \mathcal{G}$ with small $d_{H,P_{\mathcal{X}}}(\rightarrow, \rightsquigarrow)$.

In order to further specify our learning problem, we will now describe what the input of such a learning procedure will look like. For a manipulation graph $\rightarrow$, let $B_{\rightarrow} : \mathcal{X} \rightarrow 2^{\mathcal{X}}$ be the function that maps an instance $\mathbf{x}$ to its set of children, i.e., $B_{\rightarrow}(\mathbf{x}) = \{\mathbf{x}' \in \mathcal{X} : \mathbf{x} \rightarrow \mathbf{x}'\}$. We note that a manipulation graph $\rightarrow$ is uniquely defined by $B_{\rightarrow}$. Thus we will sometimes use $B_{\rightarrow}$ and $\rightarrow$ interchangeably. The input to our learning procedure will be of the form of samples $S = \{(\mathbf{x}_1, B_{\rightarrow}(\mathbf{x}_1)), \dots, (\mathbf{x}_n, B_{\rightarrow}(\mathbf{x}_n))\}$ from the true manipulation graph $\rightarrow$.

As a next step in formulating our learning problem, we will need to define a loss function. As stated above, our goal is to learn $\rightsquigarrow \in \mathcal{G}$ with small $d_{H,P_{\mathcal{X}}}(\rightarrow, \rightsquigarrow)$. As the definition of $d_{H,P_{\mathcal{X}}}(\rightarrow, \rightsquigarrow)$ contains a supremum over all $h \in \mathcal{H}$, we cannot use it as a loss directly (as a loss needs to be defined point-wise). However, we can formulate a loss that is closely related and will serve to guarantee low $d_{H,P_{\mathcal{X}}}(\rightarrow, \rightsquigarrow)$. Let the *graph loss* for a manipulation graph $\rightsquigarrow$, a domain point x , a manipulation set $B \subset \mathcal{X}$ and a hypothesis h as

$$\ell^{\text{graph}}(h, \rightsquigarrow, \mathbf{x}, B) = \begin{cases} 1 & \text{if } h(\mathbf{x}) = 0 \wedge \exists \mathbf{x}' \in B : h(\mathbf{x}') = 1 \\ & \wedge \forall \mathbf{x}'' : \mathbf{x} \rightsquigarrow \mathbf{x}'' \text{ implies } h(\mathbf{x}'') = 0 \\ 1 & \text{if } h(\mathbf{x}) = 0 \wedge \forall \mathbf{x}' \in B : h(\mathbf{x}') = 0 \\ & \wedge \exists \mathbf{x}'' : \mathbf{x} \rightsquigarrow \mathbf{x}'' \text{ and } h(\mathbf{x}) = 1 \\ 0 & \text{otherwise.} \end{cases}$$

This loss is indeed closely related to the $\mathcal{H}$ - $P_{\mathcal{X}}$ -distance as $\ell^{\text{graph}}(h, \rightsquigarrow, \mathbf{x}, B_{\rightarrow}(\mathbf{x})) = |\ell^{\rightarrow, \perp}(h, \mathbf{x}) - \ell^{\rightsquigarrow, \perp}(h, \mathbf{x})|$.

The *true graph loss* with respect to some marginal $P_{\mathcal{X}}$ and true manipulation graph $\rightarrow$ is then defined by

$$\mathcal{L}_{(P_{\mathcal{X}}, \rightarrow)}^{\text{graph}}(h, \rightsquigarrow) = \mathbb{E}_{\mathbf{x} \sim P_{\mathcal{X}}}[\ell^{\text{graph}}(h, \rightsquigarrow, \mathbf{x}, B_{\rightarrow}(\mathbf{x}))].$$

Furthermore for a sample $S = \{(\mathbf{x}_1, B_1), \dots, (\mathbf{x}_n, B_n)\}$ we define the *empirical graph loss* as

$$\mathcal{L}_S^{\text{graph}}(h, \rightsquigarrow) = \sum_{(\mathbf{x}_i, B_i) \in S} \ell^{\text{graph}}(h, \rightsquigarrow, \mathbf{x}_i, B_i).$$

Similar to previous sections, we now want to define a loss class for $\mathcal{H} \times \mathcal{G}$. We define $g(h, \rightsquigarrow)$ to be the set of all pairs $(\mathbf{x}, B) \in \mathcal{X} \times 2^{\mathcal{X}}$ on which $\ell^{\text{graph}}(h, \rightsquigarrow, \mathbf{x}, B) = 1$. Then the *graph loss class* of $\mathcal{H} \times \mathcal{G}$ is defined as

$$(\mathcal{H} \times \mathcal{G})_{\ell^{\text{graph}}} = \{g(h, \rightsquigarrow) : h \in \mathcal{H} \text{ and } \rightsquigarrow \in \mathcal{G}\}.$$

We will now show that if the VC-dimension of the loss class $(\mathcal{H} \times \mathcal{G})_{\ell^{\text{graph}}}$ is finite, we can indeed learn $\mathcal{G}$ with respect to ℓ^{graph} .

We now want to give some examples for when the VC dimension of the loss class $(\mathcal{H} \times \mathcal{G})_{\ell^{\text{graph}}}$ is indeed finite.

Example 8.5.1. 1. If for every $h \in \mathcal{H}$ and every $\rightarrow \in \mathcal{G}$, we have $VC(\{h\} \times \mathcal{G})_{\ell^{\text{graph}}} \leq d_1$ and $VC((\mathcal{H} \times \{\rightarrow\})_{\ell^{\text{graph}}}) \leq d_2$ then $VC((\mathcal{H} \times \mathcal{G})_{\ell^{\text{graph}}}) \leq d_1 d_2$.

2. Let $\mathcal{H}_{\text{lin}} = \{h : \mathbb{R}^d \rightarrow \{0, 1\} : \exists w \in \mathbb{R}^d : h(x) = 1 \text{ if and only if } x^T w \geq 0\}$ be the class of linear separators and $\mathcal{G}_{\text{lin}} = \{\rightarrow : \exists w \in \mathbb{R}^d : f_{\rightarrow}(x) = B_{x^T w}(x)\}$ be the graph class consistent of graphs such that all their neighborhood sets are balls whose radius may depend linearly on the feature vector. Then the VC-dimension of $\mathcal{H}_{\text{lin}} \times \mathcal{G}_{\text{lin}}$ is finite.

Lemma 8.5.2. Let $VC((\mathcal{H} \times \mathcal{G})_{\ell^{\text{graph}}}) = d$. Then there is $n_{\text{graph}} : (0, 1)^2 \mapsto \mathbb{N}$, such that for any marginal distribution $P_{\mathcal{X}}$ and any manipulation graph $\rightarrow$ for a sample $S = \{(\mathbf{x}_1, B_{\rightarrow}(\mathbf{x}_1)), \dots, (\mathbf{x}_n, B_{\rightarrow}(\mathbf{x}_n))\}$ of size $n \geq n(\varepsilon, \delta)$, we have with probability at least $1 - \delta$ over the sample generation $S_{\mathcal{X}} = (\mathbf{x}_1, \dots, \mathbf{x}_n) \sim P_{\mathcal{X}}^n$ for any $h \in \mathcal{H}$ and any $\rightsquigarrow \in \mathcal{G}$

$$|\mathcal{L}_{(P_{\mathcal{X}}, \rightarrow)}^{\text{graph}}(h, \rightsquigarrow) - \mathcal{L}_S^{\text{graph}}(h, \rightsquigarrow)| < \varepsilon.$$

Furthermore, $n_{\text{graph}}(\varepsilon, \delta) \in O(\frac{d + \log \frac{1}{\delta}}{\varepsilon^2})$.

We note that the above lemma is agnostic in the sense that it did not require $\rightarrow \in \mathcal{G}$. We will now introduce an empirical version of the $\mathcal{H}$ - $\mathbb{P}_{\mathcal{X}}$ -distance. This will allow us to state the main theorem of this section and show that it is indeed possible to learn $\rightsquigarrow \in \mathcal{G}$ with low $d_{\mathcal{H}, P_{\mathcal{X}}}(\rightarrow, \rightsquigarrow)$ if $\text{VC}((\mathcal{H} \times \mathcal{G})_{\ell_{\text{graph}}})$ is finite.

Definition 8.5.3. *Given a sample $S_{\mathcal{X}} = \{(\mathbf{x}_1, \dots, \mathbf{x}_n)\}$ of domain elements $\mathbf{x}_i$ and two manipulation graphs $\rightarrow$ and $\rightsquigarrow$ we can define the empirical $\mathcal{H}$ - $S_{\mathcal{X}}$ -distance as*

$$d_{\mathcal{H}, S_{\mathcal{X}}}(\rightarrow, \rightsquigarrow) = \sup_{h \in \mathcal{H}} \sum_{\mathbf{x}_i \in S_{\mathcal{X}}} \ell^{\text{graph}}(h, \rightsquigarrow, \mathbf{x}_i, B_{\rightarrow}(\mathbf{x}_i)).$$

Theorem 8.5.4. *Let $\text{VC}((\mathcal{H} \times \mathcal{G})_{\ell_{\text{graph}}}) = d$. Then there is $n_{\text{dist}} : (0, 1)^2 \mapsto \mathbb{N}$, such that for every marginal distribution $P_{\mathcal{X}}$ and every manipulation graph $\rightarrow$ for a sample $S = \{(\mathbf{x}_1, B_{\rightarrow}(\mathbf{x}_1)), \dots, (\mathbf{x}_n, B_{\rightarrow}(\mathbf{x}_n))\}$ of size $n \geq n(\varepsilon, \delta)$, we have with probability at least $1 - \delta$ over the sample generation $S_{\mathcal{X}} = (\mathbf{x}_1, \dots, \mathbf{x}_n) \sim P_{\mathcal{X}}^n$ for any $\rightsquigarrow \in \mathcal{G}$*

$$d_{\mathcal{H}, P_{\mathcal{X}}}(\rightarrow, \rightsquigarrow) < d_{\mathcal{H}, S_{\mathcal{X}}}(\rightarrow, \rightsquigarrow) + \varepsilon.$$

Furthermore, $n_{\text{dist}}(\varepsilon, \delta) \in O(\frac{d + \log \frac{1}{\delta}}{\varepsilon^2})$.

Combining Theorem 8.5.4 and Theorem 8.4.2 we can thus conclude that it is indeed possible to learn $\rightsquigarrow \in \mathcal{G}$ such that using $\ell^{\rightsquigarrow}$ as a surrogate loss function guarantees a good approximation on the true strategic loss $\ell^{\rightarrow}$.

8.6 Learning Manipulation Capabilities from Distribution Shifts

In the last section we explored learning the manipulation structure from full knowledge of the manipulation capabilities of finite randomly drawn sample. Obtaining access detailed information about the manipulation capabilities of users might not be feasible in reality. Instead, the only information accessible to a learner might be shifts in the data distribution observed after deployment of past data. In this section we will show what kinds of prior knowledge allow for estimation of manipulation structures from one manipulated and one unmanipulated sample from the (originally) same distribution. Before introducing our method for learning manipulation capabilities from distribution shifts, we first formalize the notion of manipulated sample.

Manipulated sample We assume, that if an admissible manipulation for the preferred label (i.e. the label 1) is available to an instance x given a published classifier h , then some manipulation to a positively labeled instance will occur. However, we do not assume that this is necessarily a best-response manipulation, in the sense that the instance will “move as little as possible”. The following definition formalizes this notion of classifier induced manipulations for a sample of instances.

Definition 8.6.1 (Classifier induced manipulation of a sample). *Let g be a manipulation graph and h be a hypothesis. We say $\pi : \mathcal{X} \rightarrow \mathcal{X}$ is a (g, h) -induced manipulation if*

$$\pi(x) \begin{cases} = x & \text{if } h(x) = 1 \text{ or} \\ & B_g(x) \cap h^{-1}(1) = \emptyset \\ \in B_g(x) \cap h^{-1}(1) & \text{if } h(x) = 0 \text{ and} \\ & B_g(x) \cap h^{-1}(1) \neq \emptyset \end{cases}$$

Now for a labeled sequence $S = ((x_1, y_1), \dots, (x_m, y_m))$ of instances and sequence $\Pi = (\pi_1, \dots, \pi_m)$ of (g, h) -induced manipulations $\pi_1, \dots, \pi_m$, we define the Π -manipulated sample $S^\Pi = ((\pi_1(x_1), y_1), \dots, (\pi_m(x_m), y_m))$. Similarly for an unlabeled sample $S = (x_1, x_2, \dots, x_m)$, the Π -manipulated sample is defined by $S^\Pi = (\pi_1(x_1), \dots, \pi_m(x_m))$.

Note that the above definition allows for repeated feature vectors $x_i = x_j$ (with $i \neq j$) in the sequence S to move to differing manipulated instances $\pi_i(x_i) \neq \pi_j(x_j)$. For simplicity, we will often just refer to the sequence Π as a (g, h) -induced manipulation without specifically referring to its components $\pi_1, \dots, \pi_m$.

We can now use this definition to derive a sample complexity bound for estimating the region of manipulation for any particular $h \in \mathcal{H}$ from one manipulated and one unmanipulated sample. We use the following notion of disagreement between two manipulation graphs:

Definition 8.6.2. *Given a distribution D over a domain set $\mathcal{X}$ (a.k.a. a marginal distribution), a classifier h and two manipulation graphs, g, g'*

$$\begin{aligned} Dis_{(D, h)}(g, g') = \\ D[\{x \in \mathcal{X} : \ell^{\text{mani}}(g, h, x) \neq \ell^{\text{mani}}(g', h, x)\}] \end{aligned}$$

Before giving the main result of the section, we will define the sufficient condition required for learning.

Definition 8.6.3. The loss class of $\mathcal{G} \times \mathcal{H}$ with respect to ℓ^{mani} is defined as

$$(\mathcal{G} \times \mathcal{H})_{\ell^{\text{mani}}} = \{\{x \in \mathcal{X} : \ell^{\text{mani}}(g, h, x) = 1\} : g \in \mathcal{G} \text{ and } h \in \mathcal{H}\}$$

Furthermore for a fixed h let the class $\mathcal{G}_{\ell^{\text{mani}}, h}$ be defined as

$$\mathcal{G}_{\ell^{\text{mani}}, h} = \{\{x \in \mathcal{X} : \ell^{\text{mani}}(g, h, x) = 1\} : g \in \mathcal{G}\}$$

We will define $\text{VC}(\mathcal{G}_{\ell^{\text{mani}}, \mathcal{H}}) = \sup_{h \in \mathcal{H}} \text{VC}(\mathcal{G}_{\ell^{\text{mani}}, h})$.

Observation 8.6.4. • The VC-dimension of $(\mathcal{G}_{\text{f.r.}} \times \mathcal{H}_{\text{thres}})_{\ell^{\text{mani}}}$ is 2.

- Let $\mathcal{H}_{\text{half}} = \{h_w : w \in \mathbb{R}^d : h_w(x) = 1 \text{ iff } x^T w \geq 0\}$ the hypothesis class of linear half spaces and $\mathcal{G}_{\text{f.r.}}^d = \{g_r = (\mathcal{X}, E_{g_r}) : (x, x') \in E_{g_r} \text{ iff } \|x - x'\|_2 \leq r\}$ be the class of fixed-radius balls in $\mathbb{R}^d$. Then the VC dimension of $(\mathcal{G}_{\text{f.r.}}^d \times \mathcal{H}_{\text{half}})_{\ell^{\text{mani}}}$ is at most $2d$.

Lemma 8.6.5. Let $\mathcal{G}, \mathcal{H}$ be such that $\text{VC}(\mathcal{G}_{\ell^{\text{mani}}, \mathcal{H}}) = d$. Let $A_{\text{graph}} : \mathcal{X}^* \times \mathcal{X}^* \times \mathcal{H} \rightarrow 2^{\mathcal{G}}$ be a learner following the Empirical Manipulation Estimation Paradigm (as defined in Algorithm 2). Then A_{graph} has the following success guarantee for learning the manipulation graph:

For every marginal distribution $P_{\mathcal{X}}$, every $g^{\rightarrow} \in \mathcal{G}$, every $h \in \mathcal{H}$ and every sequence of $(g^{\rightarrow}, h)$ -induced manipulations Π , and every $\varepsilon, \delta \in (0, 1)$,

if $m \geq C \frac{d + \log(\frac{1}{\delta})}{\varepsilon^2}$ (for some universal constant C) with probability at least $(1 - \delta)$, over samples $S_1 \sim P_{\mathcal{X}}^m, S_2 \sim P_{\mathcal{X}}^m$, for every $\hat{g} \in A_{\text{graph}}(S_1, S_2^{\Pi}, h)$,

$$\text{Dis}_{(P_{\mathcal{X}}, h)}(g^{\rightarrow}, \hat{g}) \leq \varepsilon.$$

and

$$g^{\rightarrow} \in A_{\text{graph}}(S_1, S_2^{\Pi}, h).$$

The key tool for proving the above lemma is the notion of a sample S_1 being ε -representative with respect to $(\mathcal{G} \times \mathcal{H})_{\ell^{\text{mani}}}$. For any $g \in \mathcal{G}$ and $h \in \mathcal{H}$ the empirical manipulated loss over such a sample is ε -close to its true loss. Standard uniform convergence theory (see, e.g., [Shalev-Shwartz and Ben-David, 2014a] Chapter 4) shows that, given a class of finite VC-dimension, for any data generating distribution, a large enough sample will be ε -representative with respect to that class.

Observation 8.6.6. If an un-manipulated sample S_1 is ε -representative with respect to $(\mathcal{G} \times \mathcal{H})_{\ell^{\text{mani}}}$, then it can be indefinitely re-used by A_{graph} for any hypothesis $h \in \mathcal{H}$ and any manipulated ε -representative samples S_2^{Π} . Thus if $\text{VC}(\mathcal{G} \times \mathcal{H})_{\ell^{\text{mani}}} = d$, then $m \geq C \frac{d + \log(\frac{1}{\delta})}{\varepsilon^2}$ (for some universal constant C), implies that with probability $1 - \delta$ any $S_1 \sim P^m$ is repeatedly reusable by A_{graph} to guarantee ε -success as in the Lemma above. We will later use this fact when introducing our batch-learning setting. In this setting we the initial unmanipulated sample in all subsequent steps.

Algorithm 2: EmpiricalManipulationEstimation (realizable)

Input: graph class $\mathcal{G}$, hypothesis h , input samples S_1 and S_2^{Π} , parameter ε

Output: set of candidate manipulation graphs $\mathcal{G}_c$

$L_{\max} \leftarrow \max_{g \in \mathcal{G}} L_{S_1}^{\text{mani}}(g, h)$ s.t. $L_{S_2}^{\text{mani}}(g, h) = 0$

$\mathcal{G}_c \leftarrow \{g \in \mathcal{G} : L_{S_1}^{\text{mani}}(g, h) \in [L_{\max} - \varepsilon, L_{\max}]\}$

8.7 Classes of manipulation graphs that allow realizable learnability

So far we have discussed how to estimate the manipulation capabilities (Section 8.5 and Section 8.6) and how certain inaccuracies in the manipulation graph might still allow for accurate learning (Section 8.4). We will now address the question of PAC learning with respect to a class of potential candidate graph class $\mathcal{G}$.

More concretely, we will consider the following learning problem:

Definition 8.7.1. We say a hypothesis class $\mathcal{H}$ is learnable with respect to a class of candidate manipulation graphs $\mathcal{G}$, if there is a learner $\mathcal{A}$ and a sample complexity function $m_{\mathcal{H}, \mathcal{G}} : (0, 1)^2 \rightarrow \mathbb{N}$, such that for every $\varepsilon, \delta \in (0, 1)$, for every distribution P and every $g \in \mathcal{G}$, if $S \sim P^m$ with $m \geq m_{\mathcal{H}, \mathcal{G}}(\varepsilon, \delta)$, then $\mathcal{L}_P^g(\mathcal{A}(S)) \leq \inf_{h \in \mathcal{H}} \mathcal{L}_P^g(h) + \varepsilon$ with probability $1 - \delta$. We say $\mathcal{H}$ is learnable with respect to $\mathcal{G}$ in the realizable case, if the guarantee holds for all $g \in \mathcal{G}$ with $\inf_{h \in \mathcal{H}} \mathcal{L}_P^g(h) = 0$.

We start with an example that shows the necessity of prior knowledge and how certain graph classes for which not knowing the exact manipulation graph from this class will make learning with respect to robust loss impossible.

Example 8.7.2. Let $\mathcal{X} = \{0, 1\} \times [0, 1]$. Let $\mathcal{H} = \{h_a \in \{0, 1\}^{\mathcal{X}} : a \in [0, 1]\}$, where for all $x \in [0, 1] : h_a((0, x)) = 0$ and $h_a((1, x)) = 1$ if $a \neq x$ and $h_a((1, a)) = 0$. Furthermore

let the graph class $\mathcal{G} = \{g_a = (\mathcal{X}, E_g), \text{ where } E_g = \{(0, x), (1, a), x \in \mathbb{R}\}\}$, that is $\mathcal{G}$ consists of manipulation graphs in which every element $(0, x)$ can manipulate to exactly one element $(1, a)$, which is the same for all x . The uncertainty in the manipulation graph is now, what the true parameter for a is. Now consider a uniform distribution P over $(\{0, 1\} \times [0, 1]) \times \{0, 1\} \subset \mathcal{X} \times \mathcal{Y}$. It is clear that for every $g_a \in \mathcal{G}$, $\min_{h \in \mathcal{H}} \mathcal{L}_P^{g_a}(h) = \mathcal{L}_P^{g_a}(h_a) = 0$. However for every h_b with $b \neq a$ we have $\mathcal{L}_P^{g_a}(h_b) = \frac{1}{2}$. Thus without knowledge of a , strategically learning $\mathcal{H}$ with respect to all $\mathcal{G}$ is impossible.

We have now seen that uncertainty in the manipulation graph can indeed render learning impossible, even if the class of considered graphs has a simple structure. To contrast this we will now consider a partial order on the graph class that reflects some of the difficulty of learning with respect to a class of candidate graphs.

We will first define the partial order over manipulation graphs that are induced by manipulation loss and a hypothesis class $\mathcal{H}$. We will then show that if a class of manipulation graphs $\mathcal{G}$ is totally ordered with respect to this order, then finite VC-classes are learnable in the realizable case with respect to $\mathcal{G}$.

Definition 8.7.3. For a hypothesis class $\mathcal{H}$, let the $\ell^{\text{mani}} - \mathcal{H}$ -induced partial order $\preceq_{\mathcal{H}}$ on manipulation graphs be defined by: $g_1 \preceq_{\mathcal{H}} g_2$, if and only if for every $h \in \mathcal{H}$ and every $x \in \mathcal{X}$, we have $\ell^{\text{mani}}(g_1, h, x) \leq \ell^{\text{mani}}(g_2, h, x)$.

A graph class $\mathcal{G}$ is totally ordered with respect to $\preceq_{\mathcal{H}}$ if for every distinct $g_1, g_2 \in \mathcal{G}$, we have that either $g_1 \preceq_{\mathcal{H}} g_2$ or $g_2 \preceq_{\mathcal{H}} g_1$. For a subset $A \subset \mathcal{G}$ of a totally ordered graph class, we define $\max_{\preceq_{\mathcal{H}}}$ as the graph $g \in A$ with $g' \preceq g$ for all $g' \in A$.

Observation 8.7.4. • A graph class $\mathcal{G}$ is totally ordered with respect to the class of all hypotheses $\mathcal{F}$ if and only if for every distinct $g_1, g_2 \in \mathcal{G}$ either g_1 is a subgraph of g_2 or g_2 is a subgraph of g_1 .

- For $\mathcal{H}_1 \subset \mathcal{H}_2$ and two graphs g_1, g_2 , we have that $g_1 \preceq_{\mathcal{H}_2} g_2$ implies $g_1 \preceq_{\mathcal{H}_1} g_2$. Thus, if a graph class $\mathcal{G}$ is totally ordered with respect to $\mathcal{H}_2$ it is also totally ordered with respect to $\mathcal{H}_1$.
- If $\mathcal{G}$ is totally ordered with respect to $\mathcal{H}$, then $\text{VC}(\mathcal{G}_{\ell^{\text{mani}}, \mathcal{H}}) = 1$.
- There are $\mathcal{G}$ and $\mathcal{H}$, such that $\text{VC}(\mathcal{H}) = d$ and $\mathcal{G}$ is totally ordered with respect to $\mathcal{H}$, but $\text{VC}((\mathcal{H} \times \mathcal{G})_{\ell^{\text{mani}}}) = \infty$.

As noted in Observation 8.7.4, it can be the case that $\text{VC}(\mathcal{H})$ and $\text{VC}(\mathcal{G})_{\ell^{\text{mani}}, \mathcal{H}}$ are finite, but $\text{VC}(\mathcal{G} \times \mathcal{H})_{\ell^{\text{mani}}}$ is still infinite. In particular, this is the case for any $\mathcal{G}$ that

contains a $g \in \mathcal{G}$ such that $\text{VC}(\mathcal{H}_{\ell^g})$ is infinite, where $\mathcal{H}_{\ell^g}$ is the loss class of $\mathcal{H}$ with respect to the strategic loss ℓ^g . Furthermore, recall that we have shown in the previous section that there are hypothesis classes $\mathcal{H}$ with finite VC-dimension but infinite VC-dimension of the loss class $\mathcal{H}_{\ell^g}$, which are not properly strategically robust learnable by any learner. The following observation is an immediate corollary.

Observation 8.7.5. There are classes $\mathcal{H}$ of finite VC-dimension and graph classes $\mathcal{G}$ which are totally ordered with respect to $\mathcal{H}$, such that there is no proper successful batch-learner for $\mathcal{H}$ and $\mathcal{G}$ for any finite sample and round complexity, not even in the realizable case.

However, similar to before, we will now aim to show that any class of finite VC dimension can be improperly strategically robustly learned with respect to a totally ordered graph class $\mathcal{G}$. We can generalize the results obtained in the case of a known manipulation graph to learning with respect to a totally ordered graph class in the realizable case. We will exploit the fact that realizability in the strategically robust sense implies that the training sample will be robustly realizable with probability 1. Now let $\mathcal{G}$ be a totally ordered graph class and $\mathcal{H}$ a hypothesis class. We define the maximally robust graph in $\mathcal{G}$ with respect to a sample S and class $\mathcal{H}$ as

$$\text{MG}_{\mathcal{H},\mathcal{G}}(S) = \max_{\preceq_{\mathcal{H}}} \{g \in \mathcal{G} : \min_{h \in \mathcal{H}} \mathcal{L}_S^g(h) = 0\}.$$

In cases where this maximum does indeed exist, we then define a *maximally robust empirical risk minimizer (MRERM)* with respect to $\mathcal{H}$ and $\mathcal{G}$ as the hypothesis in $\mathcal{H}$ minimizing the strategically robust loss with respect to $\text{MG}_{\mathcal{H},\mathcal{G}}(S)$, i.e.,

$$\text{MRERM}_{\mathcal{H},\mathcal{G}}(S) \in \arg \min_{h \in \mathcal{H}} \mathcal{L}_S^{\text{MG}_{\mathcal{H},\mathcal{G}}(S)}(h).$$

In cases where the maximum graph does not exist, we need to instead pick a hypothesis $h_S^{\max}$ that is simultaneously robust with respect to every graph in $\mathcal{G}$ for which S is $\mathcal{H}$ -realizable to achieve our guarantee. Such a hypothesis might not always exist within $\mathcal{H}$. However, we can extend $\mathcal{H}$ to a class $\mathcal{H}'$, with $\text{VC}(\mathcal{H}') = \text{VC}(\mathcal{H})$ and such that for every finite sample $S \subset \mathcal{X}$ the class $\mathcal{H}'$ contains a hypothesis $h_S^{\max}$. In order to define such a hypothesis class and general MRERM rigorously we need to use the set-theoretic concept of ultrafilters. We will elaborate on the existence of such a hypothesis class later in the section.

We note that in the realizable setting, the maximal graph used for the estimation here will always overestimate the robustness of the true manipulation graph $g^{\rightarrow}$. We use this fact to define a successful improper learner and prove the following theorem based on techniques from the literature on PAC learning with respect to adversarial perturbations [Montasser et al., 2019].

Theorem 8.7.6. *Let $\text{VC}(\mathcal{H})$ be finite and let $\mathcal{G}$ be totally ordered with respect to $\mathcal{H}$. Then there is a strategically robust (improper) PAC-learner (i.e., a PAC-learner with respect to $\ell^{g^\rightarrow}$ -loss) which is successful for every $g^\rightarrow \in \mathcal{G}$ in the strategically robustly realizable case (i.e. when $\inf_{h \in \mathcal{H}} \mathcal{L}_P^{g^\rightarrow}(h) = 0$).*

Note that the learner is successful, even without knowing the true manipulation graph $g^\rightarrow$ and without receiving any local perturbation sets as input.

8.7.1 Note on the maximally robust ERM paradigm

The proof of Theorem 8.7.6, employs the notions of a maximal graph within $\mathcal{G}$ and maximal ERM hypothesis within the class $\mathcal{H}$. Recall the definitions from Section 8.7:

- $\text{MG}_{\mathcal{H}, \mathcal{G}}(S) = \max_{\preceq_{\mathcal{H}}} \{g \in \mathcal{G} : \min_{h \in \mathcal{H}} \mathcal{L}_S^g(h) = 0\}$
- $\text{MRERM}_{\mathcal{H}, \mathcal{G}}(S) \in \arg \min_{h \in \mathcal{H}} \mathcal{L}_S^{\text{MG}_{\mathcal{H}, \mathcal{G}}(S)}(h)$

However, in general, these do not necessarily exist in these classes. In this section, we show that we can always embed the original classes $\mathcal{G}$ and $\mathcal{H}$ in such a way that the above notions are well-defined.

Consider a fixed sample $S = ((x_1, y_1), \dots, (x_m, y_m))$ that is realizable with respect to the strategic loss for some $g^\rightarrow \in \mathcal{G}$. Then the set

$$\mathcal{G}_S = \{g \in \mathcal{G} : \min_{h \in \mathcal{H}} \mathcal{L}_S^g(h) = 0\} \subseteq \mathcal{G}$$

is not empty, and we can define a new manipulation graph $\text{MG}_{\mathcal{H}, \mathcal{G}}(S)$ by taking the union of all edge sets of graphs in $\mathcal{G}_S$ as the new edge set for this maximal graph. Now, if the set

$$\arg \min_{h \in \mathcal{H}} \mathcal{L}_S^{\text{MG}_{\mathcal{H}, \mathcal{G}}(S)}(h)$$

is not empty, then we can choose any element in this set to define a maximally robust ERM hypothesis $\text{MRERM}_{\mathcal{H}, \mathcal{G}}(S)$. Note that this is always the case if the graph class $\mathcal{G}$ (and therefore the set $\mathcal{G}_S$) is finite.

Thus we now focus on the case where $\mathcal{G}_S$ induced by some S is infinite and the set $\arg \min_{h \in \mathcal{H}} \mathcal{L}_S^{\text{MG}_{\mathcal{H}, \mathcal{G}}(S)}(h)$ is empty. We will use the concept of an *ultrafilter* to define a maximal hypothesis $h_S^{\max}$ for the sample S . Finally, we will show that adding all possible (over all labelled samples S) such maximal hypotheses yields a hypothesis class $\mathcal{H}'$ with $\mathcal{H} \subset \mathcal{H}'$ and $\text{VC}(\mathcal{H}') = \text{VC}(\mathcal{H})$.

Definition 8.7.7 (Filter). Let $\mathcal{Z}$ be some set and let $\mathcal{F} \subseteq 2^{\mathcal{Z}}$ be a collection of subsets of $\mathcal{Z}$. We call $\mathcal{F}$ a filter if the following conditions are satisfied:

- $\mathcal{F} \neq \emptyset$
- $\emptyset \notin \mathcal{F}$
- $\mathcal{F}$ is upwards inclusion closed: if $A \in \mathcal{F}$ and $A \subseteq B$, then $B \in \mathcal{F}$
- $\mathcal{F}$ is closed under finite intersections: if $A \in \mathcal{F}$ and $B \in \mathcal{F}$, then $(A \cap B) \in \mathcal{F}$.

A filter $\mathcal{F}$ is an ultrafilter if for every domain subset $C \subseteq \mathcal{Z}$, either C or its complement $\bar{C}$ is a member of $\mathcal{F}$.

It can be shown (using Zorn's lemma) that every filter $\mathcal{F}$ can be extended to an ultrafilter. That is, there always exists an ultrafilter $\tilde{\mathcal{F}}$ with $\mathcal{F} \subseteq \tilde{\mathcal{F}}$.

We will start by defining a filter over $\mathcal{G}_S$. Note that $\mathcal{G}_S$ is totally ordered. A *final segment* $G \subseteq \mathcal{G}_S$ is a subset such that $g \in G$ and $g \preceq g'$ implies $g' \in G$. Now consider the collection $\mathcal{F} \subseteq 2^{\mathcal{G}_S}$ defined by:

$$\mathcal{F} = \{G \subseteq \mathcal{G}_S \mid G \text{ contains a final segment of } \mathcal{G}_S\}.$$

It is not difficult to see that $\mathcal{F}$ is indeed a filter. Let $\tilde{\mathcal{F}}$ denote an ultra-filter extending $\mathcal{F}$ (that is, $\mathcal{F} \subseteq \tilde{\mathcal{F}}$).

Recall that for every $g \in \mathcal{G}_S$ there exists at least one $h \in \mathcal{H}$ such that $\mathcal{L}_S^g(h) = 0$. Let $\tau : \mathcal{G}_S \rightarrow \mathcal{H}$ be a function that assigns every manipulation graph $g \in \mathcal{G}_S$ such a hypothesis $h_g = \tau(g)$. The image of τ is thus a subset $\mathcal{H}_S \subseteq \mathcal{H}$ of the hypothesis class of empirical risk minimizers corresponding to the graphs in $\mathcal{G}_S$:

$$\mathcal{H}_S = \{h_g \mid g \in \mathcal{G}_S\}.$$

We can consider the set $\mathcal{H}_S$ as indexed by $\mathcal{G}_S$ and therefore inheriting the order of $\mathcal{G}_S$ (and we can thus refer to final segments of $\mathcal{H}_S$ etc). Note that, if for some $x \in \mathcal{X}$, and $y \in \{0, 1\}$, there exists a final segment of $\mathcal{H}_S$, in which all functions assign x the same label y , then the set

$$\{g \in \mathcal{G}_S \mid h_g(x) = y\} \in \mathcal{F}$$

is an element of the filter $\mathcal{F}$ (since the filter is upward inclusion closed). For such cases, the limit function $h_S^{\max}$ will assign this label y to x .

Observe that for every x , the set $\{g \in \mathcal{G}_S \mid h_g(x) = y\}$ is the complement (in $\mathcal{G}_S$) of the set $\{g \in \mathcal{G}_S \mid h_g(x) = 1 - y\}$, and thus exactly one of these sets is a member of the ultrafilter $\tilde{\mathcal{F}}$ that contains $\mathcal{F}$. We can thus define the limit function on the whole domain $\mathcal{X}$ by

$$h_S^{\max}(x) = \begin{cases} 1 & \text{if } \{g \in \mathcal{G}_S \mid h_g(x) = 1\} \in \tilde{\mathcal{F}} \\ 0 & \text{else.} \end{cases}$$

That is, for every $x \in \mathcal{X}, y \in \{0, 1\}$, we have $h_S^{\max}(x) = y$ if and only if $\{g \in \mathcal{G}_S \mid h_g(x) = y\} \in \tilde{\mathcal{F}}$. The ultra-filter $\tilde{\mathcal{F}}$ serves to define a tie-breaker label for all domain elements x that are not “eventually” assigned the same label - x ’s for which every final segment of $\mathcal{G}_S$ contains both g ’s with $h_g(x) = 0$ and g ’s with $h_g(x) = 1$.

It remains to show that this so-defined limit function $h_S^{\max}(x)$ has empirical strategic loss 0 on S for all graphs in $\mathcal{G}_S$ and that adding all such limit functions to the hypothesis class $\mathcal{H}$ will not increase its VC-dimension.

For the first property, consider some $g \in \mathcal{G}_S$ and some $(x, y) \in S$. If $y = 0$, then all functions $h_{g'}$ for $g \preceq g'$ assign 0 to all points in $B_g(x)$ (since each $h_{g'}$ is an empirical risk minimizer for the empirical strategic loss $\mathcal{L}_S^{g'}$). If $y = 1$, then all these functions assign label 1 to x . Since the set $\{g' \in \mathcal{G}_S \mid g \preceq g'\}$ is a member of $\mathcal{F}$ and therefore a member of $\tilde{\mathcal{F}}$, the first case implies that $h_S^{\max}(x') = 0$ for all $x' \in B_g(x)$ and the second case implies that $h_S^{\max}(x) = 1$. In both cases $\ell^g(h_S^{\max}, (x, y)) = 0$, and since this holds for all $(x, y) \in S$ and all $g \in \mathcal{G}_S$, we have

$$\mathcal{L}_S^g(h_S^{\max}(x)) = 0.$$

Now we consider the extended hypothesis class

$$\mathcal{H}' = \mathcal{H} \cup \{h_S^{\max} \in \{0, 1\}^{\mathcal{X}} \mid S \in (\mathcal{X} \times \{0, 1\})^*\}$$

where we added the limit functions (as defined above) for all possible labeled samples S over $\mathcal{X} \times \{0, 1\}$. In order to show that $\mathcal{H}'$ has the same VC-dimension as $\mathcal{H}$, we argue that any finite set that is shattered by $\mathcal{H}'$ is already shattered by $\mathcal{H}$.

Consider domains points $x_1, x_2, \dots, x_d$, shattered by $\mathcal{H}'$, and some labels $y_1, y_2, \dots, y_d$. Assume this labeling is realized by a limit function $h_S^{\max}$ (that came from some labeled sample S), that is $h_S^{\max}(x_i) = y_i$ for all $i \in [d]$. Note that $h_S^{\max}(x_i) = y_i$ implies that for each i the set

$$\{g \in \mathcal{G}_S \mid h_g(x_i) = y_i\} \in \tilde{\mathcal{F}}$$

Since ultra-filters are closed under finite intersections, the set

$$\{g \in \mathcal{G}_S \mid h_g(x_i) = y_i \text{ for all } i \in [d]\}$$

is also a member of the ultrafilter $\tilde{\mathcal{F}}$, and therefore not empty. Thus, there exists a $g \in \mathcal{G}_S$ and corresponding ERM function $h_g \in \mathcal{H}_S \subseteq \mathcal{H}$ with $h_g(x_i) = y_i$ for all $i \in [d]$. Thus, any labeling on a finite set of points that is realized by some limit function, is also already realized by a hypothesis from $\mathcal{H}$. Therefore, $\text{VC}(\mathcal{H}') = \text{VC}(\mathcal{H})$, and we can use the larger class $\mathcal{H}'$ to define the maximally robust ERM hypothesis, we set

- $\text{MG}_{\mathcal{H}, \mathcal{G}}(S) = (\mathcal{X}, E)$ with $E = \bigcup_{\{g \in \mathcal{G} : \min_{h \in \mathcal{H}} \mathcal{L}_S^g(h) = 0\}} E_g$
- $\text{MRERM}_{\mathcal{H}}^{\mathcal{G}}(S) = \begin{cases} h \in \arg \min_{h \in \mathcal{H}} \mathcal{L}_S^{\text{MG}_{\mathcal{H}, \mathcal{G}}(S)}(h) & \text{if this set is not empty} \\ h_S^{\max} & \text{as defined above, otherwise.} \end{cases}$

8.8 Batch-Learning for strategic classification

To model this situation, we propose a novel batch-learning setting in which we use unlabeled data from previous rounds to estimate the manipulation structure. We show that in this batch-learning setting it is possible to learn a close-to-optimal classifier in terms of the strategic loss even without knowing the feasible manipulations beforehand. In line with recent advances in the strategic classification literature, we do not assume a best-response from agents but only require that observed manipulations are feasible.

To model repeated decision-making scenarios (such as the university admission task outlined in the introduction), we introduce a formal batch-learning protocol. In our protocol, a learner receives one non-manipulated labelled sample from the distribution, publishes an initial hypothesis $\hat{h}^0$, and then, in each round t , successively observes strategically manipulated (but unlabeled) samples in response to the last published hypothesis $\hat{h}^{t-1}$. Recall that we denote the true underlying manipulation graph by $g^{\rightarrow}$.

Definition 8.8.1 (Strategic batch-learning protocol). *Let $(m_i)_{i \in \mathbb{N}}$ be a sequence of sample sizes, $m_i \in \mathbb{N}$.*

Round 0: *The learner receives a labeled sample $S \sim P^{m_0}$ and, in response, publishes classifier $\hat{h}^0 : \mathcal{X} \rightarrow \{0, 1\}$.*

Round t for $t \geq 1$: *The learner receives an unlabeled sample $S' = S^{\Pi} \in \mathcal{X}^{m_t}$, which is a Π -manipulated version of some unlabeled sample $S \sim P_{\mathcal{X}}^{m_t}$, where Π is a $(g^{\rightarrow}, \hat{h}^{t-1})$ -induced manipulation. In response, the learner publishes classifier $\hat{h}^t$.*

We call a learner $\mathcal{A}$ that operates according to the above protocol, a *strategic batch learner*. Note that the learner receives only one labeled sample from the data-generating process, in the first round. And the only information about the underlying manipulation structure $g^\rightarrow$ it receives, are the unlabeled, manipulated samples in the subsequent rounds.

The goal is to output a hypothesis with low *strategic loss*.

We now define our success criterion for a strategic batch learner:

Definition 8.8.2. A strategic batch learner $\mathcal{A}$ is said to learn hypothesis class $\mathcal{H}$ under graph class $\mathcal{G}$ with sample complexity $m_{\mathcal{G},\mathcal{H}} : (0,1)^2 \rightarrow \mathbb{N}$ and round complexity $T_{\mathcal{G},\mathcal{H}} : (0,1)^2 \rightarrow \mathbb{N}$, if for every $\varepsilon, \delta \geq 0$, and every P over $X \times \{0,1\}$ and every true manipulation graph $g^\rightarrow \in \mathcal{G}$, after $T = T(\varepsilon, \delta)$ many rounds, $\mathcal{A}$ outputs hypothesis h^T satisfying $\mathcal{L}_P^{g^\rightarrow}(h^T) \leq \inf_{h' \in \mathcal{H}} \mathcal{L}_P^{g^\rightarrow}(h') + \varepsilon$, with probability at least $(1 - \delta)$ over the sample generation.

We say a learner $\mathcal{A}$ is a successful strategic batch learner in the *realizable* case with sample complexity $m_{\mathcal{G},\mathcal{H}}^{\text{real}}$ and round complexity $T_{\mathcal{G},\mathcal{H}}^{\text{real}}$, if it satisfies the above criterion for all distributions $P \in \mathcal{P}^\rightarrow$, where $\mathcal{P}^\rightarrow$ denotes the set of all distributions over $\mathcal{X} \times \{0,1\}$ with $\inf_{h \in \mathcal{H}} \mathcal{L}_{P'}^{g^\rightarrow}(h') = 0$.

8.8.1 Class of Thresholds

Algorithm 3: Strategic Batch-Learning for Thresholds

Input: parameters $\varepsilon, \varepsilon'$
receive sample $S_0 \sim P^m$
 $\mathcal{L}_0^{0/1} \leftarrow \inf_{h' \in \mathcal{H}_{\text{thres}}} \mathcal{L}_{S_0}^{0/1}(h')$
for $i = 0, \dots, \frac{1}{\varepsilon}$ **do**
 $r_i \leftarrow \max\{r : \inf_{h' \in \mathcal{H}_{\text{thres}}} \mathcal{L}_{S_0}^{g_r}(h') = \mathcal{L}_0^{0/1} + i \cdot \varepsilon\}$
end
 $\mathcal{G}^0 \leftarrow [g_{r_0}, g_{r_1}, \dots, g_{r_{\frac{1}{\varepsilon}}}]$
 $\hat{h}^0 \leftarrow \arg \min_{h' \in \mathcal{H}_{\text{thres}}} \mathcal{L}_{S_0}^{g_{r_0}}(h')$
publish $\hat{h}^0$
receive sample S_1 , where $S'_1 \sim P^m$ and $S_1 = S_1^{\Pi_1}$ for some sequence of $(g_{\rightarrow}, \hat{h}^0)$ -induced manipulations Π_1 .
 $r_{\max} \leftarrow \max\{r : g_r \in \mathcal{G}^0 \text{ and } L_{S_1}^{\text{mani}}(g_r, \hat{h}^0) = 0\}$
 $r_{\min} \leftarrow \min\{r : g_r \in \mathcal{G}^0 \text{ and } L_{S_0}^{\text{mani}}(g_r, \hat{h}^0) \geq L_{S_0}^{\text{mani}}(g_{r_{\max}}, \hat{h}^0) - \varepsilon' \text{ and } L_{S_1}^{\text{mani}}(g, \hat{h}^0) = 0\}$
 $\mathcal{G}^1 \leftarrow \{g_r \in \mathcal{G}^0 : r \in [r_{\min}, r_{\max}]\}$
 $k_1 = \lfloor \frac{|\mathcal{G}^1|}{2} \rfloor$
 $\hat{g}^1 \leftarrow \mathcal{G}^1[k_1]$, where $\mathcal{G}[k]$ refers to the k -th element of $\mathcal{G}$
 $\hat{h}^1 \leftarrow \arg \min_{h \in \mathcal{H}'} \mathcal{L}_{S_0}^{\hat{g}^1}(h)$
for rounds $t = 2, \dots, T$ **do**
 publish $\hat{h}^{t-1}$
 receive sample S_t , where $S'_t \sim P^m$ and $S_t = S_t^{\Pi_t}$ for some sequence of $(g_{\rightarrow}, \hat{h}^{t-1})$ -induced manipulations Π_t .
 $\mathcal{G}^t \leftarrow \{g \in \mathcal{G}^{t-1} : L_{S_t}^{\text{mani}}(g, \hat{h}^{t-1}) = 0\}$
 $\mathcal{G}_0^t \leftarrow \{g \in \mathcal{G}^t : L_{S_0}^{\text{mani}}(g, \hat{h}^{t-1}) \geq \max_{g' \in \mathcal{G}^t} L_{S_0}^{\text{mani}}(g', \hat{h}^{t-1}) - \varepsilon'\}$
 if $\hat{g}^{t-1} \in \mathcal{G}_0^t$ **then**
 $\mathcal{G}^t \leftarrow [\mathcal{G}^t[0], \dots, \mathcal{G}^t[k_{t-1}]] \cap \mathcal{G}_0^t$
 else
 $\mathcal{G}^t \leftarrow \mathcal{G}_0^t$
 end
 $k_t \leftarrow \lfloor \frac{|\mathcal{G}^t|}{2} \rfloor$
 $\hat{g}^t \leftarrow \mathcal{G}^t[k_t]$
 $\hat{h}^t \leftarrow \min_{h \in \mathcal{H}'} \mathcal{L}_{S_0}^{\hat{g}^t}(h)$
end

We start by showing that learning with respect to an unknown manipulation graph for the hypothesis class of thresholds with a simple graph class is possible. Consider $\mathcal{X} = \mathbb{R}$. The class of thresholds is defined as $\mathcal{H}_{\text{thres}} = \{h_{a,0} : \mathbb{R} \rightarrow \{0,1\} : h_{a,0}(x) = 1 \text{ iff } x > a\} \cup \{h_{a,1} : \mathbb{R} \rightarrow \{0,1\} : h_{a,1}(x) = 1 \text{ iff } x \geq a\}$. We now look at the graph class consisting of fixed-radius manipulation graphs g_r , which for any x has outgoing edges to every x' with $x' - x \leq r$, i.e. $\mathcal{G}_{\text{f.r.}} = \{g_r = (\mathbb{R}, E_r) : (x, x') \in E_r \text{ iff } x' \leq x + r\}$.

We show that for this simple class there can be indeed a successful strategic batch-learner. We first note that under the robust realizability assumption, the learner only needs one round to learn a close to optimal classifier and does not require access to any manipulated samples.

Observation 8.8.3. The Strategic Batch-Learning for Thresholds Algorithm (Algorithm 1) is a proper learner for the strategic-batch learning problem for $\mathcal{H}_{\text{thres}}$ and $\mathcal{G}_{\text{f.r.}}$ in the realizable case with sample complexity $m_{\mathcal{H},\mathcal{G}}^{\text{real}} = O(\frac{\log(\frac{1}{\delta})}{\varepsilon^2})$ and round complexity $T_{\mathcal{H},\mathcal{G}}^{\text{real}} = 1$.

We note that in the agnostic case gaining further knowledge of the manipulation graph is indeed necessary. To see this consider the following example.

Example 8.8.4. Let us consider $\mathcal{X} = \mathbb{R}$, $\mathcal{H}_{\text{thres}}$ and $\mathcal{G}_{\text{f.r.}}$. Furthermore let P be defined by $P((0,0)) = \frac{1}{2}$, $P((1,1)) = \frac{1}{4}$ and $P((2,1)) = \frac{1}{4}$. In this setting, if the manipulation graph is given by a radius $r^\rightarrow < 1$, the distribution P is realizable and $h_{1,1}$ is a robustly realizable classifier, i.e. $\mathcal{L}_P^{g^\rightarrow}(h_{1,1}) = 0$. However if $2 > r^\rightarrow > 1$, then $\mathcal{L}_P^{g^\rightarrow}(h_{1,1}) = \frac{1}{2}$, whereas the optimal classifier in this setting $h_{2,1}$ has only loss $\frac{1}{4}$.

We thus see that the realizable and the agnostic cases are indeed different in this setting.

Next, we show that there also is an algorithm that solves the strategic-batch learning problem for these classes in the agnostic case.

Theorem 8.8.5. The Strategic Batch-Learning for Thresholds Algorithm (Algorithm 1) is a proper learner for the strategic-batch learning problem for $\mathcal{H}_{\text{thres}}$ and $\mathcal{G}_{\text{f.r.}}$ in the hypothesis-agnostic case with sample complexity $m_{\mathcal{H},\mathcal{G}} = O(\frac{\log(\frac{1}{\delta\varepsilon})}{\varepsilon^2})$ and round complexity $T_{\mathcal{H},\mathcal{G}} = O(\log(\frac{1}{\varepsilon}))$.

This is achieved by Algorithm 1 (formal proofs are provided in the appendix). In the first step the algorithm uses the labelled sample S_0 to generate candidate graphs (which are stored as an ordered list $\mathcal{G}^0$), in such a way that the sample losses on S_0 for the corresponding optimal hypotheses increases in ε -steps. Choosing an appropriate sample size, we can guarantee that S_0 is ε -representative in terms of strategic loss ℓ^g for all $g \in \mathcal{G}_{\text{f.r.}}$.

That is, the observed sample losses on S_0 are ε -close to the corresponding expected losses according to distribution P . This then guarantees that optimizing the sample loss for one of the generated candidates yields a close-to-optimal hypothesis on the ground-truth distribution.

We further use the fact, that for any $h \in \mathcal{H}_{\text{thres}}$, any distribution P over $\mathcal{X} \times \{0, 1\}$ any sample S and any radii $r_1 \leq r_2$, we have that $\mathcal{L}_P^{g_{r_1}}(h) \leq \mathcal{L}_P^{g_{r_2}}(h)$ as well as $\mathcal{L}_S^{g_{r_1}}(h) \leq \mathcal{L}_S^{g_{r_2}}(h)$. Now let $g^{t-1} = g_r$ be the current candidate graph. Then the algorithm publishes a hypothesis $h^{t-1} = \arg \min_{h \in \mathcal{H}_{\text{thres}}} \mathcal{L}_{S_0}^{g^{t-1}}(h)$. There are two possibilities: (1) $\mathcal{L}_{S_0}^{g^{\rightarrow}}(h^{t-1})$ and $\mathcal{L}_P^{g^{\rightarrow}}(h^{t-1})$ are significantly higher than $\mathcal{L}_{S_0}^{g^{t-1}}(h^{t-1})$. Therefore we have $r < r^{\rightarrow}$. Furthermore, with high probability, we will observe a manipulated sample S_t which was manipulated more than g^{t-1} would predict. Thus, g^{t-1} would not be in the updated sets of candidate graphs $\mathcal{G}_0^t$ consistent with the observed S_t . Similarly graphs $g_{r'}$ with $r' < r$ are eliminated from the candidate set. (2) $\mathcal{L}_{S_0}^{g^{\rightarrow}}(h^{t-1})$ and $\mathcal{L}_P^{g^{\rightarrow}}(h^{t-1})$ are not significantly higher than $\mathcal{L}_{S_0}^{g^{t-1}}(h^{t-1})$. In this case, the observed sample S_t would be consistent with the current g^{t-1} . In this case all candidate graphs $g_{r''}$ with $r < r''$ are eliminated from the candidate set. In the case in which $r > r^{\rightarrow}$, this obviously does not pose a problem. Now consider the case in which $r < r^{\rightarrow}$. Then for $h^* = \arg \min_{h \in \mathcal{H}_{\text{thres}}} \mathcal{L}^{g^{\rightarrow}}(h)$ we have that $\mathcal{L}_P^{g_r}(h^*) \leq \mathcal{L}_P^{g_{r^{\rightarrow}}}(h^*)$. Now assuming that S_0 is ε'' -representative for P with respect to $\mathcal{H}_{\text{thres}}$ and loss ℓ^g for every $g \in \mathcal{G}_{g.r.}$, then $\mathcal{L}_P^{g_r}(h^{t-1}) \leq \mathcal{L}_{S_0}^{g_r}(h^{t-1}) + \varepsilon'' \leq \mathcal{L}_{S_0}^{g_r}(h^*) + \varepsilon'' \leq \mathcal{L}_P^{g_r}(h^*) + 2\varepsilon'' \leq \mathcal{L}_P^{g^{\rightarrow}}(h^*) + 2\varepsilon''$. Thus, in this case, despite $r < r^{\rightarrow}$, the selected hypothesis is still close to optimal in terms of $\ell^{g^{\rightarrow}}$. Thus, the elimination of graphs with radius greater than r does not hinder success. Thus, the distinction of the two cases can be exploited by the algorithm to do a binary search on the candidate set.

We also note that the way the candidate hypotheses are picked, we always pick the maximal radius for a given sample loss. This leads to corresponding maximally robust hypotheses, allowing for the first hypotheses to be successful in the realizable case as shown in Observation 8.8.3.

8.8.2 General VC-Classes

We will now show that similar learning guarantees in our strategic batch learning setup are possible for more general hypothesis classes and graph classes. We will again consider the partial order on graph classes introduced in the previous section. We will show that the threshold algorithm generalizes to totally ordered graph classes if $VC(\mathcal{G} \times \mathcal{H})_{\ell^{\text{mani}}}$ is finite.

We can now generalize our threshold algorithm to an algorithm for totally ordered graph classes (Algorithm 3). This algorithm essentially works in the same way: It first identifies the maximal graph g_0 for which there is a hypothesis with maximum accuracy that is still robust with respect to g_0 . The corresponding optimal hypothesis is the first hypothesis $\hat{h}^0$ published by the algorithm. For the robust-realizable case, this is sufficient to guarantee success (see Observation 8.8.6), as we get uniform convergence of $\mathcal{H}$ both in terms of ℓ^m with respect to $\mathcal{G}$ and in terms of 0/1-loss, which is sufficient to guarantee uniform convergence in terms of the strategic loss. For the agnostic case other candidate graphs are generated in a similar fashion. The upper bound on the strategic loss is increased by ε -steps, and for each such increment the maximal graph g_i , which allows for a classifier $h \in \mathcal{H}$ with corresponding loss $\mathcal{L}_{S_0}^{g_i}(h)$ to be smaller than that bound, is identified. For these $\frac{1}{\varepsilon}$ many candidate graphs $\{g_0, \dots, g_{\frac{1}{\varepsilon}}\}$ we can then perform a kind of binary search by always publishing the optimal classifier with respect to the current median classification graph. We can then update the set of candidate graphs in each round by observing the manipulations caused by the published classifiers. We are guaranteed to always observe manipulations if robustness was under-estimated, giving the algorithm sufficient feedback for the binary search to terminate successfully. This yields the guarantee in Theorem 8.8.7.

Observation 8.8.6. Let $\text{VC}(\mathcal{H} \times \mathcal{G})_{\ell^{\text{mani}}} = d_1$ and $\text{VC}(\mathcal{H}) = d_2$. Furthermore let $\mathcal{G}$ be totally ordered with respect to $\mathcal{H}$. Then Algorithm 3 is a successful proper strategic batch learner in the realizable case with sample complexity $m_{\mathcal{H}, \mathcal{G}}^{\text{real}}(\varepsilon, \delta) = O(\frac{(d_1+d_2) \log(d_1+d_2) + \frac{1}{\delta}}{\varepsilon^2})$ and round complexity $T_{\mathcal{H}, \mathcal{G}}^{\text{real}}(\varepsilon, \delta) = 1$.

Theorem 8.8.7. *Let and $\text{VC}(\mathcal{H}) = d_2$ $\text{VC}(\mathcal{H}' \times \mathcal{G})_{\ell^{\text{mani}}} = d_1$, where $\mathcal{H}' = \mathcal{H} \cup \{h_S^{\text{max}} \in \{0, 1\} : S \in (\mathcal{X} \times \{0, 1\}^*)'\}$, as defined in the previous section. Furthermore, let $\mathcal{G}$ be totally ordered with respect to $\mathcal{H}$. Then Algorithm 3 is a successful proper strategic batch learner with sample complexity $m_{\mathcal{H}, \mathcal{G}}(\varepsilon, \delta) = O(\frac{(d_1+d_2) \log(d_1+d_2) + \log(\frac{1}{\delta\varepsilon})}{\varepsilon^2})$ and round complexity $T_{\mathcal{H}, \mathcal{G}}(\varepsilon, \delta) = O(\log(\frac{1}{\varepsilon}))$.*

Algorithm 4: Strategic Batch-Learning for totally ordered graph classes

Input: parameters $\varepsilon, \varepsilon'$, hypothesis class $\mathcal{H}$, graph class $\mathcal{G}$
receive sample $S_0 \sim P^m$
 $\mathcal{L}_0^{0/1} \leftarrow \inf_{h' \in \mathcal{H}} \mathcal{L}_{S_0}^{0/1}(h')$
for $i = 0, \dots, \frac{1}{\varepsilon}$ **do**
 $g_i \leftarrow \max_{\preceq_{\mathcal{H}}} \{g : \inf_{h' \in \mathcal{H}} \mathcal{L}_{S_0}^g(h') = \mathcal{L}_0^{0/1} + i \cdot \varepsilon\}$
end
set $\mathcal{G}^0 = [g_0, g_1, g_2, \dots, g_{\frac{1}{\varepsilon}}]$
 $\hat{h}^0 \leftarrow \arg \min_{h \in \mathcal{H}} \mathcal{L}_{S_0}^{g_0}(h)$
publish $\hat{h}^0$
receive sample S_1 , where $S'_1 \sim P^m$ and $S_t = S_t^{\Pi_t}$ for some sequence of
 $(g_{\rightarrow}, h^0)$ -induced manipulations Π_1 .
 $g_{\max} \leftarrow \max_{\preceq_{\mathcal{H}}} \{g \in \mathcal{G}^0 : L_{S'_1}^{\text{mani}}(g, \hat{h}^0) = 0\}$
 $g_{\min} \leftarrow \min_{\preceq_{\mathcal{H}}} \{g \in \mathcal{G}^0 : L_{S_0}^{\text{mani}}(g, \hat{h}^0) \geq L_{S_0}^{\text{mani}}(g, \hat{h}^0) - \varepsilon' \text{ and } L_{S'_1}^{\text{mani}}(g, \hat{h}^0) = 0\}$
 $\mathcal{G}^1 \leftarrow \{g \in \mathcal{G}^0 : g \preceq_{\mathcal{H}} g_{\max} \text{ and } g_{\min} \preceq_{\mathcal{H}} g\}$
 $k_1 = \lfloor \frac{|\mathcal{G}^1|}{2} \rfloor$
 $\hat{g}^1 \leftarrow \mathcal{G}^1[k_1]$
 $\hat{h}^1 \leftarrow \arg \min_{h \in \mathcal{H}'} \mathcal{L}_{S_0}^{\hat{g}^1}(h)$
for rounds $t = 2, \dots, T$ **do**
 publish $\hat{h}^{t-1}$
 receive sample S_t , where $S'_t \sim P^m$ and $S_t = S_t^{\Pi_t}$ for some sequence of
 $(g^{\rightarrow}, h^{t-1})$ -induced manipulations Π_t .
 $\mathcal{G}^t \leftarrow \{g \in \mathcal{G}^{t-1} : L_{S'_t}^{\text{mani}}(g, \hat{h}^{t-1}) = 0\}$
 $\mathcal{G}_0^t \leftarrow \{g \in \mathcal{G}^t : L_{S_0}^{\text{mani}}(g, \hat{h}^{t-1}) \geq \max_{g' \in \mathcal{G}^t} L_{S_0}^{\text{mani}}(g', \hat{h}^{t-1}) - \varepsilon'\}$
 if $\hat{g}^{t-1} \in \mathcal{G}_0^t$ **then**
 $\mathcal{G}^t \leftarrow [\mathcal{G}^t[0], \dots, \mathcal{G}^t[k_{t-1}]] \cap \mathcal{G}_0^t$
 else
 $\mathcal{G}^t \leftarrow \mathcal{G}_0^t$
 end
 $k_t \leftarrow \lfloor \frac{|\mathcal{G}^t|}{2} \rfloor$
 $\hat{g}^t \leftarrow \mathcal{G}^t[k_t]$
 $\hat{h}^t \leftarrow \min_{h \in \mathcal{H}'} \mathcal{L}_{S_0}^{\hat{g}^t}(h)$
end

We note that in the case where the maxima expressed within the algorithm always exist for a class, the output of the learner is proper. If those maxima do not exist, we can

use the extension of the maximally robust empirical risk minimization criterion discussed in the previous section and learn with respect to the extended class $\mathcal{H}' \supset \mathcal{H}$ with $\mathcal{H}' = \mathcal{H} \cup \{h_S^{\max} : S \in (\mathcal{X} \times \{0, 1\})^*\}$.

8.9 Conclusion

In this chapter, we introduced a new strategic loss, which incentivizes correct classification under strategic manipulations. We also incorporate the idea of social burden into our notion of loss. We differentiated this loss from previous formal frameworks designed to mitigate strategic manipulation. In particular, we showed that optimizing for our strategic loss can yield satisfactory classification rules, even if there is no incentive-compatible hypothesis in the class that performs well on the classification task at hand. In addition, we showed that finite VC classes are improperly learnable with respect to this loss, if the manipulation graph is known. We then explore learning for unknown manipulation graphs. In particular, we define a similarity between manipulation graphs that takes into account our loss function, and that optimizing with respect to this loss gives reasonable guarantees for learning. We show possible ways of learning the manipulation structure either from full manipulation information over a finite set, or from a distribution shift. Lastly, we investigate learning with respect to a class of possible manipulation graphs. We show that in the realizable case, hypothesis classes finite VC classes are improperly learnable with respect to this class of manipulation graphs which is ordered by inclusion. Lastly we propose a batch-learning model where a learner tries to find a hypothesis with close to optimal loss in a setting with unknown manipulation, but where the learner can observed distribution shifts between batches. We show sufficient conditions for learnability in this setting. The setting explored here is different from settings that only optimize for accuracy, rather than both accuracy and avoidance of social burden. This setting also does not consider the possibility of improvement, but only looks at feature changes through gaming. Considering our questions inspired by the fundamental theorem, our results give rise to the following observations: For strategic classification improper and proper learning are not the same even if the manipulation graphs are known. Agnostic and realizable learnability are equivalent in the case of known manipulation graphs. However in the case of unknown manipulation graphs, realizable learnability does not imply agnostic learnability.

8.10 Proofs

Observation 8.3.2. For any $d \in \mathbb{N} \cup \{\infty\}$ there exists a class $\mathcal{H}$ and a manipulation graph $g^\rightarrow = (\mathcal{X}, E_{g^\rightarrow})$ with $\text{VC}(\mathcal{H}) = 1$ and $\text{VC}(\mathcal{H}_{\ell^\rightarrow}) \geq d$.

Proof of Observation 8.3.2: Let $\mathcal{X}$ be some infinite domain. We treat the case $d \in \mathbb{N}$ first. We consider a set $\mathcal{U} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_d\}$ of d domain points and a disjoint set $\mathcal{V} = \{\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_{2^d}\}$ of 2^d domain points. We associate each subset of $\mathcal{U}$ with exactly one point in $\mathcal{V}$. We design a manipulation graph $\mathcal{M}$ by adding an edge $\mathbf{x}_i \rightarrow \mathbf{z}_j$ if and only if $\mathbf{x}_i$ is a member of the subset associated with $\mathbf{z}_j$, and include no other edges. Now we consider the class $\mathcal{H}$ of singletons over $\mathcal{V}$, that is $\mathcal{H} = \{h_1, h_2, \dots, h_{2^d}\}$ consists of 2^d functions and we have $h_j(\mathbf{x}) = \mathbf{1}[\mathbf{x} = \mathbf{z}_j]$. That is h_j assigns label 0 to every point in the domain except for $\mathbf{z}_j$. Then we have $\text{VC}(\mathcal{H}) = 1$. However the loss class $\mathcal{H}_{\ell^\rightarrow}$ shatters the set $\{(\mathbf{x}_1, 0), (\mathbf{x}_2, 0), \dots, (\mathbf{x}_d, 0)\}$ (and also the set $\{(\mathbf{x}_1, 1), (\mathbf{x}_2, 1), \dots, (\mathbf{x}_d, 1)\}$). Thus, $\text{VC}(\mathcal{H}_{\ell^\rightarrow}) \geq d$. For the case $d = \infty$, we will use the same construction with the set $\mathcal{U} = \mathbb{N}$ and $\mathcal{V}$ being an uncountable set, again indexed by the power-set of $\mathcal{U}$. $\square$

Observation 8.3.3. For any hypothesis class $\mathcal{H}$ and any manipulation graph $g^\rightarrow = (\mathcal{X}, E_{g^\rightarrow})$, we have $\text{VC}(\mathcal{H}) \leq \text{VC}(\mathcal{H}_{\ell^\rightarrow})$.

Proof of Observation 8.3.3: Let $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_d, y_d)\}$ be a set of points shattered by the hypothesis class $\mathcal{H}$. Since all elements of $\mathcal{H}$ are functions, we can assume $y_i = 1$ for all i . The same set of domain points $\{(\mathbf{x}_1, 1), (\mathbf{x}_2, 1), \dots, (\mathbf{x}_d, 1)\}$ with all labels set to 1 is then shattered by the loss class $\mathcal{H}_{\ell^\rightarrow}$. To see this, note that for a function $h \in \mathcal{H}$ and point $\mathbf{x}$, if $h(\mathbf{x}) = 1$, then $(\mathbf{x}, 1) \notin h_{\ell^\rightarrow}$ is not in the loss set. Thus, if the class $\mathcal{H}$ can shatter those points, the loss class will shatter them as well. $\square$

Theorem 8.3.4. For every $d \in \mathbb{N} \cup \{\infty\}$, there exists a hypothesis class $\mathcal{H}$ with $\text{VC}(\mathcal{H}_{\ell^\rightarrow}) = d$ that is learnable with sample complexity $O(\log(1/\delta)/\varepsilon)$ in the realizable case.

Proof of Theorem 8.3.4: We consider the class of singletons from the proof of Observation 8.3.2. Note that, if the class is realizable with respect to the strategic loss, then there exists at most one $\mathbf{z}_j$ such that this point $(\mathbf{z}_j, 1)$ with label 1 has a positive weight under this distribution. Further, the distribution can not assign any weight to points $(\mathbf{x}_i, 1)$, and it can only assign a positive weight to points $(\mathbf{x}_i, 0)$ if $\mathbf{x}_i$ is not a member of the subset corresponding to $\mathbf{z}_j$. Further, if the point $(\mathbf{z}_j, 1)$ is a member of the training sample, the learner can output hypothesis h_j and this hypothesis then has true strategic loss 0. If the training sample does not contain any point with label 1, the learner can output a hypothesis h_0 that assigns label 0 to all points. In that case, we get $\mathcal{L}_P^g(h_0) = P(\mathbf{z}_j, 1)$.

If the probability $P(\mathbf{z}_j, 1)$ was larger than ε , then, by the standard argument, a sample of the stated size being an ε -net for singletons would have contained the point (with high probability at least $1 - \delta$). $\square$

Theorem 8.3.8. (Adaptation of Theorem 1 and Theorem 4 in [Montasser et al., 2019])

There exists a hypothesis class $\mathcal{H}$ with $\text{VC}(\mathcal{H}) = 1$ that is not learnable with respect to the strategic loss by any proper learner $\mathcal{A}$ for $\mathcal{H}$ even in the realizable case. On the other hand, every class $\mathcal{H}$ of finite VC-dimension is learnable (by some improper learner).

Proof of Theorem 8.3.8: The construction to prove the first claim can be taken exactly as is in the proof of Theorem 1 by Montasser et al. [Montasser et al., 2019] by setting the label $+1$ there to 0 and the label -1 there to 1. The positive result on general learnability with an improper learner is derived by Montasser et al. [Montasser et al., 2019] by means of adapting a general compression scheme for VC-classes developed by Moran and Yehudayoff [Moran and Yehudayoff, 2016]. We can use the same adaptation to show learnability with respect to the strategic loss with the following modification: instead of just compressing a data sample S , the compression scheme for the adversarial loss first creates an inflated sample S_U which adds the perturbation sets (and then discretizing these using Sauer’s lemma to obtain a new finite data set). For the strategic loss, we would add the label conditioned perturbation sets (as described above), that is we inflate the sample by adding points $(\mathbf{x}', y)$, for $(\mathbf{x}_i, y)$ being the data point of smallest index with $\mathbf{x}_i \rightarrow \mathbf{x}'$. The discretization step and remaining compression and decompression technique can be used identically as in the adversarial loss case. $\square$

Theorem 8.4.2. *Let $\mathcal{H}$ be any hypothesis class and $\rightarrow, \rightsquigarrow$ two manipulation graphs. Then for any distribution P over $\mathcal{X} \times \mathcal{Y}$ and any $h \in \mathcal{H}$ we have*

$$\begin{aligned} \mathcal{L}_P^g(h) &\leq \mathcal{L}_P^{0/1}(h) + \mathcal{L}_P^{\rightsquigarrow, \perp}(h) + d_{H, P_{\mathcal{X}}}(\rightarrow, \rightsquigarrow) \\ &\leq 2\mathcal{L}_P^{\rightsquigarrow}(h) + d_{H, P_{\mathcal{X}}}(\rightarrow, \rightsquigarrow). \end{aligned}$$

Furthermore, by rearranging the result, we get

$$\frac{1}{2}\mathcal{L}_P^{\rightsquigarrow}(h) - d_{H, P_{\mathcal{X}}}(\rightarrow, \rightsquigarrow) \leq \mathcal{L}_P^g(h).$$

Proof of Theorem 8.4.2:

$$\begin{aligned}
\mathcal{L}_P^g(h) &= \mathcal{L}_P^{0/1}(h) + \mathcal{L}_P^{\rightarrow, \perp}(h) \\
&\quad - \mathbb{E}_{(x,y) \sim P}[h(x) = 0, y = 1, \exists x' : x \rightarrow x' : h(x') = 1] \\
&\leq \mathcal{L}_P^{0/1}(h) + \mathcal{L}_P^{\rightsquigarrow, \perp}(h) + |\mathcal{L}_P^{\rightarrow, \perp}(h) - \mathcal{L}_P^{\rightsquigarrow, \perp}(h)| \\
&\leq \mathcal{L}_P^{0/1}(h) + \mathcal{L}_P^{\rightsquigarrow, \perp}(h) + d_{H, P_{\mathcal{X}}}(\rightarrow, \rightsquigarrow) \\
&\leq 2\mathcal{L}_P^{\rightsquigarrow}(h) + d_{H, P_{\mathcal{X}}}(\rightarrow, \rightsquigarrow)
\end{aligned}$$

By exchanging $\rightarrow$ and $\rightsquigarrow$ and rearranging we get the second inequality. $\square$

Lemma 8.5.2. *Let $\text{VC}((\mathcal{H} \times \mathcal{G})_{\ell_{\text{graph}}}) = d$. Then there is $n_{\text{graph}} : (0, 1)^2 \mapsto \mathbb{N}$, such that for any marginal distribution $P_{\mathcal{X}}$ and any manipulation graph $\rightarrow$ for a sample $S = \{(\mathbf{x}_1, B_{\rightarrow}(\mathbf{x}_1)), \dots, (\mathbf{x}_n, B_{\rightarrow}(\mathbf{x}_n))\}$ of size $n \geq n(\varepsilon, \delta)$, we have with probability at least $1 - \delta$ over the sample generation $S_{\mathcal{X}} = (\mathbf{x}_1, \dots, \mathbf{x}_n) \sim P_{\mathcal{X}}^n$ for any $h \in \mathcal{H}$ and any $\rightsquigarrow \in \mathcal{G}$*

$$|\mathcal{L}_{(P_{\mathcal{X}}, \rightarrow)}^{\text{graph}}(h, \rightsquigarrow) - \mathcal{L}_S^{\text{graph}}(h, \rightsquigarrow)| < \varepsilon.$$

Furthermore, $n_{\text{graph}}(\varepsilon, \delta) \in O(\frac{d + \log \frac{1}{\delta}}{\varepsilon^2})$.

Proof of Lemma 8.5.2: The result follows directly from the finite VC-dimension of $\text{VC}((\mathcal{H} \times \mathcal{G})_{\ell_{\text{graph}}})$ and the fact that a finite VC-dimension of a class yields uniform convergence of the class (Theorem 6.7 and Theorem 6.8 from [Shalev-Shwartz and Ben-David, 2014a]). $\square$

Theorem 8.5.4. *Let $\text{VC}((\mathcal{H} \times \mathcal{G})_{\ell_{\text{graph}}}) = d$. Then there is $n_{\text{dist}} : (0, 1)^2 \mapsto \mathbb{N}$, such that for every marginal distribution $P_{\mathcal{X}}$ and every manipulation graph $\rightarrow$ for a sample $S = \{(\mathbf{x}_1, B_{\rightarrow}(\mathbf{x}_1)), \dots, (\mathbf{x}_n, B_{\rightarrow}(\mathbf{x}_n))\}$ of size $n \geq n(\varepsilon, \delta)$, we have with probability at least $1 - \delta$ over the sample generation $S_{\mathcal{X}} = (\mathbf{x}_1, \dots, \mathbf{x}_n) \sim P_{\mathcal{X}}^n$ for any $\rightsquigarrow \in \mathcal{G}$*

$$d_{\mathcal{H}, P_{\mathcal{X}}}(\rightarrow, \rightsquigarrow) < d_{\mathcal{H}, S_{\mathcal{X}}}(\rightarrow, \rightsquigarrow) + \varepsilon.$$

Furthermore, $n_{\text{dist}}(\varepsilon, \delta) \in O(\frac{d + \log \frac{1}{\delta}}{\varepsilon^2})$.

Proof of Theorem 8.5.4: Let $\rightsquigarrow \in \mathcal{G}$ and $S_{\mathcal{X}} \sim P_{\mathcal{X}}^n$. Then using Lemma 8.5.2 we get

with probability $1 - \delta$

$$\begin{aligned}
& |d_{\mathcal{H}, P_X}(\rightarrow, \rightsquigarrow) - d_{\mathcal{H}, S_X}(\rightarrow, \rightsquigarrow)| \\
&= \left| \sup_{h \in \mathcal{H}} \mathbb{E}_{x \sim P_X} [|\ell^{\rightarrow, \perp}(h, x) - \ell^{\rightsquigarrow, \perp}(h, x)|] \right. \\
&\quad \left. - \sup_{h \in \mathcal{H}} \sum_{x_i \in S_X} \ell^{\text{graph}}(h, \rightsquigarrow, x_i, B_{\rightarrow}(x_i)) \right| \\
&\leq \sup_{h \in \mathcal{H}} |\mathbb{E}_{x \sim P_X} [|\ell^{\rightarrow, \perp}(h, x) - \ell^{\rightsquigarrow, \perp}(h, x)|] \\
&\quad - \sum_{x_i \in S_X} \ell^{\text{graph}}(h, \rightsquigarrow, x_i, B_{\rightarrow}(x_i))| \\
&\leq \sup_{h \in \mathcal{H}} |\mathcal{L}_{(P_X, \rightarrow)}^{\text{graph}}(h, \rightsquigarrow) - \mathcal{L}_S^{\text{graph}}(h, \rightsquigarrow)| < \varepsilon.
\end{aligned}$$

□

Definition 8.10.1. Let $\ell^{\text{str}} : \mathcal{G}_{\text{all}} \times \mathcal{F} \times \mathcal{X} \times \{0, 1\}$ be the loss defined by

$$\ell^{\text{str}}(g, h, x, y) = \max\{\ell^{0/1}(h, x, y), \ell^{\text{mani}}(g, h, x)\}.$$

The loss class of $\mathcal{G} \times \mathcal{H}$ with respect to ℓ^{str} is defined by

$$\begin{aligned}
& (\mathcal{G} \times \mathcal{H})_{\ell^{\text{str}}} = \\
& \{ \{(x, y) \in \mathcal{X} \times \{0, 1\} : \ell^{\text{str}}(g, h, x, y) = 1\} : h \in \mathcal{H}, g \in \mathcal{G} \}.
\end{aligned}$$

Claim 8.10.2. Let $d = \text{VC}((\mathcal{G} \times \mathcal{H})_{\ell^{\text{mani}}}) + \text{VC}(\mathcal{H})$. Then, $\text{VC}((\mathcal{G} \times \mathcal{H})_{\ell^{\text{str}}}) \leq d \log(d)$.

Proof. We follow the same argument as before. Let us denote $(g, h)_{\ell^{\text{mani}}} = \{(x, y) \in \mathcal{X} \times \{0, 1\} : \ell^{\text{mani}}(g, h, x) = 1\}$ and $h_{\ell^{0/1}} = \{(x, y) \in \mathcal{X} \times \{0, 1\} : \ell^{0/1}(h, x, y) = 1\}$. Lastly, let $(g, h)_{\ell^{\text{str}}} = \{(x, y) \in \mathcal{X} \times \{0, 1\} : \ell^{\text{str}}(g, h, x, y) = 1\}$. We can easily see that $(g, h)_{\ell^{\text{str}}} = (g, h)_{\ell^{\text{mani}}} \cup h_{\ell^{0/1}}$. Thus, $(\mathcal{G} \times \mathcal{H})_{\ell^{\text{str}}} = \{A \cup B : A \in \mathcal{H}_{\ell^{0/1}}, B \in (\mathcal{G} \times \mathcal{H})_{\ell^{\text{mani}}}\}$. Thus, by standard arguments about VC-classes (e.g. exercises in [Shalev-Shwartz and Ben-David, 2014a]), we get the claimed result. □

8.10.1 Realizable Learning

Observation 8.7.5. There are classes $\mathcal{H}$ of finite VC-dimension and graph classes $\mathcal{G}$ which are totally ordered with respect to $\mathcal{H}$, such that there is no proper successful batch-learner for $\mathcal{H}$ and $\mathcal{G}$ for any finite sample and round complexity, not even in the realizable case.

Proof. This follows directly from Theorem 8.3.8 (which is an adaptation of Theorems 1 and 4 of [Montasser et al., 2019] for the strategic loss). The theorem states that there exists a class $\mathcal{H}$ and a fixed manipulation graph g of VC-dimension 1 which cannot be properly PAC-learned with respect to strategic loss by any proper learner. If we now consider this $\mathcal{H}$ and the graph class $\mathcal{G} = \{g\}$, then it is easy to see that $\mathcal{G}$ is totally ordered with respect to $\mathcal{H}$ (as it only has one element). Furthermore $\mathcal{H}$ was picked to have VC dimension 1. Furthermore if proper batch-learning was possible for $\mathcal{H}$ and $\mathcal{G}$, then proper PAC-learning would be possible for $\mathcal{H}$ with respect to ℓ^g , which we know to be impossible. This proves the theorem. $\square$

Theorem 8.7.6. Let $\text{VC}(\mathcal{H})$ be finite and let $\mathcal{G}$ be totally ordered with respect to $\mathcal{H}$. Then there is a strategically robust (improper) PAC-learner which is successful for every $g^\rightarrow \in \mathcal{G}$ in the strategically robustly realizable case (i.e. when $\inf_{h \in \mathcal{H}} L_P^{g^\rightarrow}(h) = 0$).

Proof. This is an adaptation of Theorem 4 from [Montasser et al., 2019] (and its adaptation to strategic loss, Theorem 8.3.8). The main difference here is that those theorems focus on the robustness with respect to a fixed (and known) manipulation graph, whereas in our setting we want to guarantee robustness with respect to an unknown element of a totally ordered graph class $\mathcal{G}$. Thus we cannot use any knowledge of this graph in the learning process, which makes it impossible to use robust empirical risk minimization (RERM) with respect to the true manipulation graph. However, we can use the realizability assumption and define Maximally Robust Empirical Risk Minimization (MRERM) with respect to a totally ordered graph class $\mathcal{G}$. First let us define

$$g_{\max}(S, \mathcal{H}, \mathcal{G}) = \max_{\preceq_{\mathcal{H}}} \{g \in \mathcal{G} : \text{for all } h \in \text{RERM}_{g, \mathcal{H}}(S) : L_S^g(h) = 0\}.$$

The set of maximally robust empirical risk minimizers with respect to $\mathcal{G}$ and $\mathcal{H}$ is then defined by $\text{MRERM}_{\mathcal{G}, \mathcal{H}}(S) = \text{RERM}_{g_{\max}(S, \mathcal{H}), \mathcal{H}}(S)$. We can now replace all use of the fixed deterministic $\text{RERM}_{g, \mathcal{H}}$ algorithm for a fixed manipulation graph (or perturbation sets) in the proof of Theorem 4 of [Montasser et al., 2019] by a fixed deterministic $\text{MRERM}_{\mathcal{G}, \mathcal{H}}$ -algorithm.

We note that for any $S' \subset S$, we have $g_{\max}(S, \mathcal{H}, \mathcal{G}) \preceq_{\mathcal{H}} g_{\max}(S', \mathcal{H}, \mathcal{G})$. Furthermore, we note that under the realizability assumption, we have $g_{\max}(S, \mathcal{H}, \mathcal{G}) \geq g^\rightarrow$ and thus $L_S^{g^\rightarrow}(\text{MRERM}_{\mathcal{G}, \mathcal{H}}(S)) = 0$ with probability 1 over the sample generation. We can now follow the proof of [Montasser et al., 2019] with some modified definitions, mainly replacing RERM with MRERM and replacing the inflated sample according to the true manipulation

graph (which we don't know) with an inflated sample according to the $g_{\max}(S, \mathcal{H}, \mathcal{G})$: For a training sample $S = \{(x_1, y_1), \dots, (x_m, y_m)\}$, we can now define the inflated sample according to the maximal graph that still allows realizability according to $\mathcal{H}$: Let the inflated (potentially infinite) sample $S_{g_{\max}}$ be defined as $S_{g_{\max}} = S \cup (\bigcup_{i \in \{1, \dots, m\}: y_i=0} \{(x, 0) : x \in B_{g_{\max}(S, \mathcal{H}, \mathcal{G})}(x_i)\})$. We now want to define a discretized version of this inflated sample. From standard PAC-learning theory we know that there is a positive integer $n \in O(\text{VC}(\mathcal{H}))$ that guarantees for any distribution D over $\mathcal{X} \times \{0, 1\}$ with $\inf_{h \in \mathcal{H}} L_D^{0/1}(h) = 0$, for n i.i.d. D -distributed samples $S' = \{(x'_1, y'_1), \dots, (x'_n, y'_n)\}$, with nonzero probability, every $h \in \mathcal{H}$ satisfying $L_{S'}^{0/1}(h) = 0$ also $L_D(h) \leq \frac{1}{3}$. Now let $\hat{\mathcal{H}} = \{MRERM_{\mathcal{G}, \mathcal{H}}(L) : L \subset S \text{ and } |L| = n\}$. We note that $|\hat{\mathcal{H}}| \leq |\{MRERM_{\mathcal{G}, \mathcal{H}}(L) : L \subset S \text{ and } |L| = n\}| \leq (\frac{em}{n})^n$. Now consider the dual space $\mathcal{W}$ of function $w_{(x,y)} : \mathcal{H} \rightarrow \{0, 1\}$ defined by $w_{(x,y)}(h) = \mathbf{1}[h(x) \neq y]$ and every $(x, y) \in S_{g_{\max}}$. The VC-dimension of $\mathcal{W}$ is now at most the dual VC-dimension of $\mathcal{H}$, which is known to be upper-bounded by $\text{VC}^* \leq 2^{\text{VC}(\mathcal{H})+1}$. We now define $\hat{S}_{g_{\max}}$ to be a subset of $S_{g_{\max}}$ which includes exactly one element $(x, y) \in S_{g_{\max}}$ for each distinct classification $\{w_{(x,y)}(h)\}_{h \in \hat{\mathcal{H}}}$ of $\hat{\mathcal{H}}$ realized function of $w_{(x,y)} \in \mathcal{W}$. By the Sauer lemma we have $|\hat{S}_{g_{\max}}| \leq (\frac{e|\hat{\mathcal{H}}|}{\text{VC}^*(\mathcal{H})})^{\text{VC}^*(\mathcal{H})}$, which for $m > 2\text{VC}(\mathcal{H})$ is at most $(\frac{e^2 m}{\text{VC}(\mathcal{H})})^{\text{VC}(\mathcal{H})\text{VC}^*(\mathcal{H})}$. We can now note that for any $T \in \mathbb{N}$ and any $h_1, \dots, h_T \in \hat{\mathcal{H}}$ if $\frac{1}{T} \sum_{t=1}^T \mathbf{1}[h_t(x) = y] > \frac{1}{2}$ for every $(x, y) \in \hat{S}_{g_{\max}}$, then $\frac{1}{T} \sum_{t=1}^T \mathbf{1}[h_t(x) = y] > \frac{1}{2}$ for every $(x, y) \in S_{g_{\max}}$ as well, which would then imply $L_S^{g_{\max}(S, \mathcal{G}, \mathcal{H})}(\text{Majority}(h_1, \dots, h_T)) = 0$, which implies $L_S^{g_{\rightarrow}}(\text{Majority}(h_1, \dots, h_T)) = 0$. We can now find these functions $h_1, \dots, h_T$ in exactly the same way as in [Montasser et al., 2019] (via using the α -Boost algorithm). The resulting classifier $\hat{h} = \text{Majority}(h_1, \dots, h_T)$ satisfies $L_S^{g_{\rightarrow}}(\hat{h}) = 0$. Furthermore we note that each of the classifiers h_t is the result of $MRERM_{\mathcal{G}, \mathcal{H}}(L_t)$ for some $L_t \subset S$ with $|L_t| = n$. Thus, the classifier $\hat{h}$ is representable as the value of an (order-dependent) reconstruction function of set size

$$\begin{aligned} nT &= O(\text{VC}(\mathcal{H}) \log(|\hat{S}_{g_{\max}}|)) \\ &= O(\text{VC}(\mathcal{H})^2 \text{VC}^*(\mathcal{H}) \log(\frac{m}{\text{VC}(\mathcal{H})})). \end{aligned}$$

Thus invoking Lemma 11 of [Montasser et al., 2019]¹ with respect to $L^{g_{\rightarrow}}$ if

$$m > c\text{VC}(\mathcal{H})^2 \text{VC}^*(\mathcal{H}) \log(\text{VC}(\mathcal{H}) \text{VC}^*(\mathcal{H}))$$

(for a sufficiently large numerical constant c), we have that with probability at least $1 - \delta$,

$$L_P^{g_{\rightarrow}}(\hat{h}) \leq$$

¹We need a slight adaptation from adversarial loss to strategic loss here, but the proof for this goes through exactly as is for strategic manipulation loss as argued in the proof of Theorem 8.3.8.

$$O(\text{VC}(\mathcal{H})^2 \text{VC}^*(\mathcal{H})) \frac{1}{m} \log\left(\frac{m}{\text{VC}(\mathcal{H})}\right) \log(m) + \frac{1}{m} \log\left(\frac{1}{\delta}\right).$$

This concludes our proof. $\square$

8.10.2 Batch Learning

Class of Thresholds

Observation 8.8.3. The Strategic Batch-Learning for Thresholds Algorithm (Algorithm 3) is a proper learner for the strategic-batch learning problem for $\mathcal{H}_{\text{thres}}$ and $\mathcal{G}_{\text{f.r.}}$ in the realizable case with sample complexity $m_{\mathcal{H},\mathcal{G}}^{\text{real}} = O\left(\frac{\log(\frac{1}{\delta})}{\varepsilon^2}\right)$ and round complexity $T_{\mathcal{H},\mathcal{G}}^{\text{real}} = 1$.

Proof. This is a special case of Observation 8.8.6, as the graphs in $\mathcal{G}_{\text{f.r.}}$ are totally ordered with respect to $\mathcal{F}$ (and thus also with respect to $\mathcal{H}_{\text{thres}}$). $\square$

Theorem 8.8.5. The Strategic Batch-Learning for Thresholds Algorithm (Algorithm 1) is a proper learner for the strategic-batch learning problem for $\mathcal{H}_{\text{thres}}$ and $\mathcal{G}_{\text{f.r.}}$ in the agnostic case with sample complexity $m_{\mathcal{H},\mathcal{G}}(\varepsilon, \delta) = O\left(\frac{\log(\frac{1}{\delta\varepsilon})}{\varepsilon^2}\right)$ and round complexity $T_{\mathcal{H},\mathcal{G}}(\varepsilon, \delta) = O(\log(\frac{1}{\varepsilon}))$.

Proof. We note that the graphs in $\mathcal{G}_{\text{f.r.}}$ are totally ordered with respect to $\mathcal{F}$ (and thus also with respect to $\mathcal{H}_{\text{thres}}$). We also find that $(\mathcal{G}_{\text{f.r.}} \times \mathcal{H}_{\text{thres}})_{\ell^{\text{mani}}}$ is the class of intervals over the real line and thus $C \in \text{VC}((\mathcal{G}_{\text{f.r.}} \times \mathcal{H}_{\text{thres}})_{\ell^{\text{mani}}}) = 2$. Therefore we can treat this Theorem as a special case of Theorem 8.8.7 (and refer the reader to the proof of that theorem). $\square$

General VC-Classes

Observation 8.6.4

- The VC-dimension of $(\mathcal{G}_{\text{f.r.}} \times \mathcal{H}_{\text{thres}})_{\ell^{\text{mani}}}$ is 2.
- Let $\mathcal{H}_{\text{half}} = \{h_w : w \in \mathbb{R}^d : h_w(x) \text{ iff } x^T w \geq 0\}$ the hypothesis class of linear half spaces and $\mathcal{G}_{\text{f.r.}}^d = \{g_r = (\mathcal{X}, E_{g_r}) : (x, x') \in E_{g_r} \text{ iff } \|x - x'\|_2 \leq r\}$ be the class of fixed-radius balls in $\mathbb{R}^d$. Then the VC dimension of $(\mathcal{G}_{\text{f.r.}}^d \times \mathcal{H}_{\text{half}})_{\ell^{\text{mani}}}$ is at most $2d$.

Proof. • We note that the elements of $(\mathcal{G}_{\text{f.r.}} \times \mathcal{H}_{\text{thres}})_{\ell^{\text{mani}}}$ correspond exactly to the class of intervals over the real line. The VC-dimension of that class is known to be 2 [Shalev-Shwartz and Ben-David, 2014a].

- We note, that the elements of $(\mathcal{G}_{\text{f.r.}}^d \times \mathcal{H}_{\text{half}})_{\ell^{\text{mani}}}$ are sets $C_{w,r} = \{x \in \mathbb{R}^d : -r \leq x^T w \leq 0\}$. Now if we take any set C of $2d + 1$ points, then there is one point $x \in C$ such that x is in the convex hull of the remaining points, i.e., $x \in \text{conv}(C \setminus \{x\})$. Now it is impossible for any $C_{w,r}$ to achieve $C_{w,r} \cap C = \{x\}$. Thus the set C cannot be shattered by $(\mathcal{G}_{\text{f.r.}}^d \times \mathcal{H}_{\text{half}})_{\ell^{\text{mani}}}$. Therefore $\text{VC}((\mathcal{G}_{\text{f.r.}}^d \times \mathcal{H}_{\text{half}})_{\ell^{\text{mani}}}) \leq 2d$.

□

Lemma 8.6.5. Let $\mathcal{G}, \mathcal{H}$ be such that $\text{VC}(\mathcal{G}_{\ell^{\text{mani}}}, \mathcal{H}) = d$. Then any Empirical Manipulation Estimation learner (such as Algorithm 2 below) has the following success guarantee for learning the manipulation graph:

For every marginal distribution $P_{\mathcal{X}}$, every $g^{\rightarrow} \in \mathcal{G}$, every $h \in \mathcal{H}$ and every sequence of $(g^{\rightarrow}, h)$ -induced manipulations Π , and every $\varepsilon, \delta \in (0, 1)$,

if $m \geq C \frac{d + \log(\frac{1}{\delta})}{\varepsilon^2}$ (for some universal constant C) with probability $\geq (1 - \delta)$, over samples $S_1 \sim P_{\mathcal{X}}^m, S_2 \sim P_{\mathcal{X}}^m$, for every $\hat{g} \in A_{\text{graph}}(S_1, S_2^{\Pi}, h)$,

$$\text{Dis}_{(P_{\mathcal{X}}, h)}(g^{\rightarrow}, \hat{g}) \leq \varepsilon.$$

and

$$g^{\rightarrow} \in A_{\text{graph}}(S_1, S_2^{\Pi}, h).$$

Proof. We note it is sufficient to show that there is a constant C , such that for i.i.d P samples S_1 and S_2 of size $m \geq C \frac{d + \log(\frac{1}{\delta})}{\varepsilon^2}$ with probability $1 - \delta$, we have $L_{S_1}^{\text{mani}} \in [L_{\max} - \frac{\varepsilon}{2}, L_{\max}]$, to guarantee that if we run A_{graph} with parameter $\frac{\varepsilon}{2}$, we get:

- $\text{Dis}_{(P_{\mathcal{X}}, h)}(g^{\rightarrow}, g) \leq \varepsilon$ for all $g \in A_{\text{graph}}(S_1, S_2^{\Pi}, h)$.
- and $g^{\rightarrow} \in A_{\text{graph}}(S_1, S_2^{\Pi}, h)$.

First we note, that for any samples S_1 and S_2 and the true manipulation graph $g^\rightarrow$, we have $L_{S_2^\Pi}^{\text{mani}}(g^\rightarrow, h) = 0$, as the sequence of mappings Π replaces every sample point which can be manipulated in S_2 (i.e., all sample points with $\ell(g^\rightarrow, h) = 1$), with a sample point x' with $h(x') = 1$ (i.e., a sample point with $\ell^{\text{mani}}(g^\rightarrow, h, x') = 0$). From $g^\rightarrow \in \mathcal{G}$, it follows that we have $g^\rightarrow \in \{g \in \mathcal{G} : L_{S_2^\Pi}^{\text{mani}}(g, h) = 0\}$. Then this implies

$$\begin{aligned} L_{\max} &= \max\{L_{S_1}^{\text{mani}}(g, h) : g \in \mathcal{G}, L_{S_2^\Pi}^{\text{mani}}(g, h) = 0\} \\ &\geq L_{S_1}^{\text{mani}}(g^\rightarrow, h). \end{aligned}$$

Now from VC-theory, we know that there exists a universal constant C , such that a sample size of $m \geq \frac{d + \log(\frac{1}{\delta})}{\varepsilon^2}$ guarantees that with probability $1 - \frac{\delta}{2}$ we have that a sample $S \sim P^m$ is $\frac{\varepsilon}{8}$ -representative with respect to $\mathcal{G}_{\ell^{\text{mani}}, h}$.

Now let S_1 and S_2 be $\frac{\varepsilon}{8}$ -representative with respect to $(\mathcal{G} \times \{h\})_{\ell^{\text{mani}}}$. We want to show that $L_{\max} - \frac{\varepsilon}{2} \leq L_{S_1}^{\text{mani}}(g^\rightarrow, h)$. We note that there exists $g_{\max} \in \mathcal{G}$, such that $L_{\max} = L_{S_1}^{\text{mani}}(g_{\max}, h)$ and $L_{S_2^\Pi}^{\text{mani}}(g_{\max}, h) = 0$.

Because of the $\frac{\varepsilon}{8}$ -representativeness of both samples, we get for every $g \in \mathcal{G}$, we have

$$\begin{aligned} |L_{S_1}^{\text{mani}}(g, h) - L_{S_2}^{\text{mani}}(g, h)| &\leq \\ |L_{S_1}^{\text{mani}}(g, h) - L_{P_X}^{\text{mani}}(g, h)| + |L_{S_2}^{\text{mani}}(g, h) - L_{P_X}^{\text{mani}}(g, h)| &\leq \frac{\varepsilon}{4}. \end{aligned}$$

Furthermore we note, that for any manipulation graph g , $|L_{S_2^\Pi}^{\text{mani}}(g, h) - L_{S_2}^{\text{mani}}(g, h)| \leq \frac{1}{|S_2|} |\{x \in S_2 : x \notin S_2^\Pi\}| = L_{S_2}^{\text{mani}}(g^\rightarrow, h)$.

It now follows that:

$$\begin{aligned} L_{\max} &= L_{S_1}^{\text{mani}}(g_{\max}, h) \\ &\leq |L_{S_1}^{\text{mani}}(g_{\max}, h) - L_{S_2}^{\text{mani}}(g_{\max}, h)| \\ &\quad + |L_{S_2}^{\text{mani}}(g_{\max}, h) - L_{S_2^\Pi}^{\text{mani}}(g_{\max}, h)| \\ &\leq \frac{\varepsilon}{4} + L_{S_2}^{\text{mani}}(g^\rightarrow, h) \\ &\leq \frac{\varepsilon}{2} + L_{S_1}^{\text{mani}}(g^\rightarrow, h). \end{aligned}$$

Putting everything together, we have $L_{S_1}^{\text{mani}}(g^\rightarrow, h) \in [L_{\max} - \frac{1}{\varepsilon}, L_{\max}]$. This concludes our proof. $\square$

Observation 8.6.6

If an un-manipulated sample S_1 is ε -representative with respect to $(\mathcal{G} \times \mathcal{H})_{\ell^{\text{mani}}}$, then it can be indefinitely re-used by A_{graph} for any hypothesis $h \in \mathcal{H}$ and any manipulated ε -representative samples S_2^Π . Thus if $\text{VC}(\mathcal{G} \times \mathcal{H})_{\ell^{\text{mani}}} = d$, then $m \geq C \frac{d + \log(\frac{1}{\delta})}{\varepsilon^2}$ (for some universal constant C), implies that with probability $1 - \delta$ any $S_1 \sim P^m$ is repeatedly reusable by A_{graph} to guarantee ε -success as in the Lemma above, This allows us to reuse the initial unmanipulated sample in all subsequent steps.

Proof. We note that in order for sample S_1 to guarantee success in Lemma 1 we only required it to be $\frac{\varepsilon}{4}$ -representative with respect to $\mathcal{G}_{\ell^{\text{mani}},h}$. If $\text{VC}((\mathcal{H} \times \mathcal{G})_{\ell^{\text{mani}}}) = d$, then there exists a sample size $m \geq C \frac{d + \log(\frac{1}{\delta})}{\varepsilon^2}$ that with probability $1 - \delta$ over the sample generation $S_1 \sim P^m$, we have that S_1 is $\frac{\varepsilon}{4}$ -representative with respect to $(\mathcal{G} \times \mathcal{H})_{\ell^{\text{mani}}}$. This implies that with probability $1 - \delta$, S_1 is $\frac{\varepsilon}{4}$ -representative w.r.t. $\mathcal{G}_{\ell^{\text{mani}},h}$ for all $h \in \mathcal{H}$ simultaneously, proving the observation. □

Observation 8.7.4.

- A graph class $\mathcal{G}$ is totally ordered with respect to the class of all hypotheses $\mathcal{F}$ if and only if for every distinct $g_1, g_2 \in \mathcal{G}$ either g_1 is a subgraph of g_2 or g_2 is a subgraph of g_1 .
- For $\mathcal{H}_1 \subset \mathcal{H}_2$ and two graphs g_1, g_2 $g_1 \preceq_{\mathcal{H}_2} g_2$ implies $g_1 \preceq_{\mathcal{H}_1} g_2$. Thus, if a graph class $\mathcal{G}$ is totally ordered with respect to $\mathcal{H}_2$ it is also totally ordered with respect to $\mathcal{H}_1$.
- If $\mathcal{G}$ is totally ordered with respect to $\mathcal{H}$, then $\text{VC}(\mathcal{G}_{\ell^{\text{mani}},\mathcal{H}}) \leq 1$.
- There are $\mathcal{G}$ and $\mathcal{H}$, such that $\text{VC}(\mathcal{H}) = d$ and $\mathcal{G}$ is totally ordered with respect to $\mathcal{H}$, but $\text{VC}((\mathcal{H} \times \mathcal{G})_{\ell^{\text{mani}}}) = \infty$.

Proof. • " $\rightarrow$ ": Let $\mathcal{G}$ be totally ordered with respect to $\mathcal{F}$. This means for any two graphs $g_1, g_2 \in \mathcal{G}$, we have either $g_1 \preceq_{\mathcal{F}} g_2$ or $g_2 \preceq_{\mathcal{F}} g_1$. Without loss of generality, assume $g_1 \preceq_{\mathcal{F}} g_2$. We want to show that this implies that g_1 is a subgraph of g_2 . For the purpose of contradiction, assume the opposite. This means that there exists an edge $(x, x') \in V_{g_1}$, such that $(x, x') \notin V_{g_2}$. Now consider any function $f \in \mathcal{F}$ with $f(x) = 0$ and $f(x') = 1$. Then $\ell^{\text{mani}}(g_1, f, x) = 1 \geq 0 \geq \ell^{\text{mani}}(g_2, f, x)$, which

contradicts $g_1 \preceq_{\mathcal{F}} g_2$.

” $\leftarrow$ ” Let $\mathcal{G}$ such that for every two graphs $g_1, g_2 \in \mathcal{G}$, we have either $E_{g_1} \subset E_{g_2}$ or $E_{g_2} \subset E_{g_1}$. Without loss of generality, assume $E_{g_1} \subset E_{g_2}$. We want to show that this implies $g_1 \preceq_{\mathcal{F}} g_2$. For the sake of contradiction, we assume the opposite. This means that there exists an $f \in \mathcal{F}$ and a $x \in \mathcal{X}$, such that $\ell^{\text{mani}}(g_1, f, x) = 1$ and $\ell^{\text{mani}}(g_2, f, x) = 0$. $\ell^{\text{mani}}(g_1, f, x) = 1$ implies that $f(x) = 0$ and that there is an $x' \in \mathcal{X}$ such that $f(x') = 1$ and such that $(x, x') \in E_{g_1}$. Now this implies that $(x, x') \in E_{g_2}$ as well. From $f(x) = 0$ and $f(x') = 1$ it follows that $\ell^{\text{mani}}(g_2, f, x) = 1$, contradicting our assumption.

- This follows directly from the definitions. If for g_1 and g_2 , we have $g_1 \preceq_{\mathcal{H}_2} g_2$ then for all $h \in \mathcal{H}_2$ and all $x \in \mathcal{X}$, we have $\ell^{\text{mani}}(g_1, h, x) = 1$ implies $\ell^{\text{mani}}(g_2, h, x) = 1$. Since $\mathcal{H}_1 \subset \mathcal{H}_2$ for all $h \in \mathcal{H}_1$ we thus have $\ell^{\text{mani}}(g_1, h, x) = 1$ implies $\ell^{\text{mani}}(g_2, h, x) = 1$. Thus $g_1 \preceq_{\mathcal{H}_1} g_2$.
- Take any $h \in \mathcal{H}$ and any $\{x_1, x_2\} = C \subset \mathcal{X}$. Assume that C was shattered by $\mathcal{G}_{\ell^{\text{mani}}, h}$. Then there is some g_1 , such that $\ell^{\text{mani}}(g_1, h, x_1) = 1$ and $\ell^{\text{mani}}(g_1, h, x_2) = 0$. We know that $\mathcal{G}$ is totally ordered with respect to $\mathcal{H}$. Thus we know that for any $g_2 \in \mathcal{G}$, we either have $g_1 \preceq_H g_2$, which implies that $\ell^{\text{mani}}(g_2, h, x_1) = 1$ or $g_2 \preceq_H g_1$ which implies $\ell^{\text{mani}}(g_2, h, x_2) = 0$. Thus there is no $g_2 \in \mathcal{G}$ with $\ell^{\text{mani}}(g_2, h, x_1) = 0$ and $\ell^{\text{mani}}(g_2, h, x_2) = 1$, contradicting that C is shattered by $\mathcal{G}_{\ell^{\text{mani}}, h}$. Thus $\text{VC}(\mathcal{G}_{\ell^{\text{mani}}, h}) \leq 1$. Since $h \in \mathcal{H}$ was picked arbitrarily we have $\text{VC}((\mathcal{G})_{\ell^{\text{mani}}, \mathcal{H}}) = \sup_{h \in \mathcal{H}} (\text{VC}(\mathcal{G}_{\ell^{\text{mani}}, h})) \leq 1$.
- For $\text{VC}((\mathcal{H} \times \mathcal{G})_{\ell^{\text{mani}}})$ to be infinite it is sufficient to find one g and a hypothesis class $\mathcal{H}$ such that $\text{VC}(\mathcal{H} \times \{g\})_{\ell^{\text{mani}}} = \infty$. We have seen previously that there are classes $\mathcal{H}$ with $\text{VC}(\mathcal{H}) = 1$ and manipulation graphs g , such that $\text{VC}(\mathcal{H} \times \{g\})_{\ell^{\text{mani}}} = \infty$. Lastly we note that if we pick $\mathcal{G} = \{g\}$, then $\mathcal{G}$ is trivially totally ordered with respect to $\mathcal{H}$.

□

Observation 8.8.6. Let $\text{VC}(\mathcal{H} \times \mathcal{G})_{\ell^{\text{mani}}} = d_1$ and $\text{VC}(\mathcal{H}) = d_2$. Furthermore let $\mathcal{G}$ be totally ordered with respect to $\mathcal{H}$. Then Algorithm 4 is a successful proper strategic batch learner in the realizable case with sample complexity $m_{\mathcal{H}, \mathcal{G}}^{\text{real}}(\varepsilon, \delta) = O\left(\frac{(d_1 + d_2) \log(d_1 + d_2) + \log(\frac{1}{\delta})}{\varepsilon^2}\right)$ and round complexity $T_{\mathcal{H}, \mathcal{G}}^{\text{real}}(\varepsilon, \delta) = 1$.

Proof. We assume to be in the realizable case, i.e. $\inf_{h \in \mathcal{H}} L_P^{g^\rightarrow}(h) = 0$. Thus with probability 1, the sample S_0 will be realizable under $L^{g^\rightarrow}$ as well. This implies $\inf_{h \in \mathcal{H}} L_{S_0}^{0/1}(h) = 0$. Therefore, in line 3 of Algorithm 3, we set $L_0^{0/1}$ to 0. We then determine g_0 to be the maximal graph according to $\preceq_{\mathcal{H}}$ to yield $\inf_{h \in \mathcal{H}} L_{S_0}^{g_0}(h) = 0$ (line 5). From this and the realizability assumption, it follows that $g^\rightarrow \preceq_{\mathcal{H}} g_0$. In line 8, we now define $\hat{h}^0$, to be $\arg \min_{h \in \mathcal{H}} L_{S_0}^{g_0}(h)$. From $g^\rightarrow \preceq_{\mathcal{H}} g_0$, it follows that $L_{S_0}^{g^\rightarrow}(\hat{h}^0) = 0$ as well. Now, we can choose a sample size $m \in O(\frac{(d_1+d_2)\log(d_1+d_2)+\log(\frac{1}{\delta})}{\varepsilon^2})$, large enough to guarantee $\frac{\varepsilon}{8}$ -representativeness with respect to $\mathcal{H}_{\ell_{g^\rightarrow}}, (\mathcal{G} \times \mathcal{H})_{\ell^{\text{mani}}}$ as well as $(\mathcal{G} \times \mathcal{H})_{\ell^{\text{str}}}$ simultaneously with probability at least $1 - \frac{\delta}{2}$. Now with probability $1 - \delta$, $S_0 \sim P^m$ and $S'_1 \sim P^m$ are both such $\frac{\varepsilon}{8}$ -representative samples. First we note, that this guarantees that $L_P^{g^\rightarrow}(\hat{h}_0) \leq \frac{\varepsilon}{8}$ with probability $1 - \frac{\delta}{2}$.

This implies that an $\frac{\varepsilon}{8}$ -representative sample S'_1 at most a fraction of $\frac{\varepsilon}{4}$ can be manipulated by a $(g^\rightarrow, \hat{h}_0)$ -induced manipulation. We now assume that the parameter ε in the algorithm, is the same as the ε in our sample complexity analysis (our algorithm lets us set this parameter. We can assume that we know in advance what the size of the first sample will be.). Now if S'_1 is $\frac{\varepsilon}{8}$ -representative with respect to $\ell^{g^\rightarrow}$, then for g_1 (as chosen in line 5), we have $L_{S'_1}^{g_1}(\hat{h}_0) \geq L_{S_0}^{g_1}(\hat{h}_0) - \frac{\varepsilon}{4} \geq \inf_{h \in \mathcal{H}} L_{S_0}^{g_1}(h) - \frac{\varepsilon}{4} = \frac{3\varepsilon}{4}$. Since $L_{S_0}^{0/1}(\hat{h}_0) = 0$, we have $L_{S'_1}^{0/1}(\hat{h}_0) \leq \frac{\varepsilon}{4}$. Thus $L_{S'_1}^{\text{mani}}(g_1, \hat{h}_0) \geq \frac{3\varepsilon}{4} - \frac{\varepsilon}{4} = \frac{\varepsilon}{2}$. Now we noticed before that only at most a fraction of $\frac{\varepsilon}{4}$ samples in S'_1 get replaced in S_1 . Thus g_1 does **not** fulfill $L_{S'_1}^{\text{mani}g_1}(\hat{h}_0) = 0$. Therefore $G_1 = \{g_0\}$. Which means that for all consecutive rounds t , we get $\hat{h}^t = \inf_{h \in \mathcal{H}} L_{S_0}^{g_0}(h)$, which yields the guarantee $L_P^{g^\rightarrow}(\hat{h}^t) \leq L_P^{g_0}(\hat{h}^t) \leq \frac{\varepsilon}{8}$. □

Theorem 8.8.7.

Let $\text{VC}(\mathcal{H} \times \mathcal{G})_{\ell^{\text{mani}}} = d_1$ and $\text{VC}(\mathcal{H}) = d_2$. Furthermore let $\mathcal{G}$ be totally ordered with respect to $\mathcal{H}$. Then Algorithm 4 is a successful proper strategic batch learner with sample complexity $m_{\mathcal{H}, \mathcal{G}}(\varepsilon, \delta) = O(\frac{(d_1+d_2)(\log(d_1+d_2))+\frac{1}{\delta\varepsilon}}{\varepsilon^2})$ and round complexity $T_{\mathcal{H}, \mathcal{G}}(\varepsilon, \delta) = O(\log(\frac{1}{\varepsilon}))$.

Proof. We will first argue that among the candidate graphs $\mathcal{G}^0$ we pick in round 0, there is a candidate graph g_i , which yields a close-to-optimal hypothesis h , given some amount of ε -representativeness and for certain choices of parameters.

Assume we have a sample S_0 which is $\frac{\varepsilon}{16}$ -representative with respect to $(\mathcal{G} \times \mathcal{H})_{\ell^{\text{str}}}$. Then S_0 is $\frac{\varepsilon}{16}$ -representative with respect to $\mathcal{H}_{\ell_g}$ for every $g \in \mathcal{G}$. Now let us run the

algorithm with the parameter " ε " set as $\frac{\varepsilon}{4}$. Let $\mathcal{G}^0$ be defined as in the algorithm (line 7). Furthermore for every $g_i \in \mathcal{G}^0$, define $h_i = \arg \min_{h \in \mathcal{H}} L_{S_0}^{g_i}(h)$ and $\mathcal{H}' = \{h_0, \dots, h_{\frac{4}{\varepsilon}}\}$. Now let h^* denote the optimal hypothesis $h^* = \arg \min_{h \in \mathcal{H}} L_P(g^\rightarrow)(h)$. Now let $j = \lceil \frac{4(L_{S_0}^{g^\rightarrow}(h^*) - L_0^{0/1})}{\varepsilon} \rceil + 1$. Now we compare h^* to h_j from the candidate set $\mathcal{H}'$. By definition of g_j and h_j , we know that $\frac{j\varepsilon}{4} + L_0^{0/1} = L_{S_0}^{g_j}(h_j) \leq L_{S_0}^{g_j}(h^*)$. Furthermore, we know that $L_P^{g^\rightarrow}(h^*) \leq L_{S_0}^{g^\rightarrow}(h^*) + \frac{\varepsilon}{16} \leq (\frac{(j-1)\varepsilon}{4} + L_0^{0/1}) + \frac{\varepsilon}{16} < \frac{j\varepsilon}{4} + L_0^{0/1}$. Thus $g^\rightarrow \preceq_{\mathcal{H}_{\text{thres}}} g^j$. Therefore, $L_P^{g^\rightarrow}(h_j) \leq L_P^{g^j}(h_j) \leq L_{S_0}^{g^j}(h_j) + \frac{\varepsilon}{16} \leq L_{S_0}^{g^j}(h^*) + \frac{\varepsilon}{16} = \frac{j\varepsilon}{4} + L_0^{0/1} + \frac{\varepsilon}{16} = \frac{(j-2)\varepsilon}{4} + L_0^{0/1} + \frac{9\varepsilon}{16} \leq L_{S_0}^{g^\rightarrow}(h^*) + \frac{9\varepsilon}{16} \leq L_P^{g^\rightarrow}(h^*) + \frac{10\varepsilon}{16}$. This shows that there is indeed a hypothesis in our candidate-set that is close to optimal.

We now note that the graph elimination that happens in lines 13 to 15 and lines 20 to 21 is equivalent to the estimation of Algorithm 2, which we know to keep the optimal graph if the samples encountered are $\frac{\varepsilon'}{8}$ -representative. We further note that the update in line 23 only occurs, if the sample $S_t = S_t^{\Pi}$ was not significantly manipulated according to $\hat{h}^{t-1}$. Thus g^{t-1} already sufficiently accounted for all possible manipulation caused by $g^\rightarrow$, which means that eliminating candidate graphs g with $g^{t-1} \preceq_{\mathcal{H}} g$ will not cause a miss-estimation of $g^\rightarrow$ that causes more than $2\varepsilon'$ difference in strategic loss. Thus the elimination of graphs will yield an approximately optimal hypothesis. Furthermore this is a kind of binary search which eliminates half the candidates in each step (as the algorithm always picks the median candidate in line 15 and line 26 respectively and the elimination either eliminates all graphs smaller or all graphs greater to the current candidate.) Therefore the algorithm needs at most $O(\log(\frac{1}{\varepsilon}))$ rounds.

Lastly, we note that a sample size of $m = O(\frac{(d_1+d_2) \log(d_1+d_2) + \log(\frac{\log(\frac{1}{\delta})}{\delta})}{\varepsilon^2})$ is sufficient to guarantee that in each of the $O(\log(\frac{1}{\varepsilon}))$ rounds, the probability of receiving an $\frac{\varepsilon}{16}$ -representative sample is at least $1 - \frac{\delta}{\log(\frac{1}{\varepsilon})}$. Thus via union bound the sample size of $m = O(\frac{(d_1+d_2) \log(d_1+d_2) + \log(\frac{\log(\frac{1}{\delta})}{\delta})}{\varepsilon^2})$ is sufficient to guarantee (ε, δ) -learning success. $\square$

Chapter 9

Adversarially Robust Learning

9.1 Introduction

Adversarial perturbations, imperceptibly small manipulations to input instances that cause unexpected failures of learned predictors, pose significant safety concerns for their employment. The phenomenon, first exposed a decade ago in image classification tasks [Szegedy et al., 2014], has since received substantial attention, both in the practical and theoretical machine learning research communities. Studies in both lines of research often skip over the question of how to suitably model “imperceptible perturbations” and rather investigate how to defend against various (fixed) types of attacks. In theoretical studies, this is typically modeled by defining an *adversarial loss* that, in addition to misclassification, penalizes the existence of a perturbed instance on which a classifier assigns a different label. This type of loss definition then crucially depends on a given *perturbation type*, that is some function $u : \mathcal{X} \rightarrow 2^{\mathcal{X}}$ that assigns every domain instance a set of points that would be viewed as an admissible perturbation (often balls of some radius with respect to some norm are considered).

However, in most realistic settings, exact perturbation sets to be used by an adversary are not plausibly available to a learner. Moreover, it has been shown that encouraging robustness with respect to a perturbation type that is not suitable for the task easily degrades classification accuracy [Tsipras et al., 2019a, Zhang et al., 2019, Yang et al., 2020]. In response, some work has then explored the question of learnability when the perturbation type is not known at all in advance [Montasser et al., 2021], but information about possible perturbations can be accessed through queries. It is not difficult to be convinced though that without any restrictions or knowledge of the perturbations to be

used, robust learning from samples only is impossible. Even for a small collection of possible perturbation types, as soon as these induce inconsistencies between optimal predictors a sample based learner cannot be simultaneously successful with respect to all options.

In this work we explore a middle ground between fixed and unknown perturbation types, which we term *learning with uncertain perturbation sets*. We will assume that the true perturbation type is a member of a fixed class of perturbation types. One can think of this as knowing that we should encourage robustness with respect to some type of norm induced ball, but may not be certain about which type of norm or which radius is most suitable for the task. We then study which structures (on the interplay between perturbation type class and the class of predictors to be learned) allow for PAC-type learnability guarantees.

Given a class $\mathcal{U}$ of perturbation types and hypothesis class $\mathcal{H}$, we define an $\mathcal{H}$ -induced partial order on $\mathcal{U}$. We show that when $\mathcal{U}$ is actually totally ordered in this sense, statistical (PAC) learning is possible in the realizable case as soon as $\mathcal{H}$ has finite VC-dimension. $\mathcal{U}$ being totally ordered applies to a wide variety of settings, for example the case where the various perturbation types are balls in some metric space while the radius varies over $\mathcal{U}$ (and actually also does not need to be constant over domain $\mathcal{X}$ for each $u \in \mathcal{U}$). However, we also show that, without the realizability assumption learning from samples alone is not feasible, even when $\mathcal{U}$ is totally ordered with respect to $\mathcal{H}$.

We thus explore two natural remedies: (1) We allow the learner to interact with a *perfect attack oracle* [Montasser et al., 2021]. Given an instance x and classifier f , such an oracle certifies robustness or provides an adversarial perturbation of x for f . Previous work on learning with unknown perturbation sets has employed such an oracle and shown learnability for classes with finite Littlestone dimension [Montasser et al., 2021]. In this work, we show that provided certain structures ($\mathcal{U}$ being totally ordered or a finite union of totally ordered perturbation type classes) learning with a perfect attack oracle becomes feasible for all classes of finite VC-dimension (which is a much wider space of natural hypothesis classes than those of finite Littlestone dimension). (2) We allow the learner to output a classifier that sometimes *abstains from prediction*. For this, we define a modification of the adversarial loss, where a hypothesis does not suffer loss when it abstains on an instance that was actually manipulated. We show that this again yields learnability for VC-classes when the perturbation type class is a finite union of totally ordered classes. We then consider the case in which $\mathcal{H}$ has finite disagreement-coefficient [Hanneke, 2007] and show that such $\mathcal{H}$ can be learned with respect to every class of perturbation types. And finally, we define a notion of $(\varepsilon, \mathcal{H})$ -cover for a class of perturbation types $\mathcal{U}$, introduce learners for this kind of cover and show that a class $\mathcal{H}$ is robustly learnable with respect to $\mathcal{U}$ if there is a finite-disagreement-cover learner for $(\mathcal{H}, \mathcal{U})$.

Our results on learning with abstentions show that different strategies for guarantees for different kinds of robustness (for example different types of perturbation sets) can be combined into an abstention learner that only predicts in the agreement region. The guarantees for each learning strategy then yield a guarantee for the combined classifier that depends on the number of strategies used. This highlights a different aspect to the power of abstention in adversarially robust learning from what has been established in prior work [Balcan et al., 2020b, 2023a]. We show that abstentions can also be used to address uncertainty in perturbation types.

9.1.1 Related work

Providing PAC type learning guarantees under adversarial robustness requirements has received an enormous amount of research attention in the past decade [Feige et al., 2015, Cullina et al., 2018, Wang et al., 2018, Awasthi et al., 2019, Montasser et al., 2019, Attias et al., 2019, Montasser et al., 2020, Ashtiani et al., 2020a, Montasser et al., 2021, Gourdeau et al., 2021a, 2022a, Montasser et al., 2022, Attias and Hanneke, 2022, Attias et al., 2022a, Bhattacharjee et al., 2021, Awasthi et al., 2022b,a, 2023a, Mao et al., 2023]. We will focus here on aspects within this literature that are most relevant to our settings.

Most prior works study the sample complexity of adversarial robustness with respect to a fixed type of adversarial perturbation (often a metric ball with a certain radius). However, recent research has developed frameworks of analysis that go beyond learning with respect to a fixed type of perturbations. It has been argued that the “correct” type of admissible adversarial perturbations (which originally were “imperceptible” modifications to input instances that lead to misclassification by a learned predictor) should depend on the underlying data-generating process [Bhattacharjee and Chaudhuri, 2021, Chowdhury and Urner, 2022]. These studies define a notion of “adaptive robustness” with respect to which a predictor should be evaluated. The drawback of these notions is that the correct perturbation type is not available to the learner and thus a predictor cannot be straightforwardly evaluated with respect to this loss. A different way of relaxing the assumption of a fixed, known perturbation type is provided in the framework of *tolerance* [Ashtiani et al., 2023a, Bhattacharjee et al., 2023]. Here, a learner is evaluated with respect to some fixed perturbation type, while being compared to the optimal achievable loss with respect to a larger perturbation type, relaxing the requirement for the learner and reflecting that the exact relevant perturbation type is typically not known (or even uniquely existing).

Most relevant to our work is a recent study on PAC learning with respect to *unknown perturbation sets* [Montasser et al., 2021]. This work provides learning guarantees for

the case that the robustness requirement will be with respect to an entirely unknown perturbation type without any prior knowledge about the nature of these perturbations. Those learning bounds assume access to a perfect attack oracle (as we do in some parts of this work), and require the class to have finite Littlestone dimension. In this work, we show that adding some structure to the class of perturbation types (as well as the promise that the true perturbation type will be a member of this class), allows for sample and query complexity bounds that are independent of the hypothesis class’s Littlestone dimension.

Finally, there is a long line of work studying the benefits of abstentions for classification tasks [Bartlett and Wegkamp, 2008, El-Yaniv and Wiener, 2010, Wiener and El-Yaniv, 2015, Yuan and Wegkamp, 2010, Zhang and Chaudhuri, 2016]. Some recent studies have also shown that the ability to abstain from prediction for manipulated instances can be beneficial for adversarial robustness requirements [Balcan et al., 2020b, 2023a]. However, the former studies a different setting for the generation of adversarial perturbations, and the latter considers fixed perturbation types.

9.1.2 Overview of results

In Section 9.2, we define our setup, introduce the notions of learnability and loss we investigate and provide an overview of the different kinds of prior knowledge and oracle access we employ. In particular, we there define our partial order on classes of perturbation types.

In Section 9.3 we analyze perturbation type classes for which this order is a total order. Theorem 9.3.1 states that every hypothesis class with finite VC dimension can be robustly learned in the realizable case with respect to any totally ordered perturbation type class. We then prove that this result cannot be extended to the agnostic case (Observation 9.3.2). This motivates investigating the benefits of a perfect attack oracle. Theorem 9.3.3 shows that the impossibility can be overcome with access to a perfect attack oracle and provides a finite sample and oracle-query bound for the agnostic case.

In Section 9.4, we investigate classes of perturbation types $\mathcal{U}$ that are not necessarily totally ordered. We first extend our previous results on learning with access to a perfect-attack-oracle to the case where $\mathcal{U}$ is a union of finitely many totally ordered perturbation classes (Theorem 9.4.1). We then establish that there are classes of perturbation types and hypothesis classes which cannot be learned without abstention, even with access to a perfect attack oracle (Observation 9.4.3), but which can be learned with abstention without the need for access to a perfect-attack-oracle (Observation 9.4.4). This motivates

our investigations into learning with respect to our adversarial abstention loss (Definition 9.2.5).

We show that in the realizable case, every hypothesis class with finite VC-dimension can be learned with respect to a finite union of totally ordered perturbation types in Theorem 9.4.5. We then consider the case in which $\mathcal{H}$ has finite disagreement-coefficient and show that such $\mathcal{H}$ can be learned with respect to every class of perturbation types (Theorem 9.4.7). Lastly, we define a notion of $(\varepsilon, \mathcal{H})$ -cover for a class of perturbation types $\mathcal{U}$, which states that for every $u \in \mathcal{U}$ there is a close-to-optimal hypothesis for u in the cover (Definition 9.4.8) and introduce the notion of learners for this kind of cover (Definition 9.4.9). We then generalize our previous results by showing that a class $\mathcal{H}$ is robustly learnable with respect to $\mathcal{U}$ if there is a finite-disagreement-cover learner for $(\mathcal{H}, \mathcal{U})$ (Theorem 9.4.10). While we provide some proof sketches in the main body of the paper, the detailed proofs for all results are in the supplement.

9.2 Setup

For sets A and B , we let 2^A denote the powerset of A and A^B the set of all functions from B to A . Let $\mathcal{X}$ be a domain and $\mathcal{Y} = \{0, 1\}$ be a label space. We model the data generation as a distribution P over $\mathcal{X} \times \mathcal{Y}$, and let $P_{\mathcal{X}}$ denote its marginal over $\mathcal{X}$. A *hypothesis* or *predictor* is a function $h : \mathcal{X} \rightarrow \{0, 1, \star\}$. Thus, on input $x \in \mathcal{X}$, a hypothesis h either predicts label 0 or 1 or abstains by outputting $\star$. We call a hypothesis *non-abstaining* if $h(x) \neq \star$ for all $x \in \mathcal{X}$. A *hypothesis class* $\mathcal{H}$ is a set of non-abstaining predictors. We will let $\mathcal{F} = \{0, 1, \star\}^{\mathcal{X}}$ denote the set of all predictors over $\mathcal{X}$.

The performance of a predictor h is measured by a loss function $\ell : \mathcal{F} \times \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$. For loss ℓ , we use the notation $\mathcal{L}_P(h) = \mathbb{E}_{(x,y) \sim P}[\ell(h, x, y)]$, to denote the *expected loss* of predictor h with respect to distribution P and $\mathcal{L}_S(h) = \frac{1}{|S|} \sum_{i=1}^{|S|} \ell(h, x_i, y_i)$ the *empirical loss* on dataset $S = ((x_1, y_1), \dots, (x_m, y_m))$. Further, for a hypothesis class $\mathcal{H}$ we will let $\text{opt}_P(\mathcal{H}) = \inf_{h \in \mathcal{H}} \mathcal{L}_P(h)$ denote the *approximation error* of class $\mathcal{H}$ with respect to P and loss ℓ .

The standard loss for classification tasks is the binary loss $\ell^{0/1}(h, x, y) = \mathbf{1}[h(x) \neq y]$, where $\mathbf{1}[\cdot]$ denotes the indicator function. The standard loss for adversarial robustness additionally penalizes h on instance (x, y) for the existence of a perturbation on x that h does not label with y . This loss is defined for a specified *perturbation type*, which is a function $u : \mathcal{X} \rightarrow 2^{\mathcal{X}}$ that assigns every instance $x \in \mathcal{X}$ a *set* $u(x) \subseteq \mathcal{X}$ with $x \in u(x)$ of

admissible perturbations. The *adversarial loss* with respect to u is then defined as

$$\ell^u(h, x, y) = \mathbf{1}[\exists z \in u(x) : h(z) \neq y].$$

We will use the notation $\ell^{u,\perp}(h, x)$ to denote the *adversarial component loss*, which indicates whether a domain point x lies close to the decision boundary with respect to the perturbation type u :

$$\ell^{u,\perp}(h, x) = \mathbf{1}[\exists z \in u(x) : h(z) \neq h(x)].$$

For hypotheses that have the option to abstain from prediction, we propose a variation of the adversarial loss that allows for a predictor to abstain from prediction on perturbed instances (but not on unperturbed instances) without suffering loss. The following definition captures this modification:

Definition 9.2.1 (Adversarial abstention loss). *For a perturbation type $u : \mathcal{X} \rightarrow 2^{\mathcal{X}}$, the adversarial abstention loss is defined by*

$$\ell^{u,\text{abst}}(h, x, y) = \begin{cases} 1 & \text{if } h(x) \neq y \\ 1 & \text{if } \exists z \in u(x) \text{ such that } h(z) \neq y \text{ and } h(z) \neq \star \\ 0 & \text{otherwise} \end{cases}$$

The main difference with adversarial loss is that if $h(z) = \star$ for an adversarial point $z \in u(x)$ with $z \neq x$, then there is no penalty and $\ell^{u,\text{abst}}(h, x, y) = 0$, thus allowing the predictor to abstain on points that were perturbed. This definition models the scenario where if the predictor can correctly detect that an adversarial attack has happened and abstains, then it is not penalized. However, abstaining on non-perturbed points is still penalized. Note that in case a predictor h never abstains (that is $h(x) \neq \star$ for all $x \in \mathcal{X}$) the two definitions of adversarial loss coincide. We let $\mathcal{L}_P^u(h)$, $\mathcal{L}_P^{u,\text{abst}}(h)$, $\mathcal{L}_S^u(h)$, $\mathcal{L}_S^{u,\text{abst}}(h)$ denote the corresponding expected and empirical losses, and $\text{opt}_P^u(\mathcal{H})$, $\text{opt}_P^{u,\text{abst}}(\mathcal{H})$ the corresponding approximation errors of a class $\mathcal{H}$ and distribution P .

Uncertain perturbation type In the literature, adversarial robustness is mostly treated with respect to a fixed (known) perturbation type u , while learning with respect to an entirely unknown perturbation type has also been investigated. In this work, we introduce a setting that naturally interpolates between these two extremes. We assume that the true perturbation type u^* is unknown to the learner, but promised to lie within a fixed *class* $\mathcal{U}$ of *perturbation types*. That is, the learner has prior knowledge of a class $\mathcal{U}$ with $u^* \in \mathcal{U}$, where u^* is the perturbation type that the learner will be evaluated with (see Definition

9.2.3 below). We can think of u^* as the perturbation type that the true adversarial environment will employ to manipulate input instances and induce misclassifications. We let $\mathcal{U}^{\text{all}}$ denote the class that contains all possible perturbation types.

It is easy to see that without any restrictions on the class $\mathcal{U}$ and any capability of the learner to (interactively) gather information about u^* and no option to abstain, a learner cannot succeed (see also Observations 9.3.2 and 9.4.2 below). To gain control over potentially infinite classes of perturbation types, we will now define a partial order on a class $\mathcal{U}$ that is induced by a hypothesis class $\mathcal{H}$. This structure will prove to be conducive for learnability.

Definition 9.2.2. Let $\mathcal{H} \subseteq \{0, 1\}^{\mathcal{X}}$ be a hypothesis class. For perturbation types u_1, u_2 we say that u_1 is smaller than u_2 with respect to class $\mathcal{H}$, and write $u_1 \preceq_{\mathcal{H}} u_2$, if and only if for all $h \in \mathcal{H}$ and all $x \in \mathcal{X}$ we have

$$\ell^{u_1, \perp}(h, x) \leq \ell^{u_2, \perp}(h, x).$$

We say a set of perturbation types $\mathcal{U}$ is totally ordered, with respect to $\preceq_{\mathcal{H}}$, if for every $u_1, u_2 \in \mathcal{U}$, we have either $u_1 \preceq_{\mathcal{H}} u_2$ or $u_2 \preceq_{\mathcal{H}} u_1$.

We have $u_1 \preceq_{\mathcal{H}} u_2$ if, for every function $h \in \mathcal{H}$ the margin area (the points which receive positive adversarial component loss) with respect to u_1 is included in the margin area with respect to u_2 . This is, for example, the case if all perturbation sets with respect to u_1 are included in those induced by u_2 .

Resources to learners and learnability A learner is a function $A : (\mathcal{X} \times \mathcal{Y})^* \rightarrow \mathcal{F}$ that takes in a finite sequence $S = ((x_1, y_1), \dots, (x_m, y_m))$ of labeled examples and outputs a predictor $A(S) : \mathcal{X} \rightarrow \{0, 1, \star\}$. We will study the following PAC-type [Valiant, 1984] notion of learnability of hypothesis classes $\mathcal{H}$ with perturbation type classes $\mathcal{U}$.

Definition 9.2.3 (Learnability without abstentions). Let $\mathcal{H} \subseteq \{0, 1\}^{\mathcal{X}}$ be a hypothesis class and $\mathcal{U}$ a class of perturbation types. We say that $\mathcal{H}$ is robustly learnable with respect to perturbation class $\mathcal{U}$ if there exists a learner A and a function $m : (0, 1)^2 \rightarrow \mathbb{N}$ such that for every $\varepsilon, \delta > 0$, every $m \geq m(\varepsilon, \delta)$, every $u^* \in \mathcal{U}$ and every data-generating distribution P we have

$$\mathbb{P}_{S \sim P^m}[\mathcal{L}_P^{u^*}(A(S)) \leq \text{opt}_P^{u^*}(\mathcal{H}) + \varepsilon] \geq 1 - \delta.$$

We speak of learnability in the realizable case if the above requirement is relaxed to only hold for all distributions P with $\text{opt}_P^{u^*}(\mathcal{H}) = 0$. Without this assumption, we also speak of learnability in the agnostic case.

If the above definition is fulfilled for a class $\mathcal{H}$, then the function $m : (0, 1)^2 \rightarrow \mathbb{N}$ in the above definition is an upper bound on the *sample complexity* of the learning task. Note that, while we don't require the learner in the above definition to output a non-abstaining hypothesis, predicting $\star$ will always cause loss with respect to ℓ^u . We can thus assume all considered predictors are non-abstaining classifiers for this setting.

Note that the above definition generalizes and unifies previous notions of adversarially robust learnability. We obtain learnability with respect a *known perturbation type* (the setting that is mostly considered in prior works) when $\mathcal{U} = \{u^*\}$ is a singleton class. Learning with *unknown perturbations* [Montasser et al., 2021] on the other hand, is the setting where $\mathcal{U}$ is the set of all possible perturbation types. In this work, we are interested in studying more fine-grained conditions on the structure of (the combination of) $\mathcal{H}$ and $\mathcal{U}$ that allow for learnability. We will call the case where $\mathcal{U}$ is neither a singleton set nor the set of all possible perturbation types *learning with uncertain perturbation sets*.

We will show that, even when the class of perturbation types $\mathcal{U}$ in consideration is totally ordered with respect to hypothesis class $\mathcal{H}$, agnostically learning $\mathcal{H}$ with $\mathcal{U}$ is impossible even in very simple cases (see Observation 9.3.2). We will thus, as has been done in earlier studies [Ashtiani et al., 2020a, Montasser et al., 2021] in addition allow the learner access to a *perfect attack oracle*, which we model as follows:

Definition 9.2.4 (Perfect Attack Oracle). *A perfect attack oracle for perturbation type u is a function $\mathcal{O}_u : \mathcal{F} \times \mathcal{X} \rightarrow \mathcal{X} \cup \{\text{robust}\}$, that takes as input a non-abstaining predictor f and a domain point x and either certifies that f is robust on x with respect to u by outputting `robust` or returns an admissible perturbation of x for f . That is*

$$\mathcal{O}_u(f, x) = \begin{cases} \text{robust} & \text{if } f(z) = f(x) \text{ for all } z \in u(x) \\ z \in u(x) & \text{with } f(x) \neq f(z) \text{ otherwise} \end{cases}$$

We say that a class $\mathcal{H}$ and perturbation class $\mathcal{U}$ are *learnable with access to a perfect attack oracle* if there exists a function $n : (0, 1)^2 \rightarrow \mathbb{N}$ such that the conditions in Definition 9.2.3 are satisfied for a learner that additionally makes at most $n(\varepsilon, \delta)$ queries to the perfect attack oracle. The function $n(\cdot, \cdot)$ then is an upper bound on the *attack oracle complexity* of the learning problem.

In Section 9.4 we will consider more general pairs of hypothesis and perturbation classes $\mathcal{H}$ and $\mathcal{U}$. We will show that if $\mathcal{U}$ is not necessarily totally ordered with respect to $\mathcal{H}$ (or a finite union of such totally ordered classes) there are tasks on which no learner (even with access to a perfect attack oracle) can succeed (see Observation 9.4.3 below). For such cases we will explore learnability with respect to the adversarial abstention loss:

Definition 9.2.5 (Learnability with abstentions). *Let $\mathcal{H} \subseteq \{0, 1\}^{\mathcal{X}}$ be a hypothesis class and $\mathcal{U}$ a class of perturbation types. We say that $\mathcal{H}$ is robustly learnable with abstentions with respect to perturbation class $\mathcal{U}$ if there exists a learner A and a function $m : (0, 1)^2 \rightarrow \mathbb{N}$ such that for every $\varepsilon, \delta > 0$, every $m \geq m(\varepsilon, \delta)$, every $u^* \in \mathcal{U}$ and every data-generating distribution P we have*

$$\mathbb{P}_{S \sim P^m}[\mathcal{L}_P^{u^*, \text{abst}}(A(S)) \leq \text{opt}_P^{u^*}(\mathcal{H}) + \varepsilon] \geq 1 - \delta.$$

Note that since $\mathcal{H}$ contains only non-abstaining predictors, we have $\text{opt}_P^{u^*}(\mathcal{H}) = \text{opt}_P^{u^*, \text{abst}}(\mathcal{H})$.

Discussion on additional parameters and assumptions Throughout this paper, we will also work with various standard complexity measures, such as bounded VC-dimension or bounded Littlestone dimension of the hypothesis class $\mathcal{H}$, bounded VC-dimension of the loss class or the adversarial component loss class. We refer the reader to the appendix for a reminder of the definitions of these notions.

We will further assume that learners have the capability to identify empirical risk minimizing hypotheses from a class. It is standard (though often implicit) to assume this as oracle access for the learner as well. We consider the standard ERM oracle, a robust ERM (RERM) oracle (for a fixed perturbation type u) and a maximally robust ERM (MRERM) oracle (for a class $\mathcal{U}$ of perturbation types, and realizable samples only). We define the following oracles:

$$\begin{aligned} \text{ERM}_{\mathcal{H}} : S &\mapsto h_S \in \arg \min_{h \in \mathcal{H}} \mathcal{L}_S^{0/1}(h) \\ \text{RERM}_{\mathcal{H}}^u : S &\mapsto h_S^u \in \arg \min_{h \in \mathcal{H}} \mathcal{L}_S^u(h) \\ \text{MRERM}_{\mathcal{H}}^{\mathcal{U}} : S &\mapsto \begin{cases} (h_S^{u_S^*}, u_S^*) \text{ with } h_S^{u_S^*} \in \arg \min_{h \in \mathcal{H}} \mathcal{L}_S^{u_S^*}(h), \\ \quad \text{and } u_S^* \text{ is } \preceq_{\mathcal{H}} \text{ maximal in the set } \{u \in \mathcal{U} \mid \min_{h \in \mathcal{H}} \mathcal{L}_S^u(h) = 0\} \\ \text{error if } \mathcal{L}_S^u(h) > 0 \text{ for all } u \in \mathcal{U} \text{ and } h \in \mathcal{H} \end{cases} \end{aligned}$$

That is, we will assume that $\text{MRERM}_{\mathcal{H}}^{\mathcal{U}}$ will return an error if the input sample S is not ℓ^u -realizable for any $u \in \mathcal{U}$. If, on the other hand, there does exist a $u \in \mathcal{U}$ for which S is ℓ^u -realizable, then $\text{MRERM}_{\mathcal{H}}^{\mathcal{U}}$ will return a maximal (with respect to $\preceq_{\mathcal{H}}$) such perturbation type u_S^* and corresponding hypothesis $h_S^{u_S^*}$ with $\mathcal{L}_S^{u_S^*}(h_S^{u_S^*}) = 0$. In particular, we assume that for every non-empty sample S there exists such a maximal element $u_S^* \in \mathcal{U}$ and corresponding ERM hypothesis from $\mathcal{H}$. While these do not always exist in $\mathcal{U}$ and $\mathcal{H}$ a priori, it has recently been shown that it is possible to embed $\mathcal{U}$ and $\mathcal{H}$ into slightly

larger classes so that the $\text{MRERM}_{\mathcal{H}}^{\mathcal{U}}$ oracle is always well defined [Lechner et al., 2023b]. See Appendix Section 9.6 for more details on this.

Since we focus on studying sample complexity (independently of computational considerations) and consider learners as functions throughout, assuming access to the above oracles is not restricting the validity of our bounds. It is still interesting to understand when is their existence reasonable to assume. Both ERM and RERM oracles have been widely used in the literature and they are often computationally hard to implement. Assuming their existence, however, an MRERM oracle can be easily implemented for the case where $\mathcal{U}$ is finite and totally ordered by running a binary search over $\mathcal{U}$ and calling RERM for the comparison step. If we assume $\mathcal{U}$ is parametrized by a k -bit number, then this requires $O(k)$ queries to RERM.

9.3 Learning with a totally ordered perturbation class

We will start with investigating the case where the perturbation class $\mathcal{U}$ is totally ordered with respect to hypothesis class $\mathcal{H}$ (see Definition 9.2.2). Note that if two perturbation types u_1 and u_2 we satisfy $u_1(x) \subseteq u_2(x)$ for all $x \in \mathcal{X}$, then u_1 is smaller than u_2 , $u_1 \preceq_{\mathcal{H}} u_2$, for every hypothesis class $\mathcal{H} \subset \{0, 1\}^{\mathcal{X}}$. This is, for example, the case when u_1 and u_2 assign balls (with respect to some metric) centered at x to every domain point x and u_2 always assigns a ball of radius at least as large as the ball assigned by u_1 (while the balls assigned by u_1 and u_2 are not necessarily required to have the same radii uniformly over the space $\mathcal{X}$).

In this section, we will focus on learning with respect to the standard adversarial loss and thus only consider non-abstaining predictors. We first show that for $\mathcal{H}$ and totally ordered $\mathcal{U}$, whenever the class $\mathcal{H}$ has bounded VC-dimension we get learnability in the realizable case. This result is based on adapting a compression argument for adversarially robust learning [Montasser et al., 2019] to the case of uncertain perturbation sets. A very similar adaptation has recently been made for the related problem of strategic classification [Lechner et al., 2023b]. The proof of this result can be found in the appendix.

Theorem 9.3.1. *Let $\mathcal{U}$ be a perturbation class that is totally ordered with respect to a hypothesis class $\mathcal{H}$ with $\text{VC}(\mathcal{H}) = d < \infty$. Then $\mathcal{H}$ is learnable in the realizable case with respect to $\mathcal{U}$ with sample complexity $O(\frac{d \cdot 2^d \cdot \log(1/\delta)}{\epsilon})$.*

Recall that realizability here means $\text{opt}_P^{u^*}(\mathcal{H}) = 0$ for the true perturbation type u^* (with respect to which the learner is evaluated). We now show that without this assumption

learning guarantees are impossible without equipping the learner with additional resources (or weakening the success criteria).

Observation 9.3.2. There exists a class $\mathcal{H}$ with $\text{VC}(\mathcal{H}) = 1$ and a perturbation class $\mathcal{U}$ that is totally ordered with respect to $\mathcal{H}$ that is not learnable (in the sense of Definition 9.2.3).

We will now show that allowing a learner access to a perfect attack oracle yields learnability in the agnostic case.

Theorem 9.3.3. *Let $\mathcal{U}$ be a perturbation class that is totally ordered with respect to a hypothesis class $\mathcal{H}$ with $\text{VC}(\mathcal{H}) = d < \infty$ and assume access to a perfect attack oracle. Then $\mathcal{H}$ is learnable in the agnostic case with respect to $\mathcal{U}$ with sample complexity $m(\varepsilon, \delta) = O(\frac{d \cdot 2^d \cdot \log(1/\delta)}{\varepsilon^2})$ and oracle complexity $O(m(\varepsilon, \delta)^2)$.*

Proof. We will employ a well-known reduction from agnostic learning to realizable compression-based learning [Montasser et al., 2019]. Since our proof for Theorem 9.3.1 for the realizable case of learning with uncertain, totally ordered perturbation sets employs a compression-based learner, for the agnostic case it suffices to show that, given any sample $S = ((x_1, y_1), \dots, (x_m, y_m))$, we can identify a *largest* subsample S' of S that is realizable with respect to the underlying perturbation type u^* . We will now outline how the perfect attack oracle can be employed to achieve this.

By employing the $\text{MRERM}_{\mathcal{H}}^{\mathcal{U}}$ -oracle, the learner can generate an ordered list $T = ((S_1, h_1, u_1), \dots, (S_n, h_n, u_n))$ of subsamples of S that are realizable with respect to some $u \in \mathcal{U}$, such that $(h_i, u_i) = \text{MRERM}_{\mathcal{H}}^{\mathcal{U}}(S_i)$ for all i and $u_i \preceq_{\mathcal{H}} u_j$ for all $i \leq j$. Note that we have $u^* \preceq_{\mathcal{H}} u_i$ if and only if the perfect attack oracle returns robust for all sample points in S_i when tested on h_i , that is $\mathcal{O}_{u^*}(h_i, x) = \text{robust}$ for all x with $(x, y) \in S_i$ for some y . Thus, using a binary search over the list T , we can identify the smallest index ι for which $u^* \preceq_{\mathcal{H}} u_{\iota}$. Since we have $n \leq 2^m$ and every test in the search employs at most m calls to the perfect attack oracle, this search terminates after at most m^2 queries. Finally, note that all subsets S_i from T for which $i \geq \iota$ are realizable with respect to u^* . Thus we can choose any largest among $S_{\iota}, \dots, S_n$ as a maximal realizable subset with respect to the unknown perturbation type u^* . $\square$

In Section 9.7 we provide additional guarantees for the agnostic case.

9.4 Beyond totally ordered perturbation classes

We now consider a more general setting where $\mathcal{U}$ is not necessarily totally ordered with respect to $\mathcal{H}$.

9.4.1 Learning with the standard adversarial loss

We will start by showing that $\mathcal{H}, \mathcal{U}$ are learnable with access to a perfect attack oracle if $\mathcal{U}$ is the union of k sub-classes which are each totally ordered with respect to $\mathcal{H}$. This naturally contains the case that $\mathcal{U}$ is finite. Another natural case where this occurs is considering perturbation sets that are balls around domain points of certain radii and with respect to one of a number of possible norms. Consider $\mathcal{X} = \mathbb{R}^d$ and the set

$$\mathcal{U} = \{u_r^p : x \mapsto B_r^p(x) \mid r \in \mathbb{R}, p \in 0, 1, 2, \infty\}$$

where we let $B_r^p(x)$ denote the ℓ^p -norm ball around x . Then the above class $\mathcal{U}$ is a union of 4 totally ordered perturbation classes (one for balls of each norm) with respect to any hypothesis class $\mathcal{H}$.

Theorem 9.4.1. *Let $\mathcal{H}$ be a hypothesis class with $\text{VC}(\mathcal{H}) = d < \infty$, and let $\mathcal{U}$ be a perturbation class, which is a union $\mathcal{U} = \mathcal{U}_1 \cup \mathcal{U}_2 \cup \dots \cup \mathcal{U}_k$ of k subclasses which are each totally ordered with respect to class $\mathcal{H}$. Then $\mathcal{H}$ is learnable with access to a perfect attack oracle $\mathcal{O}_{u^*}$ in the realizable case with respect to $\mathcal{U}$ with sample complexity $O\left(\frac{d \cdot 2^d \cdot \log(k/\delta)}{\varepsilon} + \frac{\log(k) \cdot \log(1/\varepsilon) + \log(1/\delta)}{\varepsilon^2}\right)$ and query complexity $O\left(\left(\frac{\log(k) \cdot \log(1/\varepsilon) + \log(1/\delta)}{\varepsilon^2}\right)^2\right)$.*

Proof. We can generate k hypotheses $h_1, h_2, \dots, h_k$ that are the output of an (ε, δ) -successful learner for $\mathcal{H}$ with respect to each of the $\mathcal{U}_i$. Since each of the classes $\mathcal{U}_i$ are totally ordered with respect to $\mathcal{H}$, the result of Theorem 9.3.1 tells us that $O\left(\frac{d \cdot 2^d \cdot \log(k/\delta)}{\varepsilon}\right)$ sample points suffice to guarantee that, with probability at least $1 - \delta$, we have $\ell^{u^*}(h_i) \leq \varepsilon$ for all i with $u^* \in \mathcal{U}_i$. Thus the class $\mathcal{H}_k = \{h_1, h_2, \dots, h_k\}$ of these k learning outputs has approximation error at most ε with respect to the data generating distribution P . Note that, since we have $|\mathcal{H}_k| = k$, this class has also bounded Littlestone dimension at most $\log(k)$. Employing a result ([Montasser et al., 2021], Theorem 3) on agnostically learning finite Littlestone classes with access to a perfect attack oracle, an additional $O\left(\frac{\log(k) \cdot \log(1/\varepsilon) + \log(1/\delta)}{\varepsilon^2}\right)$ samples and the stated number of queries suffice to output a hypothesis h with $\mathcal{L}_P^{u^*}(h) \leq 3\varepsilon$. $\square$

It is not difficult to see that, even in the realizable case, and when $\mathcal{U}$ is the union of just two totally ordered subclasses, a learner without access to a perfect attack oracle cannot succeed with respect to the standard adversarial loss.

Observation 9.4.2. There exists a class $\mathcal{H}$ with $\text{VC}(\mathcal{H}) = 1$ and a perturbation class $\mathcal{U}$ that is the union of two totally ordered subclasses with respect to $\mathcal{H}$ that are not learnable in the realizable case (in the sense of Definition 9.2.3) without access to a perfect attack oracle.

For the proof of this observation we refer the reader to the supplementary material. Now we show that if we remove even more structure from the set of perturbation types $\mathcal{U}$ then learning success becomes impossible even with access to a perfect attack oracle. The following observation will then motivate to investigate learning with respect to the adversarial abstention loss (rather than the standard adversarial loss).

Observation 9.4.3. There exists a class $\mathcal{H}$ with $\text{VC}(\mathcal{H}) = 1$ and a perturbation class $\mathcal{U}$ that are not learnable (in the sense of Definition 9.2.3), even with access to a perfect attack oracle.

Proof. We consider the same domain, class, and distribution as in the proof of the previous observation: $\mathcal{X} = \mathbb{R}$, P with $P((-3, 1) = P((3, 0)) = 0.5$, and

$$\mathcal{H} = \{h_t : x \mapsto \mathbf{1}[x \leq t] \mid t \in \mathbb{R}\}.$$

Further, we consider the class $\mathcal{U} = \{u_{a,b} \mid a, b \in \mathbb{R}\}$ of perturbation types $u_{a,b}$, that assign all points $x \leq 0$ to a ball of radius a around x and all points $x > 0$ to a ball of radius b . Then, even when promised realizability, a learner cannot distinguish the cases where $u^* = u_{a,b}$ for some values a, b with $a + b \leq 6$ from samples and any finite number of queries to the perfect attack oracle. In particular, if $u^* = u_{a,b}$ for some a, b with $a + b = 6$, predicting with respect to one such pair a, b will induce loss 0.5 for all other such pairs. $\square$

9.4.2 Learning with the adversarial abstention loss

We will now consider learning with respect to the adversarial abstention loss. We will first show that learning with abstentions can actually help overcome some of the impossibilities shown above. For this, let us revisit the example seen in Observation 9.4.3.

Observation 9.4.4. There exists a class $\mathcal{H}$ with $\text{VC}(\mathcal{H}) = 1$ and a perturbation class $\mathcal{U}$ that are not learnable (in the sense of Definition 9.2.3), even with access to a perfect attack oracle, but that is robustly learnable with abstentions (in the sense of Definition 9.2.5) in the realizable case (without access to a perfect attack oracle).

Proof sketch. We consider the example from the proof of Observation 9.4.3. Now consider the sub-classes $\mathcal{U}_1 = \{u_{a,0} : a \in \mathbb{R}\}$ and $\mathcal{U}_2 = \{u_{0,b} : b \in \mathbb{R}\}$. Since both $\mathcal{U}_1$ and $\mathcal{U}_2$ are totally ordered we can learn successful hypotheses h_1 and h_2 respectively. We now define $h : \mathcal{X} \rightarrow \{0, 1, \star\}$ by $h(x) = y$ if $h_1(x) = h_2(x) = y$ and $h(x) = \star$ otherwise. This is a successful hypothesis. For more detail, we refer the reader to the appendix. $\square$

We will now continue by showing that, similar to the case where we have access to a perfect-attack-oracle, a finite union $\mathcal{U} = \bigcup_{i=1}^k \mathcal{U}_i$ of totally ordered perturbation sets $\mathcal{U}_i$ is robustly learnable with abstention in the robustly realizable case.

Theorem 9.4.5. *Let $\mathcal{U} = \bigcup_{i=1}^k \mathcal{U}_i$, where every $\mathcal{U}_i$ is totally ordered. Furthermore, let $\mathcal{H}$ have a finite VC dimension. Then $\mathcal{H}$ is robustly learnable with abstention with respect to $\mathcal{U}$ in the robustly realizable case with sample complexity $O(\frac{d \cdot 2^d \cdot k^2 \log(k/\delta)}{\varepsilon})$.*

Abstention learning with disagreement coefficient

The proof of Theorem 9.4.5 shows how bounding the mass of the region on which two learners with small 0/1-loss disagree, yields a robust error guarantee for learning with abstentions. In this section, we will review the definition of the *disagreement coefficient* ([Hanneke, 2007, Hanneke et al., 2014]), and show that hypothesis classes $\mathcal{H}$ with finite disagreement coefficient are robustly learnable with abstention in the realizable case with respect to every class of perturbation types $\mathcal{U}$.

For a distribution $P_{\mathcal{X}}$ over $\mathcal{X}$, we let the $P_{\mathcal{X}}$ -induced difference between two hypotheses be denoted by

$$d_{P_{\mathcal{X}}}(h, h') = \mathbb{P}_{x \sim P_{\mathcal{X}}}[h(x) \neq h'(x)].$$

The ball in $\mathcal{H}$ around a hypothesis h with radius r is then denoted by

$$B_{P_{\mathcal{X}}}(h, r) = \{h' \in \mathcal{H} : d_{P_{\mathcal{X}}}(h, h') \leq r\}.$$

Lastly, let the disagreement region of a set of hypotheses $\mathcal{H}$ be

$$\text{DIS}(\mathcal{H}) = \{x \in \mathcal{X} : \exists h, h' \in \mathcal{H} \text{ with } h(x) \neq h'(x)\}.$$

We can now state the definition of the disagreement coefficient.

Definition 9.4.6 ([Hanneke et al., 2014]). *For a distribution P over $\mathcal{X} \times \mathcal{Y}$ and a class $\mathcal{H} \subseteq \{0, 1\}^{\mathcal{X}}$, let $h^* \in \arg \min_{h' \in \mathcal{H}} \mathcal{L}_P(h')$. The disagreement-coefficient of $\mathcal{H}$ with respect to P is defined by:*

$$\theta_P(\mathcal{H}) = \sup_{r \in (0,1)} \frac{P_{\mathcal{X}}(\text{DIS}(B(h^*, r)))}{r}.$$

Furthermore, let the worst-case disagreement coefficient be $\theta(\mathcal{H}) = \sup_{P \in \Delta(\mathcal{X} \times \mathcal{Y})} \theta_P(\mathcal{H})$.

Theorem 9.4.7. *Let $\mathcal{H}$ be a hypothesis class with $\theta(\mathcal{H}) < \infty$ and $VC(\mathcal{H}) = d < \infty$. Then the class $\mathcal{H}$ is robustly learnable with abstention in the realizable case with respect to the class of all perturbation types $\mathcal{U}^{\text{all}}$ with sample complexity $O(\frac{d + \log(\frac{1}{\delta})}{(\frac{\varepsilon}{\theta(\mathcal{H})})^2})$.*

The proof can be found in Appendix 9.9.6, but the main idea is that for any point x , we return a $y \in \{0, 1\}$ if and only if every h with $\mathcal{L}_S^{0/1}(h) = 0$ agrees with y , otherwise we abstain. Because of the bound on the disagreement coefficient, one can show that for a large enough S , this leads to a hypothesis with low abstention loss.

Abstention learning via ε -cover

We will now generalize our previous results, by introducing the concept of an $(\varepsilon, \mathcal{H})$ -cover.

Definition 9.4.8 ($(\varepsilon, \mathcal{H})$ -cover of $\mathcal{U}$). *Let $\mathcal{H} \subseteq \{0, 1\}^{\mathcal{X}}$ be a hypothesis class, P a distribution over $\mathcal{X} \times \mathcal{Y}$ and $\mathcal{U}$ a class of perturbation types. A set of hypotheses $\mathcal{H}'$ is a $(\varepsilon, \mathcal{H})$ -cover for $\mathcal{U}$ with respect to P , if for every $u \in \mathcal{U}$, there exists an $h' \in \mathcal{H}'$ such that $\mathcal{L}_P^u(h') < \inf_{h \in \mathcal{H}} \mathcal{L}_P^u(h) + \varepsilon$.*

We will now provide the definition of a successful cover learner.

Definition 9.4.9. *Let $\mathcal{H} \subseteq \{0, 1\}^{\mathcal{X}}$ be a hypothesis class, and $\mathcal{U}$ be a class of perturbation types. We say a function $\mathcal{A}_{\text{cover}} : (\mathcal{X} \times \mathcal{Y})^* \rightarrow 2^{\{0, 1\}^{\mathcal{X}}}$ that maps a sample S to a hypothesis class $\hat{\mathcal{H}} \subset \mathcal{F}$ is a successful finite-disagreement-cover learner for $(\mathcal{H}, \mathcal{U})$, if there is a function $m_{(\mathcal{H}, \mathcal{U}), \text{cover}} : (0, 1)^3 \rightarrow \mathbb{N}$, such that for every $\varepsilon, \eta, \delta \in (0, 1)$, every $m \geq m_{(\mathcal{H}, \mathcal{U}), \text{cover}}(\varepsilon, \eta, \delta)$ and every distribution P over $\mathcal{X} \times \{0, 1\}$ with probability $1 - \delta$ over $S \sim P^m$ both of the following statements hold:*

- $\mathcal{A}(S)$ is an $(\varepsilon, \mathcal{H})$ -cover for $\mathcal{U}$ with respect to P
- $P_{\mathcal{X}}(\text{DIS}(\mathcal{A}(S))) \leq \eta$.

We note that any successful cover learner, that is guaranteed to output a finite set is a successful finite-disagreement cover learner. The case, where $\mathcal{U}$ is a union of k totally ordered perturbation type classes $\mathcal{U}_i$, we have seen before, can be interpreted as first learning a cover $\hat{H}$ of size k , where each of the element of the union "covers" one of the classes. The case in which $\mathcal{H}$ has a finite disagreement coefficient can furthermore be seen as a case where the set of all $\text{ERM}_{\mathcal{H}}$ hypotheses constitute an $(\varepsilon, \mathcal{H})$ -cover $\mathcal{H}'$ for every class of perturbation types $\mathcal{U}$. The mass of the disagreement region of $\mathcal{H}$ is then bounded by the disagreement coefficient. We now state our result generalizing these observations.

Theorem 9.4.10. *A class $\mathcal{H}$ is robustly learnable with abstentions with respect to $\mathcal{U}$ in the robustly realizable case, if there is a successful finite-disagreement-cover learner for $(\mathcal{H}, \mathcal{U})$. Furthermore, the sample complexity of robustly learning with abstentions is then bounded by $m_{\mathcal{H}, \mathcal{U}}(\varepsilon, \delta) \leq m_{\mathcal{H}, \mathcal{U}, \text{cover}}(\varepsilon/2, \varepsilon/2, \delta)$.*

Inspired by considerations of adaptive robustness radii [Chowdhury and Uner, 2022, Bhattacharjee and Chaudhuri, 2021], in Section 9.8 of the appendix we discuss a setting of abstention learning in which we allow the size of the perturbation sets to vary locally between different regions of the domain. We provide a learning guarantee for this setting that is not covered by our previous results.

9.5 Conclusion

In this work we relaxed the assumption that we know exactly the perturbation sets that are to be used by an adversary. Our model provides an interpolation between knowing the perturbation sets exactly and not knowing them at all. Many of our results rely on realizability. We have also shown that without realizability, learning is not possible unless the learner receives other feedback, such as access to a perfect attack oracle. It might be interesting to consider some milder relaxations of realizability. For example, instead of saying $\text{opt}_P^{u^*}(\mathcal{H}) = 0$, what if we only know that the adversarial Bayes loss is zero for some $u^* \in \mathcal{U}$?

While our work focuses on statistical aspects, it will also be interesting to understand the kinds of computation required for our results. For example, we rely on the existence of an MRERM oracle. When are they computationally feasible? Do we require accurate MRERM oracles or can approximate (and hence potentially more tractable) versions also be used for the learning task?

Another question is to find a dimension that characterizes learning in various settings. This was only recently resolved for adversarially robust learning (for the case when u is known) [Montasser et al., 2022], however, the dimension proposed in the paper is not easy to define. On the other hand, in the setting when the only information we can obtain about u is through a perfect attack oracle, the Littlestone dimension of $\mathcal{H}$ has been shown to characterize adversarially robust learning [Montasser et al., 2021]. It will be interesting to see if such a simple dimension can be obtained for the settings considered in our paper.

9.6 Maximally Robust Empirical Risk Minimization

Recall the definition of the empirical risk minimization oracles:

$$\begin{aligned} \text{ERM}_{\mathcal{H}} : S &\mapsto h_S \in \arg \min_{h \in \mathcal{H}} \mathcal{L}_S^{0/1}(h) \\ \text{RERM}_{\mathcal{H}}^u : S &\mapsto h_S^u \in \arg \min_{h \in \mathcal{H}} \mathcal{L}_S^u(h) \\ \text{MRERM}_{\mathcal{H}}^{\mathcal{U}} : S &\mapsto \begin{cases} (h_S^{u^*}, u^*) \text{ with } h_S^{u^*} \in \arg \min_{h \in \mathcal{H}} \mathcal{L}_S^{u^*}(h), \\ \quad \text{and } u^* \text{ is } \preceq_{\mathcal{H}} \text{ maximal in the set } \{u \in \mathcal{U} \mid \min_{h \in \mathcal{H}} \mathcal{L}_S^u(h) = 0\} \\ \text{error if } \mathcal{L}_S^u(h) > 0 \text{ for all } u \in \mathcal{U} \text{ and } h \in \mathcal{H} \end{cases} \end{aligned}$$

While there always exists functions h_S and h_S^u that are solutions to the queries in the first two oracles, the last one requires a bit more care.

Let S be some labeled dataset and assume that $\mathcal{H}$ and $\mathcal{U}$ are hypothesis and perturbation type classes such that S is $\mathcal{H}$ -realizable for the true perturbation type $u^* \in \mathcal{U}$. We can then define a maximal perturbation type u_S^* for S , $\mathcal{H}$ and $\mathcal{U}$ as follows:

$$u_S^*(x) = \bigcup_{u \in \mathcal{U} \text{ such that } \min_{h \in \mathcal{H}} \mathcal{L}_S^u(h) = 0} u(x).$$

This defines a perturbation type $u_S^*(x)$, which is not necessarily in $\mathcal{U}$ but does satisfy $u^* \preceq_{\mathcal{H}} u_S^*$, which suffices for our purposes.

If, given u_S^* , the set $\arg \min_{h \in \mathcal{H}} \mathcal{L}_S^{u_S^*}(h)$ is non-empty, the oracle can return any function in this set as $h_S^{u_S^*}$. We will now argue that we can define a “closure” $\bar{\mathcal{H}}$ of the hypothesis class $\mathcal{H}$, which has the same VC-dimension as $\mathcal{H}$ and such that $h_S^{u_S^*} \in \bar{\mathcal{H}}$.

We here show how such an embedding of $\mathcal{H}$ into the closure class $\bar{\mathcal{H}}$ can be constructed in the case of a countable domain $\mathcal{X}$ and a perturbation class $\mathcal{U}$ that is *separable*, in the sense that for every $\mathcal{U}' \subset \mathcal{U}$ there exists a countable subset $\mathcal{U}'' \subset \mathcal{U}'$, such that for every $x \in \mathcal{X} : \bigcup_{u \in \mathcal{U}'} u(x) = \bigcup_{u \in \mathcal{U}''} u(x)$. This is the case for a wide range of perturbation types, in particular in the case for which the sets $u(x)$ are always open balls with respect to some ℓ_p -norm in $\mathbb{R}^d$. In the general case (where $\mathcal{X}$ may be uncountable and $\mathcal{U}$ not separable in the above sense) recent work has shown that the existence of a closure class with identical VC-dimension is still guaranteed [Lechner et al., 2023b]. The more general construction there employs the set-theoretic concept of *filters*. It is shown in the context of strategic classification, but the technique is readily available to adversarial perturbation sets. We

here illustrate the intuition behind that abstract construction by focusing on countable and separable case.

We now assume that $\mathcal{X}$ is countable and let $(x_i)_{i \in \mathbb{N}}$ be an enumeration of the elements of $\mathcal{X}$. The separability assumption on the perturbation class $\mathcal{U}$ implies that for all S there is a countable subset $\mathcal{U}^c \subseteq \mathcal{U}$ such that for all $x \in \mathcal{X}$ we have

$$u_S^*(x) = \bigcup_{u \in \mathcal{U}^c \text{ such that } \min_{h \in \mathcal{H}} \mathcal{L}_S^u(h) = 0} u(x).$$

For a given sample S , let $(u_i)_{i \in \mathbb{N}}$ be an enumeration of the set

$$\{u \in \mathcal{U}^c \mid \min_{h \in \mathcal{H}} \mathcal{L}_S^u(h) = 0\}$$

and let $(h_i)_{i \in \mathbb{N}}$ be an enumeration of corresponding empirical risk minimizing hypotheses from $\mathcal{H}$:

$$h_i = \text{RERM}_{\mathcal{H}}^{u_i}(S).$$

We will now define the function h_S^* by successively choosing subsequences from the sequence $(h_i)_{i \in \mathbb{N}}$ as follows. We set $F_0 = (h_i)_{i \in \mathbb{N}}$ and $J_0 = \mathbb{N}$. Then, for every $i \in \mathbb{N}$, we define a set of indices J_i such that

- **Case 1:** There exists a $k \in J_{i-1}$ such that $h_j(x_i) = h_k(x_i)$ for all $j \geq k, j \in J_{i-1}$. Then we define:

$$J_i = \{j \in J_{i-1} \mid h_j(x_i) = h_k(x_i)\}$$

- **Case 2:** Both labels are assigned infinitely often to x_i by the sequence $(h_j)_{j \in J_{i-1}}$. Then we define

$$J_i = \{j \in J_{i-1} \mid h_j(x_i) = 1\}$$

and we define subsequence $F_i = (h_j)_{j \in J_i}$. Note that, by construction, in each resulting sequence F_i the first i points $x_1, x_2, \dots, x_i$ receive the same label by all functions in the sequence. Now, by choosing indices to select the i -th function from each sequence F_i (by choosing the first index of J_1 , the second index in J_2 , the third in J_3 and so on), and re-identifying these indices with the natural numbers, we obtain yet another subsequence $F = (f_i)_{i \in \mathbb{N}}$. In this sequence we now have for every $i \in \mathbb{N}$ and every $k \geq i$

$$f_k(x_i) = f_i(x_i).$$

That is, this sequence of functions converges pointwise and we can use the above equation to define the function h_S^* as follows:

$$h_S^{u_S^*}(x_i) = f_i(x_i).$$

It is not difficult to see that the so defined function $h_S^{u_S^*}$ satisfied $\mathcal{L}_{u_S^*}^S(h_S^{u_S^*}) = 0$. Let $\bar{\mathcal{H}}$ be defined by adding these function $h_S^{u_S^*}$ to $\mathcal{H}$ for all possible S .

$$\bar{\mathcal{H}} = \mathcal{H} \cup \{h_S^{u_S^*} \in \{0, 1\}^{\mathcal{X}} \mid S \in (\mathcal{X} \times \mathcal{Y})^*\}$$

We observe that every behavior that $\bar{\mathcal{H}}$ exhibits on a finite subset $C \subseteq \mathcal{X}$, is already witnessed by a function from $\mathcal{H}$. Thus we get $\text{VC}(\bar{\mathcal{H}}) = \text{VC}(\mathcal{H})$, and we can always let $\text{MRERM}_{\mathcal{H}}^{\mathcal{U}}$ return a function from $\bar{\mathcal{H}}$.

9.7 Agnostic learning for finite unions of total orders

Theorem 9.7.1. *Let $\mathcal{H}$ be a hypothesis class with $\text{VC}(\mathcal{H}) = d < \infty$, and let $\mathcal{U}$ be a perturbation class, which is a union $\mathcal{U} = \mathcal{U}_1 \cup \mathcal{U}_2 \cup \dots \cup \mathcal{U}_k$ of k subclasses which are each totally ordered with respect to class $\mathcal{H}$. Then $\mathcal{H}$ is agnostically learnable with access to a perfect attack oracle $\mathcal{O}_{u^*}$ with respect to $\mathcal{U}$ with sample complexity $m_1(\varepsilon, \delta) + m_2(\varepsilon, \delta)$, and query complexity $n_1(\varepsilon, \delta) + n_2(\varepsilon, \delta)$ where:*

- $m_1(\varepsilon, \delta) = O\left(\frac{d \cdot 2^d \cdot \log(k/\delta)}{\varepsilon^2}\right)$
- $m_2(\varepsilon, \delta)$ is the sample complexity of agnostically learning a hypothesis class of size k wrt $\mathcal{U}^{\text{all}}$.
- $n_1(\varepsilon, \delta) = k \cdot (m_1(\varepsilon, \delta))^2$
- $n_2(\varepsilon, \delta)$ is the query complexity of agnostically learning a hypothesis class of size k wrt $\mathcal{U}^{\text{all}}$

Proof. We can generate k hypotheses $h_1, h_2, \dots, h_k$ that are the output of an (ε, δ) -successful learner for $\mathcal{H}$ with respect to each of the $\mathcal{U}_i$. Since each of the classes $\mathcal{U}_i$ are totally ordered with respect to $\mathcal{H}$, the result of Theorem 9.3.3 (combined with a union bound) tells us that $m_1(\varepsilon, \delta)$ sample points and $n_1(\varepsilon, \delta)$ queries suffice to guarantee that, with probability at least $1 - \delta$, we have $\ell^{u^*}(h_i) \leq \text{opt}_P^{u^*}(\mathcal{H}) + \varepsilon$ for all i with $u^* \in \mathcal{U}_i$. Thus the

class $\mathcal{H}_k = \{h_1, h_2, \dots, h_k\}$ of these k learning outputs has approximation error at most $\text{opt}_P^{u^*}(\mathcal{H}) + \varepsilon$ with respect to the data generating distribution P . Note that, since we have $|\mathcal{H}_k| = k$, this class has bounded Littlestone dimension at most $\log(k)$, VC-dimension at most $\log(k)$, and dual VC-dimension at most k . Employing a result ([Montasser et al., 2021], Theorem 4) on agnostically learning finite VC classes with access to a perfect attack oracle, we get that both $m_2(\varepsilon, \delta)$ and $n_2(\varepsilon, \delta)$ are finite and $\mathcal{L}_P^{u^*}(h) \leq \text{opt}_P^{u^*}(\mathcal{H}) + 2\varepsilon$. $\square$

9.8 ε -cover Learning

9.8.1 Abstention learning for regions with different radii

Let us now explore the case, where we allow our perturbation types to consist of perturbation sets whose size varies throughout the domain. While a lot of work assumes that the size of the radius is fixed throughout the domain, it is not a priori clear that the distances in the domain correspond perfectly to what humans find perceptible. This was noted in prior works [Bhattacharjee and Chaudhuri, 2021, Chowdhury and Uner, 2022] and motivated their adaptive adversarial loss. We explore such variation of local radii for the setting of uncertain perturbation sets. We have seen examples of this in Observation 3 and Observation 4. We will now generalize these perturbation types. Let $\mathcal{X} = \mathbb{R}$. For a function $f : \mathcal{X} \rightarrow [k]$, let the perturbation type $u_{f,a_1,\dots,a_k}$ defined by:

$$u_{f,a_1,\dots,a_k}(x) = B_{a_{f(x)}}(x).$$

That is, we have k different regions in the domain, which are determined by f . Furthermore, we assume that each perturbation set within a region i is a ball with radius of a fixed size a_i , while radii can vary between regions. For a fixed function f , let us now consider

$$\mathcal{U}_f = \{u_{f,a_1,\dots,a_k} : a_1, \dots, a_k \in \mathbb{R}\}.$$

Let $\text{VC}(\mathcal{H}) \leq \text{VC}(\mathcal{H} \times \mathcal{U})_\ell \leq d$. We note that in general, that is excluding particular hypothesis sets $\mathcal{H}$ and function f , this set of perturbation sets cannot be expressed as a finite union of totally ordered sets.

For every $i \in [k]$ consider the classes $\mathcal{U}_{f,i} = \{u_{f,0,\dots,0,a_i,0,\dots,0} : a_i \in \mathbb{R}\}$. Let P be a realizable distribution with respect to $\mathcal{H}$ and $u^* \in \mathcal{U}$. Let $m(\varepsilon, \delta) = C \frac{d + \log(\frac{1}{\delta})}{\varepsilon^2}$, where C is the constant we require to get a uniform-convergence guarantee (which exists due to the corresponding VC-dimensions being finite, see [Shalev-Shwartz and Ben-David, 2014a]), and let $m \geq m(\frac{\varepsilon}{k(2k^2+1)}, \frac{\delta}{k})$. Now for a sample S and any i , let $(h_i, u_i) = \text{MRERM}_{\mathcal{U}_{f,i}}^{\mathcal{H}}(S)$. We

denote by a_i^{max} the maximum realizable radius for component i , i.e. $h_{0,\dots,0,a_i^{max},0,\dots,0} = h_i$. Now let $u^* = u_{f,a_1^*,\dots,a_k^*}$ be the ground true perturbation set. Then for every i , $a_i^* \leq a_i^{max}$. Let $u^{max} = u_{f,a_1^{max},\dots,a_k^{max}}$. We note that for every $x \in \mathcal{X}, y \in \mathcal{Y}$ and $h \in \mathcal{F}$, we have $\ell^{u^{max}}(h, x, y) \leq \max_{i \in [k]} \ell^{u_i}(h, x, y)$. Now let

$$h(x) = \begin{cases} y & \text{if for all } i : h_i(x) = y \\ \star & \text{otherwise.} \end{cases}$$

We note that by the PAC-learning guarantee with respect to 0/1-loss, with probability $1 - \delta$, for every $i, j \in [k]$ we have a pairwise disagreement between h_i and h_j of at most $\frac{2\varepsilon}{(2k^2+1)k}$. Thus the disagreement-region of $\{h_i : i \in [k]\}$ has mass at most $\frac{k^2\varepsilon}{k(2k^2+1)}$. Thus, with probability $1 - \delta$ over the sample generation the abstention loss of h we have

$$\begin{aligned} \mathcal{L}_P^{u^*, \text{abst}}(h) &\leq \mathcal{L}_P^{u^{max}, \text{abst}}(h) \leq \sum_{i=1}^k \mathcal{L}_P^{u^i, \text{abst}}(h) \leq \sum_{i=1}^k (\mathcal{L}_P^{u^i}(h_i) + P_{\mathcal{X}}(\{x : h(x) = \star\})) \\ &\leq k \left(\frac{\varepsilon}{k(2k^2+1)} + \frac{k^2\varepsilon}{k(2k^2+1)} \right) = \varepsilon. \end{aligned}$$

Note, that we can thus view $\{h\}$ as an $(\mathcal{H}, \varepsilon)$ -cover of $\mathcal{U}$ with respect to P and adversarial *abstention* loss.

9.9 Proofs

9.9.1 Proof of Theorem 1

Proof. For this result we adapt a compression based argument for the case of fixed perturbation type [Montasser et al., 2019] to the case of a class $\mathcal{U}$ of perturbation types. A similar adaptation has recently been shown for the setting of strategic classification [Lechner et al., 2023b]. The key difference to the original compression scheme is that we use the $\text{MRERM}_{\mathcal{H}}^{\mathcal{U}}$ oracle for the weak learners in each round of boosting, rather than the simple $\text{RERM}_{\mathcal{H}}^{\mathcal{U}}$ oracle as is done in the original publication [Montasser et al., 2019]. Importantly, the maximal ERM paradigm is well defined on all subsets of an original given data sample, and thus each required weak hypothesis can be encoded with a finite number of samples. Finally, since the maximal perturbation type for any subsample is always at least as large as the true perturbation type (under the realizability assumption), using $\text{MRERM}_{\mathcal{H}}^{\mathcal{U}}$ for each boosting step we obtain the required guarantee for the true perturbation type (without

the learner/compressor requiring knowledge of the true type). We thus obtain the same compression size and implied sample complexity in the case of a totally ordered class of perturbation types as for a fixed perturbation type. $\square$

9.9.2 Proof of Observation 1

Proof. Consider a finite domain $\mathcal{X} = \{x_1, x_2, x_3\}$ and distribution P over $\mathcal{X} \times \{0, 1\}$ with $P((x_1, 1)) = 1/4$, $P((x_3, 0)) = 3/4$ and $P((x, y)) = 0$ for all $(x, y) \notin \{(x_1, 1), (x_3, 0)\}$. We define a class $\mathcal{U} = \{u_1, u_2\}$ containing two perturbation types u_1 and u_2 defined as follows:

$$u_1(x_1) = \{x_1, x_2\}, \quad u_1(x_2) = \{x_2\}, \quad u_1(x_3) = \{x_3\}$$

and

$$u_2(x_1) = \{x_1, x_2\}, \quad u_2(x_2) = \{x_2\}, \quad u_2(x_3) = \{x_2, x_3\}$$

We let $h_{y_1 y_2 y_3}$ denote the function that labels x_i with y_i on this domain (there are only 8 different non-abstaining predictors over $\mathcal{X}$), and let $H = \{h_{110}, h_{100}\}$ be a hypothesis class containing only two of these predictors. Clearly $\text{VC}(\mathcal{H}) = 1$ and $u_1 \preceq_{\mathcal{H}} u_2$. Note that

$$\text{opt}_P^{u_1}(\mathcal{H}) = \mathcal{L}_P^{u_1}(h_{110}) = 0$$

while $\mathcal{L}_P^{u_1}(h_{y_1 y_2 y_3}) \geq 1/4$ for all predictors $h_{y_1 y_2 y_3} \neq h_{110}$. We further note that

$$\text{opt}_P^{u_2}(\mathcal{H}) = \mathcal{L}_P^{u_2}(h_{100}) = \mathcal{L}_P^{u_2}(h_{000}) = 1/4$$

while $\mathcal{L}_P^{u_2}(h_{y_1 y_2 y_3}) \geq 3/4$ for all predictors $h_{y_1 y_2 y_3} \notin \{h_{000}, h_{100}\}$. Samples from P will not allow distinguishing whether the true perturbation type is u_1 or u_2 (since we are not imposing a realizability assumption here). Thus, no learner will be able to output predictors that are (close to) optimal for all possible perturbation types in $\mathcal{U}$. $\square$

9.9.3 Proof of Observation 2

Proof. We consider $\mathcal{X} = \mathbb{R}$ and $\mathcal{H}$ to be the class of threshold classifiers:

$$\mathcal{H} = \{h_t : x \mapsto \mathbf{1}[x \leq t] \mid t \in \mathbb{R}\}.$$

Further consider $\mathcal{U} = \mathcal{U}_- \cup \mathcal{U}_+$, where $\mathcal{U}_-$ consists of perturbation types u_r^- that assign each x to a ball (interval) of radius r around x if $x \leq 0$ and a ball of radius $r/2$ around x if $x > 0$, and $\mathcal{U}_+$ similarly consists of perturbation types u_r^+ that assign each x to a ball (interval)

of radius r around x if $x > 0$ and a ball of radius $r/2$ around x if $x \leq 0$. Both $\mathcal{U}_-$ and $\mathcal{U}_+$ are totally ordered (with respect to any hypothesis class) but their union $\mathcal{U}$ is not totally ordered with respect to $\mathcal{H}$. Consider a distribution P with $P((-3, 1) = P((3, 0)) = 0.5$. Then, even when promised realizability, a learner cannot distinguish the cases $u^* = u_4^-$ and $u^* = u_4^+$ from samples; and performing well with respect to one of these perturbation types will induce loss 0.5 with respect to the other. $\square$

9.9.4 Proof of Observation 4

Proof. We consider the example from Observation 9.4.3 Now consider the classes $\mathcal{U}_1 = \{u_{a,0} : a \in \mathbb{R}\}$ and $\mathcal{U}_2 = \{u_{0,b} \in \mathbb{R}\}$. Since both $\mathcal{U}_1$ and $\mathcal{U}_2$ are totally ordered there are successful robust learners $\mathcal{A}_1$ for $\mathcal{H}$ with respect to $\mathcal{U}_1$ and $\mathcal{A}_2$ with respect to $\mathcal{U}_2$. We run both learners on a sample with a size to guarantee $(\varepsilon/2, \delta/2)$ -success in both cases. We denote the resulting hypotheses with h_1 and h_2 respectively. We now define $h : \mathcal{X} \rightarrow \{0, 1, \star\}$ by $h(x) = y$ if $h_1(x) = h_2(x) = y$ and $h(x) = \star$ otherwise.

We now argue that with probability $1 - \delta$, h has adversarial abstention loss less than ε . Let $u^* = u_{a^*,b^*} \in \mathcal{U}$ be the ground-truth perturbation type. Now let P be any $\ell^{u_{a^*,b^*}}$ -realizable distribution. Let $(\cdot, u_{a_1,0}) = \text{MRERM}_{\mathcal{H}}^{\mathcal{U}_1}(S)$ and $(\cdot, u_{0,b_2}) = \text{MRERM}_{\mathcal{H}}^{\mathcal{U}_2}(S)$. With probability 1, we have both $a_1 > a^*$ and $b_2 > b^*$. Further, we note every $x \in \mathcal{X}$ and every $u_{a,b} \in \mathcal{U}$, $\ell^{u_{a,b}}(h, x, y)$ is either determined by a, h and y or b, h and y , not both, since either $x \leq 0$ or $x > 0$. This means, that for every $x \in \mathcal{X}$, $\ell^{u_{a_1,b_2}}(h, x, y) \leq \max\{\ell^{u_{a_1,0}}(h, x, y), \ell^{u_{0,b_2}}(h, x, y)\}$.

Combining these observations, we get

$$\mathcal{L}_P^{u^*, \text{abst}}(h) \leq \mathcal{L}_P^{u_{a_1,b_2}, \text{abst}}(h) \leq \mathcal{L}_P^{u_{a_1,0}, \text{abst}}(h) + \mathcal{L}_P^{u_{0,b_2}, \text{abst}}(h) \leq \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon.$$

$\square$

9.9.5 Proof of Theorem 4

Proof. We assume robust realizability with respect to $\mathcal{U}$. In particular this means that we are in the non-robustly realizable case. Thus for any $\mathcal{U}_i$, we know that the set $\mathcal{U}'_i = \mathcal{U}_i \cup \{\{\{x\} : x \in \mathcal{X}\}\}$ is totally ordered and that it contains some $u_i \in \mathcal{U}'_i$ such that $\mathcal{H}$ is robustly realizable with respect to the ground truth distribution P and u_i . We know from Theorem 9.3.1, that for every i , there is a successful maximally robust learner $\mathcal{A}_i$ for $\mathcal{H}$ with respect to $\mathcal{U}'_i$ in the realizable case, since $\mathcal{U}'_i$ is totally ordered.

Now let $\mathcal{A}$ be defined by

$$A(S)(x) = \begin{cases} 1 & \text{iff for all } i \in [k] \text{ we have } \mathcal{A}_i(S)(x) = 1 \\ 0 & \text{iff for all } i \in [k] \text{ we have } \mathcal{A}_i(S)(x) = 0 \\ \star & \text{otherwise .} \end{cases}$$

Furthermore, let $(h, u_i(S)) = \text{MRERM}_{\mathcal{H}}^{u_i}(S)$. Now let $u^* \in \mathcal{U}$ be the ground-truth perturbation type. We know that there is $i^* \in [k]$, such that $u^* \preceq u_{i^*}$. Let P be a distribution with $\text{opt}_P^{u^*}(\mathcal{H}) = 0$. Now if $S \sim P^m$ with $m \geq \max_{i \in [k]} m(\frac{\varepsilon}{2k^2+1}, \frac{\delta}{k})$, then with probability $1 - \delta$, for every $i \in [k]$, $\mathcal{L}_P^{u_i}(\mathcal{A}_i(S)) \leq \frac{\varepsilon}{2k^2+1}$ and therefore, $\mathcal{L}_P^{0/1}(\mathcal{A}_i(S)) \leq \frac{\varepsilon}{2k^2+1}$. Thus, for any two $i, j \in [k]$, we have

$$P_{\mathcal{X}}(\{x \in X : \mathcal{A}_i(S)(x) \neq \mathcal{A}_j(S)(x)\}) \leq \frac{2\varepsilon}{2k^2+1}.$$

Thus,

$$\begin{aligned} \mathcal{L}_P^{u^*, \text{abst}}(\mathcal{A}(S)) &\leq \mathcal{L}_P^{u^*, \text{abst}}(\mathcal{A}_{i^*}(S)) + P_{\mathcal{X}}(\{x \in \mathcal{X} : A(S)(x) = \star\}) \\ &\leq \mathcal{L}_P^{u_{i^*}, \text{abst}}(\mathcal{A}_{i^*}(S)) + P_{\mathcal{X}}(\{x \in X : \exists i, j, \mathcal{A}_i(S)(x) \neq \mathcal{A}_j(S)(x)\}) \\ &\leq \frac{\varepsilon}{2k^2+1} + \frac{2k^2\varepsilon}{2k^2+1} = \varepsilon. \end{aligned}$$

□

9.9.6 Proof of Theorem 5

Proof. Let $u^* \in \mathcal{U}^{\text{all}}$ be any perturbation set and P any distribution with $\text{opt}_P^{u^*}(\mathcal{H}) = 0$. Furthermore, let $m \geq m_{\mathcal{H}}(\varepsilon, \delta)$ where m_H is the sample complexity function in $O(\frac{d + \log(\frac{1}{\delta})}{(\frac{\varepsilon}{\theta(\mathcal{H})})^2})$. Let $S \sim P^m$. Let $\hat{\mathcal{H}} = \{h \in \mathcal{H} : \mathcal{L}_S^{0/1}(h) = 0\}$. Now we define our output hypothesis $\hat{h}$ by

$$\hat{h} = \begin{cases} 1 & \text{if for every } h \in \hat{\mathcal{H}} : h(x) = 1 \\ 0 & \text{if for every } h \in \hat{\mathcal{H}} : h(x) = 0 \\ \star & \text{otherwise.} \end{cases}$$

We note that there is some $h^* \in \hat{\mathcal{H}}$ with $\mathcal{L}_P^{u^*}(h^*) = \mathcal{L}_P^{0/1}(h^*) = 0$. Since every $h \in \hat{H}$ is an $\text{ERM}_{\mathcal{H}}$ hypothesis and $\mathcal{H}$ has VC dimension d , the sample of ERMs gives us

$$\begin{aligned}\mathcal{L}_P^{u^*, \text{abst}}(\hat{h}) &\leq \mathcal{L}_P^{u^*, \text{abst}}(h^*) + P_{\mathcal{X}}(\{x \in \mathcal{X} : h^*(x) \neq \hat{h}(x)\}) \\ &\leq \mathcal{L}_P^{u^*}(h^*) + P_{\mathcal{X}}(\{x \in \mathcal{X} : \hat{h}(x) = \star\}) \\ &= 0 + P_{\mathcal{X}}(\text{DIS}(B(h^*, \frac{\varepsilon}{\theta(\mathcal{H})})) = \varepsilon.\end{aligned}$$

□

9.9.7 Proof of Theorem 6

Proof. Let $m \geq m_{\mathcal{U}, \mathcal{H}, \text{cover}}(\varepsilon/2, \varepsilon/2, \delta)$ and $S \sim P^m$. And let $u^* \in \mathcal{U}$ the ground-truth perturbation set. We define the abstention learner as:

$$\mathcal{A}(S)(x) = \begin{cases} 1 & \text{if for all } h \in \mathcal{A}_{\text{cover}}(S) : h(x) = 1 \\ 0 & \text{if for all } h \in \mathcal{A}_{\text{cover}}(S) : h(x) = 0 \\ \star & \text{otherwise} \end{cases}$$

Then with probability $1 - \delta$, $P_{\mathcal{X}}(\text{DIS}(\mathcal{A}_{\text{cover}}(S))) \leq \frac{\varepsilon}{2}$ and $\exists h \in \mathcal{A}_{\text{cover}}(S)$ with

$$\mathcal{L}_P^{u^*}(h) \leq \min_{h' \in \mathcal{H}} \mathcal{L}_P^{u^*}(h') + \varepsilon/2.$$

Since for every $x \in \mathcal{X}$ either $\mathcal{A}(S)(x) = h(x)$ or $x \in \mathcal{A}_{\text{cover}}(S)$. Thus,

$$\mathcal{L}_P^{u^*, \text{abst}}(\mathcal{A}(S)) \leq \mathcal{L}_P^{u^*, \text{abst}}(h) + P_{\mathcal{X}}(\text{DIS}(\mathcal{A}_{\text{cover}}(S))) \leq \min_{h' \in \mathcal{H}} \mathcal{L}_P^{u^*, \text{abst}}(h') + \varepsilon.$$

□

Part IV

Computable PAC Learning

Chapter 10

On Learnability With Computable Learners

We note that so far our discussion of learning has only looked at generalization guarantees without any algorithmic guarantees. Moreover, *learners*, have been modeled as mappings from training samples to, say, classifiers (or other possible learned objects). In this section we will investigate how changing this definition of PAC learning to also contain conditions on the *computability* of a learner impacts the learnability of a problem. Strengthening the definitions of learnability to only allow computable learners reflects our intention of modeling automated learning.

10.1 Overview of Contributions

In this chapter I present our work on Computable PAC learning [Agarwal et al., 2020] which initiated an investigation of the nature of such notions of computable learning. In this work we focussed our attention on binary classification learning with standard 0-1-loss. As discussed in Section 2.1 purely statistical Vapnik-Chervonenkis theory provided a characterization of uniform convergence by a combinatorial property of the hypothesis class (the VC-dimension) for which learners were considered as general functions. VC-theory did not impose any requirement on the learners actually being implementable by algorithms [Vapnik and Chervonenkis, 1971a]. Valiant’s computational learning theory framework of PAC learnability combined the statistical success condition with a requirement that learners are algorithms whose running time is polynomial in $\frac{1}{\epsilon}$, $\log\left(\frac{1}{\delta}\right)$ and some

parameter d of the function class, for example, the Euclidean dimension of the feature space in the case of linear classifiers [Valiant, 1984, Haussler, 1992]. In our work [Agarwal et al., 2020], we introduced an intermediate setup. While we do not impose any (polynomial) efficiency requirements on the runtime of our learners, we will require the learners to be computable. We then demonstrate settings in which such a definition leads to different conclusions than what we know for the common notion of PAC learning.

In Section 10.2.2, we develop our modified definition of PAC learnability, which incorporates a requirement of the learning algorithm and its outputs to be computable. We term this new notion CPAC learnability. Of course, restricting the set of candidate learners can only result in a decrease in the scope of learnable classes. Therefore, the first step we take is asking: are there hypotheses classes of finite VC-dimension (therefore, PAC learnable classes) that are not CPAC learnable? We then move on to analyze and identify conditions under which such classes exist. We also consider various variations of the basic setup: proper versus non-proper learning, as well as a notion of non-uniform learnability.

The main contribution of this work is to show the existence of such classes (that are learnable under the common, unrestricted definition of PAC learning, but cannot be learned by any computable learner) in the setting of *proper learning*, that is, under the additional requirement that the learner output a function from the hypothesis class; the existence of such classes shows that the characterization of PAC learnability by the VC-dimension [Vapnik and Chervonenkis, 1971a, Vapnik, 2000] crucially depends on disregarding computability requirements. These results are presented in Section 10.4. In that section, we also identify conditions under which usual PAC learnability does imply CPAC learnability, that is, under which computable learners exist. We also develop some insights into the implications of computability requirements in the more challenging (to analyze) case of *improper learning* in Section 10.6. Furthermore, in Section 10.5 we extend our study to a notion of *non-uniform learnability* and show that in this setting there are also classes that are learnable (namely the class of all computable functions), but not learnable by a computable learner.

10.2 Defining Computable PAC (CPAC) Learning

10.2.1 General background

Notation. Let $\Sigma = \{0, 1\}$ be a binary alphabet and let Σ^* be the set of all finite words over Σ . Note that we can naturally identify Σ^* with the natural numbers $\mathbb{N}$ or with the

set of all finite subsets of natural numbers. We will often implicitly assume that we fixed one such encoding. We use standard notation for sets and functions, for example, we let $[n] = \{1, 2, \dots, n\} \subset \mathbb{N}$ denote the set of the first n natural numbers

Computability. Throughout this paper, we will assume we fixed a programming language. The existence of a program or algorithm is then equivalent to the existence of a Turing machine. We use $(T_i)_{i \in \mathbb{N}}$ to denote a fixed enumeration of all Turing machines/programs. We use the notation $T \equiv S$ to indicate that two Turing Machines have the same behavior as functions (not necessarily the same encoding, thus not necessarily the same index in the enumeration). Further, for a Turing machine T and input x , we write $T(x) \downarrow$ to indicate that T halts on input x and we write $T(x) \uparrow$ to indicate that T loops (does not halt) on input x .

We say that a function $f : \Sigma^* \rightarrow \Sigma^*$ is *total computable* if there exists a program P that halts on every string $\sigma \in \Sigma^*$ with $P(\sigma) = f(\sigma)$. A set $S \subseteq \Sigma^*$ is called *decidable* (or recursive) if there exists a program P such that for every $\sigma \in \Sigma^*$, P halts on σ and outputs whether $\sigma \in S$. Finally, a set $S \subseteq \Sigma^*$ is called *recursively enumerable* (or semidecidable) if there exists a program P that enumerates all strings in S , or, equivalently, if there exists a program P that halts on every input $\sigma \in S$ and, if it halts, correctly indicates whether $\sigma \in S$.

Moreover, we will fix a language and proof system for first-order logic that is both sound and complete: a first-order formula has a proof if and only if it is a tautology. We additionally require that this language has a rich enough vocabulary (with respect to function and relation symbols) to guarantee that the set of all its tautologies is undecidable (e.g. [Mendelson \[2009\]](#), Proposition 3.54 (Church's Theorem)).

Learning We now first recall some of the definition of PAC learning for binary classification introduced in Part I. We let $X = \mathbb{N}$ denote the domain and $Y = \{0, 1\}$ denote the label space. A *hypothesis* is a function $h : X \rightarrow Y$. We will often identify such binary functions h with the subset of the domain that h maps to 1 and denote this as $X_h = h^{-1}(1)$. A *hypothesis class* $\mathcal{H} \subseteq Y^X$ is a set of hypotheses.

As is common in learning theory, we assume that data is generated by some distribution D over $X \times \{0, 1\}$. We denote the *error* of a hypothesis h with respect to the distribution D by $L_D(h) = \text{Prob}_{(x,y) \sim D}[h(x) \neq y]$. A *learner* is a function that takes in a finite sequence of labeled domain points $S = ((x_1, y_1), \dots, (x_n, y_n))$ and outputs a hypothesis h . The *empirical error* of a hypothesis h with respect to a sample $S = ((x_1, y_1), \dots, (x_n, y_n))$ is defined as $L_S(h) = \frac{1}{n} \sum_{i=1}^n \mathbf{1}[h(x_i) \neq y_i]$

Definition 10.2.1 ((Agnostic) PAC learnability). *A hypothesis class $\mathcal{H}$ is agnostic PAC learnable if there exists a learner A such that for all $\varepsilon, \delta \in (0, 1)$, there is a sample size $m(\varepsilon, \delta)$ such that, for any distribution D , if the input to A is an iid sample S from D of size at least $m(\varepsilon, \delta)$, then, with probability at least $(1 - \delta)$ over the samples, the learner outputs a hypothesis $h = A(S)$ with*

$$L_D(h) \leq \inf_{h \in \mathcal{H}} L_D(h) + \varepsilon$$

The class is said to be PAC learnable in the realizable case, if the above holds under the condition that $\inf_{h \in \mathcal{H}} L_D(h) = 0$.

Definition 10.2.2 (Proper learning). *A hypothesis class $\mathcal{H}$ is proper (agnostic) PAC learnable if it is (agnostic) PAC learnable with a learner that always outputs a hypothesis $h \in \mathcal{H}$ from the class.*

The standard notion of learnability (as in Definition 10.2.1) reflects a guarantee that holds uniformly for all functions in the hypothesis class $\mathcal{H}$ (and uniformly overall data-generating distributions). Namely, from a certain sample size on, the output of the learner has to compete with the *best* hypothesis in $\mathcal{H}$. We now introduce a notion of *non-uniform learnability* where we allow the sample size that the learner requires to depend on the hypothesis $h \in \mathcal{H}$ that it is aiming to compete with.

Definition 10.2.3 (Non-uniform learnability). *We say that a hypothesis class $\mathcal{H}$ is non-uniformly learnable if there exists a learner A and a function $m : (0, 1)^2 \times \mathcal{H} \rightarrow \{0, 1\}$ such that, for every $\varepsilon, \delta \in (0, 1)$, every hypothesis $h \in \mathcal{H}$ and for every distribution D , if $m \geq m(\varepsilon, \delta, h)$, then with probability of at least $1 - \delta$ over the choice of $S \sim D^m$, it holds that*

$$L_D(A(S)) \leq L_D(h) + \varepsilon.$$

10.2.2 CPAC learnability

Computable learning

We now develop our framework of CPAC learnability. While we do not impose any (polynomial) efficiency requirements on the runtime of our learners, we will require the learners to be computable. In addition we would like to also impose computability requirements on the output hypotheses of these learners and for the hypothesis class $\mathcal{H}$. That is, we would like to ensure that a claim of “learnability” for a hypothesis class implies the existence of an

algorithm that implements the learning process and that this learning algorithm provides an output hypothesis in a form that allows for (computably) evaluating this hypothesis on all inputs of the domain X .

Remark 1. It is not straightforward to define what a “computable class of functions” is. Computability is defined as a property of sets of finite words. Here we are considering sets of infinite objects – functions from $\mathbb{N}$ to $\{0, 1\}$. Of course, we may wish to consider only computable functions, and in that case each function can be represented by a finite program. However, while a *set of programs* may be decidable, this is not naturally implied by “simplicity” of the *set of functions* these programs encode. Note that even for a set containing only one simplest function, say the constant-zero function, the set of all programs that encode it (in any fixed given programming language) is not decidable.

In light of the above discussion, we pose as a minimal requirement for our hypothesis classes that they consist of computable functions. Further, we introduce the following two additional restrictions on the computability properties (decidable and recursively enumerable) of the set of programs that compute these functions and that the learning algorithm will use to output a hypothesis. Thus, we distinguish the following two notions for the representations for hypothesis classes:

Definition 10.2.4 (Decidable Representation (DR) of a Hypothesis class). *We say that a class of functions, $\mathcal{H}$, is Decidably Representable (DR) if there exists a decidable set of programs $\mathcal{P}$ such that the set of all functions computed by a program in $\mathcal{P}$ equals $\mathcal{H}$.*

Definition 10.2.5 (Recursively Enumerable Representation (RER) of a Hypothesis class). *We say that a class of functions $\mathcal{H}$ is Recursively Enumerably Representable (RER) if there exists a recursively enumerable set of programs $\mathcal{P}$ such that the set of all functions computed by a program in $\mathcal{P}$ equals $\mathcal{H}$.*

Remark 2. There are other ways of finitely representing functions over a countable domain. For example, we may restrict our attention to functions that have finite support, that is, functions that are constant for all large enough inputs:

$$\mathcal{F} = \{f \in \{0, 1\}^{\mathbb{N}} \mid \exists n_f \in \mathbb{N} \text{ such that } \forall i, j > n_f, f(i) = f(j)\}.$$

Functions in $\mathcal{F}$ can be represented by explicitly listing the finite set of instances on which their value differs from the value that the function converges to. Most of the hypothesis classes we introduce in Section 10.4 consist of such functions with finite support, and we will by default assume the list representation. It is easy to see that such a list could be turned into a program that computes the function, thus all functions in $\mathcal{F}$ are computable. However, subsets of $\mathcal{F}$ may not be DR or RER in the sense of the above Definitions 10.2.4 and 10.2.5.

We now introduce our notion of *computable PAC learnability*, or *CPAC learnability* for short. The minimal requirement on the output of a CPAC learner is that it uses a representation that allows for evaluating the output hypothesis on every input of the domain.

Definition 10.2.6 (CPAC learnability). *We say that a class $\mathcal{H}$ is (agnostic) CPAC learnable, if there is a computable (agnostic) PAC learner for $\mathcal{H}$ that uses a representation for the predictors it outputs, that allows it to evaluate the outputted function on each domain point. If the learner always outputs a (representation of) a hypothesis in class $\mathcal{H}$, we call it a proper CPAC learner and the class proper CPAC learnable.*

In addition, we may require that the CPAC learner uses representations according to the definitions of DR and RER classes.

Without explicitly defining these here, we note that the above also imply notions of non-uniform learnability with additional requirements for the learner to be a proper learner with respect to $\mathcal{H}$ and/or for the learner to be computable.

10.3 Summary of Results

We consider various learning scenarios and investigate whether learnability in the general sense (where learners are modeled as arbitrary functions) implies learnability with computable learners. We summarize our results in the tables below. As discussed earlier, we only consider classes of computable predictors, thus “any class” assumes only this. Note also that a class being Decidably Representable (DR) implies that it is Recursively Enumerably Representable (RER), which implies that it consists of computable functions. Thus, some of the results in the below tables are inherited via these inclusions.

Uniform proper Learning			
	Any class	RE class	DR class
Realizable	PAC $\nRightarrow$ CPAC Theorem 10.4.1	PAC $\Rightarrow$ CPAC Theorem 10.4.2	PAC $\Rightarrow$ CPAC implied by Theorem 10.4.2
Agnostic	PAC $\nRightarrow$ CPAC implied by Theorem 10.4.3	PAC $\nRightarrow$ CPAC implied by Theorem 10.4.3	PAC $\nRightarrow$ CPAC Theorem 10.4.3

Uniform improper Learning

	Any class	RE class	DR class
Realizable	open	PAC $\Rightarrow$ CPAC implied by Theorem 10.4.2	PAC $\Rightarrow$ CPAC implied by Theorem 10.4.2
Agnostic	left open later solved by Sterkenburg [2022] (PAC $\nRightarrow$ CPAC)	left open later solved by Sterkenburg [2022] (PAC $\nRightarrow$ CPAC)	left open later solved by Sterkenburg [2022] (PAC $\nRightarrow$ CPAC)

For the case of non-uniform learning, our Theorem 10.5.3 shows that there is a general class (actually the class of all computable functions) that is learnable, but not with a computable learner. We note that this had been shown earlier with a different proof [Soloveichik, 2008]. This negative result holds in the realizable case for any (not necessarily proper) learner. Whether non-uniform learnability implies non-uniform learnability with a computable learner for more restricted classes (for example DR or RER classes) is currently open.

10.4 Proper CPAC Learning

We start by considering the proper learning setup and show that, in general, even in the realizable case, PAC learnability does not imply CPAC learnability.

Theorem 10.4.1. *There exists a class of computable functions that has VC-dimension 1 and is not proper CPAC learnable (even in the realizable setup).*

Proof. Let the domain set X be the set of natural numbers, $\mathbb{N}$. Fix a recursive enumeration of all Turing machines $(T_i)_{i \in \mathbb{N}}$. For every $i \in \mathbb{N}$, define h_i as follows:

$$h_i(j) = \begin{cases} 1 & \text{if } j = 2i \\ 1 & \text{if } j = 2i + 1 \text{ and the } i\text{'th Turing machine halts on the empty input} \\ 0 & \text{if } j = 2i + 1 \text{ and the } i\text{'th Turing machine does not halt on the empty input} \\ 0 & \text{otherwise} \end{cases}$$

All of these functions have finite support, i.e. X_{h_i} is finite for any h_i . Therefore all h_i are computable functions. Finally, let $\mathcal{H}_{\text{halting}}$ be the class $\{h_i : i \in \mathbb{N}\}$. Note that $\text{VC-dimension}(\mathcal{H}_{\text{halting}}) = 1$, since the functions have disjoint support.

Assume by way of contradiction that there exists a computable proper learner for $\mathcal{H}_{\text{halting}}$. Consider the probability distribution over $X \times \{0, 1\}$ that assigns all the probability mass to $(2i, 1)$ and 0 to all other points. It is easy to see that for every $i \in \mathbb{N}$, on input training sample $(2i, 1)$ such a learner must output the function h_i . However, since we require CPAC learners to output classifiers using a representation that allows for evaluating them on every domain point, such a CPAC learner would provide a solution to whether Turing Machine T_i halts on the empty input by testing the output of h_i on the test point $2i + 1$. This is a contradiction because this implies we computably solve the halting problem, which is obviously not possible. $\square$

Note that, while the class of functions $\mathcal{H}_{\text{halting}}$ in the proof of the above theorem consists of computable functions (see Remark 2), the class is not RER or DR in the sense of Definitions 10.2.4 and 10.2.5. We next show, that, at least in the realizable case, if a class is RER, then a class is PAC learnability implies CPAC learnability. Recall that, without computability requirements, a hypothesis class is PAC learnable if and only if it has finite VC-dimension, and in this case, every empirical risk minimizing (ERM) learner is a successful PAC learner (see, for example, Theorem 6.7 in the textbook [Shalev-Shwartz and Ben-David \[2014a\]](#)).

Theorem 10.4.2. *Every recursively enumerably representable (RER) hypothesis class $\mathcal{H}$ that has finite VC-dimension is proper CPAC learnable (in the realizable case).*

Proof. We will argue that for such classes there is an algorithm that implements ERM (empirical risk minimization) with respect to this class. Namely, given any realizable labeled sample S , run an algorithm that generates all members of $\mathcal{H}$ one by one, and, for each, compute its empirical error on S . Since we are in the realizable setting, we are guaranteed to eventually find some $h \in \mathcal{H}$ with zero empirical error. As soon as we reach such an h , we halt and h is the output of the learner. $\square$

Next, we will show that the above result crucially relies on the realizability assumption. We show that even if a hypothesis class meets the stronger requirement of being decidable representable (DR), PAC learnability does not imply proper CPAC learnability in the agnostic case.

Theorem 10.4.3. *There exists a decidable representable (DR) hypothesis class of finite VC-dimension (of functions with finite support) that is not proper CPAC agnostically learnable.*

Proof. Let the domain set X be the set of natural numbers, $\mathbb{N}$. Fix a proof system for first order logic over a rich enough vocabulary that is sound and complete (i.e., every first order formula of that language has a proof if and only if it is a logical truth). By “rich enough vocabulary” we mean a finite set of functions symbols and relation symbols so that the set of all its logical truths is undecidable, for example a language for the natural numbers with the ordering, addition, and multiplication. For every $i, j \in \mathbb{N}$, we define a function h_{ij} as follows:

$$h_{ij}(k) = \begin{cases} 1 & \text{if } k = 2i \\ 1 & \text{if } k = 2j + 1 \\ 0 & \text{otherwise} \end{cases}$$

Enumerate all proofs and logical statements of the proof system. Let

$$\mathcal{H}_{LT} = \{h_{ij} : \text{The } i\text{'th proof in our logic is a proof for the } j\text{'th formula}\}.$$

Note that $\mathcal{H}_{LT}$ is a decidable class. Each function in $\mathcal{H}_{LT}$ has a finite support, and we may therefore assume a list representation for these functions (see Remark 2). Now, to decide if some finite support function is in $\mathcal{H}_{LT}$, we first check if the function maps exactly one even natural number and one odd natural number to 1 and maps all other natural numbers to 0. Then we check if the proof corresponding to the even number, proves the logical statement corresponding to the odd number. Since every function in $\mathcal{H}_{LT}$ assigns the value 1 to only two inputs, $\text{VC}(\mathcal{H}_{LT}) \leq 2$.

Now we will argue that this class is not agnostic CPAC learnable. By way of contradiction, assume that there exists a proper CPAC learner for $\mathcal{H}_{LT}$ in the agnostic setting. For every $i \in \mathbb{N}$, consider a distribution D_i that has all probability mass on the point $(2i + 1, 1)$. Thus, every training sample from this distribution is a sequence containing only this point with label 1. Note that if statement $2i + 1$ is a logical truth, then a CPAC learner needs to output function $h_{j, 2i+1}$, there proof j certifies the truth of statement $2i + 1$. Otherwise, the CPAC learner may output any function from the class (since all functions in $\mathcal{H}_{LT}$ have error 1 in that case).

Thus, by feeding this learner a sequence of points $(2i + 1, 1)$ and observing the label of the output predictor of this learner on points $2i + 1$, we developed a computable procedure to decide whether a statement is a logical truth, in contradiction to our assumption on the richness of the language. $\square$

10.4.1 Computable (approximations to) empirical risk minimization

Note that the class $\mathcal{H}_{LT}$ from the proof of Theorem 10.4.3 is proper CPAC learnable in the realizable case. Since it is DR, it is also RER, and this thus follows by Theorem 10.4.2. In particular, observe that in the realizable case, an input sample point $(2i + 1, 1)$ can only occur if the i 'th formula is a logical truth. The learner can then go over all finite sequences in lexicographic order until it finds a proof for the i 'th formula. Similarly, the argument in the proof of Theorem 10.4.2 tells us that if a class is RER, then the recursive enumerability allows us to search for an ERM hypothesis in the class if we are given a promise that there exists a hypothesis in $\mathcal{H}$ with zero empirical error. However, lacking such a promise in the agnostic case, the search is not guaranteed to terminate, leading to the negative conclusions for $\mathcal{H}_{LT}$ in Theorem 10.4.3.

In this subsection we explore intermediate scenarios. Specifically, we identify other conditions under which such a search will terminate. In some cases, this implies computable empirical risk minimization also in the agnostic case.

Theorem 10.4.4. *If $\mathcal{H}$ is an RER class and S is a training sample for which the learner is given a promise that*

$$\min_{h \in \mathcal{H}} L_S(h) \leq \varepsilon$$

for some $\varepsilon \in [0, 1)$, then there is an algorithm that outputs an $h \in \mathcal{H}$ with $L_S(h) \leq \varepsilon$.

Proof. Similarly to the realizable case, an algorithm can iterate over the hypothesis class, and evaluate the empirical loss of every h on S of every hypothesis and terminate, when it finds a hypothesis h for which $L_S(h) \leq \varepsilon$ holds. It is guaranteed to halt by the promise in the statement that such an $h \in \mathcal{H}$ exists. $\square$

The above result does not guarantee computable empirical risk minimization (as long as there is no promise on the actual minimal empirical risk). However, if $\mathcal{H}$ is RER and has finite VC-dimension, and we are given a bound α on the approximation error of the class (that is, we know $\inf_{h \in \mathcal{H}} L_P(h) \leq \alpha$) and, in addition, we required the learner to halt and output a hypothesis only with high probability over the sample, then we can use the above result to find a hypothesis in $\mathcal{H}$ that has error close to α . In that case, by standard uniform convergence results for classes of finite VC-dimension, we have

$$\min_{h \in \mathcal{H}} L_S(h) \leq \alpha + \varepsilon$$

with high probability over the draw of sample S and in this case Theorem 10.4.4 yields a hypothesis h with $L_P(h) \leq \inf_{h \in \mathcal{H}} L_P(h) + 2\varepsilon$.

Next, we present a situation, in which ERM is computably possible even in the (fully) agnostic case. Note that the condition below holds, for example for so-called *maximum classes* [Floyd and Warmuth, 1995], in which the number of behaviors on each finite domain subset achieves the bound in Sauer's lemma (see Lemma 6.10 in Shalev-Shwartz and Ben-David [2014a]). Recall that, for a hypothesis h , we let $X_h \subseteq X$ denote the domain points that h labels with 1.

Theorem 10.4.5. *Let $\mathcal{H}$ be an RER class such that for every finite domain subset $W \subseteq X$, the number of subsets of the form $W \cap X_h$ for $h \in \mathcal{H}$ can be computed. Then there is an algorithm that computes ERM for $\mathcal{H}$ (even in the non-realizable case). In particular, if $\mathcal{H}$ in addition has finite VC-dimension, then $\mathcal{H}$ is (agnostically) CPAC learnable.*

Proof. Given a sample $S = ((x_1, y_1), \dots, (x_n, y_n))$, we let $S_X = \{x_1, \dots, x_n\}$ denote the set of (unlabeled) domain points that occur in S . Since $\mathcal{H}$ is RER, there is an algorithm that enumerates functions in $\mathcal{H}$ and, for each can then compute the intersection $S_X \cap X_h$. Similar to the arguments in the proofs of Theorems 10.4.2 and 10.4.4, the number of subsets of the form $S_X \cap X_h$ gives a stopping criterion for the search for an ERM. $\square$

We immediately get the following corollary for maximum classes:

Corollary 10.4.6. *If $\mathcal{H}$ is an RER class, has bounded VC-dimension and is a maximum class, then it is CPAC learnable.*

Finally, we show that for classes of initial segments over some partial ordering are CPAC learnable even in the agnostic case.

Theorem 10.4.7. *If $\mathcal{H}$ is the class of all initial segments under any (partial) ordering relation (containing also the constant 0 hypothesis), and $\mathcal{H}$ is RER then there is a CPAC ERM learner for $\mathcal{H}$ (even in the agnostic case).*

Proof. First, note that, since $\mathcal{H}$ is a class of initial segments, the VC-dimension of $\mathcal{H}$ is 1.

Let $\prec$ denote the partial order, and for $x \in X$, let $\text{init}(x)$ denote the initial segment corresponding to x , that is $\text{init}(x) = \{y \in X \mid y \prec x\}$. Since $\mathcal{H}$ consists of initial segments, for each $h \in \mathcal{H}$, there is an $x_h \in X$ such that $X_h = \text{init}(x_h)$ (recall that X_h denotes the set

of all domain points that are labeled 1 by h). And since we assume that $\mathcal{H}$ is the class of *all* initial segments over $\prec$, we get

$$\mathcal{H} = \{\text{init}(x) \mid x \in X\}.$$

Now note that for every pair of points $(x, y) \in X^2$, the class $\mathcal{H}$ has exactly three behaviors. If x and y are comparable by $\prec$, say without loss of generality $x \prec y$, then $\mathcal{H}$ contains the labelings $(0, 0)$, $(0, 1)$ and $(1, 1)$ for (x, y) . If x and y are not comparable, then $\mathcal{H}$ contains the labelings $(0, 0)$, $(0, 1)$ and $(1, 0)$ for (x, y) .

This, again, can be turned into a stopping criterion for a search for an empirical risk minimizer. Given a sample S , there is an algorithm enumerates all functions in $\mathcal{H}$, until for each pair of domain points in S , three labeling behaviors were observed. Then, it can compute the empirical error of each of these functions over S and output an ERM hypothesis $h \in \mathcal{H}$. Since the VC-dimension of $\mathcal{H}$ is 1, this is a CPAC learner for $\mathcal{H}$. $\square$

10.5 Computable non-uniform Learning

We now consider the non-uniform learning setup (recall Definition 10.2.3), and show that, here as well, there are classes that are learnable (when learners can be arbitrary functions), not by a computable learner. A similar result had been shown in Soloveichik [2008], however, we provide a different proof. We first note that a hypothesis class $\mathcal{H}$ is non-uniformly learnable if and only if it is a countable union of finite VC-classes.

Theorem 10.5.1 (Theorem 7.2 in Shalev-Shwartz and Ben-David [2014a]). *$\mathcal{H}$ is non-uniformly learnable if and only if there exist a sequence of classes $(\mathcal{H}_i)_{i \in \mathbb{N}}$, each of finite VC-dimension, such that $\mathcal{H} = \bigcup_{i \in \mathbb{N}} \mathcal{H}_i$.*

Corollary 10.5.2. *The class $\mathcal{H}_{\text{comp}}$ of all computable functions from $\mathbb{N}$ to $\{0, 1\}$ is non-uniformly learnable.*

Proof. Since each function h in $\mathcal{H}_{\text{comp}}$ is computable, there exists a program computing h . Thus, $\mathcal{H}_{\text{comp}}$ is countable, and therefore Theorem 10.5.1 applies. $\square$

However, we will show that the class $\mathcal{H}_{\text{comp}}$ cannot be learned by any computable learner (proper or non-proper).

Theorem 10.5.3 (Also see Soloveichik [2008]). *No computable learner can non-uniformly learn the class $\mathcal{H}_{\text{comp}}$.*

We will now provide an alternative proof for this result. We start by proving a “computable version” of the classic No-Free-Lunch Theorem, which states that for every learner, for a fixed sample size m , there exists a (simple) distribution over $2m$ domain points, on which this learner will suffer a high expected loss. We show that, if the learner is computable, there is also an algorithm that finds this distribution. The Lemma and proof below are modifications of Theorem 5.1 in [Shalev-Shwartz and Ben-David \[2014a\]](#).

Lemma 10.5.4 (Computable No-Free-Lunch). *For any computable learner A , and for any $m \in \mathbb{N}$, any domain X of size at least $2m$, any subset $\{x_1, \dots, x_{2m}\} \subseteq X$ of size $2m$, we can computably find a function $f : \{x_1, \dots, x_{2m}\} \rightarrow \{0, 1\}$ such that*

$$\mathbb{E}_{S \sim D^m}[L_D(A(S))] \geq \frac{1}{4}, \quad \text{and} \quad \text{Prob}_{S \sim D^m} \left[L_D(A(S)) \geq \frac{1}{8} \right] \geq \frac{1}{7},$$

where D is a uniform distribution over $\{(x_1, f(x_1)), \dots, (x_{2m}, f(x_{2m}))\}$.

Proof. The existence of f and D with the stated properties follow from the original No-Free-Lunch Theorem. All that is left to show here, is that we can computably find the function f .

There are $T = 2^{2m}$ functions from $\{x_1, \dots, x_{2m}\}$ to $\{0, 1\}$, which we denote by $f_1, \dots, f_T$. For each $1 \leq i \leq T$, define D_i to be the uniform distribution over $\{(x_1, f_i(x_1)), \dots, (x_{2m}, f_i(x_{2m}))\}$. We know one of these are the function f and distribution D that we are looking for.

For each $1 \leq i \leq T$, there are $k = (2m)^m$ possible sequences of m samples drawn from the distribution D_i . Let us denote these sequences as $S_1^i, \dots, S_k^i$. Each of these sample set occurs with equal probability. Since A is computable, we can calculate the expected loss of the learner with respect to distribution D_i as follows:

$$\mathbb{E}_{S \sim D_i^m}[L_{D_i}(A(S))] = \frac{1}{k} \sum_{j=1}^k L_{D_i}(A(S_j^i)) = \frac{1}{k} \sum_{j=1}^k \left[\frac{1}{2m} \sum_{l=1}^{2m} \mathbf{1}[A(S)(x_l) \neq f_i(x_l)] \right]$$

From the No-Free-Lunch theorem, we know that one of the expected losses is at least $\frac{1}{4}$. We can compute this quantity for each i from 1 to T until we find the quantity at least $\frac{1}{4}$. This gives the function f that we seek. The high probability statement follows from the statement about high expected loss (see [Shalev-Shwartz and Ben-David \[2014a\]](#) for details). $\square$

With this, we proceed to prove that $\mathcal{H}_{comp}$ is not learnable by a computable learner.

Proof of Theorem 10.5.3. Assume by way of contradiction that there is a computable learner A that non-uniformly learns $\mathcal{H}_{comp}$. By Definition 10.2.3, this means that for any computable function h , and any $\varepsilon, \delta \in (0, 1)$, there is a natural number $n_A(\varepsilon, \delta, h)$ such that for any distribution D and any $n \geq n_A(\varepsilon, \delta, h)$, we get

$$\text{Prob}_{S \sim D^n} [L_D(A(S)) \leq L_D(h) + \varepsilon] \geq 1 - \delta.$$

Consider the following partition of the natural numbers $\mathbb{N} = \cup_{i \in \mathbb{N}} P_i$ where each $P_i = [l_i, u_i]$. For each i , $u_i - l_i + 1 = 10i$, $l_{i+1} = u_i + 1$ and $l_1 = 1$. For each $i \in \mathbb{N}$ and for $m_i = 5i$, by Lemma 10.5.4, there exists a function f_i and a distribution D_i on P_i such that for the distribution D'_i over $\mathbb{N} \times \{0, 1\}$, the learner A fails to uniformly learn for $\varepsilon = \frac{1}{8}$ and $\delta = \frac{1}{8}$. Here, D'_i is such that

$$D'_i((x, y)) = \begin{cases} D_i((x, y)) & \text{if } x \in P_i \\ 0 & \text{otherwise} \end{cases}$$

Consider a function $h_0 : \mathbb{N} \rightarrow \{0, 1\}$ such that, for any $x \in \mathbb{N}$, $h_0(x) = f_j(x)$ where j is such that $x \in P_j$ (we can compute such a j given an x). Since each f_i can be found computably, h_0 is a computable function. Hence $h_0 \in \mathcal{H}_{comp}$.

There is a $k \in \mathbb{N}$ such that $|P_k| > 2n_A(\frac{1}{8}, \frac{1}{7}, h_0)$. For the distribution D'_k , with probability at least $\frac{1}{7}$ over samples of size $2n_A(\frac{1}{8}, \frac{1}{7}, h_0)$ drawn from D'_k , $L_{D'_k}(A(S)) > \frac{1}{8}$. This contradicts our assumption that A non-uniformly learned the class $\mathcal{H}_{comp}$. $\square$

Remark 3. Similar impossibility results as those in Theorem 10.5.3, can be shown for certain subclasses of the class $\mathcal{H}_{comp}$. For example, the same proof technique will show that the class of all computable functions with a bounded run time cannot non-uniformly be learned by a learner with an accordingly bounded run time.

10.6 Improper CPAC Learning

To understand the implications of CPAC-learnability in the more general setup of non-proper learners, we obtain the following reduction inspired by a reduction in [Daniely et al. \[2014\]](#).

For a hypothesis class $\mathcal{H}$ and a sequence $S = ((x_1, y_1), (x_2, y_2), \dots, (x_m, y_m))$ of labeled domain points, we say that S has a *realizable labeling* if there is an $h \in \mathcal{H}$ such that $y_i = h(x_i)$ of all pairs $(x_i, y_i) \in S$. We say that S has a *random labeling* if each y_i was the result of a random coin flip where $\text{Prob}(y_i = 1) = \text{Prob}(y_i = 0) = \frac{1}{2}$. With this terminology, we introduce the following task.

Definition 10.6.1 (The Distinguishing Problem). *Given a hypothesis class $\mathcal{H}$, we say that a (potentially randomized) function $A : \bigcup_{n \in \mathbb{N}} (X \times \{0, 1\})^n \rightarrow \{\text{"realizable"}, \text{"unrealizable"}\}$ solves the distinguishing problem if for any $\delta > 0$, there is an $M \in \mathbb{N}$ such that for any sequence $S = (x_1, \dots, x_M) \in X^M$, we have*

- for T - a realizable labeling of S , $\Pr(A(T) = \text{"realizable"}) \geq 1 - \delta$,
- for T - a random labeling of S , $\Pr(A(T) = \text{"unrealizable"}) \geq 1 - \delta$,

where the probability is taken over the randomization of A and the random coin flips that generated the labels in the latter case.

Theorem 10.6.2. *If there is an (agnostic) CPAC learner of a class $\mathcal{H}$ whose range is a class of finite VC-dimension, we can solve the distinguishing problem for $\mathcal{H}$ with a computable distinguisher.*

Proof. We let F be a CPAC learner that learns the class $\mathcal{H}$ and has a range with finite VC-dimension. We let $\mathcal{H}'$ denote this class of output hypotheses of F , and let d denote the VC-dimension of $\mathcal{H}'$.

By the Sauer's Lemma, we know that the number of behaviors $\mathcal{H}'$ exhibits on a set of size m is bounded by a function $p(m)$ that is polynomial in m . We will now show that we can construct an algorithm A to solve the distinguishing problem for $\mathcal{H}$.

For a given $\delta > 0$, choose $M > d$ and large enough so that $p(M) \left(\frac{e}{4}\right)^{\frac{M}{4}} < \delta$. Here $p(M) = \left(\frac{eM}{d}\right)^d$ which is an upper bound to the growth function of $\mathcal{H}'$ for M .

For a labeled sequence $S = ((x_1, y_1), \dots, (x_M, y_M)) \in (X \times \{0, 1\})^M$, we let D_S denote the uniform distribution over S . Now we describe an algorithm A that solves the distinguishing problem. Given S , the algorithm A will

1. By uniformly sub-sampling from S , generate a sample from distribution D_S to run learner F with error parameter $\frac{1}{4}$ and confidence parameter δ .
2. Compute $L_S(h)$ output hypothesis h of F .
3. If $L_S(h) < \frac{1}{4}$, return "realizable". Else return "unrealizable".

Note that for a realizable labeling S , we are guaranteed $L_S(h) < \frac{1}{4}$ with probability at least $1 - \delta$.

The set of hypotheses that F may return has at most $p(M)$ behaviors on any set of instances of size M . The probability that any behavior yields error less than $\frac{1}{4}$ on S is bounded by

$$p(M) \cdot \frac{\binom{M}{\frac{M}{4}}}{2^M} \leq p(M) \cdot \left(\frac{e}{4}\right)^{\frac{M}{4}} < \delta$$

So, with probability at least $1 - 2\delta$ (over the random coin flips and the sampling from D_S), the output hypothesis makes more than $\frac{1}{4}$ error on S and is hence identified as "unrealizable" with at least this probability. $\square$

Observe that the classes $\mathcal{H}_{\text{halting}}$ and $\mathcal{H}_{LT}$ that we used for the negative results on proper CPAC learning in Section 10.4, are actually CPAC learnable if we remove the properness requirement. All functions in these classes map at most two domain-points to 1 and the rest of the domain to 0. They are thus both subclasses of $\mathcal{H}_2$, the class of all hypotheses h with $|X_h| \leq 2$. This class is actually properly CPAC learnable, thus both $\mathcal{H}_{\text{halting}}$ and $\mathcal{H}_{LT}$ are (improperly) CPAC learnable.

Conjectures and (formally) open problems

Inspired by our observations about improper learning, we stated some conjectures and open problems.

Conjecture 10.6.3. *If a class $\mathcal{H}$ is (improperly) CPAC learnable, then there is a superclass $\mathcal{H}' \supseteq \mathcal{H}$ such that $\mathcal{H}'$ is proper CPAC learnable.*

Theorem 10.6.2, along with Conjecture 10.6.3 leads to the following corollary.

Corollary 10.6.4. *If $\mathcal{H}$ is (improper) CPAC learnable, then $\mathcal{H}$ is computably distinguishable.*

We also conjecture the following:

Conjecture 10.6.5. *There is a class $\mathcal{H}$ consisting of computable hypotheses, and with finite VC-dimension, such that $\mathcal{H}$ is not computably distinguishable.*

Corollary 10.6.4, along with Conjecture 10.6.5 would imply the existence of a class $\mathcal{H}$ of finite VC-Dimension that is not (improperly) CPAC learnable.

We also published as open problems the questions of whether there are decidable representable classes of finite VC-dimension which cannot be CPAC learned in the improper case, conjecturing that such a class indeed exists. We proposed the following candidate class.

Let the domain set X be the set of natural numbers, $\mathbb{N}$. Fix a proof system for first order logic over a rich enough vocabulary that is sound and complete (i.e., every first order formula of that language has a proof if and only if it is a logical truth). By “rich enough vocabulary” we mean a finite set of functions symbols and relation symbols so that the set of all its logical truths is undecidable, for example a language for the natural numbers with the ordering, addition, and multiplication. Enumerate all proofs and logical statements of the proof system. We can now define an order $\prec$ over theorems (i.e., statements with a proof), by $i \prec j$ if and only if the proof of i has lower number in the enumeration of proofs than the proof of j . Let

$$h_i(x) = \begin{cases} 1 & \text{if the logical statement with number } x \text{ is a theorem and } x \prec i \\ 0 & \text{otherwise} \end{cases}.$$

We can now define the class $\mathcal{H}_{init} = \{h_i : i \in \mathbb{N}\}$. It is easy to see that the class $\mathcal{H}_{init}$ is decidable and has VC-dimension 1.

Conjecture 10.6.6. *The class $\mathcal{H}_{init}$ is not improperly CPAC learnable.*

[Sterkenburg \[2022\]](#) showed that there are indeed DR classes of VC-dimension 1 which cannot be agnostically CPAC learned, even by improper learners. The construction is similar to our class $\mathcal{H}_{init}$, but uses Turing machines (and their halting after t steps) in order to construct a similar class of initial segments. The proof then uses an n -fold recursion theorem to get a contradiction to a necessary condition for CPAC learnability, which they derive from our computable no-free-lunch theorem in the previous section. This necessary condition for the CPAC learnability of a class $\mathcal{H}$ states that there exists an algorithm, that given any input sample S of sufficiently large size outputs a labelling on S that is not $\mathcal{H}$ -realizable.

We also posed as open question whether there is a characterization of CPAC learnable classes. [Delle Rose et al. \[2023\]](#) indeed provided such a characterization showing that the necessary condition proposed by [Sterkenburg \[2022\]](#) is also a sufficient condition for CPAC learnability.

10.7 Conclusion and Overview over Related Work

We showed that in the case of CPAC learning, improper learnability does not imply proper learnability, contrasting the results from binary classification without computational constraints. We further show that the characterization of proper learnability by the VC dimension breaks down in the agnostic case: we showed that there are decidable representable classes of VC-dimension 1, which are not CPAC learnable. However, in the proper realizable case we can show that it is sufficient for a class to have finite VC-dimension and be recursively enumerable in order to be CPAC learned. This is due to the fact that the realizable case provides a stopping criterion for an algorithm implementing the ERM criterion, as it can search until a hypothesis which realizes the labelling on the training sample is found. Thus we show that for proper CPAC learning, the agnostic and the realizable case are different. It was later shown by Sterkenburg [2022], that there are also decidable representable classes of VC-dimension 1 which are not *agnostically* learnable in the improper case. This combined with our result on realizable CPAC learning showed that there is also a separation between agnostic and realizable learnability if learners are not required to be proper. The result followed a similar construction as our class $\mathcal{H}_{\text{init}}$ proposed in Section 10.6. The proof relied on a result which showed that if a class is (agnostically improperly) CPAC learnable, then there is an algorithm that for every set of sufficient size outputs a labelling that is not realized by the class. This result was shown by using our computable No-Free-Lunch Theorem in Section 10.5. Delle Rose et al. [2023] later showed that the existence of an oracle which finds a non-realizable labeling of the class on any set of size $d + 1$ for large enough d is also a sufficient condition for CPAC learnability. Thus, they provided a characterization of (agnostic improper) CPAC learnability of a class by what they coined its *effective VC-dimension*. The non-realizable-labelling oracle, can be thought of as a computable witness function for the fact that a set is not shattered by the class. Thus one can indeed think of the effective VC-dimension as a computable version of the VC-dimension: In order to show that the VC dimension of a class $\mathcal{H}$ is smaller or equal to d , one needs to show that *for every set S of size $d + 1$, there is a labelling of that set that is not realized by $\mathcal{H}$* . The computable or effective VC dimension, now replaces the "for all" quantifier in that statement with an arbitrary input and the "exists" quantifier with an algorithm that computes such a labelling for every input set of size $d + 1$. Other recent works have extended the study of computability in learning to other learning settings such as continuous domains [Ackerman et al., 2022] and online learning [Hasrati and Ben-David, 2023].

10.7.1 Computable dimensions

Looking at the definition for effective VC-dimension, a natural question that arises is, whether there can always be such "computable" dimensions, for each dimension fulfilling our notion of *finitary characterization* (Definition 3.1.3). If the formulas α_d in W are indeed first-order formulas, one can think of a similar approach: Given such a formula α_d , take its negation $\neg\alpha_d$ and write it in a prefix-free way. Then turn all $\forall$ -quantifiers into potential inputs to an algorithm and all $\exists$ -classifiers to outputs of that algorithm, with the guarantee that they satisfy the corresponding statement in $\neg\alpha_d$. This works well, if $\neg\alpha_d$ can be written in such a way that it is prefix-free and starts with only $\forall$ -quantifiers followed with $\exists$ -quantifiers. In turn, we would require α_d to be prefix-free and start with a sequence of $\exists$ -quantifiers followed by a sequence of $\forall$ -quantifiers. Thus if finitary dimension W consists only of formulas using exists quantifiers over elements of $\mathcal{X}$ and over elements of $\mathcal{H}$, then they also give rise to a computational version. Further exploring which characterizing dimensions allow for a computation version that characterizes the corresponding learning problem with a computability constraint is an exciting direction for future research.

Chapter 11

On the Computability of Robust PAC Learning

In this chapter, I present the work ”On the Computability of Robust PAC Learning” (joint work with Pascale Gourdeau and Ruth Uerner, COLT 2024). In this work, we initiate the study of computable PAC learning in the (adversarially) robust setting. Adversarial robust PAC learning has by now been extensively studied, e.g., by [Montasser et al. \[2019, 2021\]](#), [Gourdeau et al. \[2021b\]](#), [Awasthi et al. \[2023b\]](#), [Lechner et al. \[2023a\]](#) and recent work has provided a characterization of learnability (again, in the purely information-theoretic sense) by a parameter of the one inclusion graph of the learning problem [[Montasser et al., 2022](#)]. We here explore how adding a requirement of computability adds more subtle aspects to the questions of robust PAC learnability, and expose some perhaps surprising aspects of this setup.

We start by setting up a formal framework for robust learning under computability constraints on the learners and hypotheses, and also the perturbation sets employed. As a warm-up, we provide some simple sufficient conditions for learnability in our setting, but also show that the question of robust CPAC learnability is more subtle than the combination of its components: we prove that there exist classes that are CPAC learnable and robustly PAC learnable with respect to perturbation types that are decidable, yet are not robustly CPAC learnable.

We then explore the role of computability of the robust loss for robust CPAC learnability. Perhaps surprisingly, the robust loss being computably evaluable is neither a necessary nor a sufficient requirement for robust learnability. The insufficiency result relies on showing that there are CPAC learnable classes with computably evaluable robust loss that do

not admit a robust empirical risk minimization (RERM) oracle [Montasser et al., 2019] which would return a hypothesis with minimal empirical robust risk on any sample) in the robust agnostic setting. This result is significant as many works assume access to an RERM oracle. In particular, the standard reduction of agnostic to realizable learning in the robust learning setting requires such an oracle [Montasser et al., 2019], and we here demonstrate an example where learnability in the robust realizable case does not extend to the agnostic case through this reduction, since RERM is not computable.

Finally, we explore the role of dimensions for computable robust learnability. We introduce a novel, computable version of the robust shattering dimension. Finiteness of the robust shattering dimension has been shown to be necessary for robust learnability [Montasser et al., 2019], however it has remained an open question whether it is also a sufficient condition [Montasser et al., 2022]. For the computable version, we prove that the finiteness of our *computable robust shattering dimension* is necessary, yet not sufficient, to ensure robust CPAC learnability. This result might provide novel insights for the case without computability constraints. Our results involve deriving a new no free lunch theorem for robust learning, which might also be of independent interest.

11.1 Related Work

The related work for computable learning is discussed in the previous chapter. We will now give a discussion of the related work for adversarially robust learning, recalling some of the discussion of Chapter 9 and giving additional focus to computational aspects.

Robust Learning. There is a rich learning theory literature on adversarial robustness. Earlier work focused on the existence of adversarial examples for the stability notion of robustness [Fawzi et al., 2016, 2018a,b, Gilmer et al., 2018, Shafahi et al., 2019, Tsipras et al., 2019b], and for its true-label counterpart [Diochnos et al., 2018, Mahloujifar et al., 2019]. Many of the works in the former category highlighted the incompatibility between robustness and accuracy, while Bhattacharjee and Chaudhuri [2021] and Chowdhury and Uner [2022] later argued that these two objectives should not be in conflict by proposing notions of adaptive robustness. Some studies focus on the sample complexity of robust learning in a PAC setting through a notion called the adversarial VC dimension [Cullina et al., 2018]. Some upper bounds depend on the number of perturbations allowed for each instance [Attias et al., 2022b], others on the VC and dual VC dimension of a hypothesis class obtained through an improper learner [Montasser et al., 2019], thus also covering

the case of infinite perturbation sets. [Ashtiani et al. \[2020b\]](#) later gave an upper bound for *proper* robust learning in terms of the VC dimensions of the class and induced margin class. [Montasser et al. \[2022\]](#) exhibited a characterization of robust learning based on the one-inclusion graph of [Haussler et al. \[1994\]](#) adapted to robust learning. The sample complexity of robust learning has also been studied in the semi-supervised setting [[Attias et al., 2022a](#)], through the lens of transformation invariances [[Shao et al., 2022](#)], through Rademacher complexity bounds [[Khim et al., 2019](#), [Yin et al., 2019](#), [Awasthi et al., 2020](#)] and by the use of online learning algorithms [[Diakonikolas et al., 2020](#), [Bhattacharjee et al., 2021](#)]. Robust risk relaxations have been studied by [Viallard et al. \[2021\]](#), [Ashtiani et al. \[2023b\]](#), [Bhattacharjee et al. \[2023\]](#), [Raman et al. \[2023\]](#), while [Balcan et al. \[2022, 2023b\]](#) offered reliability guarantees for robust learning under various notions of robustness. In terms of the true-label robust risk, sample complexity upper and lower bounds with access to random examples only have been derived with respect to distributional assumptions [[Diochnos et al., 2020](#), [Gourdeau et al., 2021b, 2022c](#)], as this set-up does not allow distribution-free robust learning [[Gourdeau et al., 2021b](#)]. To circumvent computational or information-theoretic obstacles of robust learning with random examples only, recent work has studied robust learning with the help of oracles [[Montasser et al., 2021](#), [Gourdeau et al., 2022b](#), [Lechner et al., 2023a](#)], while other work has instead looked at curtailing the (computational) power of the adversary [[Mahloujifar and Mahmood, 2019](#), [Garg et al., 2020](#)]. Prior works have also analyzed the role of computational efficiency in robust learnability [[Bubeck et al., 2019](#), [Degwekar and Vaikuntanathan, 2019](#), [Gourdeau et al., 2021b](#)], however we are the first to exhibit effects of issues arising from the basic, essential requirement of computability.

11.1.1 Robust CPAC Learning: Definition

We recall the definition of adversarially robust learning from a previous chapter.

Adversarial robustness. Let $u : \mathcal{X} \rightarrow 2^{\mathcal{X}}$ be a *perturbation type*, assigning each point $x \in \mathcal{X}$ a region $u(x)$ accessible to an adversary, with the convention $x \in u(x)$. We define the robust risk as:

$$\mathcal{L}_D^u(h) = \mathbb{P}_{(x,y) \sim D} (\exists z \in u(x) . h(z) \neq y) .$$

Similarly as in the standard setting, we will consider the robust loss function $\ell^u : \mathcal{H} \times \mathcal{X} \times \{0, 1\} \rightarrow \{0, 1\}$ defined as $\ell^u(h, x, y) = \mathbf{1}[\exists z \in u(x) . h(x) \neq y]$, and denote the empirical robust risk of predictor h over a sample S by $\mathcal{L}_S^u(h)$.

We will here work with this *stability* notion of robustness with known perturbation sets, while prior work has also explored other notions of robustness with emphasis on label correctness [Mahloujifar et al., 2019, Gourdeau et al., 2021b, Bhattacharjee and Chaudhuri, 2021, Chowdhury and Urner, 2022].

Definition 11.1.1 (Agnostic Robust PAC Learnability [Montasser et al., 2019]). *A hypothesis class $\mathcal{H}$ is u -robustly PAC learnable in the agnostic setting if there exists a learner $\mathcal{A}$ and a function $m(\cdot, \cdot)$ such that for all $\varepsilon, \delta \in (0, 1)$ and for every distribution D , if the input to $\mathcal{A}$ is an i.i.d. sample S from D of size at least $m(\varepsilon, \delta)$, then, with probability at least $(1 - \delta)$ over the samples, the learner outputs a hypothesis $\mathcal{A}(S)$ with $\mathcal{L}_D^u(\mathcal{A}(S)) \leq \inf_{h \in \mathcal{H}} \mathcal{L}_D^u(h) + \varepsilon$. The class is said to be u -robustly PAC learnable in the realizable setting if the above holds under the condition that $\inf_{h \in \mathcal{H}} \mathcal{L}_D^u(h) = 0$.*

Computable robust learnability. We can straightforwardly adapt the definition of CPAC learnability to its robust counterpart:

Definition 11.1.2 (Robust CPAC Learnability). *We say that a class $\mathcal{H}$ is u -robustly (agnostic) CPAC learnable, if there exists a u -robust PAC learner for $\mathcal{H}$ that also satisfies the computability requirements of a CPAC learner.*

Similarly to the requirements for $\mathcal{H}$, we can require the perturbation type u also be DR or RER:

Definition 11.1.3 (Representations of Perturbation Types). *Let $u : \mathcal{X} \rightarrow 2^{\mathcal{X}}$ be a perturbation type. Then u is said to be decidably representable (or recursively enumerable) if the set $\{(x, z) . x \in \mathcal{X}, z \in u(x)\}$ is decidable (or recursively enumerable, respectively).*

Asking for perturbation types to be decidably representable is quite a natural requirement. Indeed, we show in Appendix 11.6.2 an example that uses a perturbation type u that is not DR and thus makes the impossibility of u -robust CPAC learning trivial.

As opposed to the binary loss, which depends only on a simple comparison between two labels, and can thus always be evaluated for computable predictors, the robust loss is not always computably evaluable. In Section 11.3.2, we explore the role of the following notion of loss computability.

Definition 11.1.4 (Robust loss computably evaluable). *Given a hypothesis class $\mathcal{H}$, and perturbation type u , we say that the robust loss for u is computably evaluable on $\mathcal{H}$ if $\ell^u : \mathcal{H} \times \mathcal{X} \times \{0, 1\} \rightarrow \{0, 1\}$ is a total computable function.*

Proper learnability and empirical risk minimization. We also recall that for all introduced notions of PAC learnability, we call the class $\mathcal{H}$ *properly PAC/CPAC/ u -robustly PAC/ u -robustly CPAC learnable* if there exists a learner that satisfies the corresponding definition of learnability and always outputs functions from $\mathcal{H}$. A learner $\mathcal{A}$ is said to *perform empirical risk minimization (ERM)*, or *implement an $\text{ERM}_{\mathcal{H}}$ oracle*, if it always outputs a predictor from the hypothesis class of minimal empirical risk, that is for all $m \in \mathbb{N}$ and samples $S \in (\mathcal{X} \times \mathcal{Y})^m$,

$$\mathcal{A}(S) \in \arg \min_{h \in \mathcal{H}} \mathcal{L}_S(h) .$$

Robust empirical risk minimization (RERM) is defined analogously for the robust risk $\mathcal{L}_S^u$. We now outline different types of RERM oracles that we will study throughout this work. A labelled sample S is said to be *robustly realizable* if $\min_{h \in \mathcal{H}} \mathcal{L}_S^u(h) = 0$, and *robustly non-realizable* otherwise. We will distinguish between three cases:

- **weak-realizable $\text{RERM}_{\mathcal{H}}^u$ oracle:** for any robustly realizable sample S , the oracle outputs h with $\mathcal{L}_S^u(h) = 0$. On robustly non-realizable samples such an oracle is allowed not to halt.
- **strong-realizable $\text{RERM}_{\mathcal{H}}^u$ oracle:** for any robustly realizable sample S , the oracle outputs h with $\mathcal{L}_S^u(h) = 0$. For any robustly non-realizable sample, it halts and outputs “not-robustly-realizable”.
- **agnostic $\text{RERM}_{\mathcal{H}}^u$ oracle:** the oracle performs RERM on both robustly realizable and robustly non-realizable samples.

Note that, if the robust loss is computably evaluable, then the existence of a computable strong realizable $\text{RERM}_{\mathcal{H}}^u$ oracle is equivalent to the existence of an agnostic $\text{RERM}_{\mathcal{H}}^u$ oracle. Moreover, a strong-realizable $\text{RERM}_{\mathcal{H}}^u$ oracle always implies the existence of an agnostic $\text{RERM}_{\mathcal{H}}^u$ oracle: One can first run the oracle on the whole sample, if the sample is realizable, the learner will output an ERM hypothesis. If not, we can run the learner on all subsets of S of size $|S| - 1$. If the strong realizable oracle outputs a hypothesis that is realizable on one of these subsamples, this hypothesis is also ERM on S itself. If there is no such subsample, now consider subsamples of size $|S| - 2$. We go on iterating through all possible subsamples of S in order of decreasing size. This guarantees that the first subsample on which the strong-realizable ERM oracle returns a hypothesis realizing the subsamples labelling, it is a realizable subsample of maximal size. This guarantees the returned hypothesis is (agnostic) ERM on S as well.

Useful facts. Recall the following sufficient condition for CPAC learnability, which we later use:

Fact 11.1.5 ([Agarwal et al., 2020, Sterkenburg, 2022]). If $\text{VC}(\mathcal{H})$ is finite and there exists a total computable function that implements an $\text{ERM}_{\mathcal{H}}$ oracle, then $\mathcal{H}$ is CPAC learnable.

We now recall the following result by Ashtiani et al. [2020b], which outlines sufficient conditions for the proper robust learnability of a class. Throughout this text, we will use the notation of Ashtiani et al. [2020b], where the *margin class* $\mathcal{H}_{\text{mar}}^u := \{\text{mar}_h^u\}_{h \in \mathcal{H}}$ consists of the margin sets $\text{mar}_h^u := \{x \in \mathcal{X} \mid \exists z \in u(x) . h(x) \neq h(z)\}$, i.e., given $\mathcal{H}$ and u , the instances in $\mathcal{X}$ that incur a robust loss of 1 due to a perturbation that is labeled differently by h (rather than due to mislabeling).

Fact 11.1.6 (Theorem 7 in [Ashtiani et al., 2020b]). If both the VC dimension of a class $\mathcal{H}$ and the VC dimension of $\mathcal{H}_{\text{mar}}^u$ are finite, then $\mathcal{H}$ is properly (agnostically) u -robustly PAC learnable.

11.2 Simple Sufficient Conditions for Robust CPAC Learnability

We start by exhibiting conditions ensuring (proper) robust CPAC learnability in the agnostic setting.

Fact 11.2.1. Let hypothesis class $\mathcal{H}$ and perturbation type u be such that $\text{VC}(\mathcal{H}) + \text{VC}(\mathcal{H}_{\text{mar}}^u) < \infty$ and there exists a total computable function that implements an agnostic $\text{RERM}_{\mathcal{H}}^u$ oracle. Then $\mathcal{H}$ is properly u -robustly CPAC learnable. If there exists a computable function that implements a weak-realizable $\text{RERM}_{\mathcal{H}}^u$ oracle, then $\mathcal{H}$ is u -robustly CPAC learnable in the realizable case.

Proof. By Fact 11.1.6, $D := \text{VC}(\mathcal{H}) + \text{VC}(\mathcal{H}_{\text{mar}}^u) < \infty$ implies that a finite sample of size $O\left(\frac{D \log D + \log 1/\delta}{\varepsilon^2}\right)$ is sufficient to guarantee robust generalization w.r.t. u , whenever a robust risk minimizer for a given sample is returned. Since robust ERM is computably implementable, we are done. $\square$

Fact 11.2.2. Let hypothesis class $\mathcal{H}$ be RER and perturbation type u be such that ℓ^u is computably evaluable on $\mathcal{H}$. Then $\mathcal{H}$ admits a weak realizable RERM oracle.

Proof. Let S be a robustly realizable input sample. The following procedure computes $\text{RERM}_{\mathcal{H}}^u$. Since the class is RER, we can iterate through the class $\mathcal{H} = (h_i)_{i \in \mathbb{N}}$ and for each h_i :

- Compute $\mathcal{L}_S^u(h_i)$, which is possible since the robust loss is computably evaluable;
- Check whether $\mathcal{L}_S^u(h_i) = 0$. If so, halt and output h_i , otherwise continue.

As we know that there exists $h \in \mathcal{H}$ with $\mathcal{L}_S^u(h) = 0$, we know that the algorithm will halt. $\square$

Remark 4. Combining Facts 11.2.1 and 11.2.2 shows that for a RER hypothesis class $\mathcal{H}$ and a perturbation type u , the conditions $\text{VC}(\mathcal{H}) + \text{VC}(\mathcal{H}_{\text{mar}}^u) < \infty$ and ℓ^u on $\mathcal{H}$ being computably evaluable are sufficient to guarantee proper robust CPAC learnability in the robustly realizable case.

Now, we show that having access to a computable online learner (see Definition 15 in [Hasrati and Ben-David, 2023]) and a counterexample oracle, together with the margin class having finite VC dimension, is sufficient to guarantee robust learnability. The counterexample oracle we will use is the Perfect Attack Oracle (PAO) of [Montasser et al., 2021], which takes as input $h, x, y \in \mathcal{H} \times \mathcal{X} \times \{0, 1\}$ and returns $z \in u(x)$ such that $h(z) \neq y$, if such a counterexample exists. The proof is included in Appendix 11.6.1.

Proposition 11.2.3. *Let $\mathcal{H}$ be computably online learnable, and let u be such that $\text{VC}(\mathcal{H}_{\text{mar}}^u) < \infty$. Then $\mathcal{H}$ is u -robustly CPAC learnable with access to the PAO in the realizable setting.*

11.3 Relating CPAC and Robust CPAC Learning

In this section, we study the relationship between CPAC learnability, robust CPAC learnability and the computable evaluability of the robust loss. We start in Section 11.3.1 with examples where we have CPAC learnability, but not robust CPAC learnability. The constructions in that section also show that the robust loss being computably evaluable is not sufficient for robust CPAC learnability. In Section 11.3.2, we show that the robust loss being computably evaluable is in general also not necessary to ensure robust CPAC learning, which might be unexpected.

11.3.1 Robust CPAC learnability is not implied by Robust PAC + CPAC learnability

In this section, we first look at two examples of hypothesis classes and perturbation types where CPAC learnability and robust learnability do not imply robust CPAC learnability. In both cases, the perturbation types are DR, in contrast to Example 11.6.1. Our first

result holds in the robust agnostic case for general, improper learners, and the second, in the robust realizable case for proper learners.

Theorem 11.3.1. *There exists a hypothesis class $\mathcal{H}$ and perturbation type u such that (i) $\mathcal{H}$ is properly CPAC learnable, (ii) u is decidable representable and (iii) $\mathcal{H}$ is properly u -robustly PAC learnable, but $\mathcal{H}$ is not (improperly) u -robustly CPAC learnable in the agnostic setting.*

Sketch. Fix a proof system for first-order logic over a rich enough vocabulary that is sound and complete. Let $\{\varphi_i\}_{i \in \mathbb{N}}$ and $\{\pi_j\}_{j \in \mathbb{N}}$ be enumerations of all theorems and proofs, respectively. We define the following hypothesis class on $\mathbb{N}$:

$$\mathcal{H} = \{h_a : \forall i \in \mathbb{N}, h_a(2i) = \mathbf{1}[i \leq a] \wedge h_a(2i+1) = 1\}_{a \in \mathbb{N} \cup \{\infty\}} .$$

Thus, $\mathcal{H}$ is the the concept class where each function defines a threshold on even integers, and is the constant function 1 on odd integers.

The perturbation sets are defined as follows. For any $i \in \mathbb{N}$ we set:

$$\begin{aligned} u(6i) &= \{6i\} \cup \{2j+1\}_{j \in \mathbb{N}} , \\ u(6i+2) &= \{6i+2\} , \\ u(6i+4) &= \{6i+2, 6i+4\} \cup \{2j+1 \mid \pi_j \text{ is a proof of theorem } \varphi_i\}_{j \in \mathbb{N}} , \\ u(2i+1) &= \{2i+1\} . \end{aligned}$$

The properties (i)-(iii) are fulfilled by this construction, as we will show in detail in the appendix. Furthermore, we note that the question of whether $u(6i) \cap u(6i+4)$ is empty is equivalent to whether there exists a proof for theorem φ_i , and thus is undecidable. We now show that this undecidable problem can be reduced to agnostically robustly CPAC learning $\mathcal{H}$ with respect to u . To see this define the distributions D_i on $\mathbb{N} \times \{0, 1\}$ as follows:

$$D_i((6i, 1)) = 1/2, \quad D_i((6i+2, 1)) = 1/6, \quad D_i((6i+4, 0)) = 1/3 .$$

For a learner to succeed on all the D_i , it needs to decide whether $u(6i) \cap u(6i+4) = \emptyset$ for all i . □

For more detail and the full proof we refer the reader to Appendix [11.7.1](#).

The result above holds in the improper agnostic setting. Next, we look at the robustly realizable setting, and show an impossibility result in case we require a *proper* robust learning algorithm.

Theorem 11.3.2. *There exists a hypothesis class $\mathcal{H}$ and perturbation type u such that (i) $\mathcal{H}$ is properly CPAC learnable, (ii) u is decidablely representable, and (iii) $\mathcal{H}$ is properly u -robustly PAC learnable, but not properly u -robustly CPAC learnable in the realizable setting.*

Proof. Let $(T_i)_{i \in \mathbb{N}}$ be an enumeration of Turing Machines. We define the hypothesis class on $\mathbb{N}$:

$$\mathcal{H} = \{h_{a,b,c} : \forall i (h_{a,b,c}(2i) = 1 \text{ iff } i \in \{a, b\}) \wedge (h_{a,b,c}(2i+1) = c)\}_{a,b \in \mathbb{N}, c \in \{0,1\}} ,$$

i.e., $\mathcal{H}$ is the class of functions that only maps at most two even numbers to 1 and are constant on the odd numbers. First note that $\text{VC}(\mathcal{H}) = 3$, as for any shattered set $X \subseteq \mathbb{N}$, we can have at most one odd integer in X , and at most two even ones. It is straightforward to show that any set of the form $\{2i, 2j, 2k+1\}$ for $i, j, k \in \mathbb{N}$ can be shattered. This implies, by Theorem 6 in [Montasser et al., 2019], that $\mathcal{H}$ is (improperly) robustly learnable for any perturbation type u . We will argue below that, for the perturbation type we employ, $\mathcal{H}$ is in fact properly u -robustly CPAC learnable. Second, as ERM is easily implementable for $\mathcal{H}$, we have that $\mathcal{H}$ is properly CPAC learnable.

We define the perturbation sets as follows:

$$\begin{aligned} u(2i) &= \{2i\} \cup \{2j+1 : T_i(i) \text{ halts within } j \text{ steps}\} , \\ u(2i+1) &= \{2i+1\} . \end{aligned}$$

Note that u is decidablely representable, as for any two points x_1, x_2 , the program outlined in Appendix 11.7.2 decides whether $x_2 \in u(x_1)$.

We now show that for this u , $\mathcal{H}$ is *properly* u -robustly learnable. To see this, we again use the terminology of Ashtiani et al. [2020b], and consider the margin class $\mathcal{H}_{\text{mar}}^u$, where each $h_{a,b,c} \in \mathcal{H}$ fixes a set $\text{mar}_{h_{a,b,c}}^u$, which we will denote by $\text{mar}_{a,b,c}$ for brevity. Note that, from the property $k \subseteq u(k)$ that holds for all $k \in \mathbb{N}$, only even integers can belong in some set $\text{mar}_{a,b,c}$. Recall that $h_{a,b,c}$ maps at most two even numbers a and b to 1 and all odd numbers to $c \in \{0, 1\}$.

We distinguish two cases:

- $c = 0$: then $\text{mar}_{a,b,0} = \{2a \mid T_a(a) \text{ halts}\} \cup \{2b \mid T_b(b) \text{ halts}\}$,
- $c = 1$: then $\text{mar}_{a,b,1} = \{2i \mid T_i(i) \text{ halts}\} \setminus \{2a, 2b\}$.

From this, we can see that shattered sets must only contain even integers k such that $T_{k/2}(k/2)$ halts. We now argue that $\text{VC}(\mathcal{H}_{\text{mar}}^u) = 5$. Consider a set $X = \{k_1, \dots, k_6\}$ of size 6 such that k_i is even and $T_{k_i/2}(k_i/2)$ halts for all $i = 1, \dots, 6$. Then the subset $\{k_1, k_2, k_3\}$ cannot be part of the projection of $\mathcal{H}_{\text{mar}}^u$ onto X , and thus $\{k_1, \dots, k_6\}$ cannot be shattered. Indeed, note that for $c = 0$, the margin set $\text{mar}_{a,b,0}$ has size 2 only for any $a, b \in \mathbb{N}$. Thus $\{k_1, k_2, k_3\} \in \text{mar}_{a,b,c}$ would imply that there exists $a, b \in \mathbb{N}$ such that $k_1, k_2, k_3 \in \text{mar}_{a,b,1}$. However then, by definition, there are at most two of k_4, k_5, k_6 that are not in $\text{mar}_{a,b,1}$, a contradiction. Finally, consider a set $X = \{k_1, \dots, k_5\}$ of size 5 such that k_i is even and $T_{k_i/2}(k_i/2)$ halts for all $i = 1, \dots, 5$. Any subset $X' \subseteq X$ is contained in the projection of $\mathcal{H}_{\text{mar}}^u$ onto X : if $|X'| \geq 3$ there exists $a, b \in \mathbb{N}$ such that $X' \in \text{mar}_{a,b,1}$, and otherwise if $|X'| \leq 2$ there exists $a, b \in \mathbb{N}$ such that $X' \in \text{mar}_{a,b,0}$, and thus X is shattered.

Now, for any $i, k \in \mathbb{N}$, with $i \neq k$, we define a distribution $D_{i,k}$ on $\mathbb{N} \times \{0, 1\}$ as follows:

$$D_{i,k}(2i, 1) = D_{i,k}(2k, 0) = 1/2 \ .$$

We will now show that $\mathcal{H}$ is not properly u -robustly CPAC learnable with respect to the set of distributions $\{D_{i,k}\}_{i,k \in \mathbb{N}}$ in the realizable case. Note that robustly realizability here implies that at most one of the sets $u(2i)$ or $u(2k)$ is not a singleton. Indeed, if $u(2i) = \{2i\}$ then $T_i(i)$ does not halt, whereas if $\{2i\} \subset u(2i)$ (strict inequality) then $T_i(i)$ halts, and similarly for $2k$. Hence, since we have $D_{i,k}(2i, 1) = D_{i,k}(2k, 0) = 1/2$, to choose between $h_{i,i',0}$ and $h_{i,i',1}$ for some fixed $i' \in \mathbb{N}$ with $i' \neq k$ and $i' \neq i$ we need to know whether either of $u(2i)$ or $u(2k)$ is a singleton.

Now consider the following decision problem, which we term **TwoHalt**. Given two Turing machines T_i and T_j , the task is to correctly predict which run out of $T_i(i)$ and $T_j(j)$ halts if only one of them does. If both runs loop, the solver should still halt and produce an output; in case both halt the decider may run indefinitely or halt and produce some output. That is, we only require a correct decision if exactly one of $T_i(i)$ and $T_j(j)$ halts, and require the program to halt if both loop. We show below (see Lemma 11.7.1) that no algorithm can satisfy these requirements.

We now argue that a u -robust proper CPAC learner can be used to solve **TwoHalt**, and thus, by invoking Lemma 11.7.1, which is stated and proved in Appendix 11.7.3, conclude that no such learner exists. If a u -robust proper CPAC learner for $\mathcal{H}$ existed, there is a sample size M that suffices for robust loss at most $\varepsilon = 1/3$ with probability $1 - \delta = 2/3$ over all M size samples from $D_{i,k}$. We then solve **TwoHalt** by running the learner on all M -length sequences over the points $(2i, 1)$ and $(2k, 0)$, and taking the majority for the label that the resulting predictor gives on the odd numbers.

□

11.3.2 Robust CPAC learnability and its relationship to the computability of the robust loss and RERM

One of the obstacles with robust CPAC learning is that, even if a class consists of computable functions, the robust loss might not be pointwise computable. This is in contrast to the CPAC setting for which the binary loss is always computable for computable functions. It is then natural to ask what relationship exists between the computability of the robust loss and the robust CPAC learnability of a class. The following result demonstrates that the computability of the pointwise robust loss for all hypotheses in the class is not a necessary criterion for robust CPAC learnability.

Theorem 11.3.3. *There exist $\mathcal{H}$ and u such that (i) $\mathcal{H}$ is properly CPAC learnable, (ii) u is decidablely representable, and (iii) $\mathcal{H}$ is properly u -robustly CPAC learnable in the agnostic setting and admits a computable agnostic $\text{RERM}_{\mathcal{H}}^u$ oracle, but the function ℓ^u is not computably evaluable on $\mathcal{H}$.*

Sketch. This theorem is proven with the following construction. Let $\mathcal{X}$, $(\varphi_i)_{i \in \mathbb{N}}$ and $(\pi_i)_{i \in \mathbb{N}}$ be defined as in the construction for Theorem 11.3.1. Let $(i, 0)$, represent the i -th formula and for any i, j let (i, j) represent the j -th proof (independently of i). Let $\mathcal{H} = \{h_{a,b}\}_{a,b \in \mathbb{N} \cup \{0, \infty\}}$, where

$$h_{a,b}(i, j) = \begin{cases} 1 & \text{if } i < a \text{ or } (i = a \text{ and } j \leq b) \\ 0 & \text{otherwise} \end{cases}.$$

It is clear that $\text{VC}(\mathcal{H}) = 1$ and that ERM is computably implementable, so $\mathcal{H}$ is properly CPAC learnable. We define the perturbation regions as follows:

$$\begin{aligned} u(i, 0) &= \{(i, 0)\} \cup \{(i, j) : \pi_j \text{ proves } \varphi_i\}, \\ u(i, j) &= \{(i, 0), (i, j)\}. \end{aligned}$$

It is also easy to verify that this perturbation type is DR. In the full proof, we show that in general there is no algorithm that evaluates the loss of each hypotheses $h_{a,b}$ on every point. However, there is an RER subclass $\mathcal{H}' = \{h_{a,\infty} : a \in \mathbb{N}\} \subset \mathcal{H}$ such that every hypothesis in $\mathcal{H}'$ is computably evaluable and such that on any distribution, $\mathcal{H}'$ contains an optimal (with respect to $\mathcal{H}$) hypothesis. It is thus possible to robustly CPAC learn $\mathcal{H}$ by performing RERM over $\mathcal{H}'$. $\square$

The full proof is included in Appendix 11.8.1. We next show that, even if the robust loss is computably evaluable on $\mathcal{H}$, and the class $\mathcal{H}$ is properly CPAC learnable and properly

robustly PAC learnable, neither the existence of a robust ERM oracle nor robust CPAC learnability are implied.

Theorem 11.3.4. *There exist a DR hypothesis class $\mathcal{H}$ and a DR perturbation type u such that (i) $\mathcal{H}$ is properly CPAC learnable, (ii) $\mathcal{H}$ is properly u -robustly PAC learnable, and (iii) the robust loss is computably evaluable, but there is no strong realizable $\text{RERM}_{\mathcal{H}}^u$ -oracle and $\mathcal{H}$ is not properly agnostically u -robustly CPAC learnable.*

Sketch. We prove this theorem with the following construction. Let the instance space be $\mathcal{X} = \mathbb{N} \times \mathbb{N}$. Let the hypothesis class be $\mathcal{H} = \{h_{i,j} : i, j \in \mathbb{N}\}$, where

$$h_{i,j}(x) = \begin{cases} 1 & \text{if } x = (i, k) \text{ with } k \leq j \\ 0 & \text{otherwise} \end{cases},$$

i.e., $\mathcal{H}$ is the class of initial segments on $\{(i, j)\}_{j \in \mathbb{N}}$ for all $i \in \mathbb{N}$. We set the perturbation types to:

$$\begin{aligned} u((i, 0)) &= \{(i, 0)\} \cup \{(i, k) : T_i \text{ does not halt after } k \text{ steps on the empty input}\} , \\ u((i, j)) &= \begin{cases} \{(i, j), (i, 0)\} & T_i \text{ does not halt after } j \text{ steps on the empty input} \\ \{(i, j)\} & \text{otherwise.} \end{cases} \end{aligned}$$

The proof of the impossibility of agnostic robust CPAC learning and of the non-existence of a strong realizable RERM oracle relies on reductions from the Halting problem. $\square$

The detailed proof shows that the construction in the sketch above fulfills all the requirements for proving the theorem and is included in Appendix 11.8.2. Recall from Fact 11.2.2 that the class being RER (a weaker assumption than being DR) and the robust loss being computably evaluable imply the existence of a weak realizable $\text{RERM}_{\mathcal{H}}^u$ -oracle. We therefore demonstrated a separation between the existence of RERM oracles in the robust realizable versus the agnostic case. Our proof also shows that the existence of weak realizable RERM oracles does not imply the existence of agnostic RERM oracles. Furthermore, note that the existence of a strong realizable RERM oracle implies computable evaluability of robust loss. Thus, the result of Theorem 11.3.3 shows that the existence of an agnostic RERM oracle does not imply the existence of a strong realizable RERM oracle.

In the non-robust setting, proper *strong* CPAC (SCPAC) learnability (which is CPAC learnability with the additional requirement that the sample complexity function $m(\cdot, \cdot, \cdot)$ be computable) implies the existence of an $\text{ERM}_{\mathcal{H}}$ oracle (Sterkenburg [2022], Theorem 8).

We finish this section by showing that a similar result holds in robust learning if the robust loss is computably evaluable. The proof of Proposition 11.3.5 below is included in Appendix 11.8.3.

Proposition 11.3.5. *Let $\mathcal{H}$ be a hypothesis class and u a perturbation type such that ℓ^u is computably evaluable. Then, if $\mathcal{H}$ is properly u -robustly SCPAC learnable in the agnostic setting, $\mathcal{H}$ admits a computable agnostic $\text{RERM}_{\mathcal{H}}^u$ oracle; and if $\mathcal{H}$ is properly u -robustly SCPAC learnable in the realizable setting, then $\mathcal{H}$ admits a computable weak realizable $\text{RERM}_{\mathcal{H}}^u$ oracle.*

11.4 Necessary Conditions for Robust CPAC Learning

In this section, we define a complexity measure, denoted by $c\text{-dim}_u$, and prove its finiteness is implied by robust CPAC learnability for RER perturbation types. We later study its relationship with the VC dimension, and show that it cannot characterize robust CPAC learnability.

11.4.1 No-Free-Lunch Theorems

Prior work has shown that standard CPAC learnability is equivalent to the hypothesis class having finite *effective VC dimension*, which can be thought of as a computable version of the VC dimension [Sterkenburg, 2022, Delle Rose et al., 2023]. The effective VC dimension of $\mathcal{H}$ is the smallest k , such that there exists a computable function $w : \mathbb{N}^{k+1} \rightarrow \{0, 1\}^{k+1}$ that, when given a set of domain points of size $k + 1$, outputs a labelling that cannot be achieved by any $h \in \mathcal{H}$. The function w is called a *witness function*. The complexity measure $c\text{-dim}_u(\mathcal{H})$ that we introduce in this section is a computable version of the robust shattering dimension $\text{dim}_u(\mathcal{H})$ which has been shown to lowerbound the robust sample complexity [Montasser et al., 2019].

Definition 11.4.1 (u -robust shattering dimension [Montasser et al., 2019]). *A set $X = \{x_i\}_{i=1}^m$ is u -robustly shattered by $\mathcal{H}$ if there exists a set $Z = \{z_i^0, z_i^1\}_{i=1}^m$ with $x_i \in u(z_i^0) \cap u(z_i^1)$ for all $i = 1, \dots, m$ such that for all $y_1, \dots, y_m \in \{0, 1\}$ there exists $h \in \mathcal{H}$ such that for all $i = 1, \dots, m$ and $z' \in u(z_i^{y_i})$ we have that $h(z') = y_i$. The robust shattering dimension of $\mathcal{H}$, denoted $\text{dim}_u(\mathcal{H})$, is the largest integer k such that there exists a set of size k that is u -robustly shattered. If arbitrarily large u -robustly shattered sets exist, then $\text{dim}_u(\mathcal{H}) = \infty$.*

Note that, for X to be robustly shattered, for all $i \neq j$ the regions $u(z_i^1)$ and $u(z_j^0)$ must be disjoint. Now, contrary to the standard notion of shattering, there are two factors at play in a set being robustly shattered. The first one is whether a set is robustly “shatterable”, namely whether, by definition of u —and without considering $\mathcal{H}$ —the set X *could* be robustly shattered by an arbitrary set of functions (which holds trivially in the standard case). The second factor, if robust shatterability holds, is as in the standard case: whether there does in fact exist a set of hypotheses from $\mathcal{H}$ that can realize the robust shattering. To highlight this distinction, we introduce the following property:

Definition 11.4.2 (Robust shatterability). *A set $X = \{x_1, \dots, x_m\} \subseteq \mathcal{X}$ is called u -robustly shatterable if there exists a set $Z = \{z_i^0, z_i^1\}_{i=1}^m \subseteq \mathcal{X}$ with $x_i \in u(z_i^0) \cap u(z_i^1)$ and for all $i \neq j$ the regions $u(z_i^1)$ and $u(z_j^0)$ are disjoint. In this case, Z is said to admit a u -robustly shatterable set.*

We now define the computable version of Definition 11.4.1, which also uses the notion of a witness function. If $\dim_u(\mathcal{H}) < k$ then for all $m \geq k$, for all $X = \{x_i\}_{i=1}^m$ and for all $Z = \{z_i^0, z_i^1\}_{i=1}^m$ with $x_i \in u(z_i^0) \cap u(z_i^1)$ there exists a labelling that is *not* u -robustly realizable by $\mathcal{H}$. Thus:

Definition 11.4.3 (Computable robust shattering dimension). *A k -witness of u -robust shattering dimension for $\mathcal{H}$ is a function $w : \mathbb{N}^{2(k+1)} \rightarrow \{0, 1\}^{k+1}$ such that when given a set $Z = \{(z_i^0, z_i^1)\}_{i=1}^{k+1}$ that admits a robustly shatterable set, it outputs $y_1, \dots, y_{k+1} \in \{0, 1\}$ such that for all $h \in \mathcal{H}$ there exist $i = 1, \dots, m$ and $z' \in u(z_i^{y_i})$ with $h(z') \neq y_i$. The computable robust shattering dimension of $\mathcal{H}$, denoted $c\text{-dim}_u(\mathcal{H})$, is the smallest integer k such that there exists a computable k -witness of robust shattering dimension. If no such k exists, $c\text{-dim}_u(\mathcal{H}) = \infty$.*

We note that, by definition, if u does not admit a shatterable set of size k , then any function is a k -witness. Towards showing that finiteness of the computable robust shattering dimension is a necessary condition for u -robust CPAC learnability, we first formulate a No-Free-Lunch theorem for robust learnability, leaving aside computability considerations for the time being.

Lemma 11.4.4 (Robust No-Free-Lunch Theorem). *Let $\mathcal{X}$ be the domain and $u : \mathcal{X} \rightarrow 2^{\mathcal{X}}$ a perturbation type. Let $\mathcal{A}$ be any computable learner. Let m be a number smaller than $|\mathcal{X}|/4$ representing the training set size, with the property that there exist $4m$ points $Z = \{z_i^0, z_i^1\}_{i=1}^{2m}$ such that Z admits a u -robustly shatterable set. Then there exists a distribution D over $\mathcal{X} \times \{0, 1\}$ such that (i) there exists a function $f : \mathcal{X} \rightarrow \{0, 1\}$ such that $\mathcal{L}_D^u(f) = 0$; and (ii) with probability at least $1/7$ over $S \sim D^m$, $\mathcal{L}_D^u(\mathcal{A}(S)) \geq 1/8$.*

The proof of Lemma 11.4.4 is included in Appendix 11.9.1. We comment on the robust shatterability in the impossibility results from Section 11.3 in Appendix 11.10.

Remark 5. First note that we work with $\mathcal{X} = \mathbb{N}$, so the condition $m \leq |\mathcal{X}|/4$ is immediately satisfied for any m . Second, note that the robust shatterability condition in the theorem, which depends on the perturbation type u , is satisfied if the regions $u(x)$ are, e.g., balls of finite radii.

We are now ready to state the computable version of Lemma 11.4.4, which involves not only the existence of a distribution and function that are robustly realizable on which the learner performs poorly, but also the ability to computably find such a pair:

Lemma 11.4.5 (Computable Robust No-Free-Lunch Theorem). *Fix instance space $\mathcal{X}$ and perturbation type $u : \mathcal{X} \rightarrow 2^{\mathcal{X}}$. Let $\mathcal{A}$ be any computable learner and furthermore suppose that the perturbation type u is recursively enumerably representable (RER). Then for any $m \in \mathbb{N}$, any domain $\mathcal{X}$ of size at least $4m$ and any subset $Z = \{z_i^0, z_i^1\}_{i=1}^{2m}$ that admits a u -robustly shatterable set, we can computably find a function $f : \mathcal{X} \rightarrow \{0, 1\}$ that is computable on Z such that*

$$\mathbb{E}_{S \sim D^m} [\mathcal{L}_D^u(\mathcal{A}(S))] \geq 1/4 \quad \text{and} \quad \mathbb{P}_{S \sim D^m} (\mathcal{L}_D^u(\mathcal{A}(S)) \geq 1/8) \geq 1/7, \quad (11.1)$$

where D is the uniform distribution on $\{(z_i^{y_i}, y_i)\}_{i=1}^{2m}$ for some $y \in \{0, 1\}^{2m}$ such that for all $i = 1, \dots, 2m$ and for all $z' \in u(z_i^{y_i})$ we have that $f(z') = y_i$.

The proof of Lemma 11.4.5 follows a similar argument as the proof of the computable No-Free-Lunch Theorem shown in Section 10.5, but the computation of the robust loss differs:

Proof. We first computably find a shatterable set $X = \{x_1, x_2, \dots, x_{2m}\}$ corresponding to the set of pairs $Z = \{z_i^0, z_i^1\}_{i=1}^{2m}$. That is $x_i \in u(z_i^0) \cap u(z_i^1)$ for all $i \in [2m]$. Since the sets $u(z_i^0)$ and $u(z_i^1)$ are recursively enumerable, we can computably find such an x_i in their intersection for each i . We simply enumerate the instances from $u(z_i^0)$ and $u(z_i^1)$ by alternating between the two sets until we find an instance that belongs to both enumerations. Because we are guaranteed that Z admits a u -robustly shatterable set, this process is guaranteed to halt.

The existence of a function f and distribution D satisfying Equation 11.1 follows directly from Lemma 11.4.4, so it remains to show that, using $\mathcal{A}$ we can computably find such an f and D . Note that the construction of the pairs (f_j, D_j) in Lemma 11.4.4 corresponds to pairs of u -robustly realizable distributions. There are $T = 2^{2m}$ such pairs, as

putting mass on $z_i^{y_i}$ implies not putting mass on $z_i^{-y_i}$. Second, for a given (f_j, D_j) and a fixed sample size m , there are $k = (2m)^m$ sequences $\{S_l^j\}_{l=1}^k$ of m instances drawn from D_j , each equally likely as D_j is uniform on its support. We now run $\mathcal{A}$ on all these sequences, and compute a lower bound on the expected robust loss of $\mathcal{A}$ on (f_j, D_j) by employing the labels of the learner's output on the shatterable set X :

$$\begin{aligned} \mathbb{E}_{S \sim D_j^m} \left[\mathcal{L}_{D_j}^u(\mathcal{A}(S)) \right] &= \frac{1}{k} \sum_{l=1}^k \mathcal{L}_{D_j}^u(\mathcal{A}(S_l^j)) = \frac{1}{k} \sum_{l=1}^k \frac{1}{2m} \sum_{i=1}^{2m} \ell^u \left(\mathcal{A}(S_l^j), z_i^{(j)}, y_i^{(j)} \right) \\ &\geq \frac{1}{k} \sum_{l=1}^k \frac{1}{2m} \sum_{i=1}^{2m} \mathbf{1} \left[\mathcal{A}(S_l^j)(x_i) \neq y_i^{(j)} \right] . \end{aligned}$$

As argued in the proof of Lemma 11.4.4, there exist f_j, D_j such that

$$\frac{1}{k} \sum_{l=1}^k \frac{1}{2m} \sum_{i=1}^{2m} \mathbf{1} \left[\mathcal{A}(S_l^j)(x_i) \neq y_i^{(j)} \right] \geq 1/8 .$$

Furthermore, we note that for every j , the set of sequences $\{S_l^j\}_{l=1}^k$ can be computably generated and that the expression $\frac{1}{k} \sum_{l=1}^k \frac{1}{2m} \sum_{i=1}^{2m} \mathbf{1} \left[\mathcal{A}(S_l^j)(x_i) \neq y_i^{(j)} \right]$ can be computably evaluated. Thus by iterating through $j = 1, \dots, T$ until finding j that satisfies $\frac{1}{k} \sum_{l=1}^k \frac{1}{2m} \sum_{i=1}^{2m} \mathbf{1} \left[\mathcal{A}(S_l^j)(x_i) \neq y_i^{(j)} \right] \geq 1/8$, we can computably find (f_j, D_j) with $\mathbb{E}_{S \sim D_j^m} \left[\mathcal{L}_{D_j}^u(\mathcal{A}(S)) \right] \geq 1/8$. We further note that the corresponding labelling f_j can be computably evaluated on all elements of Z as required. $\square$

Note that if the perturbation sets are DR, then the function f above is total computable on $\mathcal{X}$. We now state the main result of this section:

Theorem 11.4.6. *Let $\mathcal{H}$ be (improperly) u -robustly CPAC learnable and suppose the perturbation type u is RER. Then the computable u -robust shattering dimension $c\text{-dim}_u(\mathcal{H})$ of $\mathcal{H}$ is finite, i.e., $\mathcal{H}$ admits a computable k -witness of u -robust shattering dimension for some $k \in \mathbb{N}$.*

The proof, included in Appendix 11.9.2, follows the reasoning of Lemma 9 in Sterkenburg [2022].

11.4.2 Relationship between the effective VC dimension and effective shattering dimension

It has been shown that the dimension $\dim_u(\mathcal{H})$ yields a lower bound on the sample complexity for u -robust learning [Montasser et al., 2019]. However, it has remained an open question whether this dimension also provides an upper bound on the sample complexity, thus whether it actually characterizes u -robust learnability. Here we show that a corresponding conjecture in the CPAC setting is not true. We first relate the effective VC dimension, which we will identify as $c\text{-VC}$ to denote computability, to the computable robust shattering dimension.

Lemma 11.4.7. *For any class $\mathcal{H}$ and perturbation type u admitting a recursively enumerable representation, we have $c\text{-dim}_u(\mathcal{H}) \leq c\text{-VC}(\mathcal{H})$.*

Proof. Let $\mathcal{H}$ have finite effective VC dimension k . Then there exist a computable function $w : \mathbb{N}^{k+1} \rightarrow \{0, 1\}^{k+1}$ such that w is a k -witness of VC dimension for $\mathcal{H}$. Now, let $Z = \{z_i^0, z_i^1\}_{i=1}^{k+1}$ admit a robustly shatterable set. Because u is RER, we can find a robustly shatterable set X of size $k+1$ for Z . It suffices to output the labelling $w(X)$, which is not achievable by any $h \in \mathcal{H}$, and thus not robustly achievable by any $h \in \mathcal{H}$ as well, and so $c\text{-dim}_u(\mathcal{H}) \leq k$, as required. $\square$

We know that CPAC learnability of a class $\mathcal{H}$ implies $c\text{-VC}(\mathcal{H}) < \infty$ [Sterkenburg, 2022], and thus, by the above lemma, CPAC learnability implies $c\text{-dim}_u(\mathcal{H}) \leq c\text{-VC}(\mathcal{H}) < \infty$. On the other hand, we have shown in this work that there exist $\mathcal{H}$, u such that CPAC learnability does not imply u -robust CPAC learnability (see Theorem 11.3.1). Thus there are classes that are not u -robust CPAC learnable while having finite effective robust shattering dimension, meaning that this dimension does not characterize robust CPAC learnability:

Corollary 11.4.8. *The dimension $c\text{-dim}_u(\mathcal{H})$ does not characterize u -robust CPAC learnability.*

11.5 Conclusions for Robust CPAC Learning

In this chapter, we extended our analysis of computable PAC learning to adversarially robust CPAC learning, a notion that combines the requirement of adversarial robustness with the computational requirement of CPAC learning. We showed that the landscape of

robust CPAC learning can not be reduced to it's parts: that is we showed that there are classes which are properly agnostically CPAC learnable and properly agnostically robustly learnable, but not even improperly agnostically learnable. We further show that there are such proper agnostically CPAC learnable, proper agnostically robustly learnable classes, which are not properly realizably robustly CPAC learnable. Whether there are such classes which fulfill these requirements and are also not improperly robustly CPAC learnable in the realizable case is still open. One crucial difference between the robust CPAC and the CPAC setting is, that the robust loss can not necessarily be evaluated for total computable function. However, we show that the difference between CPAC and robust CPAC learning cannot simply explained by the computability of the loss as we show that for CPAC learnable classes the computability of the loss is neither a sufficient nor a necessary criterion for a class to be (agnostically properly) robustly CPAC learnable. Lastly we showed a lower bound for improper realizable robust CPAC learning via the *effective u -dimension*, a computational version of the $\mathcal{U}$ -dimension introduced in [Montasser et al. \[2019\]](#).

11.6 Remaining Proofs and Remarks from Section 11.2

11.6.1 Proof of Theorem 11.2.3

Proof. We will use the computable online algorithm $\mathcal{A}$ as a black-box. Let $D := \text{VC}(\mathcal{H}) + \text{VC}(\mathcal{H}_{\text{mar}}^u)$ and note that online learnability implies that $\text{VC}(\mathcal{H}) \leq \text{Lit}(\mathcal{H}) < \infty$, and so $D < \infty$.

Let $S = \{(x_i, y_i)\}_{i=1}^m$ be drawn i.i.d. from an arbitrary robustly-realizable distribution D on $\mathcal{X} \times \{0, 1\}$. We outline below a robust CPAC algorithm, which is essentially the CycleRobust algorithm of [\[Montasser et al., 2021\]](#) with the online algorithm $\mathcal{A}$ being used, implements robust ERM on S :

Algorithm 5: u -robust CPAC algorithm

Input: $S = \{(x_i, y_i)\}_{i=1}^m$, computable online algorithm $\mathcal{A}$ for $\mathcal{H}$
Output: $h \in \mathcal{H}$ such that h achieves zero robust loss on S
 $K \leftarrow 0, h \leftarrow \mathcal{A}(\emptyset);$
while $K \leq \text{Lit}(\mathcal{H})$ **do**
 for $i = 1, \dots, m$ **do**
 if $(z, y) \leftarrow \text{PAO}(h, x_i, y_i)$ **then** $\triangleright$ robust loss is 1, get counterexample
 $K++$;
 $h \leftarrow \mathcal{A}((z, y))$ $\triangleright$ Update mistake count
 Break $\triangleright$ Update hypothesis
 $\triangleright$ Exit for-loop
 else if $i = m$ **then**
 return h $\triangleright h$ is robustly consistent on S
 $\triangleright$ No counterexample $\Rightarrow$ go to next index
 end
end
return h

Note that, because of robust realizability, we are guaranteed that $\mathcal{A}$ makes at most $\text{Lit}(\mathcal{H})$ mistakes, after which h will be u -robustly consistent on S . The program above makes a finite number of calls to $\mathcal{A}$, and thus is computable by the fact that $\mathcal{A}$ is itself a computable online learner. By Fact 11.2.1, we are done. $\square$

11.6.2 Decidable Representation of Perturbation Types

Example 11.6.1. We exhibit $\mathcal{H}, u$ such that $\mathcal{H}$ is CPAC learnable and u -robustly PAC learnable but $\mathcal{H}$ is not (improperly) u -robustly CPAC learnable. In this example, u is not DR. The hypothesis class is as follows:

$$\mathcal{H} := \{h_{a,b}(i) = \mathbf{1}[i \in \{a, b\}]\} \quad .$$

Clearly $\text{VC}(\mathcal{H}) = 2$ and ERM is computably implementable, so by Fact 11.1.5, $\mathcal{H}$ is CPAC learnable. Moreover, $\mathcal{H}$ having finite VC dimension implies that it is u -robustly PAC learnable [Montasser et al., 2019].

We now define the perturbation type:

$$\begin{aligned} u(2i) &= \{2i\} \cup \{2i+1 \mid T_i \text{ halts on the empty word}\} \quad , \\ u(2i+1) &= \{2i+1\} \cup \{2i \mid T_i \text{ halts on the empty word}\} \quad . \end{aligned}$$

We now define distributions D_i :

$$D_i(2i, 1) = D_i(2i + 1, 0) = 1/2 \ .$$

Clearly, the labelling $l(2i) = 1$ and $l(2i + 1) = 0$ is optimal (and gives zero robust risk) if and only if T_i does not halt on the empty word. Otherwise, if T_i halts, l incurs a robust risk of 1, and any function satisfying $h(2i) = h(2i + 1)$ is a robust risk minimizer incurring a robust risk of $1/2$.

If $\mathcal{H}$ were (improperly) u -robustly CPAC learnable, there would be a robust learning algorithm $\mathcal{A}$ for $\mathcal{H}$ with sample complexity $m(\varepsilon, \delta)$. Letting $M \in \mathbb{N}$ being an upper bound on $m(1/3, 1/2)$, then with probability at least $2/3$, $\mathcal{A}$ returns a hypothesis with robust risk within $1/3 < 1/2$ of the robust risk minimizer in $\mathcal{H}$ when run on a sample of size M . Then, running $\mathcal{A}$ on all 2^M sequences of length M with elements in $\{2i, 2i + 1\}$ (which are equally probable), we end up with optimal hypotheses on two thirds of the runs. Checking, for each hypothesis h , whether $h(2i) = h(2i + 1)$ and taking the majority vote over all sequences, we get a decider for whether T_i halts on the empty word. And so $\mathcal{H}$ is not (improperly) u -robustly CPAC learnable. Note here that if we had a way to know whether $k \in u(2i)$, i.e., if u was DR, we would immediately be able to implement robust ERM for samples from the distributions defined above. Not requiring perturbation regions to be DR thus makes deriving impossibility results for robust CPAC learning relatively trivial.

11.7 Proofs and Algorithms from Section 11.3.1

11.7.1 Proof of Theorem 11.3.1

Proof. Fix a proof system for first-order logic over a rich enough vocabulary that is sound and complete. Let $\{\varphi_i\}_{i \in \mathbb{N}}$ and $\{\pi_j\}_{j \in \mathbb{N}}$ be enumerations of all theorems and proofs, respectively. We define the following hypothesis class on $\mathbb{N}$:

$$\mathcal{H} = \{h_a : \forall i \in \mathbb{N}, h_a(2i) = \mathbf{1}[i \leq a] \wedge h_a(2i + 1) = 1\}_{a \in \mathbb{N} \cup \{\infty\}} \ .$$

Thus, $\mathcal{H}$ is the the concept class where each function defines a threshold on even integers, and is the constant function 1 on odd integers.

First, to show (i), note that $\text{VC}(\mathcal{H}) = 1$, as for any set X of two even integers, the labelling $h(x) = 0$, $h(z) = 1$ cannot be realized on two even integers $x < z$, and odd

integers can only have labelling 1, and thus cannot be in a shattered set. As ERM is easily implementable for $\mathcal{H}$, we have that $\mathcal{H}$ is properly CPAC learnable by Fact 11.1.5.

Now, we define the perturbation sets as follows. For any $i \in \mathbb{N}$ we set:

$$\begin{aligned} u(6i) &= \{6i\} \cup \{2j + 1\}_{j \in \mathbb{N}} , \\ u(6i + 2) &= \{6i + 2\} , \\ u(6i + 4) &= \{6i + 2, 6i + 4\} \cup \{2j + 1 \mid \pi_j \text{ is a proof of theorem } \varphi_i\}_{j \in \mathbb{N}} , \\ u(2i + 1) &= \{2i + 1\} . \end{aligned}$$

Note that the question of whether $u(6i) \cap u(6i + 4)$ is empty is equivalent to whether there exists a proof for theorem φ_i , and thus is undecidable.

It is straightforward to show (ii), as for every integer $i \in \mathbb{N}$, we can represent $u(6i + 4)$ with the program P_{6i+4} , which we have included below.

Algorithm 6: Program P_{6i+4} to decide whether $x \in u(6i + 4)$

```

Input:  $x, i \in \mathbb{N}$ 
Output: Whether  $x \in u(6i + 4)$ 
if  $x \in \{6i + 2, 6i + 4\}$  then
  | output yes
else if  $x$  is even then
  | output no
else
  |  $j \leftarrow \frac{x-1}{2}$ ;
  | if  $\pi_j$  proves  $\varphi_i$  then
  | | output yes
  | else
  | | output no
  | end
end

```

Now, for all other k , deciding whether $x \in u(k)$ is also easily representable as a program P_k (we can just assume $u(x) = \{x\}$ if there does not exist $i \in \mathbb{N}$ such that $k \in \{6i, 6i + 2, 6i + 4\}$, in order for u to be well-defined).

To show (iii), we follow the notation of [Ashtiani et al., 2020b] and show that $\text{VC}(\mathcal{H}_{\text{mar}}^u) = 1$, which implies proper robust learnability by Fact 11.2.1 (also Fact 11.1.6). Recall that the class $\mathcal{H}_{\text{mar}}^u$ consists of the sets $\text{mar}_h^u := \{x \in \mathcal{X} \mid \exists z \in u(x) . h(x) \neq h(z)\}$, defined for

each $h \in \mathcal{H}$. Now, fix $a \in \mathbb{N}$, which induces the function h_a and the set $\text{mar}_{h_a}^u$, which we will denote by mar_a for simplicity. Because of the condition $h(x) \neq h(z)$ in the definition of mar_a and the property that $\{k\} \subseteq u(k)$, the only integers $k \in \mathbb{N}$ that can belong to mar_a are those for which $\{k\}$ is a proper subset of $u(k)$, i.e., those for which there exists $i \in \mathbb{N}$ such that $k = 6i$ or $k = 6i + 4$. We can see that

$$\begin{aligned} \text{mar}_a = & \{6i \mid 6i > 2a\}_{i \in \mathbb{N}} \cup \{6i + 4 \mid \varphi_i \text{ is a tautology} \wedge 6i + 4 > 2a\}_{i \in \mathbb{N}} \\ & \cup \{6i + 4 \mid \varphi_i \text{ not a tautology} \wedge 6i + 2 = 2a\}_{i \in \mathbb{N}} , \end{aligned}$$

as these are precisely the instances which incur a robust loss of 1 with respect to h_a :

- If $k = 6i$ then since $u(6i) = \{6i\} \cup \{2j + 1\}_{j \in \mathbb{N}}$, we have that $k \in \text{mar}_a$ iff $h_a(k) = 0$,
- If $k = 6i + 4$ and φ_i is a tautology, then since

$$u(6i + 4) = \{6i + 2, 6i + 4\} \cup \{2j + 1 \mid \pi_j \text{ is a proof of theorem } \varphi_i\}_{j \in \mathbb{N}} ,$$

we have that $k \in \text{mar}_a$ iff $h_a(k) = 0$ (we cannot have $h_a(6i + 4) = 1$ while $h_a(6i + 2) = 0$),

- Else, if $k = 6i + 4$ and φ_i is not a tautology, $u(6i + 4) = \{6i + 2, 6i + 4\}$, and $k \in \text{mar}_a$ iff $h(6i + 2) \neq h(6i + 4)$ iff $6i + 2 = 2a$.

Now, it suffices to consider the subsets of $X = \{6i, 6i + 4\}_{i \in \mathbb{N}}$ to find the largest shattered set, as these are the only instances that can belong to some set mar_a . We partition X into two parts P_1 and P_2 , where $P_1 = \{6i\}_{i \in \mathbb{N}} \cup \{6i + 4 \mid \varphi_i \text{ is a tautology}\}_{i \in \mathbb{N}}$ and $P_2 = \{6i + 4 \mid \varphi_i \text{ is not a tautology}\}_{i \in \mathbb{N}}$. On the one hand, $\{\text{mar}_a\}_{a \in \mathbb{N}}$ restricted to P_1 is simply thresholds on P_1 , and so at most one element from P_1 can be in a shattered set. On the other hand, $\{\text{mar}_a\}_{a \in \mathbb{N}}$ restricted to P_2 is the class of singletons, so again at most one element from P_2 can be in a shattered set. Thus, we have established that $\text{VC}(\mathcal{H}_{\text{mar}}^u) \leq 2$. To see that $\text{VC}(\mathcal{H}_{\text{mar}}^u) = 1$ consider arbitrary $k_1 \in P_1$ and $k_2 = 6i + 4 \in P_2$ for some $i, i' \in \mathbb{N}$. It is easy to see that $\{k_1, k_2\}$ or $\{k_2\}$ won't be in the projection of $\mathcal{H}_{\text{mar}}^u$ onto $\{k_1, k_2\}$. Indeed, k_2 being in a set in the projection implies that there is a unique mar_a such that $k_2 \in \text{mar}_a$, and k_1 is either in mar_a or not.

Finally, to show that $\mathcal{H}$ not (improperly) u -robustly CPAC learnable in the agnostic setting we define a set of distributions for which no computable learner for $\mathcal{H}$ succeeds. For each $i \in \mathbb{N}$ we define a distribution D_i on $\mathbb{N} \times \{0, 1\}$ as follows:

$$\begin{aligned} D_i((6i, 1)) &= 1/2 , \\ D_i((6i + 2, 1)) &= 1/6 , \\ D_i((6i + 4, 0)) &= 1/3 . \end{aligned}$$

To determine optimal predictors on these distributions, we distinguish two cases:

1. $u(6i) \cap u(6i + 4) = \emptyset$: this implies that $u(6i + 4) = \{6i + 2, 6i + 4\}$. Since $u(6i + 2) \cap u(6i + 4) \neq \emptyset$ and these points have different labels under distribution D_i , no predictor can achieve robust loss 0 on both points simultaneously. Thus no predictor can achieve robust risk less than $1/6$. Note that $h_{3i,\infty}$ has robust risk $1/6$ on D_i , and is therefore an optimal predictor on D_i . Furthermore, note that any optimal predictor h on D_i must satisfy $h(6i + 2) = h(6i + 4) = 0$.
2. $u(6i) \cap u(6i + 4) \neq \emptyset$: in this case, since $6i$ and $6i + 4$ have different labels under distribution D_i and intersecting perturbation sets, no predictor can achieve robust loss 0 on both points and thus no predictor can achieve robust loss less than $1/3$ on D_i . Note that $h_{\infty,\infty}$ (the constant function 1) has robust risk $1/3$ on D_i and is thus a robust risk minimizer in this case. Furthermore, any optimal predictor h on D_i must satisfy $h(6i) = h(6i + 2) = 1$, to not incur additional robust loss.

We thus established that we have $h(6i + 2) = 1$ for an optimal predictor on D_i (and any predictor that has robust loss less than $1/6$ more than the optimal) if and only if $u(6i) \cap u(6i + 4) \neq \emptyset$ and thus if and only if φ_i has a proof.

We now argue that for any $i \in \mathbb{N}$, a u -robust CPAC learner $\mathcal{A}$ for $\mathcal{H}$, proper or not, can be used to determine whether $u(6i) \cap u(6i + 4) = \emptyset$, and thus can be used to decide if φ_i is a tautology or not: we let $\mathcal{A}$ be a robust computable learner for $\mathcal{H}$ and let $M \in \mathbb{N}$ be an upper bound for learning $\mathcal{H}$ with $\varepsilon = 1/7 < 1/6$ and $\delta = 1/3$. Running $\mathcal{A}$ on all sequences of length M over the set $\{(6i, 1), (6i + 2, 1), (6i + 4, 0)\}$ and evaluating the resulting predictors has to yield a $1/7$ -close to optimal hypothesis on two thirds of the sample. Thus taking the majority vote over the predictions on $(6i + 2)$ is a decider for whether φ_i is a tautology or not. $\square$

11.7.2 Algorithm for the Decidable Representation in Theorem 11.3.2

Algorithm 7: Program to decide whether $x_2 \in u(x_1)$

Input: $x_1, x_2 \in \mathbb{N}$
Output: Whether $x_2 \in u(x_1)$

```

if  $x_1$  is odd then
  if  $x_2 = x_1$  then
    | output  $x_2 \in u(x_1)$ 
  else
    | output  $x_2 \notin u(x_1)$ 
  end
else ▷  $x_1$  is even
  if  $x_2$  is odd then
    | run  $T_{x_1}(x_1)$  for at most  $\frac{x_2-1}{2}$  steps
    if  $T_{x_1}(x_1)$  halted then
      | output  $x_2 \in u(x_1)$ 
    else ▷  $T_{x_1}(x_1)$  is still running after  $\frac{x_2-1}{2}$  steps
      | output  $x_2 \notin u(x_1)$ 
    end
  else ▷  $x_2$  is even
    if  $x_2 = x_1$  then
      | output  $x_2 \in u(x_1)$ 
    else
      | output  $x_2 \notin u(x_1)$ 
    end
  end
end

```

11.7.3 Useful Lemmas for Theorem 11.3.2

Lemma 11.7.1. *There is no program that solves the problem TwoHalt, that is, no program with the following behavior: given a pair of indices of Turing Machines (i, j) as input:*

- *if $T_i(i)$ halts and $T_j(j)$ loops, output 1*
- *if $T_i(i)$ loops and $T_j(j)$ halts, output 2*
- *if both $T_i(i)$ and $T_j(j)$ loop, halt and output 1 or output 2*

- if both $T_i(i)$ and $T_j(j)$ halt, output 1 or output 2 or loop

Before proving Lemma 11.7.1, let us state and prove the following result, which will be used in the proof of Lemma 11.7.1.

Lemma 11.7.2. *For any two 2-place total computable functions f_1, f_2 there are indices c_1, c_2 , such that $T_{c_1} \equiv T_{f_1(c_1, c_2)}$ and $T_{c_2} \equiv T_{f_2(c_1, c_2)}$*

Proof. We start by defining the following Turing Machines:

$$T_a(x_1, x_2, y_1, y_2) = \begin{cases} T_{T_{x_1}(x_1, x_2)}(y_1, y_2) & \text{if } T_{x_1}(x_1, x_2) \text{ halts} \\ \uparrow & \text{otherwise.} \end{cases}$$

and

$$T_b(x_1, x_2, y_1, y_2) = \begin{cases} T_{T_{x_2}(x_1, x_2)}(y_1, y_2) & \text{if } T_{x_2}(x_1, x_2) \text{ halts} \\ \uparrow & \text{otherwise.} \end{cases}$$

By the S_m^n -Theorem, there exist total computable functions h_1 and h_2 , such that $T_a(x_1, x_2, y_1, y_2) = T_{h_1(x_1, x_2)}(y_1, y_2)$ and $T_b(x_1, x_2, y_1, y_2) = T_{h_2(x_1, x_2)}(y_1, y_2)$ hold for all x_1, x_2, y_1, y_2 . Now let e_1 be the index of the Turing machine that computes the function

$$g_1(x_1, x_2) = f_1(h_1(x_1, x_2), h_2(x_1, x_2)) ,$$

and e_2 be the index of the Turing machine that computes the function

$$g_2(x_1, x_2) = f_2(h_1(x_1, x_2), h_2(x_1, x_2)) .$$

Since g_1 is a total computable function, we have $T_{h_1(e_1, e_2)} \equiv T_{T_{e_1}(e_1, e_2)}$. By definition, we have that $T_{e_1}(e_1, e_2)$ computes $f_1(h_1(e_1, e_2), h_2(e_1, e_2))$. Thus $T_{h_1(e_1, e_2)} \equiv T_{f_1(h_1(e_1, e_2), h_2(e_1, e_2))}$. Furthermore, by the same argument, we have that $T_{h_2(e_1, e_2)} \equiv T_{T_{e_2}(e_1, e_2)} \equiv T_{f_2(h_1(e_1, e_2), h_2(e_1, e_2))}$. Thus if we choose $c_1 = h_1(e_1, e_2)$ and $c_2 = h_2(e_1, e_2)$, we get $T_{c_1} \equiv T_{f_1(c_1, c_2)}$ and $T_{c_2} \equiv T_{f_2(c_1, c_2)}$, as required. $\square$

We are now ready to prove Lemma 11.7.1, which completes the result of this section.

of Lemma 11.7.1. By way of contradiction, let's assume that there existed a program/Turing Machine T^{TwoHalt} that solved the TwoHalt problem. Recall that we had fixed an enumeration $(T_i)_{i \in \mathbb{N}}$ of all Turing Machines (or programs), and that we use the notation $T(i) \downarrow$ to indicate that a Turing Machine T halts on input i and $T(i) \uparrow$ to indicate that T loops on

input i . We can now define two natural numbers l_1 and l_2 as being the indices of Turing Machines T_{l_1} and T_{l_2} which have the following behavior:

$$T_{l_1}(i, j, z) = \begin{cases} 1 & \text{if } z = 0 \\ \uparrow & \text{if } z = i > 0 \text{ and } T^{\text{TwoHalt}}(i, j) = 1 \\ 1 & \text{if } z = j > 0 \text{ and } T^{\text{TwoHalt}}(i, j) = 1 \\ \uparrow & \text{if } z = j > 0 \text{ and } T^{\text{TwoHalt}}(i, j) = 2 \\ 2 & \text{if } z = i > 0 \text{ and } T^{\text{TwoHalt}}(i, j) = 2 \\ \uparrow & \text{otherwise} \end{cases}$$

$$T_{l_2}(i, j, z) = \begin{cases} 2 & \text{if } z = 0 \\ \uparrow & \text{if } z = i > 0 \text{ and } T^{\text{TwoHalt}}(i, j) = 1 \\ 1 & \text{if } z = j > 0 \text{ and } T^{\text{TwoHalt}}(i, j) = 1 \\ \uparrow & \text{if } z = j > 0 \text{ and } T^{\text{TwoHalt}}(i, j) = 2 \\ 2 & \text{if } z = i > 0 \text{ and } T^{\text{TwoHalt}}(i, j) = 2 \\ \uparrow & \text{otherwise} \end{cases}$$

Now by S_m^n -Theorem, there are total computable functions f_1 and f_2 such that

$$T_{f_1(i,j)}(z) = \begin{cases} 1 & \text{if } z = 0 \\ \uparrow & \text{if } z = i > 0 \text{ and } T^{\text{TwoHalt}}(i, j) = 1 \\ 1 & \text{if } z = j > 0 \text{ and } T^{\text{TwoHalt}}(i, j) = 1 \\ \uparrow & \text{if } z = j > 0 \text{ and } T^{\text{TwoHalt}}(i, j) = 2 \\ 2 & \text{if } z = i > 0 \text{ and } T^{\text{TwoHalt}}(i, j) = 2 \\ \uparrow & \text{otherwise} \end{cases}$$

and

$$T_{f_2(i,j)}(z) = \begin{cases} 2 & \text{if } z = 0 \\ \uparrow & \text{if } z = i \text{ and } T^{\text{TwoHalt}}(i, j) = 1 \\ 1 & \text{if } z = j \text{ and } T^{\text{TwoHalt}}(i, j) = 1 \\ \uparrow & \text{if } z = j \text{ and } T^{\text{TwoHalt}}(i, j) = 2 \\ 2 & \text{if } z = i \text{ and } T^{\text{TwoHalt}}(i, j) = 2 \\ \uparrow & \text{otherwise} \end{cases}$$

Now, by Lemma 11.7.2, there are $c_1, c_2 > 0$, such that $T_{f_1(c_1, c_2)} \equiv T_{c_1}$ and $T_{f_2(c_1, c_2)} \equiv T_{c_2}$. Recall that we use $T \equiv S$ to indicate that two Turing Machines have the same behavior as functions.

Now let us look at $T^{\text{TwoHalt}}(c_1, c_2)$. There are three options:

- $T^{\text{TwoHalt}}(c_1, c_2) = 1$. Then by definition $T_{c_1}(c_1) = T_{f_1(c_1, c_2)}(c_1) \uparrow$ and $T_{c_2}(c_2) = T_{f_2(c_1, c_2)}(c_2) \downarrow$. Thus T^{TwoHalt} would output 1, the wrong answer in a case where 2, the correct answer, was required.
- $T^{\text{TwoHalt}}(c_1, c_2) = 2$. Then by definition $T_{c_1}(c_1) = T_{f_1(c_1, c_2)}(c_1) \downarrow$ and $T_{c_2}(c_2) = T_{f_2(c_1, c_2)}(c_2) \uparrow$. Thus T^{TwoHalt} would output 2, the wrong answer in a case where 1, the correct answer, was required.
- $T^{\text{TwoHalt}}(c_1, c_2) \uparrow$. Note that, by definition, the Turing Machines $T_{f_1(c_1, c_2)}$ and $T_{f_2(c_1, c_2)}$ only halt and produce an output if either their input was 0 or the call to T^{TwoHalt} inside their definition halts and produces an output. Thus, for this case, we get $T_{c_1}(c_1) = T_{f_1(c_1, c_2)}(c_1) \uparrow$ and $T_{c_2}(c_2) = T_{f_2(c_1, c_2)}(c_2) \uparrow$. However, if both $T_{c_1}(c_1)$ and $T_{c_2}(c_2)$ don't halt, T^{TwoHalt} was supposed to halt. Thus T^{TwoHalt} also has the wrong behaviour in this case.

In summary, the assumption that T^{TwoHalt} existed lead to a contradiction in all cases and thus the problem **TwoHalt** does not admit a solution. $\square$

11.8 Proofs from Section 11.3.2

11.8.1 Proof of Theorem 11.3.3

Proof. Let the instance space be $\mathcal{X} = (\mathbb{N} \times (\{0\} \cup \mathbb{N}))$. Let $(\varphi_i)_{i \in \mathbb{N}}$ be an enumeration of formulas and $(\pi_j)_{j \in \mathbb{N}}$ be an enumeration of proofs. Let $(i, 0)$, represent the i -th formula and for any i, j let (i, j) represent the j -th proof (independently of i). Let $\mathcal{H} = \{h_{a,b}\}_{a,b \in \mathbb{N} \cup \{0\}}$, where

$$h_{a,b}(i, j) = \begin{cases} 1 & \text{if } i < a \text{ or } (i = a \text{ and } j \leq b) \\ 0 & \text{otherwise} \end{cases}.$$

It is clear that $\text{VC}(\mathcal{H}) = 1$ and that ERM is computably implementable, so $\mathcal{H}$ is properly CPAC learnable.

We define the perturbation region as follows:

$$\begin{aligned}\mathcal{U}(i, 0) &= \{(i, 0)\} \cup \{(i, j) : \pi_j \text{ proves } \varphi_i\} , \\ \mathcal{U}(i, j) &= \{(i, 0), (i, j)\} ,\end{aligned}$$

which is clearly DR, as to check whether $(i, j) \in \mathcal{U}((i, 0))$ we can check whether π_j proves φ_i .

It is immediate to see that ℓ^u is not computable for all hypotheses in $\mathcal{H}$. Indeed, for every $i \in \mathbb{N}$, deciding whether for the formula φ_i is a tautology reduces to deciding whether

$$\ell^u(h_{i,0}, (i, 0), 1) = 0 .$$

Now, it remains to show that $\mathcal{H}$ is properly u -robustly CPAC learnable. To this end, we will first show that $\text{VC}(\mathcal{H}_{\text{mar}}^u)$ is finite, and second, show that robust ERM is computable in the agnostic setting. By Fact 11.2.1, we will have robust CPAC learnability.

To show $\text{VC}(\mathcal{H}_{\text{mar}}^u) < \infty$, we consider the sets $\text{mar}_{h_{a,b}}^u$, which we will denote by $\text{mar}_{a,b}$ for brevity. First note that either $a = \infty$ or $b = \infty$ implies that $\text{mar}_{a,b} = \emptyset$. Let $a, b \in \mathbb{N}$ and observe that only instances of the form (a, j) can belong to $\text{mar}_{a,b}$, and thus a shattered set cannot contain both (a, j) and (a', j') where $a \neq a'$. We distinguish two cases:

- φ_a is not a tautology: then $\text{mar}_{a,b} = \{(a, 0)\}$,
- φ_a is a tautology: then $\text{mar}_{a,b} = \{(a, 0)\} \cup \{(a, j) \mid j > b \wedge \pi_j \text{ proves } \varphi_a\}$.

From this, it is clear that $\text{VC}(\mathcal{H}_{\text{mar}}^u) = 1$.

Now, to show that robust ERM is computable, let $S \subseteq \mathcal{X} \times \{0, 1\}$ be an arbitrary labelled sample of size m . For a fixed $i \in \mathbb{N}$ and $y \in \{0, 1\}$, consider the subset S_i^y of S , consisting of “ i -instances” with label y , i.e., $S_i^y := \{((i, j), y) \mid ((i, j), y) \in S\}$, and let $S_i = S_i^0 \cup S_i^1$. We can see that $h_{i,\infty}$ incurs robust loss 1 on all the instances in S_i^0 and robust loss 0 on all the instances in S_i^1 . Furthermore, while every $h_{i,b}$ incurs robust loss 1 on all S_i^0 , it might also incur loss 1 on some instances S_i^1 . Finally, for $b \in \mathbb{N}$, on every instance in $S \setminus S_i$, the hypotheses $h_{i,b}$ and $h_{i,\infty}$ incur the same robust loss. Thus $h_{i,\infty}$ always “dominates” $h_{i,b}$ in the sense that $\mathcal{L}_S^u(h_{i,\infty}) \leq \mathcal{L}_S^u(h_{i,b})$. We furthermore note that for every i , the loss of the function $h_{i,\infty}$ can be evaluated on every point. Thus, performing RERM boils down to comparing all candidate hypotheses of the form $h_{i,\infty}$, which can be done computably. $\square$

Remark 6. In the above proof, let $\mathcal{H}' \subset \mathcal{H}$ be the set of all hypotheses in $\mathcal{H}$ with $b = \infty$, i.e., $\mathcal{H}' = \{h_{a,b} \in \mathcal{H} \mid b = \infty\}$. Then it is clear that, in the robust case, because of the definition of $\mathcal{U}$, we can get away with only outputting hypotheses from $\mathcal{H}'$, no matter what the underlying distribution is, and that for all hypotheses in $\mathcal{H}'$ the robust loss is computable. In general, for some hypothesis class $\mathcal{H}$, let $\mathcal{H}'$ be a class, such that (i) for every $h \in \mathcal{H}$, there is $h' \in \mathcal{H}'$, such that for all $(x, y) \in \mathcal{X} \times \mathcal{Y}$: $\ell^{\mathcal{U}}(h', x, y) \leq \ell^{\mathcal{U}}(h, x, y)$, (ii) the robust loss of every $h' \in \mathcal{H}'$ is computably evaluable. Then if $\mathcal{H}'$ is realizable/agnostic robustly CPAC learnable, so is $\mathcal{H}$. Furthermore, if $\mathcal{H}' \subset \mathcal{H}$ and $\mathcal{H}'$ is *proper* realizable/agnostic robustly learnable, so is $\mathcal{H}$.

11.8.2 Proof of Theorem 11.3.4

Proof. Let the instance space be $\mathcal{X} = \mathbb{N} \times \mathbb{N}$. Let the hypothesis class be $\mathcal{H} = \{h_{i,j} : i, j \in \mathbb{N}\}$, where

$$h_{i,j}(x) = \begin{cases} 1 & \text{if } x = (i, k) \text{ with } k \leq j \\ 0 & \text{otherwise} \end{cases},$$

i.e., $\mathcal{H}$ is the class which defines initial segments on the sets $\{(i, j)\}_{j \in \mathbb{N}}$ for all $i \in \mathbb{N}$.

We define the perturbation types as follows.

$$\begin{aligned} \mathcal{U}((i, 0)) &= \{(i, 0)\} \cup \{(i, k) : T_i \text{ does not halt after } k \text{ steps on the empty input}\}, \\ \mathcal{U}((i, j)) &= \begin{cases} \{(i, j), (i, 0)\} & T_i \text{ does not halt after } j \text{ steps on the empty input} \\ \{(i, j)\} & \text{otherwise.} \end{cases}, \end{aligned}$$

We now prove properties (i)-(iii) from the theorem statement.

To prove (i), i.e., that $\mathcal{H}$ is properly CPAC learnable, it suffices to observe that $\text{VC}(\mathcal{H}) = 1$ and that ERM is easily implementable (in both the agnostic and the realizable cases). In order to verify $\text{VC}(\mathcal{H}) = 1$, let us take to arbitrary distinct points (i_1, j_1) and (i_2, j_2) . If $i_1 \neq i_2$, then the labelling $((i_1, j_1), 1), ((i_2, j_2), 1)$ cannot be achieved by $\mathcal{H}$. If $i_1 = i_2$, then $j_1 \neq j_2$. Let us assume without loss of generality that $j_1 < j_2$. Then the labelling $((i_1, j_1), 0), ((i_2, j_2), 1)$ cannot be achieved. For computably realizing ERM, we note that for any sample S it is sufficient to do search over the finite class $\mathcal{H}_S = \{h_{i,j} : ((i, j), 1) \in S\} \cup \{h_0^S\}$, where $h_0^S = h_{i,0}$, with i being the smallest index for which there is no $((i, j), 1) \in S$. Searching over these finitely many candidate hypotheses can be done computably.

To prove (ii), i.e., that $\mathcal{H}$ is properly u -robustly learnable, we show that the VC dimension $\text{VC}(\mathcal{H}_{\text{mar}}^u)$ of the margin class is finite. To this end, we identify the sets $\text{mar}_{\mathcal{H}}^u$ and distinguish two cases:

- T_i does not halt after j steps on the empty input: then

$$\text{mar}_{h_{i,j}}^u = \{(i, 0)\} \cup \{(i, k) : k > j \text{ and } T_i \text{ does not halt after } k \text{ steps}\} ,$$

- T_i halts on the empty input after j steps: then $\text{mar}_{h_{i,j}}^u = \emptyset$.

We now argue that $\text{VC}(\mathcal{H}_{\text{mar}}^u) = 1$. To this end, let $x_1 = (i_1, j_1)$ and $x_2 = (i_2, j_2)$ be two distinct domain points, and note that, to be shattered, we need $i_1 = i_2$, as otherwise there is no $\text{mar}_h^u \in \mathcal{H}_{\text{mar}}^u$ with $x_1, x_2 \in \text{mar}_h^u$. Now, without loss of generality, let $j_1 < j_2$. There are three cases:

- If T_{i_1} halts after j_2 steps or less, then there is no $\text{mar}_h^u \in \mathcal{H}_{\text{mar}}^u$ with $x_2 \in \text{mar}_h^u$.
- If T_{i_1} does not halt after j_2 and $j_1 \neq 0$, then there is no $\text{mar}_h^u \in \mathcal{H}_{\text{mar}}^u$ with $x_1 \in \text{mar}_h^u$ and $x_2 \notin \text{mar}_h^u$.
- If T_{i_1} does not halt after j_2 steps and $j_1 = 0$, then there is no $\text{mar}_h^u \in \mathcal{H}_{\text{mar}}^u$ with $x_2 \in \text{mar}_h^u$ and $x_1 \notin \text{mar}_h^u$.

To prove (iii), i.e., that the pointwise robust loss is computably evaluable, let $h_{i,j} \in \mathcal{H}$, $x \in \mathcal{X}$, $y \in \mathcal{Y}$ be arbitrary, and consider the following procedure:

- If $x = (i, 0)$ and $y = 1$, we have that $\ell^u(h_{i,j}, x, y) = 1$ if and only if $(i, j+1) \in u((i, 0))$, which can be determined by running T_i for $j+1$ steps and checking whether it halts.
- If $x = (i, 0)$ and $y = 0$, then $\ell^u(h_{i,j}, x, y) = 1$.
- If $x = (i, k)$ with $k > 0$ and $y = 1$, check whether $k \leq j$. If so, $\ell^u(h_{i,j}, x, y) = 0$, otherwise $\ell^u(h_{i,j}, x, y) = 1$.
- If $x = (i, k)$ with $k > 0$ and $y = 0$, check whether $k \leq j$. If so, $\ell^u(h_{i,j}, x, y) = 1$. Otherwise $\ell^u(h_{i,j}, x, y) = 0$ if and only if $(i, 0) \notin u((i, j))$, which can be checked by running T_i for k many steps and checking whether it halts.
- If $x = (i', k)$ with $i' \neq i$, then $\ell^u(h_{i,j}, x, y) = y$.

Finally, we show that there is no strong realizable $\text{RERM}_{\mathcal{H}}^u$ -oracle and that $\mathcal{H}$ is not properly agnostically u -robustly CPAC learnable. We first show the non-existence of a computable strong realizable $\text{RERM}_{\mathcal{H}}^u$ -oracle or agnostic $\text{RERM}_{\mathcal{H}}^u$ -oracle. We prove this with a reduction from the Halting problem: for all $i \in \mathbb{N}$, a strongly realizable $\text{RERM}_{\mathcal{H}}^u$

oracle (or agnostic $\text{RERM}_{\mathcal{H}}^u$ -oracle) ran on the single labelled instance $((i, 0), 1)$ outputs a hypothesis h with robust loss $\ell^u(h, (i, 0), 1) = 0$ if and only if the Turing Machine T_i halts on the empty input. As a side note, we observe that a weak realizable $\text{RERM}_{\mathcal{H}}^u$ -oracle does exist for this construction.

We show that $\mathcal{H}$ is not agnostically properly u -robustly CPAC learnable in a similar way. Assume there was an agnostically properly u -robustly CPAC learner $\mathcal{A}$ with sample complexity function m . For all $i \in \mathbb{N}$, define D_i as $D_i((i, 0), 1) = 1$. Let $m = m(1/8, 1/8)$. By the learning guarantee of $\mathcal{A}$, we have that $\mathcal{A}(S) = h \in \mathcal{H}$ with $\mathcal{L}_{D_i}^u(h) \leq \inf_{h \in \mathcal{H}} \mathcal{L}_{D_i}^u(h) + \frac{1}{8}$ with probability $7/8$ over $S \sim D_i^m$. Observe that $\inf_{h \in \mathcal{H}} \mathcal{L}_{D_i}^u(h) = 0$ if and only if T_i halts on the empty input, and otherwise $\inf_{h \in \mathcal{H}} \mathcal{L}_{D_i}^u(h) = 1$. We note that $S' = (((i, 0), 1), \dots, ((i, 0), 1))$, where $|S'| = m$ is the only possible sample drawn from D_i . Thus we have $\mathcal{A}(S') = h' \in \mathcal{H}$ with $\mathcal{L}_{D_i}^u(h') \leq 1/8$ if and only if T_i halts on the empty input. Furthermore, $\mathcal{L}_{D_i}^u(h') = \ell^u(h', (i, 0), 1) \in \{0, 1\}$ for all $h' \in \mathcal{H}$. Thus we have $\mathcal{L}_{D_i}^u(\mathcal{A}(S'), (i, 0), 1) = \ell^u(\mathcal{A}(S'), (i, 0), 1) = 0$ if and only if T_i halts on the empty input. As established above, the robust loss ℓ^u can be computably evaluated on any $h \in \mathcal{H}$. Thus a proper agnostic learner $\mathcal{A}$ yields a computable procedure for solving the Halting problem, which is a contradiction to the computable undecidability of the Halting problem. Thus $\mathcal{H}$ is not agnostically properly u -robustly CPAC learnable. $\square$

11.8.3 Proof of Proposition 11.3.5

Proof. Let $S = ((x_1, y_1), \dots, (x_k, y_k))$ be an input sample for $\text{RERM}_{\mathcal{H}}^u$, and let $\mathcal{A}$ be the robust SCPAC learning algorithm with computable sample complexity function $m_{\mathcal{H}, u}$. Consider the uniform distribution D_S over S . Let $m \geq m_{\mathcal{H}, u}(\frac{1}{k+1}, 1/7)$, which is computable by the SCPAC guarantee. There are exactly k^m different possible samples $S_1, \dots, S_{k^m}$ of size m that can be drawn from D_S , all of which being equally likely. Using the proper learner $\mathcal{A}$ on all samples S_i , we can generate the class $\hat{\mathcal{H}} = \{\mathcal{A}(S_i) : i \in [k^m]\}$. By the robust learning guarantee, we know that at least $7/8$ of the hypotheses in $\hat{\mathcal{H}}$ are successful hypotheses, i.e., hypotheses within robust risk $1/(k+1)$ of the optimal one. Furthermore, since $\varepsilon = \frac{1}{k+1}$, any successful hypothesis must have optimal loss on S (as any error on a sample point would incur loss $\frac{1}{k}$). It now suffices to use ℓ^u on $\hat{\mathcal{H}}$ and return any one of the hypotheses with minimal empirical robust risk on S . Note that, for the robust realizable setting (the second case in the theorem statement), we only get a weak realizable oracle $\text{RERM}_{\mathcal{H}}^u$, as if the sample is not robustly realizable, we don't have any guarantees on $\mathcal{A}$'s behaviour. $\square$

11.9 Proofs from Section 11.4

11.9.1 Proof of Lemma 11.4.4

Proof. The idea is similar to the standard No-Free-Lunch Theorem. We let m and $Z = \{z_i^0, z_i^1\}_{i=1}^{2m}$ as in the theorem statement, meaning that for all i , $u(z_i^0) \cap u(z_i^1) \neq \emptyset$ and for all $i \neq j$ the regions $u(z_i^1)$ and $u(z_j^0)$ are disjoint. We now look at a robust labelling $y \in \{0, 1\}^{2m}$. There are $T = 2^{2m}$ such labellings, and for each $j \in [T]$ associated with labelling $y^{(j)} \in \{0, 1\}^{2m}$, there exists a function f_j such that for all $i \in [2m]$, for all $z' \in u(z_i^{y_i^{(j)}})$ we have that $f_j(z') = y_i^{(j)}$. Namely, for $i \in [2m]$, the label $y_i^{(j)}$ dictates which of z_i^0 or z_i^1 will have its perturbation region constantly (and thus robustly) labelled. For each function f_j and its robust labelling $y^{(j)}$, we define the distribution D_j as follows:

$$D_j((z, y)) = \begin{cases} 1/2m & \text{if } y = y_i^{(j)} \text{ and } z = z_i^{y_i^{(j)}} \text{ for some } i \in [2m] \\ 0 & \text{otherwise} \end{cases}.$$

It is easy to check that $\mathcal{L}_{D_j}^u(f_j) = 0$.

Let $X = \{x_i\}_{i=1}^m$ be some u -robustly shatterable set corresponding to the set of pairs $Z = \{z_i^0, z_i^1\}_{i=1}^{2m}$, that is $x_i \in u(z_i^0) \cap u(z_i^1)$ for each $i \in [m]$. Now note any hypothesis $h : \mathcal{X} \rightarrow \{0, 1\}$ (that the learner may output) has to assign labels to the the points in the shatterable set X and $h(x_i) = 1$ implies $\ell^u(h, z_i^0) = 1$ while $h(x_i) = 0$ implies $\ell^u(h, z_i^1) = 1$. Thus each hypothesis incurs robust loss on at least one of the members $\{z_i^0, z_i^1\}$ of each pair, and such a member can be deduced from observing $h(x_i)$. Now the remaining argument to show that there exist f_j, D_j such that the conditions of the theorem statement are met is exactly as in the proof of the standard No-Free-Lunch theorem (see, e.g., Theorem 5.1 in [Shalev-Shwartz and Ben-David \[2014a\]](#) for details). We note that the argument there implies that the high expected loss of the learner's output is witnessed by its behaviours on the shatterable set X . Namely, there exists f_j, D_j such that

$$\begin{aligned} \mathbb{E}_{S \sim D_j^m} [\mathcal{L}_{D_j}^u(\mathcal{A}(S))] &= \frac{1}{k} \sum_{l=1}^k \mathcal{L}_{D_j}^u(\mathcal{A}(S_l^j)) = \frac{1}{k} \sum_{l=1}^k \frac{1}{2m} \sum_{i=1}^{2m} \ell^u(\mathcal{A}(S_l^j), z_i^{(j)}, y_i^{(j)}) \quad , \\ &\geq \frac{1}{k} \sum_{l=1}^k \frac{1}{2m} \sum_{i=1}^{2m} \mathbf{1}[\mathcal{A}(S_l^j)(x_i) \neq y_i^{(j)}] \geq 1/8 \end{aligned}$$

where $k = (2m)^m$ is the number of possible sequences of length m drawn from $\text{supp}(D_j) := \{(z_i^{(j)}, y_i^{(j)})\}_{i=1}^{2m}$, and where we have denoted by S_l^j the l -th such sequence. $\square$

11.9.2 Proof of Theorem 11.4.6

Proof. First note that, if the perturbation type u is such that there is an upper bound M on the maximal number of pairs $Z = \{z_i^0, z_i^1\}_{i=1}^{2m}$ admitting a robustly shatterable set, then the computable robust shattering dimension is vacuously finite for any hypothesis class $\mathcal{H}$. Otherwise, let $\mathcal{A}$ be a u -robust CPAC learner for $\mathcal{H}$ with sample complexity function $m(\varepsilon, \delta)$. Then, fixing $\varepsilon = 1/8$ and $\delta = 1/7$, we have that for all distributions on $\mathcal{X} \times \{0, 1\}$, for any sample of size $m := m(1/8, 1/7)$

$$\mathbb{P}_{S \sim D^m} \left(\mathcal{L}_D^u(\mathcal{A}(S)) \geq \min_{h \in \mathcal{H}} \mathcal{L}_D^u(h) + 1/8 \right) < 1/7 . \quad (11.2)$$

Now, let $Z = \{z_i^0, z_i^1\}_{i=1}^{2m}$ admit a robustly shatterable set X , and note that, as explained in the proof of Lemma 11.4.5, we can computably find such a set X . Note that $\mathcal{H}$ and u satisfying the conditions of the theorem statement imply, by Lemma 11.4.5, that we can find a distribution D on $Z \times \{0, 1\}$ such that

$$\mathbb{P}_{S \sim D^m} (\mathcal{L}_D^u(\mathcal{A}(S)) \geq 1/8) \geq 1/7 , \quad (11.3)$$

and whose support is on some $\{z_i^{y_i}\}_{i=1}^{2m}$ for $y \in \{0, 1\}^{2m}$. This support induces a labelling y on X that is not robustly achievable with respect to Z by any $h \in \mathcal{H}$, as we would otherwise have $\min_{h \in \mathcal{H}} \mathcal{L}_D^u(h) = 0$, implying that Equation 11.3 and $\mathbb{P}_{S \sim D^m} (\mathcal{L}_D^u(\mathcal{A}(S)) \geq 1/8) < 1/7$ must hold simultaneously, a contradiction. $\square$

11.10 Remarks on the robust shatterability in the impossibility results from Section 11.3

We now comment on the robust shatterability in the impossibility results from Section 11.3. On the one hand, the perturbation type in Theorem 11.3.2 does not allow for sets of arbitrary size to admit a robustly shatterable set. Indeed, any sequence of halting Turing machines (which are associated with the only instances in $\mathbb{N}$ that have a strict inequality $k \subset u(k)$) will have an ordering such that their perturbation regions form an ascending chain, which makes the condition $i \neq j \implies u(z_i^1) \cap u(z_j^0) = \emptyset$ unsatisfiable for any candidate set $Z = \{z_i^0, z_i^1\}_{i=1}^{2m}$. On the other hand, the perturbation type in Theorem 11.3.1 satisfies the robust shatterability requirement. Indeed, we will add the mild assumption that each proof proves at most one formula. This implies that for each $i \neq i'$, $u(6i + 4)$

and $u(6i' + 4)$ are disjoint, thus, letting $T = (\varphi_{i_k})_{k \in \mathbb{N}}$ be an enumeration of the tautologies, the set $Z = \{(6i_k + 2, 6i_k + 4)\}_{k \in \mathbb{N}}$ admits the robustly shatterable set $X = \{6i_k + 2\}_{k \in \mathbb{N}}$, while the sets $u(6i + 4)$ may be infinite.

Part V

Other Learning Tasks

Chapter 12

Impossibility of Characterizing Other Learning Tasks

In this section we consider two learning problems different from distribution learning. In Section 12.1 we consider classification learning for fixed distribution classes, as proposed by Benedek and Itai [1991], and in Section 12.2 we consider learning with real-valued functions and real-valued losses. For both of these settings we show that there is no scale-invariant dimension which characterizes these learning tasks. Our results do not rule out the existence of scale-sensitive characterization that depend on any fixed accuracy parameter.

12.1 Classification Learning with respect to a restricted class of distributions

The standard PAC learning definitions for classification learning are ‘distribution free’. Namely, learnability requires success with respect to any data-generating probability distributions. Benedek and Itai [1991] proposed a variant of that notion that requires success only w.r.t. some fixed probability distribution. They then raise the idea of extending the definitions by considering a *class* of distributions. This idea is then picked up in Dudley et al. [1994] with the following definition:

Definition 2.2.2. *A hypothesis class H is PAC-learnable with respect to a class of marginal distributions $\mathcal{D}$, if there is a learner $\mathcal{A}$ and a sample complexity function $m_{H,\mathcal{D}} : (0, 1)^2 \rightarrow$*

$\mathbb{N}$, if for every $\varepsilon, \delta > 0$, every $D \in \mathcal{D}$ and every labeling rule $h \in H$ and every $m \geq m_{H, \mathcal{D}}^{0/1}(\varepsilon, \delta)$, with probability $1 - \delta$ over $S \sim (D, h)^m$, we have

$$L_P^{0/1}(\mathcal{A}(S)) < \varepsilon.$$

(where $L_P^{0/1}$ is the usual 0/1 loss).

We will say a class $\mathcal{P} = \{(D, h) : D \in \mathcal{D}, h \in H\}$ of distribution classes is classification learnable, if the class of labelling rules H is learnable with respect to $\mathcal{D}$.

We will denote $\mathcal{D}^{0/1} = \{(D, h) : D \in \mathcal{D}, h \in H_{\text{all}}\}$, where $H_{\text{all}} = 2^{\mathcal{X}}$ is the class of all labelling rules. We note, that while in this definition, the marginal and the labelling rules are independent, an impossibility for characterizing this learning problem yields an impossibility for the more general setting in which the marginal contains information about the labelling rule.

We will now construct classes and show uncharacterizability of both of these learning problems in the sense of the previous chapters. Thus we show that the learning problem proposed by Benedek-Itai (Benedek and Itai [1991]) cannot be characterized in the sense of our notions of scale-invariant characterizing dimension. Using our results from the previous sections, we can get the following uncharacterizability results for this learning task.

Theorem 3.1.12. *There is no weak sample complexity dimension for classification learning of distribution classes.*

Theorem 3.1.17. *There is no finitary characterization of classification learning for distribution classes.*

Theorem 3.1.12 follows from Lemma 3.1.7 and Theorem 3.1.17 follows as a corollary of Theorem 3.1.15. Both theorems use the same class $\mathcal{Q}_{\bar{\gamma}, \bar{n}}$ from Chapter 5. That is we consider learning the class of all labelling functions $H_{\text{all}} = 2^{\mathcal{X}}$ with respect to the class of marginal distributions $\mathcal{Q}_{\bar{\gamma}, \bar{n}}$. In other words we consider the classification learnability of the class $\mathcal{Q}_{\bar{\gamma}, \bar{n}}^{0/1} = \{(D, f) : D \in \mathcal{Q}_{\bar{\gamma}, \bar{n}}, f \in H_{\text{all}}\}$. For the proof of Theorem 3.1.16 we use the classes $H_0 = \{h_a : a \in \{0, 1\}, \text{ where } h_a(x) = a \text{ for all } x \in \mathcal{X}\}$ and $H_k = \mathcal{Q}_{\bar{\gamma}_k, \bar{n}}^{0/1} = \{(D, h), D \in \mathcal{Q}_{\bar{\gamma}_k, \bar{n}}, h \in H_{\text{all}}\}$, where for all $i \in \mathbb{N}$, we have $\bar{n}(i) = i$ and $\bar{\gamma}_k(i) = \frac{1}{k}$. We note that H_0 is an ε -approximation for every H_k in the sense of the second bullet-point of Definition 3.1.14.

Lemma 12.1.1. *If $\lim_{i \rightarrow \infty} \bar{\gamma}(i) = 0$, then the class $H_{\text{all}} = 2^{\mathcal{X}}$ of all labelling rules is PAC learnable with respect to the class $\mathcal{Q} = \mathcal{Q}_{\bar{\gamma}, \bar{n}}$ is classification PAC learnable.*

The proof follows the same idea as the proof of Lemma 5.1.2. The proof can be found in Section 12.3.

Furthermore, we can show a lower bound on the sample complexity for a given $P_{\gamma,2n}$.

Lemma 12.1.2. *For $\mathcal{Q} = P_{\gamma,2n}$ and $H = H_{all}$, we have $m_{H,\mathcal{Q}}^{0/1}(\frac{\gamma}{8}, \frac{1}{7}) \geq n$.*

This Lemma follows directly from the proof of the no-free-lunch theorem in Shalev-Shwartz and Ben-David [2014a].

These two lemmas can now be used to show the theorems of this section. For more details on the proof on this section we refer the reader to Section 12.3.

12.2 Learning Real-Valued Functions with Real-Valued Losses

We will now PAC learning of real-valued functions with continuous losses. Let $\ell^g : 2^{\mathcal{X}} \times \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}_0^+$ be a (point-wise) loss, such there is a continuous function $g : \mathbb{R} \rightarrow \mathbb{R}$ with

- for all $x \in \mathcal{X}, y \in Y, h \in 2^{\mathcal{X}}$: $\ell^g(h, x, y) = g(|h(x) - y|)$.
- $g(0) = 0$, i.e. perfect prediction incurs loss 0.
- There is $a > 0$, such that $g(a) > 0$, i.e. some level of miss-estimation will incur positive loss.

We now analyse PAC-learnability of a class $H \subset \mathcal{Y}^{\mathcal{X}}$ with respect to $L^g(h, P) = \mathbb{E}_{(x,y) \sim P}[\ell^g(h, x, y)]$.

Definition 12.2.1. *We say a class $H \subset \mathcal{Y}^{\mathcal{X}}$ is 1-agnostic PAC-learnable w.r.t. L^g , if there exists a learner $\mathcal{A}$ and a sample complexity function $m_H : (0, 1)^2 \rightarrow \mathbb{N}$, such that for every $\varepsilon, \delta > 0$ and every distribution P over $\mathcal{X} \times \mathcal{Y}$, we have for every $m \geq m_H(\varepsilon, \delta)$,*

$$Pr_{S \sim P^m}[L_P^g(\mathcal{A}(S)) \leq \inf_{h \in H} L_P^g(h) + \varepsilon] \leq 1 - \delta.$$

We say a class H is PAC-learnable w.r.t. L^g in the realizable case if it is learnable with respect to all distributions P with $\inf_{h \in H} L^g(h) = 0$. The sample complexity in the realizable case will be denoted by m_H^{rlzb} .

We now state the main theorems of this subsection.

Theorem 3.1.13. *There is no weak sample complexity dimension for PAC learning real-valued function classes with respect to L^g (in neither the realizable nor the 1-agnostic case).*

Theorem 3.1.18. *There is no finitary characterization of PAC learning real-valued function classes with respect to L^g (in neither the realizable nor the 1-agnostic case).*

We note that these results do not stand in contradiction to the positive result on characterizing learnability for real-valued functions given by [Alon et al. \[1997\]](#), as this result gives a characterization for ε -weak learnability for every ε , rather than a characterization for PAC-learnability. To show our theorems, we need a similar construction as before, which we then use to apply Lemma 3.1.7 and Theorem 3.1.15 respectively.

We will now state the needed construction and then prove learnability as well as a lower bound on the sample complexity needed for the theorems. Let $g_{\max} = \min\{\max_{a>0} g(a), 1\}$ and $g^{-1} : [0, g_{\max}] \rightarrow \mathbb{R}_0^+$.

For every $\gamma \in [0, g_{\max}]$, $n \in \mathbb{N}$ and $A \subset \{1, \dots, n\}$, let

$$f_{\gamma,n}^A(x) = \begin{cases} g^{-1}(\gamma) & \text{if } x \in A \\ 0 & \text{otherwise} \end{cases}$$

Now for a fixed γ and a fixed n , we define

$$F_{\gamma,n} = \{f_{\gamma,n}^A : A \subset \{1, \dots, n\}\}.$$

We then define $H_{\bar{\gamma},\bar{n}}$ for sequences $\bar{\gamma} : \mathbb{N} \rightarrow [0, g_{\max}]$ and $\bar{n} : \mathbb{N} \rightarrow \mathbb{N}$, as

$$H_{\bar{\gamma},\bar{n}} = \bigcup_{i=1}^{\infty} \{f_{\gamma,n}^A : A \subset \{1, \dots, \bar{n}(i)\}\}.$$

Lemma 12.2.2. *If $\lim_{i \rightarrow \infty} \bar{\gamma}(i) = 0$, then $H_{\bar{\gamma},\bar{n}}$ is classification is 1-agnostic PAC learnable with respect to L^g .*

Lemma 12.2.3. *For $F = F_{\gamma,2n}$, we have $m_F(\frac{\gamma}{8}, \frac{1}{7}) \geq m_F^{rlzb}(\frac{\gamma}{8}, \frac{1}{7}) \geq n$.*

Now following the same proof strategy as for our previous results, this construction can be used fulfill the requirements needed to prove the theorems of this section. In particular we use Theorem 3.1.15 with classes $H_0 = \{h_0 : h_0(x) = 0 \text{ for all } x \in \mathcal{X}\}$ and $H_k = H_{\bar{\gamma}_k, \bar{n}}$ with $\bar{n}(i) = i$ and $\bar{\gamma}_k = \frac{1}{k}$. For the proofs of this section, we refer the reader to Section 12.3 at the end of the paper.

12.3 Proofs

Lemma 12.1.1. *If $\lim_{i \rightarrow \infty} \gamma(i) = 0$, then the class $H_{all} = 2^{\mathcal{X}}$ of all labelling rules is PAC learnable with respect to the class $\mathcal{Q} = \mathcal{Q}_{\bar{\gamma}, \bar{n}}$ is classification PAC learnable.*

Proof of Lemma 12.1.1. The proof is equivalent to the proof of Lemma 5.1.2. Let $\varepsilon > 0$. $\mathcal{Q}_{\frac{\varepsilon}{2}}^{0/1} = \bigcup_{i: \bar{\gamma}(i) > \frac{\varepsilon}{2}} P_{\bar{\gamma}(i), \bar{n}(i)}^{0/1}$. From $\lim_{i \rightarrow \infty} \bar{\gamma}(i) = 0$, we know that this class is finite. For a class $\mathcal{Q}$, define the hypothesis class $H(\mathcal{Q}) = \{h \in \{0, 1\}^{\mathcal{X}} : \exists q \in \mathcal{Q} \text{ with } q(x, 1) \geq q(x, 0) \text{ if and only if } h(x) = 1\}$. Now let us consider $H = H(\mathcal{Q}_{\frac{\varepsilon}{2}}^{0/1})$. By construction, this class is finite and can therefore be PAC learned (in the binary classification sense). Furthermore, we have constructed H in such a way that for every $q \in \mathcal{Q}^{0/1}$, we can bound $\inf_{h \in H} L_q^{0/1}(h) \leq \inf_{p \in \mathcal{Q}_{\frac{\varepsilon}{2}}} (TV(p, q) + \inf_{h \in H} L_p^{0/1}(h)) \leq \frac{\varepsilon}{2}$. Thus, if we have a learner that $(\varepsilon/2, \delta)$ -successfully learns H , we can use it to successfully (ε, δ) learned $\mathcal{Q}^{0/1}$ w.r.t. to $L^{0/1}$. $\square$

Lemma 12.1.2. *For $\mathcal{Q} = P_{\gamma, 2n}$ and $H = H_{all}$, we have $m_{H, \mathcal{Q}}^{0/1}(\frac{\gamma}{8}, \frac{1}{7}) \geq n$.*

Proof of Lemma 12.1.2. Let us consider $P \in P_{\gamma, 4n}^{0/1} = \{(D, h) : D = (1 - \gamma)\delta_0 + \gamma U\}$. For any $h \in 2^{\mathbb{N}}$, we can decompose the loss $L_{(D, h')}^{0/1}(h) = (1 - \gamma)\mathbf{1}[h(0) \neq h'(0)] + \gamma L_U^{0/1}(h) \geq \gamma L_U^{0/1}(h)$. Now for every learner $\mathcal{A}$, we can derive the lower bound $\max_{P \in P_{\gamma, 4n}} \mathbb{E}_{S \sim P}[L_{U, h}^{0/1}(A(S))] \geq \frac{1}{4}$, according to the same argument as in the no-free-lunch theorem in [Shalev-Shwartz and Ben-David \[2014a\]](#). Thus we have $\max_{P \in P_{\gamma, 4n}} \mathbb{E}_{S \sim P}[L_P^{0/1}(A(S))] \geq \frac{\gamma}{4}$. Thus by Lemma B.1 of [Shalev-Shwartz and Ben-David \[2014a\]](#), we get the claimed result. $\square$

Theorem 3.1.12. *There is no weak sample complexity dimension for classification learning of distribution classes.*

Proof of Theorem 3.1.12. Let $\mathcal{C}$ be the collection of all classification-learnable distribution classes. According to Lemma 3.1.7 it is sufficient to show that $\{m_{\mathcal{Q}}(\frac{1}{k}, \frac{1}{7}) : \mathcal{Q} \in \mathcal{C}\}$ is cofinal in $\mathbb{N}^{\mathbb{N}}$. Let $g \in \mathbb{N}^{\mathbb{N}}$ be arbitrary. Now consider the class $\mathcal{Q} = \mathcal{Q}_{\bar{\gamma}, \bar{n}}$ with $\bar{\gamma}(k) = \frac{8}{k}$ and $\bar{n}(k) = 8(g(k) + 1)$. Then, according to Lemma 12.1.1 is learnable. Furthermore, since for every $k \in \mathbb{N}$, we have $P_{\frac{8}{k}, \bar{n}(k)} \subset \mathcal{Q}$, by Lemma 12.1.2 we have $g(k) \leq \frac{\bar{n}(k)}{4} < m_{\mathcal{Q}}(1/k, 1/7)$. This shows that $\{m_{\mathcal{Q}}(\frac{1}{k}, \frac{1}{7}) : \mathcal{Q} \in \mathcal{C}\}$ is indeed cofinal in $\mathbb{N}^{\mathbb{N}}$, which concludes the proof of our claim. $\square$

Theorem 3.1.17. *There is no finitary characterization of classification learning for distribution classes.*

Proof of Theorem 3.1.17. We will again use Theorem 3.1.15 to show this claim. Thus we only need to show the conditions for Theorem 3.1.15 hold.

- *Any union of finitely many learnable classes is learnable:* Let us denote this class by $Q = \bigcup_{i=1}^k Q^i$. For every Q^i there is a learner A_i with sample complexity m_i . Let $\varepsilon > 0, \delta > 0$ and let $m_{\max} = \max(\{m_i(\varepsilon/2, \delta/2) : 1 \leq i \leq k\} \cup \{\frac{4(\log(k) + \frac{2}{\delta})}{\varepsilon^2}\})$. Now for some $q \in Q$, let $S \sim q^{m_{\max}}$. For every $i \in \{1, \dots, k\}$, we run A_i on S and denote the output hypothesis by h_i . We know that there is $j \in \{1, \dots, k\}$, such that $q \in Q^j$. By the success-guarantee of A_j , we know that with probability $1 - \frac{\delta}{2}$, $L_q^{0/1}(h_j) \leq \frac{\varepsilon}{2} + L_q^{0/1}(f_q)$. We can now use a PAC-learner for the finite class $H = \{h_i : 1 \leq i \leq k\}$ (which we know to exist from PAC learnability of binary classification of hypothesis classes (see Shalev-Shwartz and Ben-David [2014a])). We know that a sample complexity of $m_{\max}$ guarantees a $(\varepsilon/2, \delta/2)$ -learning success for learning H . Taking everything together, we have constructed a learner that guarantees (ε, δ) -success for learning Q .
- We let $H_0 = \{h_a : a \in \{0, 1\} \text{ where for all } x \in \mathbb{N} : h_a(x) = a\}$. Furthermore, we let H_k be $H_k = \mathcal{Q}_{\bar{\gamma}_k, \bar{n}}$, where $\bar{\gamma}_k(i) = 1/k$ and $n(i) = i$ for all $i \in \mathbb{N}$. By Lemma 12.1.2 all H_k are not learnable. Furthermore, by construction, we have that for every $\varepsilon_k = \frac{1}{k}$, H_0 is an ε_k approximation for H_k . Since $\lim_{k \rightarrow \infty} \varepsilon_k = 0$, the second condition of Theorem 3.1.15 is fulfilled. This concludes our proof.

□

Lemma 12.2.2. *If $\lim_{i \rightarrow \infty} \bar{\gamma}(i) = 0$, then $H_{\bar{\gamma}, \bar{n}}$ is classification is 1-agnostic PAC learnable with respect to L^g .*

Proof of Lemma 12.2.2. Let h_0 be the all-zero function, i.e. $h_0(x) = 0$ for all $x \in \mathbb{N}$. We start by noting that for every γ , every n and every $A \subset \{1, \dots, n\}$ if $P = (D, f_{\gamma, n}^A)$ for any marginal D over $\mathbb{N}$, then $L_P^g(h_0) \leq g(g^{-1}(\gamma)) = \gamma$. Furthermore, we know that finite hypothesis classes of hypotheses with finite range are learnable due to uniform convergence (which we get from first using Hoeffding on each of the elements of the class and then using a union bound). We can now use the same proof idea as in Lemma 5.1.2. Let $\varepsilon > 0$. Since $\lim_{i \rightarrow \infty} \bar{\gamma}(i) = 0$, there is N , such that for every $M \geq N$, $\bar{\gamma}(M) \leq \frac{\varepsilon}{2}$. Thus we know that for every P and every $h \in \mathcal{F}_{\bar{\gamma}, \bar{n}}$, there is $h' \in \bigcup_{i=1}^N \{f_{\bar{\gamma}(i), \bar{n}(i)}^A : A \subset \{1, \dots, n\}\}$, such that

$L_P^g(h') \leq L_P^g(h) + \frac{\varepsilon}{2}$. We now use the fact that $H' = \bigcup_{i=1}^N \{f_{\bar{\gamma}(i), \bar{n}(i)}^A : A \subset \{1, \dots, n\}\}$ is a finite class of hypotheses with finite range and can therefore be successfully PAC-learned w.r.t. L^g . Now we can use the learner for H' with sample complexity $m_{H'}$ on an i.i.d. sample of size $m \geq m_{H'}(\varepsilon/2, \delta)$, to guarantee (ε, δ) -success for learning $H_{\bar{\gamma}, \bar{n}}$. This $\square$

Lemma 12.2.3. *For $F = F_{\gamma, 2n}$, we have $m_F(\frac{\gamma}{8}, \frac{1}{7}) \geq m_F^{rlzb}(\frac{\gamma}{8}, \frac{1}{7}) \geq n$.*

Proof of Lemma 12.2.3. For this lower bound, let us only consider the realizable case. The agnostic case follows directly from it. We note that for a fixed γ and n , for every $h_1, h_2 \in F_{\gamma, 2n}$, every $x \in \mathbb{N}$, we have either $\ell^g(h_1, x, h_2(x)) = \gamma$ or $\ell^g(h_1, x, h_2(x)) = 0$. We can thus treat l^g as a binary loss. We see that the statement now becomes equivalent to the No-Free-Lunch Theorem for binary classification (see Theorem 5.1 in [Shalev-Shwartz and Ben-David \[2014a\]](#)) with the ε -parameter in the sample complexity being scaled by γ . Thus we can conclude $m_F^{rlzb}(\frac{\gamma}{8}, \frac{1}{7}) \geq n$. This concludes our proof. $\square$

Theorem 3.1.13. *There is no weak sample complexity dimension for PAC learning real-valued function classes with respect to L^g (in neither the realizable nor the 1-agnostic case).*

Proof of Theorem 3.1.13. Let $\mathcal{C}$ be the class of all PAC-learnable function classes with respect to L^g . We know from Lemma 3.1.7 that it is sufficient to show that $\{m_H(1/\cdot, 1/7) : H \in \mathcal{C}\}$ is cofinal in $\mathbb{N}^{\mathbb{N}}$. Now let $g \in \mathbb{N}^{\mathbb{N}}$ be arbitrary. We now construct a class H , such that $m_H(\frac{1}{\cdot}, \frac{1}{7})$ eventually dominates g . Let $H = H_{\bar{\gamma}, \bar{n}}$, with $\bar{\gamma}(k) = 8/k$ and $\bar{n} = 4(g(k) + 1)$ for every $k \in \mathbb{N}$. From Lemma 12.1.1 we know that the class is learnable as $\lim_{k \rightarrow \infty} \frac{8}{k} = 0$. From Lemma 12.2.3 we get $g(k) < \frac{\bar{n}(k)}{4} < m_H(1/k, 1/7)$. We note that the notion of "learnability" used here can either refer to the realizable case and sample complexity function m_H^{rlzb} or the 1-agnostic case and sample complexity function m_H . In either case, the statements are true, giving us both claims. $\square$

Theorem 3.1.18. *There is no finitary characterization of PAC learning real-valued function classes with respect to L^g (in neither the realizable nor the 1-agnostic case).*

Proof of Theorem 3.1.18. We use Theorem 3.1.15 to prove the claim. We thus only need to show that the two conditions of the theorem are fulfilled. We will focus on the sub-task of learning function classes with range bounded by 1. Since we only use function classes of the form $H_{\bar{\gamma}, \bar{n}}$ and those all consist of functions with range in $[0, 1]$, this does not cause any issue.

- *Every finite union of learnable classes of hypotheses is learnable:* Let $H = \bigcup_{i=1}^k H^i$ be a union of learnable classes. Let A_i denote the learner and m_i denote the sample complexity for learning H_i . Now define $m_{\max}(\varepsilon, \delta) = \max(\{m_i(\varepsilon/2, \delta/2)\} \cup \{\frac{4k+4\log 2/\delta}{\varepsilon^2}\})$. Let A be the learner that first runs every A_i on an input sample to create a finite hypothesis class $H'(S) = \{A_i(S) : 1 \leq i \leq k\}$ of candidates of size k and then runs ERM on the finite hypothesis class. Now P be some distribution over $\mathcal{X} \times \mathcal{Y}$ and let $m \geq m_{\max}(\varepsilon, \delta)$. Furthermore let $S \sim P^m$. We know that there is j , such that $\inf_{h \in H^j} L_P^g(h) = \inf_{h \in H} L_P^g(h)$. Now A_j guarantees that $\inf_{h \in H'(S)} L_P^g \leq \frac{\varepsilon}{2}$. Furthermore, from Hoeffding's inequality and union bound, we get that a sample size of $m_{\max}(\varepsilon, \delta)$ is sufficient for an ERM to guarantee $(\frac{\varepsilon}{2}, \frac{\delta}{2})$ -success, when learning any finite hypothesis class of size k with functions with range $[0, 1]$. Thus the learner A successfully PAC learns H with sample complexity $m_{\max}$.
- We define $H_0 = \{h_0\}$, with $h_0(x) = 0$ for all $x \in \mathbb{N}$. Furthermore we let $H_k = H_{\bar{\gamma}_k, \bar{n}}$, where $\bar{\gamma}_k(i) = 1/k$ and $\bar{n}(i) = i$. From Lemma 12.2.3 we know that none of the classes H_k are learnable. Furthermore for every $k \in \mathbb{N}$, H_0 is an $\varepsilon_k = \frac{1}{k}$ approximation of H_k , as argued in the proof of Lemma 12.2.2. Lastly $\lim_{k \rightarrow \infty} \varepsilon_k = 0$. This concludes our proof.

□

References

- Jayadev Acharya, Moein Falahatgar, Ashkan Jafarpour, Alon Orlitsky, and Ananda Theertha Suresh. Maximum selection and sorting with adversarial comparators. *Journal of Machine Learning Research*, 19(1):2427–2457, 2018.
- Jayadev Acharya, Ziteng Sun, and Huanyu Zhang. Differentially private assouad, fano, and le cam. In *Proceedings of the 32nd International Conference on Algorithmic Learning Theory*, ALT ’21, pages 48–78. JMLR, Inc., 2021.
- Nathanael Ackerman, Julian Asilis, Jieqi Di, Cameron Freer, and Jean-Baptiste Tristan. Computable pac learning of continuous features. In *Proceedings of the 37th Annual ACM/IEEE Symposium on Logic in Computer Science*, pages 1–12, 2022.
- Ishaq Aden-Ali, Hassan Ashtiani, and Gautam Kamath. On the sample complexity of privately learning unbounded high-dimensional gaussians. In *Proceedings of the 32nd International Conference on Algorithmic Learning Theory*, ALT ’21, pages 185–216. JMLR, Inc., 2021.
- Sushant Agarwal, Nivasini Ananthakrishnan, Shai Ben-David, Tosca Lechner, and Ruth Uerner. On learnability with computable learners. In *Algorithmic Learning Theory*, pages 48–60. PMLR, 2020.
- Saba Ahmadi, Hedyeh Beyhaghi, Avrim Blum, and Keziah Naggita. The strategic perceptron. In Péter Biró, Shuchi Chawla, and Federico Echenique, editors, *EC ’21: The 22nd ACM Conference on Economics and Computation, Budapest, Hungary, July 18-23, 2021*, pages 6–25. ACM, 2021.
- Saba Ahmadi, Avrim Blum, and Kunhe Yang. Fundamental bounds on online strategic classification. In *EC 2023*, pages 22–58. ACM, 2023.

- Saba Ahmadi, Kunhe Yang, and Hanrui Zhang. Strategic littlestone dimension: Improved bounds on online strategic classification, 2024. URL <https://arxiv.org/abs/2407.11619>.
- Daniel Alabi, Pravesh K Kothari, Pranay Tankala, Prayaag Venkat, and Fred Zhang. Privately estimating a Gaussian: Efficient, robust and optimal. In *Proceedings of the 55th Annual ACM Symposium on the Theory of Computing*, STOC '23, New York, NY, USA, 2023. ACM.
- Noga Alon, Shai Ben-David, Nicolò Cesa-Bianchi, and David Haussler. Scale-sensitive dimensions, uniform convergence, and learnability. *J. ACM*, 44(4):615–631, 1997.
- Noga Alon, Steve Hanneke, Ron Holzman, and Shay Moran. A theory of PAC learnability of partial concept classes. In *62nd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2021, Denver, CO, USA, February 7-10, 2022*, pages 658–671. IEEE, 2021.
- Hassan Ashtiani, Shai Ben-David, Nicholas Harvey, Christopher Liaw, Abbas Mehrabian, and Yaniv Plan. Nearly tight sample complexity bounds for learning mixtures of Gaussians via sample compression schemes. In *Advances in Neural Information Processing Systems 31*, NeurIPS '18, 2018.
- Hassan Ashtiani, Vinayak Pathak, and Ruth Urner. Black-box certification and learning under adversarial perturbations. In *Proceedings of the 37th International Conference on Machine Learning, ICML, pages 388–398*, 2020a.
- Hassan Ashtiani, Vinayak Pathak, and Ruth Urner. Black-box certification and learning under adversarial perturbations. In *International Conference on Machine Learning*, pages 388–398. PMLR, 2020b.
- Hassan Ashtiani, Vinayak Pathak, and Ruth Urner. Adversarially robust learning with tolerance. In *International Conference on Algorithmic Learning Theory, ALT*, pages 115–135, 2023a.
- Hassan Ashtiani, Vinayak Pathak, and Ruth Urner. Adversarially robust learning with tolerance. In *International Conference on Algorithmic Learning Theory*, pages 115–135. PMLR, 2023b.
- Idan Attias and Steve Hanneke. Adversarially robust pac learnability of real-valued functions. *arXiv preprint arXiv:2206.12977*, 2022.

- Idan Attias, Aryeh Kontorovich, and Yishay Mansour. Improved generalization bounds for robust learning. In *Algorithmic Learning Theory, ALT*, pages 162–183, 2019.
- Idan Attias, Steve Hanneke, and Yishay Mansour. A characterization of semi-supervised adversarially robust pac learnability. *Advances in Neural Information Processing Systems*, 35:23646–23659, 2022a.
- Idan Attias, Aryeh Kontorovich, and Yishay Mansour. Improved generalization bounds for adversarially robust learning. *Journal of Machine Learning Research*, 23(175):1–31, 2022b.
- Marco Avella-Medina. The role of robust statistics in private data analysis. *Chance*, 33(4):37–42, 2020.
- Pranjal Awasthi, Abhratanu Dutta, and Aravindan Vijayaraghavan. On robustness to adversarial examples and polynomial optimization. In *Advances in Neural Information Processing Systems, NeurIPS*, pages 13760–13770, 2019.
- Pranjal Awasthi, Natalie Frank, and Mehryar Mohri. Adversarial learning guarantees for linear hypotheses and neural networks. In *International Conference on Machine Learning*, pages 431–441. PMLR, 2020.
- Pranjal Awasthi, Anqi Mao, Mehryar Mohri, and Yutao Zhong. H-consistency bounds for surrogate loss minimizers. In *International Conference on Machine Learning*, pages 1117–1174. PMLR, 2022a.
- Pranjal Awasthi, Anqi Mao, Mehryar Mohri, and Yutao Zhong. Multi-class h -consistency bounds. *Advances in neural information processing systems*, 35:782–795, 2022b.
- Pranjal Awasthi, Anqi Mao, Mehryar Mohri, and Yutao Zhong. Theoretically grounded loss functions and algorithms for adversarial robustness. In *International Conference on Artificial Intelligence and Statistics*, pages 10077–10094. PMLR, 2023a.
- Pranjal Awasthi, Anqi Mao, Mehryar Mohri, and Yutao Zhong. Theoretically grounded loss functions and algorithms for adversarial robustness. In *International Conference on Artificial Intelligence and Statistics*, pages 10077–10094. PMLR, 2023b.
- Ainesh Bakshi, Ilias Diakonikolas, He Jia, Daniel M Kane, Pravesh K Kothari, and Santosh S Vempala. Robustly learning mixtures of k arbitrary Gaussians. In *Proceedings of the 54th Annual ACM Symposium on the Theory of Computing, STOC '22*, pages 1234–1247, New York, NY, USA, 2022. ACM.

- Maria-Florina Balcan, Avrim Blum, and Santosh S. Vempala. Efficient representations for lifelong learning and autoencoding. In *Proceedings of The 28th Conference on Learning Theory, COLT*, pages 191–210, 2015.
- Maria-Florina Balcan, Avrim Blum, and Vaishnavh Nagarajan. Lifelong learning in costly feature spaces. *Theor. Comput. Sci.*, 808:14–37, 2020a.
- Maria-Florina Balcan, Avrim Blum, Dravyansh Sharma, and Hongyang Zhang. On the power of abstention and data-driven decision making for adversarial robustness. *CoRR*, abs/2010.06154, 2020b.
- Maria-Florina Balcan, Avrim Blum, Steve Hanneke, and Dravyansh Sharma. Robustly-reliable learners under poisoning attacks. In *Conference on Learning Theory*, pages 4498–4534. PMLR, 2022.
- Maria-Florina Balcan, Steve Hanneke, Rattana Pukdee, and Dravyansh Sharma. Reliable learning for test-time attacks and distribution shift. *CoRR*, abs/2304.03370, 2023a.
- Maria-Florina F Balcan, Steve Hanneke, Rattana Pukdee, and Dravyansh Sharma. Reliable learning in challenging environments. *Advances in Neural Information Processing Systems*, 36, 2023b.
- Peter L. Bartlett and Marten H. Wegkamp. Classification with a reject option using a hinge loss. *Journal of Machine Learning Research*, 9(59):1823–1840, 2008.
- Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Mach. Learn.*, 79(1-2):151–175, 2010.
- Shai Ben-David, Pavel Hrubes, Shay Moran, Amir Shpilka, and Amir Yehudayoff. A learning problem that is independent of the set theory ZFC axioms. *CoRR*, abs/1711.05195, 2017a.
- Shai Ben-David, Pavel Hrubes, Shay Moran, Amir Shpilka, and Amir Yehudayoff. A learning problem that is independent of the set theory ZFC axioms. *CoRR*, abs/1711.05195, 2017b.
- Shai Ben-David, Pavel Hrubes, Shay Moran, Amir Shpilka, and Amir Yehudayoff. Learnability can be undecidable. *Nature Machine Intelligence*, 1:44–48, 2019.

- Shai Ben-David, Alex Bie, Gautam Kamath, and Tosca Lechner. Distribution learnability and robustness. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *NeurIPS 2023*, 2023.
- Shai Ben-David, Alex Bie, Gautam Kamath, and Tosca Lechner. Distribution learnability and robustness. In *NeurIPS*, 2023.
- Gyora M. Benedek and Alon Itai. Learnability with respect to fixed distributions. *Theor. Comput. Sci.*, 86(2):377–390, 1991.
- Robi Bhattacharjee and Kamalika Chaudhuri. Consistent non-parametric methods for maximizing robustness. In *Advances in Neural Information Processing Systems*, volume 34, pages 9036–9048, 2021.
- Robi Bhattacharjee, Somesh Jha, and Kamalika Chaudhuri. Sample complexity of robust linear classification on separated data. In *International Conference on Machine Learning*, pages 884–893. PMLR, 2021.
- Robi Bhattacharjee, Max Hopkins, Akash Kumar, Hantao Yu, and Kamalika Chaudhuri. Robust empirical risk minimization with tolerance. In *International Conference on Algorithmic Learning Theory ALT*, pages 182–203, 2023.
- Battista Biggio, Blaine Nelson, and Pavel Laskov. Poisoning attacks against support vector machines. In *Proceedings of the 29th International Conference on Machine Learning, ICML '12*, pages 1467–1474. JMLR, Inc., 2012.
- Guy Blanc, Jane Lange, Ali Malik, and Li-Yang Tan. On the power of adaptivity in statistical adversaries. In *Proceedings of the 35th Annual Conference on Learning Theory, COLT '22*, pages 5030–5061, 2022.
- Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *Journal of the ACM*, 36(4):929–965, 1989.
- Olivier Bousquet, Daniel M. Kane, and Shay Moran. The optimal approximation factor in density estimation. In *Proceedings of the 32nd Annual Conference on Learning Theory, COLT '19*, pages 318–341, 2019.
- Olivier Bousquet, Mark Braverman, Gillat Kol, Klim Efremenko, and Shay Moran. Statistically near-optimal hypothesis selection. In *Proceedings of the 62nd Annual IEEE Symposium on Foundations of Computer Science, FOCS '21*, pages 909–919. IEEE Computer Society, 2022.

- Gavin Brown, Marco Gaboardi, Adam Smith, Jonathan Ullman, and Lydia Zakyntinou. Covariance-aware private mean estimation without private covariance estimation. In *Advances in Neural Information Processing Systems 34*, NeurIPS '21. Curran Associates, Inc., 2021.
- Michael Brückner and Tobias Scheffer. Stackelberg games for adversarial prediction problems. In *Proceedings of the 17th ACM International Conference on Knowledge Discovery and Data Mining SIGKDD*, pages 547–555, 2011.
- Nataly Brückner, Daniel Carmon, Irit Dinur, Shay Moran, and Amir Yehudayoff. A characterization of multiclass learnability. In *Foundations of Computer Science, FOCS*, 2022.
- Sebastien Bubeck, Yin Tat Lee, Eric Price, and Ilya Razenshteyn. Adversarial examples from computational constraints. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pages 831–840. PMLR, 2019.
- Mark Bun and Thomas Steinke. Average-case averages: Private algorithms for smooth sensitivity and mean estimation. In *Advances in Neural Information Processing Systems 32*, NeurIPS '19, pages 181–191. Curran Associates, Inc., 2019.
- Mark Bun, Gautam Kamath, Thomas Steinke, and Zhiwei Steven Wu. Private hypothesis selection. In *Advances in Neural Information Processing Systems 32*, NeurIPS '19, pages 156–167. Curran Associates, Inc., 2019.
- T Tony Cai, Yichen Wang, and Linjun Zhang. The cost of privacy: Optimal rates of convergence for parameter estimation with differential privacy. *The Annals of Statistics*, 49(5):2825–2850, 2021.
- Siu On Chan, Ilias Diakonikolas, Rocco A. Servedio, and Xiaorui Sun. Learning mixtures of structured distributions over discrete domains. In *Proceedings of the 24th Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '13, pages 1380–1394, Philadelphia, PA, USA, 2013. SIAM.
- Siu On Chan, Ilias Diakonikolas, Rocco A. Servedio, and Xiaorui Sun. Efficient density estimation via piecewise polynomial approximation. In *Proceedings of the 46th Annual ACM Symposium on the Theory of Computing*, STOC '14, pages 604–613, New York, NY, USA, 2014a. ACM.
- Siu On Chan, Ilias Diakonikolas, Rocco A. Servedio, and Xiaorui Sun. Near-optimal density estimation in near-linear time using variable-width histograms. In *Advances in Neural*

- Information Processing Systems 27*, NIPS '14, pages 1844–1852. Curran Associates, Inc., 2014b.
- Yatong Chen, Jialu Wang, and Yang Liu. Linear classifiers that encourage constructive adaptation, 2021.
- Yiling Chen, Yang Liu, and Chara Podimata. Learning strategy-aware linear classifiers. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NeurIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.
- Sadia Chowdhury and Ruth Uner. Robustness should not be at odds with accuracy. In *3rd Symposium on Foundations of Responsible Computing, FORC*, pages 5:1–5:20, 2022.
- Daniel Cullina, Arjun Nitin Bhagoji, and Prateek Mittal. PAC-learning in the presence of evasion adversaries. *Advances in Neural Information Processing Systems*, 2018.
- Nilesh N. Dalvi, Pedro M. Domingos, Mausam, Sumit K. Sanghai, and Deepak Verma. Adversarial classification. In *Proceedings of the Tenth ACM International Conference on Knowledge Discovery and Data Mining SIGKDD*, pages 99–108, 2004.
- Amit Daniely and Shai Shalev-Shwartz. Optimal learners for multiclass problems. In *Conference on Learning Theory, COLT*, 2014.
- Amit Daniely, Nati Linial, and Shai Shalev-Shwartz. From average case complexity to improper learning complexity. In *Symposium on Theory of Computing, STOC*, pages 441–448, 2014.
- Amit Daniely, Sivan Sabato, Shai Ben-David, and Shai Shalev-Shwartz. Multiclass learnability and the ERM principle. *J. Mach. Learn. Res.*, 16:2377–2404, 2015.
- Constantinos Daskalakis and Gautam Kamath. Faster and sample near-optimal algorithms for proper learning mixtures of Gaussians. In *Proceedings of the 27th Annual Conference on Learning Theory, COLT '14*, pages 1183–1213, 2014.
- Constantinos Daskalakis, Ilias Diakonikolas, and Rocco A. Servedio. Learning Poisson binomial distributions. In *Proceedings of the 44th Annual ACM Symposium on the Theory of Computing, STOC '12*, pages 709–728, New York, NY, USA, 2012. ACM.
- Ofir David, Shay Moran, and Amir Yehudayoff. Supervised learning through the lens of compression. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett, editors, *NeurIPS*, pages 2784–2792, 2016.

- Akshay Degwekar and Vinod Vaikuntanathan. Computational limitations in robust classification and win-win results. *arXiv preprint arXiv:1902.01086*, 2019.
- Valentino Delle Rose, Alexander Kozachinskiy, Cristóbal Rojas, and Tomasz Steifer. Find a witness or shatter: the landscape of computable pac learning. In *The Thirty Sixth Annual Conference on Learning Theory*, pages 511–524. PMLR, 2023.
- Luc Devroye and Gábor Lugosi. A universally acceptable smoothing factor for kernel density estimation. *The Annals of Statistics*, 24(6):2499–2512, 1996.
- Luc Devroye and Gábor Lugosi. Nonasymptotic universal smoothing factors, kernel complexity and Yatracos classes. *The Annals of Statistics*, 25(6):2626–2637, 1997.
- Luc Devroye and Gábor Lugosi. *Combinatorial methods in density estimation*. Springer, 2001.
- Ilias Diakonikolas. Learning structured distributions. In Peter Bühlmann, Petros Drineas, Michael J. Kane, and Mark J. van der Laan, editors, *Handbook of Big Data*, pages 267–283. Chapman and Hall/CRC, 2016.
- Ilias Diakonikolas and Daniel Kane. *Algorithmic High-Dimensional Robust Statistics*. Cambridge University Press, 2022.
- Ilias Diakonikolas, Gautam Kamath, Daniel M. Kane, Jerry Li, Ankur Moitra, and Alistair Stewart. Robust estimators in high dimensions without the computational intractability. In *Proceedings of the 57th Annual IEEE Symposium on Foundations of Computer Science*, FOCS ’16, pages 655–664, Washington, DC, USA, 2016. IEEE Computer Society.
- Ilias Diakonikolas, Gautam Kamath, Daniel M. Kane, Jerry Li, Ankur Moitra, and Alistair Stewart. Being robust (in high dimensions) can be practical. In *Proceedings of the 34th International Conference on Machine Learning*, ICML ’17, pages 999–1008. JMLR, Inc., 2017a.
- Ilias Diakonikolas, Daniel M. Kane, and Alistair Stewart. Statistical query lower bounds for robust estimation of high-dimensional Gaussians and Gaussian mixtures. In *Proceedings of the 58th Annual IEEE Symposium on Foundations of Computer Science*, FOCS ’17, pages 73–84, Washington, DC, USA, 2017b. IEEE Computer Society.
- Ilias Diakonikolas, Gautam Kamath, Daniel M. Kane, Jerry Li, Ankur Moitra, and Alistair Stewart. Robustly learning a Gaussian: Getting optimal error, efficiently. In *Proceedings of the 29th Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA ’18, Philadelphia, PA, USA, 2018. SIAM.

- Ilias Diakonikolas, Gautam Kamath, Daniel M. Kane, Jerry Li, Jacob Steinhardt, and Alistair Stewart. Sever: A robust meta-algorithm for stochastic optimization. In *Proceedings of the 36th International Conference on Machine Learning, ICML '19*, pages 1596–1606. JMLR, Inc., 2019.
- Ilias Diakonikolas, Daniel M Kane, and Pasin Manurangsi. The complexity of adversarially robust proper learning of halfspaces with agnostic noise. *Advances in Neural Information Processing Systems*, 33:20449–20461, 2020.
- Dimitrios Diochnos, Saeed Mahloujifar, and Mohammad Mahmoody. Adversarial risk and robustness: General definitions and implications for the uniform distribution. In *Advances in Neural Information Processing Systems*, 2018.
- Dimitrios I. Diochnos, Saeed Mahloujifar, and Mohammad Mahmoody. Lower bounds for adversarially robust PAC learning under evasion and hybrid attacks. In *2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 717–722, 2020. doi: 10.1109/ICMLA51294.2020.00117.
- Jinshuo Dong, Aaron Roth, Zachary Schutzman, Bo Waggoner, and Zhiwei Steven Wu. Strategic classification from revealed preferences. In *Proceedings of the 2018 ACM Conference on Economics and Computation, EC*, pages 55–70, 2018.
- R. M. Dudley, Sanjeev R. Kulkarni, T. J. Richardson, and Ofer Zeitouni. A metric entropy bound is not sufficient for learnability. *IEEE Trans. Inf. Theory*, 40(3):883–885, 1994.
- Cynthia Dwork and Jing Lei. Differential privacy and robust statistics. In *Proceedings of the 41st Annual ACM Symposium on the Theory of Computing, STOC '09*, pages 371–380, New York, NY, USA, 2009. ACM.
- Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the 3rd Conference on Theory of Cryptography, TCC '06*, pages 265–284, Berlin, Heidelberg, 2006. Springer.
- Ran El-Yaniv and Yair Wiener. On the foundations of noise-free selective classification. *J. Mach. Learn. Res.*, 11:1605–1641, 2010.
- Alhussein Fawzi, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard. Robustness of classifiers: from adversarial to random noise. In *Advances in Neural Information Processing Systems*, pages 1632–1640, 2016.

- Alhussein Fawzi, Hamza Fawzi, and Omar Fawzi. Adversarial vulnerability for any classifier. *Advances in neural information processing systems*, 31, 2018a.
- Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Analysis of classifiers robustness to adversarial perturbations. *Machine Learning*, 107(3):481–508, 2018b.
- Uriel Feige, Yishay Mansour, and Robert Schapire. Learning and inference in the presence of corrupted inputs. In *Conference on Learning Theory, COLT*, pages 637–657, 2015.
- Sally Floyd and Manfred K. Warmuth. Sample compression, learnability, and the vapnik-chervonenkis dimension. *Machine Learning*, 21(3):269–304, 1995.
- Sanjam Garg, Somesh Jha, Saeed Mahloujifar, and Mahmoody Mohammad. Adversarially robust learning could leverage computational hardness. In *Algorithmic Learning Theory*, pages 364–385. PMLR, 2020.
- Jonas Geiping, Liam Fowl, W Ronny Huang, Wojciech Czaja, Gavin Taylor, Michael Moeller, and Tom Goldstein. Witches’ brew: Industrial scale data poisoning via gradient matching. In *Proceedings of the 9th International Conference on Learning Representations*, ICLR ’21, 2021.
- Kristian Georgiev and Samuel B Hopkins. Privacy induces robustness: Information-computation gaps and sparse mean estimation. In *Advances in Neural Information Processing Systems 35*, NeurIPS ’22. Curran Associates, Inc., 2022.
- Justin Gilmer, Luke Metz, Fartash Faghri, Samuel S Schoenholz, Maithra Raghu, Martin Wattenberg, and Ian Goodfellow. Adversarial spheres. *arXiv preprint arXiv:1801.02774*, 2018.
- Micah Goldblum, Dimitris Tsipras, Chulin Xie, Xinyun Chen, Avi Schwarzschild, Dawn Song, Aleksander Madry, Bo Li, and Tom Goldstein. Dataset security for machine learning: Data poisoning, backdoor attacks, and defenses. *arXiv preprint arXiv:2012.10544*, 2020.
- Sivakanth Gopi, Gautam Kamath, Janardhan Kulkarni, Aleksandar Nikolov, Zhiwei Steven Wu, and Huanyu Zhang. Locally private hypothesis selection. In *Proceedings of the 33rd Annual Conference on Learning Theory, COLT ’20*, pages 1785–1816, 2020.
- Pascale Gourdeau, Varun Kanade, Marta Kwiatkowska, and James Worrell. On the hardness of robust classification. *J. Mach. Learn. Res.*, 22:273:1–273:29, 2021a.

- Pascale Gourdeau, Varun Kanade, Marta Kwiatkowska, and James Worrell. On the hardness of robust classification. *The Journal of Machine Learning Research*, 22(1):12521–12549, 2021b.
- Pascale Gourdeau, Varun Kanade, Marta Kwiatkowska, and James Worrell. Sample complexity bounds for robustly learning decision lists against evasion attacks. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI*, pages 3022–3028, 2022a.
- Pascale Gourdeau, Varun Kanade, Marta Kwiatkowska, and James Worrell. When are local queries useful for robust learning? *Advances in Neural Information Processing Systems*, 35:33920–33933, 2022b.
- Pascale Gourdeau, Varun Kanade, Marta Kwiatkowska, and James Worrell. Sample complexity bounds for robustly learning decision lists against evasion attacks. In *International Joint Conference in Artificial Intelligence*, 2022c.
- Pascale Gourdeau, Tosca Lechner, and Ruth Urner. On the computability of robust pac learning. In *COLT*, 2024.
- Nika Haghtalab, Nicole Immorlica, Brendan Lucier, and Jack Z. Wang. Maximizing welfare with incentive-aware evaluation mechanisms. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI*, pages 160–166, 2020.
- Steve Hanneke. A bound on the label complexity of agnostic active learning. In *Machine Learning, Proceedings of the Twenty-Fourth International Conference ICML*, pages 353–360, 2007.
- Steve Hanneke et al. Theory of disagreement-based active learning. *Foundations and Trends® in Machine Learning*, 7(2-3):131–309, 2014.
- Moritz Hardt, Nimrod Megiddo, Christos H. Papadimitriou, and Mary Wootters. Strategic classification. In *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science, ITCS*, pages 111–122, 2016.
- Niki Hasrati and Shai Ben-David. On computable online learning. In *International Conference on Algorithmic Learning Theory*, pages 707–725. PMLR, 2023.
- David Haussler. Decision theoretic generalizations of the PAC model for neural net and other learning applications. *Inf. Comput.*, 100(1):78–150, 1992.

- David Haussler and Emo Welzl. epsilon-nets and simplex range queries. *Discret. Comput. Geom.*, 2:127–151, 1987.
- David Haussler, Nick Littlestone, and Manfred K Warmuth. Predicting $\{0, 1\}$ -functions on randomly drawn points. *Information and Computation*, 115(2):248–292, 1994.
- Max Hopkins, Daniel M Kane, Shachar Lovett, and Gaurav Mahajan. Realizable learning is all you need. In *Proceedings of the 35th Annual Conference on Learning Theory, COLT '22*, pages 3015–3069, 2022a.
- Samuel B. Hopkins and Jerry Li. Mixture models, robustness, and sum of squares proofs. In *Proceedings of the 50th Annual ACM Symposium on the Theory of Computing*, STOC '18, pages 1021–1034, New York, NY, USA, 2018. ACM.
- Samuel B Hopkins, Gautam Kamath, and Mahbod Majid. Efficient mean estimation with pure differential privacy via a sum-of-squares exponential mechanism. In *Proceedings of the 54th Annual ACM Symposium on the Theory of Computing*, STOC '22, pages 1406–1417, New York, NY, USA, 2022b. ACM.
- Samuel B Hopkins, Gautam Kamath, Mahbod Majid, and Shyam Narayanan. Robustness implies privacy in statistical estimation. In *Proceedings of the 55th Annual ACM Symposium on the Theory of Computing*, STOC '23, New York, NY, USA, 2023. ACM.
- Lily Hu, Nicole Immorlica, and Jennifer Wortman Vaughan. The disparate effects of strategic manipulation. In *Proceedings of the Conference on Fairness, Accountability, and Transparency, FAT**, pages 259–268, 2019.
- Peter J. Huber. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, 35(1):73–101, 1964.
- Meena Jagadeesan, Celestine Mendler-Dünner, and Moritz Hardt. Alternative microfoundations for strategic classification. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021*, pages 4687–4697, 2021.
- Ayush Jain, Alon Orlitsky, and Vaishakh Ravindrakumar. Robust estimation algorithms don't need to know the corruption level. *arXiv preprint arXiv:2202.05453*, 2022.
- He Jia, Pravesh K Kothari, and Santosh S Vempala. Beyond moments: Robustly learning affine transformations with asymptotically optimal error. *arXiv preprint arXiv:2302.12289*, 2023.

- Gautam Kamath and Jonathan Ullman. A primer on private statistics. *arXiv preprint arXiv:2005.00010*, 2020.
- Gautam Kamath, Jerry Li, Vikrant Singhal, and Jonathan Ullman. Privately learning high-dimensional distributions. In *Proceedings of the 32nd Annual Conference on Learning Theory*, COLT '19, pages 1853–1902, 2019.
- Gautam Kamath, Vikrant Singhal, and Jonathan Ullman. Private mean estimation of heavy-tailed distributions. In *Proceedings of the 33rd Annual Conference on Learning Theory*, COLT '20, pages 2204–2235, 2020.
- Gautam Kamath, Argyris Mouzakis, and Vikrant Singhal. New lower bounds for private estimation and a generalized fingerprinting lemma. In *Advances in Neural Information Processing Systems 35*, NeurIPS '22. Curran Associates, Inc., 2022a.
- Gautam Kamath, Argyris Mouzakis, Vikrant Singhal, Thomas Steinke, and Jonathan Ullman. A private and computationally-efficient estimator for unbounded gaussians. In *Proceedings of the 35th Annual Conference on Learning Theory*, COLT '22, pages 544–572, 2022b.
- Vishesh Karwa and Salil Vadhan. Finite sample differentially private confidence intervals. In *Proceedings of the 9th Conference on Innovations in Theoretical Computer Science*, ITCS '18, pages 44:1–44:9, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.
- Michael Kearns, Yishay Mansour, Dana Ron, Ronitt Rubinfeld, Robert E. Schapire, and Linda Sellie. On the learnability of discrete distributions. In *Proceedings of the 26th Annual ACM Symposium on the Theory of Computing*, STOC '94, pages 273–282, New York, NY, USA, 1994. ACM.
- Justin Khim, Varun Jog, and Po-Ling Loh. Adversarial influence maximization. In *2019 IEEE International Symposium on Information Theory (ISIT)*, pages 1–5. IEEE, 2019.
- Daniel Kifer, Shai Ben-David, and Johannes Gehrke. Detecting change in data streams. In Mario A. Nascimento, M. Tamer Özsu, Donald Kossmann, Renée J. Miller, José A. Blakeley, and K. Bernhard Schiefer, editors, *(e)Proceedings of the Thirtieth International Conference on Very Large Data Bases, VLDB 2004, Toronto, Canada, August 31 - September 3 2004*, pages 180–191. Morgan Kaufmann, 2004.

- Pravesh Kothari, Jacob Steinhardt, and David Steurer. Robust moment estimation and improved clustering via sum of squares. In *Proceedings of the 50th Annual ACM Symposium on the Theory of Computing*, STOC '18, pages 1035–1046, New York, NY, USA, 2018. ACM.
- Pravesh K Kothari, Pasin Manurangsi, and Ameya Velingker. Private robust estimation by stabilizing convex relaxations. In *Proceedings of the 35th Annual Conference on Learning Theory*, COLT '22, pages 723–777, 2022.
- Sanjeev R. Kulkarni and M. Vidyasagar. Learning decision rules for pattern classification under a family of probability measures. *IEEE Trans. Inf. Theory*, 43(1):154–166, 1997. URL <https://doi.org/10.1109/18.567668>.
- Kevin A. Lai, Anup B. Rao, and Santosh Vempala. Agnostic estimation of mean and covariance. In *Proceedings of the 57th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '16, pages 665–674, Washington, DC, USA, 2016. IEEE Computer Society.
- Tosca Lechner and Ruth Uerner. Learning losses for strategic classification. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022*, pages 7337–7344, 2022.
- Tosca Lechner, Vinayak Pathak, and Ruth Uerner. Adversarially robust learning with uncertain perturbation sets. In *Advances in Neural Information Processing Systems*, 2023a.
- Tosca Lechner, Ruth Uerner, and Shai Ben-David. Strategic classification with unknown user manipulations. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 18714–18732. PMLR, 2023b.
- Sagi Levanon and Nir Rosenfeld. Generalized strategic classification and the case of aligned incentives. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 12593–12618. PMLR, 17–23 Jul 2022.
- Jerry Li and Ludwig Schmidt. Robust proper learning for mixtures of Gaussians via systems of polynomial inequalities. In *Proceedings of the 30th Annual Conference on Learning Theory*, COLT '17, pages 1302–1382, 2017.

- Allen Liu and Ankur Moitra. Settling the robust learnability of mixtures of gaussians. In *Proceedings of the 53rd Annual ACM Symposium on the Theory of Computing*, STOC '21, pages 518–531, New York, NY, USA, 2021. ACM.
- Allen Liu and Ankur Moitra. Learning GMMs with nearly optimal robustness guarantees. In *Proceedings of the 35th Annual Conference on Learning Theory*, COLT '22, pages 2815–2895, 2022.
- Xiyang Liu, Weihao Kong, Sham Kakade, and Sewoong Oh. Robust and differentially private mean estimation. In *Advances in Neural Information Processing Systems 34*, NeurIPS '21. Curran Associates, Inc., 2021.
- Xiyang Liu, Weihao Kong, and Sewoong Oh. Differential privacy and robust statistics in high dimensions. In *Proceedings of the 35th Annual Conference on Learning Theory*, COLT '22, pages 1167–1246, 2022.
- Yiwei Lu, Gautam Kamath, and Yaoliang Yu. Exploring the limits of model-targeted indiscriminate data poisoning attacks. In *Proceedings of the 40th International Conference on Machine Learning*, ICML '23, pages 22856–22879. JMLR, Inc., 2023.
- Satyaki Mahalanabis and Daniel Stefankovic. Density estimation in linear time. In *Proceedings of the 21st Annual Conference on Learning Theory*, COLT '08, pages 503–512, 2008.
- Saeed Mahloujifar and Mohammad Mahmoody. Can adversarially robust learning leverage computational hardness? In *Algorithmic Learning Theory*, pages 581–609. PMLR, 2019.
- Saeed Mahloujifar, Dimitrios I Diochnos, and Mohammad Mahmoody. The curse of concentration in robust learning: Evasion and poisoning attacks from concentration of measure. *AAAI Conference on Artificial Intelligence*, 2019.
- Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation: Learning bounds and algorithms. In *The 22nd Conference on Learning Theory, COLT*, 2009.
- Anqi Mao, Mehryar Mohri, and Yutao Zhong. Cross-entropy loss functions: Theoretical analysis and applications. *arXiv preprint arXiv:2304.07288*, 2023.
- Elliott Mendelson. *Introduction to mathematical logic (5. ed.)*. Chapman and Hall, 2009.
- John Miller, Smitha Milli, and Moritz Hardt. Strategic classification is causal modeling in disguise. In *Proceedings of the 37th International Conference on Machine Learning, ICML*, pages 6917–6926, 2020.

- Smitha Milli, John Miller, Anca D. Dragan, and Moritz Hardt. The social cost of strategic classification. In *Proceedings of the Conference on Fairness, Accountability, and Transparency, FAT**, pages 230–239, 2019.
- Omar Montasser, Steve Hanneke, and Nathan Srebro. VC classes are adversarially robustly learnable, but only improperly. In *Conference on Learning Theory, COLT*, pages 2512–2530, 2019.
- Omar Montasser, Surbhi Goel, Ilias Diakonikolas, and Nathan Srebro. Efficiently learning adversarially robust halfspaces with noise. *arXiv preprint arXiv:2005.07652*, 2020.
- Omar Montasser, Steve Hanneke, and Nathan Srebro. Adversarially robust learning with unknown perturbation sets. In *Conference on Learning Theory, COLT*, pages 3452–3482, 2021.
- Omar Montasser, Steve Hanneke, and Nati Srebro. Adversarially robust learning: A generic minimax optimal learner and characterization. *Advances in Neural Information Processing Systems*, 35:37458–37470, 2022.
- Shay Moran and Amir Yehudayoff. Sample compression schemes for VC classes. *J. ACM*, 63(3):21:1–21:10, 2016.
- Anastasia Pentina and Shai Ben-David. Multi-task and lifelong learning of kernels. In *Algorithmic Learning Theory - 26th International Conference, ALT*, pages 194–208, 2015.
- Anastasia Pentina and Christoph H. Lampert. A pac-bayesian bound for lifelong learning. In *Proceedings of the 31th International Conference on Machine Learning, ICML*, pages 991–999, 2014.
- Anastasia Pentina and Ruth Urner. Lifelong learning with weighted majority votes. In *Advances in Neural Information Processing Systems 29, NIPS*, pages 3612–3620, 2016.
- Juan C. Perdomo, Tijana Zrnic, Celestine Mendler-Dünner, and Moritz Hardt. Performative prediction. In *Proceedings of the 37th International Conference on Machine Learning, ICML*, pages 7599–7609, 2020.
- Vinod Raman, Unique Subedi, and Ambuj Tewari. On proper learnability between average- and worst-case robustness. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

- Kelly Ramsay, Aukosh Jagannath, and Shoja’eddin Chenouri. Concentration of the exponential mechanism and differentially private multivariate medians. *arXiv preprint arXiv:2210.06459*, 2022.
- Ali Shafahi, W Ronny Huang, Christoph Studer, Soheil Feizi, and Tom Goldstein. Are adversarial examples inevitable? In *7th International Conference on Learning Representations (ICLR 2019)*, 2019.
- Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning - From Theory to Algorithms*. Cambridge University Press, 2014a.
- Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning - From Theory to Algorithms*. Cambridge University Press, 2014b.
- Han Shao, Omar Montasser, and Avrim Blum. A theory of pac learnability under transformation invariances. *Advances in Neural Information Processing Systems*, 35:13989–14001, 2022.
- Han Shao, Avrim Blum, and Omar Montasser. Strategic classification under unknown personalized manipulation. In *NeurIPS, 2023*, 2023.
- Yonadav Shavit, Benjamin L. Edelman, and Brian Axelrod. Causal strategic linear regression. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 8676–8686. PMLR, 2020.
- Bernard W. Silverman. *Density Estimation for Statistics and Data Analysis*. Springer, 1986.
- Aleksandra Slavkovic and Roberto Molinari. Perturbed M-estimation: A further investigation of robust statistics for differential privacy. In Alicia L. Carriquiry, Judith M. Tanur, and William F. Eddy, editors, *Statistics in the Public Interest: In Memory of Stephen E. Fienberg*, pages 337–361. Springer, 2022.
- David Soloveichik. Statistical learning of arbitrary computable classifiers. *arXiv preprint arXiv:0806.3537*, 2008.
- Jacob Steinhardt, Pang Wei W Koh, and Percy S Liang. Certified defenses for data poisoning attacks. In *Advances in Neural Information Processing Systems 30*, NeurIPS ’17, pages 3520–3532. Curran Associates, Inc., 2017.

- Jacob Steinhardt, Moses Charikar, and Gregory Valiant. Resilience: A criterion for learning in the presence of arbitrary outliers. In *Proceedings of the 9th Conference on Innovations in Theoretical Computer Science*, ITCS '18, pages 45:1–45:21, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.
- Tom F Sterkenburg. On characterizations of learnability with computable learners. In *Conference on Learning Theory*, pages 3365–3379. PMLR, 2022.
- Ravi Sundaram, Anil Vullikanti, Haifeng Xu, and Fan Yao. Pac-learning for strategic classification. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021*, pages 9978–9988, 2021.
- Ananda Theertha Suresh, Alon Orlitsky, Jayadev Acharya, and Ashkan Jafarpour. Near-optimal-sample estimators for spherical Gaussian mixtures. In *Advances in Neural Information Processing Systems 27*, NIPS '14, pages 1395–1403. Curran Associates, Inc., 2014.
- Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *2nd International Conference on Learning Representations, ICLR*, 2014.
- Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. In *7th International Conference on Learning Representations, ICLR*, 2019a.
- Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. In *International Conference on Learning Representations*, 2019b.
- Stratis Tsirtsis and Manuel Gomez Rodriguez. Decisions, counterfactual explanations and strategic behavior. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems NeurIPS*, 2020.
- John W. Tukey. A survey of sampling from contaminated distributions. *Contributions to Probability and Statistics: Essays in Honor of Harold Hotelling*, pages 448–485, 1960.
- Leslie G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.
- V. N. Vapnik and A. Ya. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability & Its Applications*, 16(2):264–280, 1971a.

- Vladimir Vapnik. *The Nature of Statistical Learning Theory*. Statistics for Engineering and Information Science. Springer, 2000.
- Vladimir Vapnik and Alexey Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. In *Theory of Probability and Its Applications*. Society for Industrial and Applied Mathematics, 1971b.
- Paul Viallard, Eric Guillaume VIDOT, Amaury Habrard, and Emilie Morvant. A pac-bayes analysis of adversarial robustness. *Advances in Neural Information Processing Systems*, 34, 2021.
- M Vidyasagar, S Balaji, and Barbara Hammer. Closure properties of uniform convergence of empirical means and pac learnability under a family of probability measures. *Systems I& amp; control letters*, 2001.
- Yizhen Wang, Somesh Jha, and Kamalika Chaudhuri. Analyzing the robustness of nearest neighbors to adversarial examples. In *Proceedings of the 35th International Conference on Machine Learning, ICML*, pages 5120–5129, 2018.
- Yair Wiener and Ran El-Yaniv. Agnostic pointwise-competitive selective classification. *Journal of Artificial Intelligence Research*, 52:171–201, 2015.
- Yao-Yuan Yang, Cyrus Rashtchian, Hongyang Zhang, Ruslan Salakhutdinov, and Kamalika Chaudhuri. A closer look at accuracy vs. robustness. In *Advances in Neural Information Processing Systems 33, NeurIPS*, 2020.
- Yannis G. Yatracos. Rates of convergence of minimum distance estimators and Kolmogorov’s entropy. *The Annals of Statistics*, 13(2):768–774, 1985.
- Dong Yin, Ramchandran Kannan, and Peter Bartlett. Rademacher complexity for adversarially robust generalization. In *International conference on machine learning*, pages 7085–7094. PMLR, 2019.
- Ming Yuan and Marten H. Wegkamp. Classification methods with reject option based on convex risk minimization. *J. Mach. Learn. Res.*, 11:111–130, 2010.
- Chicheng Zhang and Kamalika Chaudhuri. The extended littlestone’s dimension for learning with mistakes and abstentions. In *Proceedings of the 29th Conference on Learning Theory, COLT*, pages 1584–1616, 2016.
- Hanrui Zhang and Vincent Conitzer. Incentive-aware PAC learning. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI*, pages 5797–5804, 2021.

Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P. Xing, Laurent El Ghaoui, and Michael I. Jordan. Theoretically principled trade-off between robustness and accuracy. In *Proceedings of the 36th International Conference on Machine Learning, ICML*, pages 7472–7482, 2019.