On the Hardness of Robust Classification (Extended Version)*

Pascale Gourdeau PASCALE.GOURDEAU@CS.OX.AC.UK
Varun Kanade VARUNK@CS.OX.AC.UK
Marta Kwiatkowska MARTA.KWIATKOWSKA@CS.OX.AC.UK
James Worrell JAMES.WORRELL@CS.OX.AC.UK
Department of Computer Science, University of Oxford
Parks Road
Oxford, OX1 3QD
UK

Abstract

It is becoming increasingly important to understand the vulnerability of machine learning models to adversarial attacks. In this paper we study the feasibility of adversarially robust learning from the perspective of computational learning theory, considering both sample and computational complexity. In particular, our definition of robust learnability requires polynomial sample complexity. We start with two negative results. We show that no non-trivial concept class can be robustly learned in the distribution-free setting against an adversary who can perturb just a single input bit. We show, moreover, that the class of monotone conjunctions cannot be robustly learned under the uniform distribution against an adversary who can perturb $\omega(\log n)$ input bits. However, we also show that if the adversary is restricted to perturbing $O(\log n)$ bits, then one can robustly learn the class of 1-decision lists (which subsumes monotone conjunctions) with respect to the class of log-Lipschitz distributions. We then extend this result to show learnability of 2-decision lists and monotone k-decision lists in the same distributional and adversarial setting. Finally, we provide a simple proof of the computational hardness of robust learning on the boolean hypercube. Unlike previous results of this nature, our result does not rely on a more restricted model of learning, such as the statistical query model, nor on any hardness assumption other than the existence of an (average-case) hard learning problem in the PAC framework; this allows us to have a clean proof of the reduction, and the assumption is no stronger than assumptions that are used to build cryptographic primitives.

Keywords: learning theory, hardness of learning, decision lists, robustness.

1. Introduction

There has been considerable interest in adversarial machine learning since the seminal work of Szegedy et al. (2013), who coined the term *adversarial example* to denote the result of applying a carefully chosen perturbation that causes a classification error to a previously correctly classified datum. Biggio et al. (2013) independently observed this phenomenon. However, as pointed out by Biggio and Roli (2018), adversarial machine learning has been considered much earlier in the context of spam filtering (Dalvi et al. (2004); Lowd and Meek (2005a,b)). Their survey also distinguished two settings: *evasion attacks*, where an

*. Accepted to the Journal of Machine Learning Research

adversary modifies data at test time, and *poisoning attacks*, where the adversary modifies the training data.[1]

Several different definitions of adversarial learning exist in the literature and, unfortunately, in some instances the same terminology has been used to refer to different notions (for some discussion see e.g., (Dreossi et al. (2019); Diochnos et al. (2018))). Our goal in this paper is to take the most widely-used definitions and consider their implications for robust learning from a statistical and computational viewpoint. For simplicity, we will focus on the setting where the input space is the boolean hypercube $\mathcal{X} = \{0,1\}^n$ and consider the *realizable* setting, i.e., the labels are consistent with a target concept in some concept class.

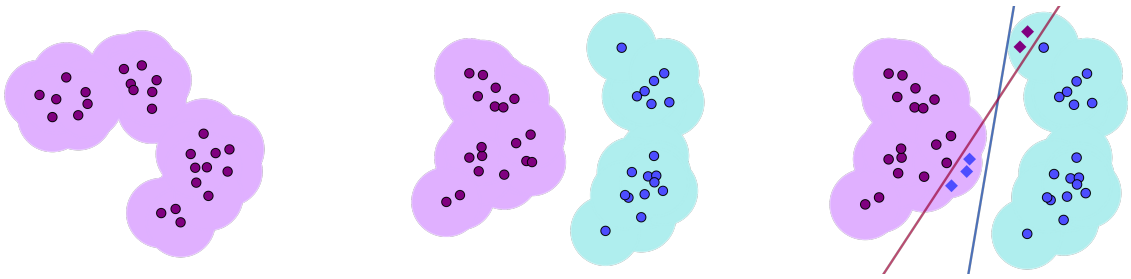
An *adversarial example* is constructed from a *natural example* by adding a perturbation. Typically, the power of the adversary is curtailed by specifying an upper bound on the perturbation under some norm; in our case, the only meaningful norm is the Hamming distance. Then, for us, the *perturbation budget* of an adversary is the number of bits the adversary is allowed to flip. For a point $x \in \mathcal{X}$, let $B_\rho(x)$ denote the Hamming ball of radius ρ around x. Given a distribution D on $\mathcal{X}$, we consider the *adversarial risk* of a hypothesis h with respect to a target concept c and perturbation budget ρ. We focus on two definitions of risk. The *exact in the ball* risk $\mathsf{R}^E_\rho(h,c)$ is the probability $\Pr_{x\sim D}(\exists y \in B_\rho(x), \cdot h(y) \neq c(y))$ that the adversary can perturb a point x drawn from distribution D to a point y such that $h(y) \neq c(y)$. The *constant-in-the-ball* risk $\mathsf{R}^C_\rho(h,c)$ is the probability $\Pr_{x\sim D}(\exists y \in B_\rho(x) \cdot h(y) \neq c(x))$ that the adversary can perturb a point x drawn from distribution D to a point y such that $h(y) \neq c(x)$. These definitions encode two different interpretations of robustness. In the first view, robustness speaks about the fidelity of the hypothesis to the target concept, whereas in the latter view robustness is concerned with the sensitivity of the output of the hypothesis to corruptions of the input. In fact, the latter view of robustness can in some circumstances be in conflict with accuracy in the traditional sense (Tsipras et al. (2019)).

1.1 Overview of Our Contributions

We view our conceptual contributions to be at least as important as the technical results and believe that the issues highlighted in our work will result in more concrete theoretical frameworks being developed to study adversarial learning.

Impossibility of Robust Learning in Distribution-Free PAC Setting: We first consider the question of whether achieving *zero* (or low) robust risk is possible under either of the two definitions. If the *balls* of radius ρ around the data points intersect so that the total region is connected, then unless the target function is constant, it is impossible to achieve $\mathsf{R}^C_\rho(h,c) = 0$ (see Figure 1). In particular, in most cases $\mathsf{R}^C_\rho(c,c) \neq 0$, i.e., even the target concept does not have zero risk with respect to itself. We show that this is the case for extremely simple concept classes such as *dictators* or *parities*. When considering the *exact-in-the-ball* notion of robust learning, we at least have $\mathsf{R}^E_\rho(c,c) = 0$; in particular, any concept class that can be exactly learned can be robustly learned in this sense. However, even in this case we show that no "non-trivial" class of functions can be robustly learned under arbitrary distributions. We highlight that these results show that a polynomial-size

1. For an in-depth review and definitions of different types of attacks, the reader may refer to Biggio and Roli (2018); Dreossi et al. (2019).



(a) (b) (c)

Figure 1: (a) The support of the distribution is such that $\mathsf{R}^C_\rho(h, c) = 0$ can only be achieved if c is constant. (b) The ρ-expansion of the support of the distribution (i.e., the points that differ by at most ρ bits from points in the support of the distribution) and target c admit hypotheses h such that $\mathsf{R}^C_\rho(h, c) = 0$. (c) An example where R^C_ρ and R^E_ρ differ. The red concept is the target, while the blue one is the hypothesis. The dots are the support of the distribution and the shaded regions represent their ρ-expansion. The diamonds represent perturbed inputs which cause $\mathsf{R}^E_\rho > 0$.

sample from the unknown distribution is not sufficient, even if the learning algorithm has arbitrary computational power (in the sense of Turing computability).[2]

Robust Learning of Parities and Decision Lists: Given the impossibility of distribution-free robust learning, we consider robust learning under specific distributions. In Section 2, we discuss the assumptions underlying different notions of robustness and note that, while the constant-in-the-ball risk has been extensively studied in the literature, the exact-in-the-ball risk is less well understood. For this reason, we decide to focus on the latter for the rest of the paper. We first remark that, under the class of log-Lipschitz distributions (which includes the uniform distribution), parity functions are efficiently robustly learnable over $\mathcal{X} = \{0, 1\}^n$. A distribution is said to be α-log-Lipschitz if the logarithm of the probability mass function is $\log(\alpha)$-Lipschitz with respect to the Hamming distance. This means that by changing one bit of the input, the probability mass function can be changed at most by a multiplicative factor α.

We then consider *decision lists* under log-Lipschitz distributions, and show that the class of 1-decision lists is efficiently robustly learnable provided that $\rho = O(\log n)$. We extend this result, with a more elaborate argument, to encompass also the classes of 2-decision lists and monotone k-decision lists, for every fixed k. On the other hand, against a stronger adversary, with budget $\rho = \omega(\log n)$, we show that even the class of monotone conjuctions (a special case of 1-decision lists, and one of the simplest concept classes studied in PAC learning) is not efficiently robustly learnable with polynomial sample complexity.

Our results apply in the setting where the learning algorithm only receives random labeled examples. On the other hand, a more powerful learning algorithm that has access to membership queries can exactly learn monotone conjunctions and as a result can also

2. We do require that any operation performed by the learning algorithm is computable; the results of Bubeck et al. (2018b) imply that an algorithm that can potentially evaluate *uncomputable* functions can always robustly learn using a polynomial-size sample. See the discussion on computational hardness below.

robustly learn with respect to the *exact-in-the-ball* loss. We also note that our positive results use PAC learning algorithms as black boxes, as we implicitly derive bounds for the robust risk in terms of the standard risk. Section 5.4, which expands our decision list result to decision trees, makes the relationship between the standard and robust risks explicit. While we use known PAC learning algorithms as robust learning algorithms, we remark that analyzing their sample complexity greatly differs from standard PAC-learning arguments, as we cannot rely on the VC dimension of a concept class to show its robustness in our setting.

Computational Hardness of Robust PAC Learning: Finally, we consider computational aspects of robust learning. Our focus is on two questions: *computability* and *computational complexity*. Recent work by Bubeck et al. (2018b) provides a result that states that minimizing the robust loss on a polynomial-size sample suffices for robust learning. However, because of the existential quantifier over the ball implicit in the definition of the *exact-in-the-ball* loss, the empirical risk cannot be *computed* as this requires enumeration over the *reals*. Even if one restricted attention to concepts defined over $\mathbb{Q}^n$, computing the loss would be *recursively enumerable*, but not *recursive*. In the case of functions defined over finite instance spaces, such as the boolean hypercube, the loss can be evaluated provided the learning algorithm has access to a membership query oracle; for the *constant-in-the-ball* loss membership queries are not required. For functions defined on $\mathbb{R}^n$ it is unclear how either loss function can be evaluated even if the learner has access to membership queries, since in principle it requires enumerating over the reals. Under strong assumptions of *inductive bias* on the target and hypothesis class, it may be possible to evaluate the loss functions; however, this would have to be handled on a case by case basis. For e.g., properties of the target and hypothesis, such as being Lipschitz or having a large margin, could be used to compute the exact-in-the-ball loss in finite time.

Second, we consider the computational complexity of robust learning. Bubeck et al. (2018a) and Degwekar et al. (2019) have shown that there are concept classes that are hard to robustly learn under cryptographic assumptions, even when robust learning is information-theoretically feasible. Bubeck et al. (2018b) establish super-polynomial lower bounds for robust learning in the *statistical query* framework. We give an alternative simpler proof of hardness, based simply on the assumption that there exist concept classes that are hard to PAC learn in the average case. The reduction transforms a hard PAC-learning problem into a problem that is easy to PAC learn, but that is hard to robustly learn. In particular, our reduction also implies that robust learning is hard even if the learning algorithm is allowed membership queries, provided the concept class that we reduce from is hard to learn using membership queries. Some of the ideas we use parallel those used in the construction by Bubeck et al. (2018a, 2019); however, starting from the average-case hardness of PAC learning allows for a simpler proof. This assumption is no stronger than the cryptographic assumptions used by Bubeck et al. (2018a), as the existence of one-way functions suffices to construct average-case hard to learn concept classes (Goldreich et al., 1986); on the other hand, Blum et al. (1993) have also shown how to construct cryptographic primitives using average-case hardness of PAC learning.[3]

3. It is believed that the existence of worst-case hard to PAC learn concept classes is not sufficient to construct one-way functions (Applebaum et al., 2008).

1.2 Related Work on the Existence of Adversarial Examples

There is a considerable body of work that studies the inevitability of adversarial examples, e.g., Fawzi et al. (2016, 2018b,a); Gilmer et al. (2018); Shafahi et al. (2018). These papers characterize robustness in the sense that a classifier's output on a point should not change if a perturbation of a certain magnitude is applied to it. Among other things, these works study geometrical characteristics of classifiers and statistical characteristics of classification data that lead to adversarial vulnerability. Another line of work considers the robust learnability of certain concept classes. Montasser et al. (2019) show that VC classes are robustly learnable with sample complexity polynomial in the VC dimension and dual VC dimension,[4] and that improper learning is necessary for robust learnability in some cases, in the sense that there does not exist a proper robust learning algorithm for these tasks. Ashtiani et al. (2020) expand on these results by showing that VC classes are properly robustly learnable if their margin class also has finite VC dimension. We note that these results and techniques do not translate to the exact-in-the-ball notion of robust risk, as they rely on inflating a sample S with points in the balls around points in S and giving them the same label as their original instance.

Closer to the present paper are Diochnos et al. (2018); Mahloujifar and Mahmoody (2019); Mahloujifar et al. (2019), which work with the exact-in-the-ball notion of robust risk. In particular, Diochnos et al. (2018) considers the robustness of monotone conjunctions under the uniform distribution on the boolean hypercube for this notion of risk (therein called the *error region* risk[5]). However, Diochnos et al. (2018) does not address the sample and computational complexity of learning: their results rather concern the ability of an adversary to magnify the missclassification error of *any* hypothesis with respect to *any* target function by perturbing the input. For example, they show that an adversary who can perturb $O(\sqrt{n})$ bits can increase the missclassification probability from 0.01 to 1/2. The main tool used in Diochnos et al. (2018) is the isoperimetric inequality for the boolean hypercube, which gives lower bounds on the volume of the expansions of arbitrary subsets. On the other hand, we use the probabilistic method to establish the existence of a single hard-to-learn target concept for any given algorithm with polynomial sample complexity.

Finally, Diochnos et al. (2019) show an exponential lower bound on the sample complexity of robust PAC learning of a wide family of concept classes[6] under Normal Lévy distributions (which include product distributions under the Hamming distance in $\{0,1\}^n$) against all adversaries that can perturb up to $o(n)$ bits. Closer to our results of Section 5, they also show a superpolynomial lower bound in sample complexity against all adversaries that can perturb $\widetilde{O}(\sqrt{n})$ bits. Our paper improves this result in the special case of the uniform distribution: we show that a weaker adversary, who can perturb only $\omega(\log n)$ bits, renders it impossible to robustly learn monotone conjunctions (and any superclass) with polynomial sample complexity. In fact, we show that $\Theta(\log n)$ is indeed the threshold for efficient robust PAC learning in this setting.

4. This gives a general upper bound that is exponential in the VC dimension.
5. They also refer to the *constant-in-the-ball* risk as *corrupted instance* risk, which refers back to the work of Feige et al. (2015) that introduced the wording *corrupted instance*.
6. The classes must be α-*close*, meaning that there must exist two concepts in the class that have (standard) error α.

This paper is the extended version of Gourdeau et al. (2019), which appeared in NeurIPS 2019.

2. Definition of Robust Learning

The notion of robustness can be accommodated within the basic set-up of PAC learning by adapting the definition of risk function. In this section we review two of the main definitions of *robust risk* that have been used in the literature. For concreteness we consider an input space $\mathcal{X} = \{0,1\}^n$ with metric $d : \mathcal{X} \times \mathcal{X} \to \mathbb{N}$, where $d(x,y)$ is the Hamming distance of $x, y \in \mathcal{X}$. Given $x \in \mathcal{X}$, we write $B_\rho(x)$ for the ball $\{y \in \mathcal{X} : d(x,y) \leq \rho\}$ with center x and radius $\rho \geq 0$.

The first definition of robust risk asks that the hypothesis be exactly equal to the target concept in the ball $B_\rho(x)$ of radius ρ around a "test point" $x \in \mathcal{X}$:

Definition 1 *Given respective hypothesis and target functions $h, c : \mathcal{X} \to \{0,1\}$, distribution D on $\mathcal{X}$, and robustness parameter $\rho \geq 0$, we define the "exact-in-the-ball" robust risk of h with respect to c to be*

$$\mathsf{R}^E_\rho(h,c) = \Pr_{x \sim D}\left(\exists z \in B_\rho(x) : h(z) \neq c(z)\right) .$$

While this definition captures a natural notion of robustness, an obvious disadvantage is that evaluating the risk function requires the learner to have knowledge of the target function outside of the training set, e.g., through membership queries. Nonetheless, by considering a learner who has oracle access to the predicate $\exists z \in B_\rho(x) : h(z) \neq c(z)$, we can use the exact-in-the-ball framework to analyze sample complexity and to prove strong lower bounds on the computational complexity of robust learning.

A popular alternative to the exact-in-the-ball risk function in Definition 1 is the following *constant-in-the-ball risk* function:

Definition 2 *Given respective hypothesis and target functions $h, c : \mathcal{X} \to \{0,1\}$, distribution D on $\mathcal{X}$, and robustness parameter $\rho \geq 0$, we define the "constant-in-the-ball" robust risk of h with respect to c as*

$$\mathsf{R}^C_\rho(h,c) = \Pr_{x \sim D}\left(\exists z \in B_\rho(x) : h(z) \neq c(x)\right) \quad .$$

An obvious advantage of the constant-in-the-ball risk over the exact-in-the-ball version is that, in the former, evaluating the loss at point $x \in \mathcal{X}$ requires only knowledge of the correct label of x and the hypothesis h. In particular, this definition can also be carried over to the non-realizable setting, in which there is no target. However, from a foundational point of view the constant-in-the-ball risk has some drawbacks: recall from the previous section that under this definition it is possible to have strictly positive, and even sometimes constant,[7] robust risk in the case that $h = c$. (Let us note in passing that the risk functions R^C_ρ and

7. For example, under the uniform distribution, for $c \in$ MON-CONJ of constant length k, $\mathsf{R}^C_1(c,c) = \frac{k+1}{2^k}$, and in the case of decision lists, any list c of the form $((x_i,0),(x_j,1),\dots)$ satisfies $\mathsf{R}^C_1(c,c) \geq \Pr_{x \sim D}(x_j = 1) = 1/2$. In the case of parity functions, it suffices to flip one bit of the index set to switch the label, so under any distribution $\mathsf{R}^C_\rho(c,c) = 1$ for any $\rho \geq 1$.

R^E_ρ are in general incomparable. Figure 1c gives an example in which $\mathsf{R}^C_\rho = 0$ and $\mathsf{R}^E_\rho > 0$.) Additionally, when we work in the hypercube, or a bounded input space, as ρ becomes larger, we eventually require the function to be constant in the whole space. Essentially, to ρ-robustly learn in the realizable setting, we require concept and distribution pairs to be represented as two sets D_+ and D_- whose ρ-expansions don't intersect, as illustrated in Figures 1a and 1b. We finish by pointing out that, in some cases in the realizable setting, the target c is not the robust risk minimizer for $\rho = 1$: the constant concept is! This is easy to see for parity functions, as $\mathsf{R}^C_1(c, 0) = \mathsf{R}^C_1(c, 1) = 1/2$ under the uniform distribution and $R^C_1(c, c) = 1$. A similar result holds for monotone conjunctions (see Appendix B).

The discussion above, which pertains to the boolean hypercube, makes apparent the fact that the exact-in-the-ball and constant-in-the-ball definitions of robust risk both rely on different distributional and concept class assumptions. The constant-in-the-ball notion of robust risk relies on a strong distributional assumption (for e.g., a margin condition) or on the stability of functions in the concept class. The exact-in-the-ball is more relevant in cases where we cannot assume that the probability mass near the boundary is small, and wish to be correct with respect to the target function. The behavior of the constant-in-ball risk is much better understood than for the exact-in-the-ball risk: most papers we have cited in Section 1.2 have used the former. For this reason, we will work with the latter.

Having settled on a risk function, we now formulate the definition of robust learning. For our purposes a *concept class* is a family $\mathcal{C} = \{\mathcal{C}_n\}_{n\in\mathbb{N}}$, with $\mathcal{C}_n$ a class of functions from $\{0,1\}^n$ to $\{0,1\}$. Likewise, a *distribution class* is a family $\mathcal{D} = \{\mathcal{D}_n\}_{n\in\mathbb{N}}$, with $\mathcal{D}_n$ a set of distributions on $\{0,1\}^n$. Finally, a *robustness function* is a function $\rho : \mathbb{N} \to \mathbb{N}$.

Definition 3 *Fix a function $\rho : \mathbb{N} \to \mathbb{N}$. We say that an algorithm $\mathcal{A}$* efficiently ρ-robustly learns *a concept class $\mathcal{C}$ with respect to distribution class $\mathcal{D}$ if there exists a polynomial $poly(\cdot,\cdot,\cdot,\cdot)$ such that for all $n \in \mathbb{N}$, all target concepts $c \in \mathcal{C}_n$, all distributions $D \in \mathcal{D}_n$, and all accuracy and confidence parameters $\epsilon, \delta > 0$, if $m \geq poly(n, 1/\epsilon, 1/\delta, size(c))$, whenever $\mathcal{A}$ is given access to a sample $S \sim D^m$ labelled according to c, it outputs a polynomially evaluatable function $h : \{0,1\}^n \to \{0,1\}$ such that $\Pr_{S\sim D^m}\left(\mathsf{R}^E_{\rho(n)}(h, c) < \epsilon\right) > 1 - \delta$.*

Note that the definition of robust learning requires polynomial sample complexity and allows improper learning (the hypothesis h need not belong to the concept class $\mathcal{C}_n$).

In the standard PAC framework, a hypothesis h is considered to have zero risk with respect to a target concept c when $\Pr_{x\sim D}(h(x) \neq c(x)) = 0$. We have remarked that exact learnability implies robust learnability; next we give an example of a concept class $\mathcal{C}$ and distribution D such that $\mathcal{C}$ is PAC learnable under D with zero risk and yet cannot be robustly learned under D (regardless of the sample complexity).

Lemma 4 *The class of dictators is not 1-robustly learnable (and thus not robustly learnable for any $\rho \geq 1$) with respect to the robust risk of Definition 1 in the distribution-free setting.*

Proof Let c_1 and c_2 be the dictators on variables x_1 and x_2, respectively. Let D be such that $\Pr_{x\sim D}(x_1 = x_2) = 1$ and $\Pr_{x\sim D}(x_k = 1) = \frac{1}{2}$ for $k \geq 3$. Draw a sample $S \sim D^m$ and label it according to $c \sim U(c_1, c_2)$. By the choice of D, the elements of S will have the same label regardless of whether c_1 or c_2 was picked. However, for $x \sim D$, it suffices to flip any of the

first two bits to cause c_1 and c_2 to disagree on the perturbed input. We can easily show that, for any $h \in \{0,1\}^{\mathcal{X}}$, $\mathsf{R}_1^E(h, c_1) + \mathsf{R}_1^E(h, c_2) \geq \mathsf{R}_1^E(c_1, c_2) = 1$. Then

$$\mathop{\mathbb{E}}_{c \sim U(c_1,c_2)} \mathop{\mathbb{E}}_{S \sim D^m} \left[\mathsf{R}_1^E(h, c)\right] \geq 1/2 \ .$$

We conclude that one of c_1 or c_2 has robust risk at least $1/2$. ■

Note that a PAC learning algorithm with error probability threshold $\varepsilon = 1/3$ will either output c_1 or c_2 and will hence have standard risk zero.

3. No Distribution-Free Robust Learning in $\{0,1\}^n$

In this section, we show that no non-trivial concept class is efficiently 1-robustly learnable in the boolean hypercube. Such a class is thus not efficiently ρ-robustly learnable for any $\rho \geq 1$. Efficient robust learnability then requires access to a more powerful learning model or distributional assumptions.

Let $\mathcal{C}_n$ be a concept class on $\{0,1\}^n$, and define $\mathcal{C} = \bigcup_{n \geq 1} \mathcal{C}_n$. We say that a class of functions is trivial if $\mathcal{C}_n$ has at most two functions, which moreover differ on every point.

Theorem 5 *For any concept class $\mathcal{C}$, $\mathcal{C}$ is efficiently distribution-free robustly learnable iff it is trivial.*

The proof of the theorem relies on the following lemma:

Lemma 6 *Let $c_1, c_2 \in \{0,1\}^{\mathcal{X}}$ and fix a distribution on $\mathcal{X}$. Then for all $h : \{0,1\}^n \to \{0,1\}$*

$$\mathsf{R}_\rho^E(c_1, c_2) \leq \mathsf{R}_\rho^E(h, c_1) + \mathsf{R}_\rho^E(h, c_2) \ .$$

Proof Let $x \in \{0,1\}^n$ be arbitrary, and suppose that c_1 and c_2 differ on some $z \in B_\rho(x)$. Then either $h(z) \neq c_1(z)$ or $h(z) \neq c_2(z)$. The result follows. ■

The idea of the proof of Theorem 5 (which can be found in Appendix C) is a generalization of the proof of Lemma 4 that dictators are not robustly learnable. However, note that we construct a distribution whose support is all of $\mathcal{X}$. It is possible to find two hypotheses c_1 and c_2 and create a distribution such that c_1 and c_2 will with high probability look identical on samples of size polynomial in n but have robust risk $\Omega(1)$ with respect to one another. Since any hypothesis h in $\{0,1\}^{\mathcal{X}}$ will disagree either with c_1 or c_2 on a given point x if $c_1(x) \neq c_2(x)$, by choosing the target hypothesis c at random from c_1 and c_2, we can guarantee that h won't be robust against c with positive probability. Finally, note that an analogous argument can be made for a more general setting (e.g., for the input space $\mathbb{R}^n$).

4. Parity Functions

In light of Theorem 5, we turn our attention to settings where distributional assumptions allow us to efficiently robustly learn certain concept classes. In this section, we show that parity functions are efficiently exactly learnable under log-Lipschitz distributions. As these

distributions have support on the whole input space, it follows that this implies efficient robust learning of parities. Recall that parity functions are of the form $f(x) = \sum_i a_i x_i + b$ (in modulo 2), where $a_i, b \in \{0, 1\}$. The idea to show robust learnability of parity functions is to show that, for a class of α-log-Lipschitz distributions, a proper PAC-learning algorithm can be used as a black box for exact learning.

Theorem 7 PARITY *is exactly learnable under α-log-Lipschitz distributions.*

Proof Consider a proper PAC-learning algorithm $\mathcal{A}$ with sample complexity poly$(\cdot)$ for PARITY (see e.g., Goldberg (2006)). Let $\mathcal{D}$ be a family of α-log-Lipschitz distributions and let $D \in \mathcal{D}$ be arbitrary. Let $\epsilon, \delta > 0$ be the accuracy and confidence parameters, n be the input dimension, and $c = (\sum_i a_i x_i + b) \bmod 2$ be the target concept. For any $h(x) = \sum_i a'_i x_i + b'$, letting $I = \{i \in [n] \mid a_i \neq a'_i\}$, we have that if I is non-empty,

$$\Pr_{x \sim D}(h(x) \neq c(x)) = \Pr_{x \sim D}\left(\sum_{i \in I} a_i x_i + b \neq \sum_{i \in I} a'_i x_i + b'\right) \geq \frac{1}{1+\alpha} \ .$$

This follows from Lemma 32(ii): for some $i \in I$, the marginal of x_i conditioned on the points $\{x_j \mid j \in I \setminus \{i\}\}$ is also α-log-Lipschitz. Then no matter what value the points in $\{x_j \mid j \in I \setminus i\}$ take, we know that the probability that x_i causes a mismatch in parity is bounded below by $1/(1+\alpha)$ by Lemma 32(i). In the case I is empty, but $b \neq b'$, $\Pr_{x \sim D}(h(x) \neq c(x)) = 1$. Then, any proper PAC-learning algorithm [8] with accuracy parameter $\epsilon < 1/(1+\alpha)$ will return c with probability at least $1 - \delta$. ■

Corollary 8 PARITY *is ρ-robustly learnable under α-log-Lipschitz distributions for any ρ.*

5. Decision Lists

From the robust learnability point of view, the previous section's concept classes are not very interesting, since we simply learn them exactly, and thus robustly for any robustness parameter. In this section, we show that certain subclasses of decision lists are robustly (but not necessarily exactly) efficiently learnable for robustness parameter $\rho = O(\log n)$ under log-Lipschitz distributions. Moreover, in Section 5.4, we extend our results to the class of decision trees by characterizing the relationship between the standard and robust risks for concepts in this class.

Decision lists were introduced in Rivest (1987), where they were shown to be efficiently PAC learnable. We denote by k-DL the class of decision lists with conjunctive clauses of size at most k at each decision node. Decision lists generalize formulas in disjunctive normal form (DNF) and conjunctive normal form (CNF): $k\text{-}DNF \cup k\text{-}CNF \subset k\text{-}DL$, where k refers to the number of literals in each clause. Formally, a decision list is a list L of pairs

$$(f_1, v_1), \ldots, (f_r, v_r) \ ,$$

8. E.g., performing Gaussian elimination on the matrix $\mathbf{X}$ of examples and label vector $\mathbf{y}$ and returning a possible solution vector $\mathbf{z} \in \{0,1\}^n$ (i.e., $\mathbf{Xz} = \mathbf{y}$), where $a_i = 1$ if and only if $\mathbf{z}_i = 1$, would be a proper learning algorithm.

where f_j is a term in C_k^n, the set of all conjunctions of size at most k with literals drawn from $\{x_1, \bar{x_1}, \ldots, x_n, \bar{x_n}\}$, v_j is a value in $\{0,1\}$, and f_r is `true`. The output $L(x)$ of L on x is v_j, where j is the least index such that $f_j(x) = 1$. Decision lists can be seen as the logical formulation of special cases of $\texttt{if} - \texttt{then} - \texttt{elif} - \cdots - \texttt{else}$ statements.

We will start with 1-DL, as we will generalize our result to 2-DL and monotone k-DL in Section 5.3 by using some results from the 1-DL case. The next two sections will be dedicated to proving the following theorem.

Theorem 9 *The class 1-DL is efficiently ρ-robustly learnable, i.e. with polynomial sample complexity, under the class of α-log-Lipschitz distributions with robustness threshold $\rho = \Theta(\log n)$.*

In particular, Section 5.1 will show that an adversary with a perturbation budget $\omega(\log n)$ renders efficient robust learning impossible under the uniform distribution and Section 5.2 will show that efficient robust learning of 1-DL is possible against adversaries with perturbation budget $O(\log n)$.

Remark 10 *The robustness threshold $\rho = \Theta(\log n)$ is an artefact of requiring a sample complexity that is polynomial in the input dimension and learning parameters. In general, for a family of α-log-Lipschitz distributions and a requirement $r(n)$ on the robust sample complexity function (that is also satisfied by the standard sample complexity function), we can guarantee "efficient" robust learning as long as $\rho(n) \leq \frac{\log(r(n))}{4(1+\alpha)\log(1+\alpha)}$.*

5.1 Non-Robust Learnability Through Monotone Conjunctions

In contrast to Theorem 5, it turns out that we do not need recourse to "bad" distributions to show that very simple classes of functions are not efficiently robustly learnable for a sufficiently powerful adversary. As we demonstrate in this section, MON-CONJ, the class of monotone conjunctions, is not efficiently robustly learnable *even under the uniform distribution* for robustness parameters that are superlogarithmic in the input dimension. Since monotone conjunctions are a subclass 1-DL, if we can show that they are not robustly learnable for a certain robustness parameter ρ, then we have that 1-DL is not efficiently ρ-robustly learnable as well.

The idea to show that MON-CONJ is not efficiently robustly learnable is in the same vein as the proof of Theorem 5. We first start by proving the following lemma, which gives a lower bound on the robust risk of two disjoint monotone conjunctions.

Lemma 11 *Under the uniform distribution, for any $n \in \mathbb{N}$, disjoint $c_1, c_2 \in$ MON-CONJ of length $3 \leq l \leq n/2$ on $\{0,1\}^n$ and robustness parameter $\rho \geq \lceil l/2 \rceil$, we have that $\mathsf{R}_\rho^E(c_1, c_2)$ is bounded below by a constant that can be made arbitrarily close to $\frac{1}{2}$ as l increases.*

Proof For a hypothesis $c \in$ MON-CONJ, let I_c be the set of variables in c. Let $c_1, c_2 \in \mathcal{C}$ be as in the theorem statement. Then the robust risk $\mathsf{R}_\rho^E(c_1, c_2)$ is bounded below by

$$\Pr_{x \sim D}\left(c_1(x) = 0 \wedge x \text{ has at least } \lfloor l/2 \rfloor \text{ 1's in } I_{c_2}\right) \geq (1 - 2^{-l})/2 \ .$$

■

Now, the following lemma shows that if we choose the length of the conjunctions c_1 and c_2 to be super-logarithmic in n, then, for a sample of size polynomial in n, c_1 and c_2 will agree on S with probability at least $1/2$. The proof can be found in Appendix D.1.

Lemma 12 *For any functions $l(n) = \omega(\log(n))$ and $m(n) = poly(n)$, for any disjoint monotone conjunctions c_1, c_2 such that $|I_{c_1}| = |I_{c_2}| = l(n)$, there exists n_0 such that for all $n \geq n_0$, a sample S of size $m(n)$ sampled i.i.d. from D will have that $c_1(x) = c_2(x) = 0$ for all $x \in S$ with probability at least $1/2$.*

We are now ready to prove our main result of the section.

Theorem 13 *MON-CONJ is not efficiently ρ-robustly learnable for $\rho(n) = \omega(\log(n))$ under the uniform distribution.*

Proof Fix any algorithm $\mathcal{A}$ for learning MON-CONJ. We will show that the expected robust risk between a randomly chosen target function and any hypothesis returned by $\mathcal{A}$ is bounded below by a constant. Fix a function $\text{poly}(\cdot,\cdot,\cdot,\cdot,\cdot)$, and note that, since $\text{size}(c)$ and ρ are both at most n, we can simply consider a function $\text{poly}(\cdot,\cdot,\cdot)$ in the variables $1/\epsilon$, and $1/\delta, n$ instead. Let $\delta = 1/2$, and fix a function $l(n) = \omega(\log(n))$ that satisfies $l(n) \leq n/2$, and let $\rho(n) = l(n)/2$ (n is not yet fixed). Let n_0 be as in Lemma 12, where $m(n)$ is the fixed sample complexity function. Then Equation 7 in the proof of Lemma 12, which can be found in Appendix D.1, holds for all $n \geq n_0$.

Now, let D be the uniform distribution on $\{0,1\}^n$ for $n \geq \max(n_0, 3)$, and choose c_1, c_2 as in Lemma 11. Note that $\mathsf{R}_\rho^E(c_1, c_2) > \frac{5}{12}$ by the choice of n. Pick the target function c uniformly at random between c_1 and c_2, and label $S \sim D^m$ with c, where $m = \text{poly}(1/\epsilon, 1/\delta, n)$. By Lemma 12, c_1 and c_2 agree with the labeling of S (which implies that all the points have label 0) with probability at least $\frac{1}{2}$ over the choice of S.

Define the following three events for $S \sim D^m$:

$$\mathcal{E} : \ c_{1|S} = c_{2|S} \ , \ \ \mathcal{E}_{c_1} : \ c = c_1 \ , \ \ \mathcal{E}_{c_2} : \ c = c_2 \ .$$

Then, by Lemmas 12 and 6,

$$\begin{aligned}
\mathop{\mathbb{E}}_{c,S}\left[\mathsf{R}_\rho^E(\mathcal{A}(S), c)\right] &\geq \Pr_{c,S}(\mathcal{E}) \mathop{\mathbb{E}}_{c,S}\left[\mathsf{R}_\rho^E(\mathcal{A}(S), c) \mid \mathcal{E}\right] \\
&> \frac{1}{2}\left(\Pr_{c,S}(\mathcal{E}_{c_1}) \mathop{\mathbb{E}}_{S}\left[\mathsf{R}_\rho^E(\mathcal{A}(S), c) \mid \mathcal{E} \cap \mathcal{E}_{c_1}\right] + \Pr_{c,S}(\mathcal{E}_{c_2}) \mathop{\mathbb{E}}_{S}\left[\mathsf{R}_\rho^E(\mathcal{A}(S), c) \mid \mathcal{E} \cap \mathcal{E}_{c_2}\right]\right) \\
&= \frac{1}{4} \mathop{\mathbb{E}}_{S}\left[\mathsf{R}_\rho^E(\mathcal{A}(S), c_1) + \mathsf{R}_\rho^E(\mathcal{A}(S), c_2) \mid \mathcal{E}\right] \\
&\geq \frac{1}{4} \mathop{\mathbb{E}}_{S}\left[\mathsf{R}_\rho^E(c_2, c_1)\right] \\
&= \frac{5}{48} \ .
\end{aligned}$$

■

5.2 Robust Learnability Against a Logarithmically-Bounded Adversary

We show that it is possible to efficiently robustly learn 1-DL if the class of distributions is α-log-Lipschitz, i.e., there exists a universal constant $\alpha \geq 1$ such that for all $n \in \mathbb{N}$, all distributions D on $\{0,1\}^n$ and for all input points $x, x' \in \{0,1\}^n$, if $d_H(x,x') = 1$, then $|\log(D(x)) - \log(D(x'))| \leq \log(\alpha)$ (see Appendix A.3 for further details and useful facts).

Theorem 14 *Let $\mathcal{D} = \{\mathcal{D}_n\}_{n\in\mathbb{N}}$, where $\mathcal{D}_n$ is a set of α-log-Lipschitz distributions on $\{0,1\}^n$ for all $n \in \mathbb{N}$. Then the class of 1-decision lists is ρ-robustly learnable with respect to $\mathcal{D}$ for robustness function $\rho(n) = O(\log n)$.*

This combined with Theorem 13 shows that $\rho(n) = \Theta(\log(n))$ is essentially the threshold for efficient robust learnability of the class 1-DL. To prove this result, we first need the following definitions and lemmas, whose proofs can be found in Appendix D.

Definition 15 *Given a 1-decision list $c = ((l_1, v_1), \ldots, (l_r, v_r))$ and $x \in \mathcal{X}$, we say that x* activates node $i \in \{1, \ldots, r\}$ *in c if $x \models l_i$ and $x \not\models l_j$ for all j such that $1 \leq j < i$.*

The following definition will play a role in our analysis of 1-decision lists.

Definition 16 *Let c and h be decision lists. Given $d \in \mathbb{N}$, we say that h is* consistent with c up to depth d, *denoted $c =_d h$, if $c(x) = h(x)$ for all $x \in \mathcal{X}$ such that the nodes in c and h respectively activated by x have level at most d.*

Note that, given a 1-decision list $f = ((l_1, v_1), \ldots, (l_r, v_r))$, we can assume without loss of generality that f is in a minimal representation, namely that

(i) A literal l only appears once in the list (otherwise we can remove all occurrences of l except the first one without changing the output of the list),

(ii) There does not exist $1 \leq i < j \leq d$ such that $l_i = \bar{l_j}$, as otherwise it is impossible to go past l_j in the list (note that if there exists $1 \leq i < d$ such that $l_d = \bar{l_i}$, we can simply set l_d to true).

We will henceforth assume that all decision lists are in their minimal representation.

Now, under log-Lipschitz distributions, if two 1-decision lists have an error below a certain threshold, they must be consistent up to a certain depth.

Lemma 17 *Let $h, c \in$ 1-DL and let D be an α-log-Lipschitz distribution. If $\Pr_{x\sim D}(h(x) \neq c(x)) < (1+\alpha)^{-2d}$, then $c =_d h$.*

Proof We will show the contrapositive. Let $c = ((l_1, v_1), \ldots, (l_r, v_r))$ and $h = ((l'_1, v'_1), \ldots, (l'_s, v'_s))$ be 1-decision lists. Let $c \neq_d h$, meaning that there exists $x \in \mathcal{X}$ such that x activates node i_0 in c and node i_1 in h such that $v_{i_0} \neq v'_{i_1}$. In particular, the following must hold

$$\begin{aligned} &x \models \neg l_i \qquad 1 \leq i < i_0 \ , \\ &x \models \neg l'_i \qquad 1 \leq i < i_1 \ , \\ &x \models l_{i_0} \wedge l_{i_1} \ . \end{aligned}$$

By Lemma 32, the probability of drawing such an x is at least $(1+\alpha)^{-i_c-i_h} \geq (1+\alpha)^{-2d}$. ■

The next step in the argument is to derive an upper bound on the robust loss $\mathsf{R}^E_\rho(c,h)$ under the condition that $c =_d h$. To this end, the key technical lemma is as follows:

Lemma 18 *Let D be an α-log-Lipschitz distribution on the n-dimensional boolean hypercube and let φ be a conjunction of d literals. Set $\eta = \frac{1}{1+\alpha}$. Then for all $0 < \varepsilon < 1/2$, if $d \geq \max\left\{\frac{4}{\eta^2}\log\left(\frac{1}{\varepsilon}\right), \frac{2\rho}{\eta}\right\}$, then $\Pr_{x\sim D}\left((\exists y \in B_\rho(x) \cdot y \models \varphi)\right) \leq \varepsilon$.*

Proof Write $\varphi = \ell_1 \wedge \cdots \wedge \ell_d$. Draw a point $x \sim D$ from distribution D. Let $X_1, \ldots, X_d \in \{0,1\}$ be indicator random variables, respectively denoting whether x satisfies the literals $\ell_1, \ldots, \ell_d$. Note that we do not assume the X_i's to be independent from each other. Writing $Y := \sum_{i=1}^d X_i$, our goal is to show that $\Pr_{x\sim D}(Y + \rho \geq d) \leq \varepsilon$.

Let D_i be the marginal distribution of X_i conditioned on $X_1, \ldots, X_{i-1}$. This distribution is also α-log-Lipschitz by Lemma 32, and hence,

$$\Pr_{X_i\sim D_i}(X_i = 1) \leq 1 - \eta \ . \tag{1}$$

Since we are interested in the random variable Y representing the number of 1's in $X_1, \ldots, X_d$, we define the random variables $Z_1, \ldots, Z_d$ as follows:

$$Z_k = \left(\sum_{i=1}^k X_i\right) - k(1-\eta) \ ,$$

with the convention that $Z_0 = 0$. The sequence $Z_1, \ldots, Z_d$ is a supermartingale with respect to $X_1, \ldots, X_d$:

$$\begin{aligned}
\mathbb{E}\left[Z_{k+1} \mid X_1, \ldots, X_k\right] &= \mathbb{E}\left[Z_k + X_{k+1} - (1-\eta) \mid X_1, \ldots, X_k\right] \\
&= Z_k + \Pr\left(X'_{k+1} = 1 \mid X_1, \ldots, X_k\right) - (1-\eta) \\
&\leq Z_k \ . && \text{(by Equation 1)}
\end{aligned}$$

Now, note that all Z_k's satisfy $|Z_{k+1} - Z_k| \leq 1$, and that $Z_d = Y - d(1-\eta)$. We can thus apply the Azuma-Hoeffding (A.H.) Inequality to get

$$\begin{aligned}
\Pr\left(Y \geq d - \rho\right) &\leq \Pr\left(Y \geq d(1-\eta) + \sqrt{2\log(2/\varepsilon)d}\right) \\
&= \Pr\left(Z_d - Z_0 \geq \sqrt{2\log(2/\varepsilon)d}\right) \\
&\leq \exp\left(-\frac{\sqrt{2\log(1/\varepsilon)d}^2}{2d}\right) && \text{(A.H.)} \\
&= \varepsilon \ ,
\end{aligned}$$

where the first inequality holds from the given bounds on d and ρ:

$$\begin{aligned} d - \rho &= (1-\eta)d + \frac{\eta d}{2} + \frac{\eta d}{2} - \rho \\ &\geq (1-\eta)d + \frac{\eta d}{2} && \text{(since } \rho \leq \tfrac{\eta d}{2}\text{)} \\ &\geq (1-\eta)d + \sqrt{2\log(1/\varepsilon)d}\ . && \text{(since } d \geq \tfrac{8}{\eta^2}\log(\tfrac{1}{\varepsilon})\text{)} \end{aligned}$$

■

We are now ready to prove that 1-DL is efficiently ρ-robustly learnable for $\rho = O(\log n)$.

Proof [of Theorem 14] Let $\mathcal{A}$ be the (proper) PAC-learning algorithm for 1-DL as in Rivest (1987), with sample complexity $\mathrm{poly}(\cdot)$. Fix the input dimension n, target concept c and distribution $D \in \mathcal{D}_n$, and let $\rho = \log n$. Fix the accuracy parameter $0 < \varepsilon < 1/2$ and confidence parameter $0 < \delta < 1/2$ and let $\eta = 1/(1+\alpha)$. Let $d_0 = \max\left\{\frac{2}{\eta}\log n, \frac{4}{\eta^2}\log\frac{2}{\varepsilon}\right\}$ and let $m = \lceil \mathrm{poly}(n, 1/\delta, \eta^{-2d_0})\rceil$, and note that this is polynomial in n, $1/\delta$ and $1/\varepsilon$.

Let $S \sim D^m$ and $h = \mathcal{A}(S)$. Then $\Pr_{x\sim D}(h(x) \neq c(x)) < \eta^{2d_0}$ with probability at least $1-\delta$. But, by Lemma 17, $\Pr_{x\sim D}(h(x) \neq c(x)) < \eta^{2d_0}$ implies that then $c =_{d_0} h$. Hence $c =_{d_0} h$ with probability at least $1-\delta$. Then, to cause an error, an adversary must activate a node at depth greater than d_0 in either h or c.

We now apply Lemma 18 to show that the probability to activate a node at depth greater than d_0 in c is at most $\varepsilon/2$ (and symmetrically for h), which suffices to conclude that $\mathsf{R}^E_\rho(c,h) < \varepsilon$ with probability at least $1-\delta$. Indeed, writing $c = ((l_1, v_1), \ldots, (l_r, v_r))$ and $\varphi := \neg l_1 \wedge \cdots \wedge \neg l_{d_0}$, observe that

$$\Pr_{x\sim D}((\exists y \in B_\rho(x) \cdot x \models \varphi)) \tag{2}$$

is precisely the probability for the adversary to be able to activate a node at depth $> d_0$ in c. Now to apply Lemma 18 we note that by definition of d_0 we have $d_0 \geq \frac{4}{\eta^2}\log\frac{2}{\varepsilon}$, and, since $\rho = \log n$, we furthermore have $d_0 \geq \frac{2\rho}{\eta}$; thus the lemma implies that Equation 2 is at most $\varepsilon/2$, as we require. ■

5.3 Generalizing from 1-DL to k-DL

This section is concerned with robust learning for k-DL. In the non-adversarial setting, learnability of k-DL can be reduced to learnability of 1-DL. We start by observing that it is apparently not straightforward to apply this reduction in the presence of an adversary.

The classical reduction of learning k-DL to 1-DL involves an embedding $\Phi : \mathcal{X}_n \to \mathcal{X}_{n'}$, for $n' := O(n^k)$, that maps valuations of a collection of n propositional variables to valuations of the collection of k-clauses over these variables. Then, for any function $c : \mathcal{X}_n \to \{0,1\}$ computed by a k-decision list, there is a function $c' : \mathcal{X}_{n'} \to \{0,1\}$ computed by a 1-decision list such that $c' \circ \Phi = c$. On a positive note, the image under Φ of an α-log-Lipschitz

distribution D on $\mathcal{X}_n$ remains log-Lipschitz on $\mathcal{X}_{n'}$, albeit with a slightly larger constant. The problem is that the map Φ is not Lipschitz with respect to the Hamming metric—indeed the image under Φ of two points with Hamming distance $\log n$ in $\mathcal{X}_n$ can have distance $\Omega(n)$ in $\mathcal{X}_{n'}$, which is not logarithmic in the dimension $n' = O(n^k)$.

We therefore take a direct approach to establishing robust learnability of k-DL in this section. The argument follows a similar pattern to the previous section, in particular involving a suitable generalization of Lemma 18. There are new ingredients relating to the hypergraph structure of propositional formulas in conjunctive normal form. These additional factors entail that we can only establish robust learnability (again relative to a $O(\log n)$-bounded adversary) in the case of 2-DL and monotone k-DL.

We start with some background on propositional logic. We regard a formula φ in conjunctive normal form (CNF) as being a set of clauses, with each clause being a set of literals. A k-CNF is a CNF formula where all clauses contain at most k literals. We say that φ is *closed under resolution* if, for any two clauses in φ, their resolvent also belongs to φ. The resolution closure of CNF formula φ, denoted $\mathrm{Res}^*(\varphi)$, is the smallest resolution-closed set of clauses that contains φ.

We can consider a CNF formula as a hypergraph whose vertices are literals and whose hyperedges are clauses. With this identification in mind, define a *cover* of a CNF formula φ is a set of literals C such that every clause in φ contains a literal from C. Define also a *matching* of φ to be a set M of clauses such that no two clauses in M contain the same literal. By a well known result for hypergraphs, for a minimal cover C and maximal matching M we have that $|C| \leq k|M|$, where k is the maximum number of literals in any clause of φ Füredi (1988). Assume now that φ is closed under resolution. We claim that a minimal cover is satisfiable as a set of literals. Suppose for a contradiction that C is a minimal cover that is not satisfiable, i.e., such that $p, \neg p \in C$ for some variable p. By minimality of C, φ contains clauses $\{p\} \cup f$ and $\{\neg p\} \cup f'$ such that C meets neither f nor f'. But then the resolvent $f \cup f'$ is also a clause of φ, and since C is a cover we must have that C meets $f \cup f'$—a contradiction. The claim is established.

Definition 19 *Fix decision lists $c, h \in k$-DL, where $c = ((K_1, v_1), \ldots, (K_r, v_r))$ and $h = ((K'_1, v'_1), \ldots, (K'_s, v'_s))$ and the clauses K_i are conjunctions of k literals. Given $i \in \{1, \ldots, r\}$ and $j \in \{1, \ldots, s\}$, define a CNF formula $\varphi_{i,j}^{(c,h)}$ by writing*

$$\varphi_{i,j}^{(c,h)} := \mathrm{Res}^*((\neg K_1 \wedge \cdots \wedge \neg K_{i-1} \wedge K_i) \wedge (\neg K'_1 \wedge \cdots \wedge \neg K'_{j-1} \wedge K'_j)) .$$

Notice that the formula $\varphi_{i,j}^{(c,h)}$ represents the set of inputs $x \in \mathcal{X}$ that respectively activate vertex i in c and vertex j in h.

Our reliance on the following proposition is the reason that the results in this section apply only to the classes 2-DL and monotone k-DL. [9]

Proposition 20 *Let $c, h \in k$-DL. Then $\varphi_{i,j}^{(c,h)}$ is a k-CNF formula for all i and j in case either $k = 2$ or c and h are both monotone.*

9. It is easy to construct an example of a non-monotone k-CNF where $\varphi_{i,j}^{(c,h)}$ is not a k-CNF: the 3-CNF $\varphi := (x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee x_4 \vee x_5)$ has resolvent $(x_2 \vee x_3 \vee x_4 \vee x_5)$, so $\mathrm{Res}^*(\varphi)$ is a 4-CNF.

Proof If $k = 2$ then $\varphi_{i,j}^{(c,h)}$ is the resolution closure of a 2-CNF formula, which remains a 2-CNF formula. Similarly, if c and h are monotone then $\varphi_{i,j}^{(c,h)}$ is the resolution closure of a k-CNF in which positive literals only appear in singleton clauses. It is clear that the latter is again a k-CNF formula. ■

We now have the following definition, in the spirit of Definition 16.

Definition 21 *Given $s \in \mathbb{N}$, we say that $c, h \in k$-DL are* equivalent to cover-size s, *denoted $c \equiv_s h$, if $c(x) = h(x)$ for all $x \in \mathcal{X}$ and for all nodes i, j such that $\varphi_{i,j}^{(c,h)}$ has a cover of size at most s and $x \models \varphi_{i,j}^{(c,h)}$.*

Next we argue that if the error between c and h is sufficiently small then they are equivalent to a suitably large cover depth.

Lemma 22 *Let D be an α-log-Lipschitz distribution and let c and h be decision lists. If $\Pr_{x \sim D}(h(x) \neq c(x)) < (1+\alpha)^{-s}$ then $c \equiv_s h$.*

Proof We prove the contrapositive. Suppose $c \not\equiv_s h$. By definition, there exist i, j such that $\varphi_{i,j}^{(c,h)}$ has a minimum satisfiable cover C of size at most s and $v_i \neq v_j'$. In particular, we have that $c(x) \neq h(x)$ for all $x \in \mathcal{X}$ that satisfy $\varphi_{i,j}^{(c,h)}$. But the probability that $x \sim D$ satisfies $\varphi_{i,j}^{(c,h)}$ is at least the probability that x satisfies C. Since C is minimal it does not contain complementary literals. Hence, the probability that $x \sim D$ satisfies C is at least $(1+\alpha)^{-s}$ by Lemma 32, which can be found in Appendix A.3. ■

The following is a generalization of Lemma 18.

Lemma 23 *Let φ be a k-CNF formula that has no cover of size s. Let D be an α-log-Lipschitz distribution on valuations for φ. Let $0 < \varepsilon < 1/2$ be arbitrary and set $\eta := \left(\frac{1}{1+\alpha}\right)^k$. If $\frac{s}{k(k+1)} \geq \max\left\{\frac{4}{\eta^2}\log\left(\frac{1}{\varepsilon}\right), \frac{2\rho}{\eta}\right\}$ then $\Pr_{x \sim D}(\exists y \in B_\rho(x) \cdot y \models \varphi) \leq \varepsilon$.*

Proof Since φ has no cover of size s, it has a matching M such that $|M| \geq \frac{s}{k}$. By definition, each literal appears in at most one clause in M, hence, by removing at most a fraction $\frac{k}{k+1}$ of the clauses in M, we can assume without loss of generality that each variable occurs in at most one clause of M and M has cardinality $d := \frac{s}{k(k+1)}$.

Consider the map $\Phi : \mathcal{X}_n \to \mathcal{X}_d$, where $\Phi(x)$ encodes the truth values of the clauses in M under the assignment x. Since the clauses in M are variable-disjoint, Φ is non-expansive under the respective Hamming metrics on $\mathcal{X}_n$ and $\mathcal{X}_d$, meaning that $d_H(\Phi(x), \Phi(y)) \leq d_H(x, y)$ for all $x, y \in \mathcal{X}_n$. Thus for all $x \in \mathcal{X}_n$,

$$\exists y \in B_\rho(x) \cdot y \models \varphi \implies \mathbf{1} \in B_\rho(\Phi(x)).$$

It will suffice to show that the probability over $x \sim D$ that the right-hand condition of the above implication holds true is at most ε.

Define a distribution D' on $\mathcal{X}_d$ by $D'(y) := \sum_{x \in \Phi^{-1}(y)} D(x)$. By Lemma 34, which can be found in Appendix D.2, we have that D' is α'-log-Lipschitz for $\alpha' := (\alpha+1)^k - 1$. We wish to upper-bound the probability over $x' \sim D'$ that $\mathbf{1} \in B_\rho(x')$. For this, we will apply Lemma 18 over the space $\mathcal{X}_d$ with distribution D'. Indeed, our assumptions on η and s entail that $\eta = \frac{1}{1+\alpha'}$ and $d \geq \max\left\{\frac{4}{\eta^2}\log\left(\frac{1}{\varepsilon}\right), \frac{2\rho}{\eta}\right\}$. Thus Lemma 18 gives that $\Pr_{x' \sim D'}(\mathbf{1} \in B_\rho(\Phi(x'))) \leq \varepsilon$. This concludes the proof. ■

We are now ready to prove the main result of the section.

Theorem 24 *Let $\mathcal{D} = \{\mathcal{D}_n\}_{n\in\mathbb{N}}$, where $\mathcal{D}_n$ is a set of α-log-Lipschitz distributions on $\{0,1\}^n$ for all $n \in \mathbb{N}$. Then the classes of 2-decision lists and monotone k-decision lists (for every fixed k) are ρ-robustly learnable with respect to $\mathcal{D}$ for robustness function $\rho(n) = O(\log n)$.*

Proof Let $\mathcal{A}$ be the (proper) PAC-learning algorithm for k-DL as in Rivest (1987), with sample complexity poly$(\cdot)$. Fix the input dimension n, target concept c and distribution $D \in \mathcal{D}_n$, and let $\rho = \log n$. Fix the accuracy parameter $0 < \varepsilon < 1/2$ and confidence parameter $0 < \delta < 1/2$ and let $\eta = 1/(1+\alpha)$. Let $s_0 = k(k+1)\max\left\{\frac{4}{\eta^2}\log\left(\frac{e^4 n^{2k+2}}{16\varepsilon}\right), \frac{2\rho}{\eta}\right\}$, write $m = \lceil \text{poly}(n, 1/\delta, \eta^{-s_0})\rceil$, and note that m is polynomial in n, $1/\delta$ and $1/\varepsilon$.

Let $S \sim D^m$ and $h = \mathcal{A}(S)$. Then $\Pr_{x\sim D}(h(x) \neq c(x)) < \eta^{-s_0}$ with probability at least $1-\delta$. But, by Lemma 22, $\Pr_{x\sim D}(h(x) \neq c(x)) < \eta^{s_0}$ implies that then $c \equiv_{s_0} h$. Hence $c \equiv_{s_0} h$ with probability at least $1-\delta$.

In case $c \equiv_{s_0} h$, an input $x \in \mathcal{X}$ only leads to a classification error if it activates nodes i and j in c and h respectively such that the formula $\varphi_{i,j}^{(c,h)}$ has no cover of cardinality s_0. Fix i and j such that $\varphi_{i,j}^{(c,h)}$ has no cover of cardinality s_0. Now $\varphi_{i,j}^{(c,d)}$ is a k-CNF formula by Proposition 20. Hence the probability that a ρ-bounded adversary can make $\varphi_{i,j}^{(c,d)}$ true is at most $\frac{16\varepsilon}{e^4 n^{2k+2}}$ by Lemma 23. Taking a union bound over all possible choices of i and j (there are $\sum_{i=1}^k \binom{n}{k} \leq k\left(\frac{en}{k}\right)^k$ possible clauses in k-decision lists, which gives us a crude estimate of $k^2\left(\frac{en}{k}\right)^{2k} \leq \frac{e^4 n^{2k+2}}{16}$ choices of i and j) we conclude that $\mathsf{R}_\rho^E(h,c) < \varepsilon$.

■

5.4 Decision Trees

In this section, we show that, under α-log-Lipschitz distributions, for any two decision trees and perturbation budget $\rho(n) = O(\log n)$, the ρ-robust risk is bounded above by a polynomial in the number n of propositional variables, the combined size m of the trees, and their standard risk. This result makes explicit the relationship between both notions of risk, while it was implicitly derived for decision lists in the previous section.

Despite the fact that it is not known whether the class of decision trees is PAC-learnable, relating the standard and robust risks for this class is still of interest if we can show that a small enough standard risk only incurs a polynomial blowup in the robust risk. This

could be particularly compelling in the local membership query model of Awasthi et al. (2013), where an algorithm can request labels for points that are $O(\log(n))$ bits away from a point in the training sample. The authors showed that, in this framework, the class of polynomial-sized decision trees is learnable (in polynomial time) under product distributions using $O(\log(n))$-local membership queries. Moreover, O'Donnell and Servedio (2007) show that monotone decision trees are PAC learnable under the uniform distribution, so our result holds in this setting as well.

Terminology. A decision tree c over n propositional variables is a finite binary tree whose internal nodes are labeled by elements of the set $\{1, \dots, n\}$ and whose leaves are labeled either 0 or 1. The depth of a leaf is the number of internal nodes of the tree in the (unique) path from the root to the given leaf. An input $x \in \mathcal{X} = \{0,1\}^n$ determines a path through such a tree, starting at the root, as follows: at an internal node with label i descend to the left child if $x_i = 0$ and descend to the right child if $x_i = 1$. We say that $x \in \mathcal{X}$ *activates a given leaf node* if the path determined by x leads to the given leaf. In this way a decision tree c determines a function $c : \mathcal{X} \to \{0,1\}$, where $c(x)$ is the label of the leaf activated by x.

Given two decision trees c, h, both over n propositional variables, and given $d \in \mathbb{N}$, we write $c =_d h$ if for all $x \in \mathcal{X}$ that activates leaves of depth at most d in both c and h, we have $c(x) = h(x)$. In the same vein as Lemma 17, given $d \in \mathbb{N}$ we have that $c =_d h$ provided that $\Pr_{x \sim D}(h(x) \neq c(x))$ is sufficiently small.

Lemma 25 *Let D be a α-log-Lipschitz distribution. If $\Pr_{x \sim D}(h(x) \neq c(x)) < (1+\alpha)^{-2d}$ then $c =_d h$.*

We omit the proof of Lemma 25, which follows that of Lemma 17 *mutatis mutandis*.

We can now bound the robust risk between decision trees as a polynomial in the of the number of propositional variables, the log-Lipschitz constant, their combined size, and their standard risk.

Theorem 26 *Let c and h be two decision trees on n propositional variables with at most m nodes in total. Let D be an α-log-Lipschitz distribution on $\mathcal{X}_n$ and $\rho = \log n$. There is a fixed polynomial $\mathrm{poly}(\cdot,\cdot,\cdot)$ such that for all $0 < \varepsilon < \frac{1}{2}$, if $\Pr_{x \sim D}(h(x) \neq c(x)) < \mathrm{poly}(\frac{1}{m}, \frac{1}{n}, \varepsilon)$, then $\mathsf{R}^E_\rho(c,h) < \varepsilon$.*

Proof Write $d := \max\left\{\frac{4}{\eta^2}\log\left(\frac{m}{\varepsilon}\right), \frac{2\rho}{\eta}\right\}$ and define $\mathrm{poly}(\frac{1}{m}, \frac{1}{n}, \varepsilon) := (1+\alpha)^{-d}$.

The assumption that $\Pr_{x \sim D}(h(x) \neq c(x)) < (1+\alpha)^{-d}$ implies that c and h are consistent to depth d. This means that $c(x) \neq h(x)$ only on those inputs $x \in \mathcal{X}$ that activate some leaf node of depth strictly greater than d, either in c or h. By Lemma 18, for each such node the probability that a ρ-bounded adversary can activate the node by perturbing the bits of a randomly generated input $x \sim D$ is at most $\frac{\varepsilon}{m}$. Taking a union bound over the nodes of depth $> d$ (there are at most m of them), we conclude that $\mathsf{R}^E_\rho(h,c) \leq \varepsilon$. ■

6. Computational Hardness of Robust Learning

In this section, we establish that the computational hardness of PAC-learning a concept class $\mathcal{C}$ with respect to a distribution class $\mathcal{D}$ implies the computational hardness of robustly learning a family of concept-distribution pairs from a related class $\mathcal{C}'$ and a restricted class of distributions $\mathcal{D}'$, which are themselves computationally easy to non-robustly learn. This is essentially a version of the main result of Bubeck et al. (2018b). Our proof also uses the Bubeck et al. (2018b) trick of encoding a point's label in the input for the robust learning problem. Interestingly, our proof does not rely on any assumption other than the existence of an average-case hard learning problem in the PAC framework and is valid under *both* notions of robust risk (cf. Definitions 1 and 2).

Construction of $\mathcal{C}'$ and $\mathcal{D}'[c']$. Suppose we are given $\mathcal{C} = \{\mathcal{C}_n\}_{n\in\mathbb{N}}$ and $\mathcal{D} = \{\mathcal{D}_n\}_{n\in\mathbb{N}}$ with $\mathcal{C}_n$ and $\mathcal{D}_n$ defined on $\mathcal{X}_n = \{0,1\}^n$. Given $k \in \mathbb{N}$, we define the family of concept-distribution pairs $\mathcal{P}^k_{\mathcal{C},\mathcal{D}} = \{(c', D') \mid c' \in \mathcal{C}', D' \in \mathcal{D}'[c']\}$,[10] where $\mathcal{C}'$ and $\mathcal{D}'$ are defined in the following. First define $\mathcal{C}' = \{\mathcal{C}'_{(k,n)}\}_{k,n\in\mathbb{N}}$ on $\mathcal{X}'_{k,n} = \{0,1\}^{(2k+1)n+1}$ as follows. Let $\mathrm{maj}_k : \mathcal{X}'_{k,n} \to \mathcal{X}_n$ be the function that yields $x \in \mathcal{X}_n$ obtained by taking a majority vote on each of the n consecutive blocks of $2k+1$ bits and ignoring the last bit. We define $\mathcal{C}'_{(k,n)} = \left\{c \circ \mathrm{maj}_{2k+1} \mid c \in \mathcal{C}_n\right\}$. Let $\varphi_{k,c} : \mathcal{X}_n \to \mathcal{X}'_{k,n}$ be defined as

$$\varphi_{k,c}(x) := \underbrace{x_1 \dots x_1 x_2 \dots x_{n-1} x_n \dots x_n}_{2k+1 \text{ copies of each } x_i} c(x) \ , \quad \varphi_{k,c}(S) := \{\varphi_{k,c}(x) \mid x \in S\} \ ,$$

for $x = x_1 x_2 \dots x_n \in \mathcal{X}_n$ and $S \subseteq \mathcal{X}_n$. This definition implies that if $c' = c \circ \mathrm{maj}_{2k+1}$, then $c(x) = c'(\varphi_{k,c}(x))$ for every $x \in \mathcal{X}_n$. For a concept $c \in \mathcal{C}_n$ and the associated $c' = c \circ \mathrm{maj}_{2k+1} \in \mathcal{C}'_{(k,n)}$, each $D \in \mathcal{D}_n$ induces a distribution $D' \in \mathcal{D}'[c']$, where $D'(z) = D(x)$ if $z = \varphi_{k,c}(x)$, and $D'(z) = 0$ otherwise.

This set up allows us to see that any algorithm (computationally efficient or not) for learning $\mathcal{C}_n$ with respect to $\mathcal{D}_n$ yields an algorithm for learning the family of concept-distribution pairs $\mathcal{P}^k_{\mathcal{C},\mathcal{D}}$. Furthermore, any such algorithm is also robust to any adversary that flips at most k bits; this is because flipping k bits cannot change the majority on any of the n blocks of $2k+1$ bits for any point in the support of D'. As we show below in Theorem 28, any *efficient* robust learning algorithm for learning $\mathcal{P}^k_{\mathcal{C},\mathcal{D}}$ also yields an efficient algorithm for learning $\mathcal{C}_n$ with respect to $\mathcal{D}_n$. Thus, choosing $\mathcal{C}_n$ to be a class that is hard to learn computationally, but easy to learn statistically, implies the *hardness* or robustly learning the corresponding class, $\mathcal{P}^k_{\mathcal{C},\mathcal{D}}$, of concept-distribution pairs computationally, but not statistically. On the other hand, it is easy to learn $\mathcal{P}^k_{\mathcal{C},\mathcal{D}}$ non-robustly, by simply outputting the *last* bit of any input which is always equal to the target label. We remark that the last part of the argument is identical to the construction of Bubeck et al. (2018b).

Before proving the main result, let us first prove the following proposition. This shows that the family of concept-distribution pairs, $\mathcal{P}^k_{\mathcal{C},\mathcal{D}}$, constructed above is *statistically* efficiently learnable.

10. This is analogous to the family of pairs of distributions (D_0, D_1) used by Bubeck et al. (2018b).

Proposition 27 *For any concept class $\mathcal{C}$ and distribution class $\mathcal{D}$, the class of concept-distribution pairs $\mathcal{P}^k_{\mathcal{C},\mathcal{D}} = \{(c', D') \mid c' \in C', D' \in \mathcal{D}'[c']\}$ as constructed above can be k-robustly (under either notion of risk) learned using $O\left(\frac{1}{\epsilon}\left(\log|\mathcal{C}_n| + \log\frac{1}{\delta}\right)\right)$ examples.*

Proof Let (c', D') be the target concept-distribution pair. Suppose that $c' \in \mathcal{C}'_{(k,n)}$, then let $c \in C_n$ be the (unique) concept that produced c' and let $D \in D_n$ be the (unique) distribution that generated D'. Given a random example $(x', c'(x'))$ where $x' \in \mathcal{X}'_{(k,n)}$, the corresponding example $(x, c(x))$ with $x \in \mathcal{X}_n$ can be easily and efficiently constructed. Thus, we have an example oracle for learning c under the distribution D.

Then note that, since $\mathcal{C}_n$ is finite, we can use PAC-learning sample bounds for the realizable setting (see for example Mohri et al. (2012)) to get that the sample complexity of learning $\mathcal{C}_n$ is $O\left(\frac{1}{\epsilon}(\log|\mathcal{C}_n| + \log\frac{1}{\delta})\right)$. Now, if we have PAC-learned $\mathcal{C}_n$ with respect to $\mathcal{D}_n$, and h is the hypothesis returned on a sample labeled according to a target concept $c \in \mathcal{C}_n$, we can compose it with the function maj_k to get a hypothesis h' for which any perturbation of at most k bits of $x' \sim D'$ (where D' is the distribution induced by the target concept c and distribution D) will not change $h'(x')$. Clearly, this produces a hypothesis that is k-robust for either notion of robust risk. ■

We now prove the main result of this section.

Theorem 28 *For any concept class $\mathcal{C}_n$, family of distributions $\mathcal{D}_n$ over $\{0,1\}^n$ and $k \in \mathbb{N}$, there exists a family of concept-distribution pairs $\{(c', D') \mid c' \in \mathcal{C}', D' \in \mathcal{D}'[c']\}$, such that (i) this concept-distribution pairs family is* efficiently *PAC learnable and (ii)* efficient *k-robust learnability of this concept-distribution pairs family under either of the robust risk functions R^C_k or R^E_k implies* efficient *PAC-learnability of $\mathcal{C}_n$ with respect to $\mathcal{D}_n$.*

Proof Given $\mathcal{C}_n$ and $\mathcal{D}_n$, let $\mathcal{P}^k_{\mathcal{C},\mathcal{D}} = \{(c', D') \mid c' \in \mathcal{C}'_{(k,n)}, D' \in \mathcal{D}'[c']\}$ be constructed as above.

For part (i), simply output the last bit of a given input, which is always equal to the target label.

For part (ii), suppose that we are given an algorithm $\mathcal{A}'$ to *computationally efficiently* k-robustly learn $\mathcal{P}$ with sample complexity $m = m(n, \epsilon, \delta)$.

Let $\epsilon, \delta > 0$ be arbitrary and $c \in \mathcal{C}_n$ be an arbitrary target concept and let $c' \in \mathcal{C}'_{(k,n)}$ be such that $c' = c \circ \mathrm{maj}_{2k+1}$. Let $D \in \mathcal{D}_n$ be a distribution on $\mathcal{X}_n$, and let $D' \in \mathcal{D}'_{c'}$ be its induced distribution on $\mathcal{X}'_{k,n}$ using the function $\varphi_{k,c}$.

A PAC-learning algorithm for $\mathcal{C}_n$ is as follows. Draw a sample $S \sim D^m$ and let $S' = \varphi_{k,c}(S)$. Note that this simulates a sample $S' \sim D'^m$, and that c' will give the same label to all points in the Hamming k-ball centered at x' for any x' in the support of D'.

Since $\mathcal{A}'$ k-robustly learns the concept-distribution pairs $\mathcal{P}^k_{\mathcal{C},\mathcal{D}}$, with probability at least $1-\delta$ over S', for any $x \sim D$, we have that h' will output a value different of $c(x)$ on the point $\varphi_{k,\mathbf{0}}(x)$ with probability at most ϵ, where $\mathbf{0}$ represents the function that outputs 0 on every $x \in X_n$. Thus, we may simply output $h = h' \circ \varphi_{k,\mathbf{0}}$, and we have an algorithm to PAC-learn $\mathcal{C}_n$ with respect to the distribution family $\mathcal{D}_n$. Clearly, all required computations in this construction can be done in polynomial time and hence computational efficiency is preserved. ■

Remark 29 *One of the parameters of interest is the ratio k/n, i.e., the fraction of adversarial corruption that any algorithm may tolerate. The above construction shows a computational vs statistical separation for an adversary that can corrupt $\Theta(1/n)$ fraction of the bits. This can easily be boosted to show such a separation when the adversary can corrupt $\Theta(1)$ fraction of the bits. This is achieved by using error correcting codes such as those introduced by Guruswami and Indyk (2001). Since this use of error correcting codes is identical to that in the recent work of Degwekar et al. (2019), we omit the detailed description.*

7. Conclusion

We have studied robust learnability from a computational learning theory perspective and have shown that efficient robust learning can be hard—even in very natural and apparently straightforward settings. Rather straightforwardly, classes that can be *exactly* learned can also be *robustly* learned and we show that the class of parities falls into this category under the class of log-Lipschitz distributions. We also give a tight characterization of the strength of an adversary to prevent robust learning of monotone decision lists, again under certain distributional assumptions. We moreover show that, against a logarithmically-bounded adversary, the robust risk between two decision trees is polynomial in their size and standard risk. Lastly, we have provided a simpler proof of the previously established result of the computational hardness of robust learning.

An interesting avenue for future work is to see whether our positive robust learning results can be extended to other classes of functions. Another interesting line of inquiry is to see whether efficient robust learning can still be guaranteed under less stringent distributional assumptions. Indeed, we have shown that efficient robust learning is impossible in the distribution-free setting except for trivial concept classes, while it is possible to efficiently robustly learn certain concept classes against a logarithmically-bounded adversary under the uniform distribution or log-Lipschitz distributions. The intermediate picture remains unknown, and positive results could lead to novel algorithms, e.g., ones that rely on (local) membership queries on specific parts of the input space. Finally, given that the classes we have studied so far are all efficiently robustly learnable against a logarithmically-bounded adversary under the uniform distribution, an open problem is to determine whether this holds for all efficiently PAC-learnable concept classes, or if there exists such a concept class with robustness threshold $o(\log n)$ under the uniform distribution.

In light of our results, it seems to us that more thought needs to be put into what we want out of robust learning in terms of computational efficiency and sample complexity, which will inform our choice of risk functions. Indeed, while robust learning definitions that have appeared in prior work initially seem natural and reasonable, their inadequacies surface when viewed under the lens of computational learning theory. Given our negative results in the context of the current robustness models, one may surmise that requiring a classifier to be correct in an entire ball near a point is asking for too much. Under such a requirement, we can only solve "easy problems" with strong distributional assumptions. Nevertheless, it may still be of interest to study these notions of robust learning in different learning models, e.g., where one has access to membership queries.

Acknowledgments

Pascale Gourdeau was funded by the Clarendon Fund and the Natural Sciences and Engineering Research Council of Canada (NSERC). Varun Kanade was supported in part by the Alan Turing Institute under the EPSRC grant EP/N510129/1. Marta Kwiatkowska has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 834115, project FUN2MODEL). James Worrell was funded by EPSRC Fellowship EP/N008197/1.

A. Preliminaries

A.1 The PAC framework

We study the problem of robust classification in the realizable setting and where the input space is the boolean cube $\mathcal{X}_n = \{0,1\}^n$. For clarity, we first recall the definition of the PAC learning framework from Valiant (1984).

Definition 30 (PAC Learning) *Let $\mathcal{C}_n$ be a concept class over $\mathcal{X}_n$ and let $\mathcal{C} = \bigcup_{n\in\mathbb{N}} \mathcal{C}_n$. We say that $\mathcal{C}$ is* PAC learnable using hypothesis class $\mathcal{H}$ *and sample complexity function $p(\cdot,\cdot,\cdot,\cdot)$ if there exists an algorithm $\mathcal{A}$ that satisfies the following: for all $n \in \mathbb{N}$, for every $c \in \mathcal{C}_n$, for every D over $\mathcal{X}_n$, for every $0 < \epsilon < 1/2$ and $0 < \delta < 1/2$, if whenever $\mathcal{A}$ is given access to $m \geq p(n, 1/\epsilon, 1/\delta, \mathit{size}(c))$ examples drawn i.i.d. from D and labeled with c, $\mathcal{A}$ outputs a polynomially evaluatable $h \in \mathcal{H}$ such that with probability at least $1-\delta$,*

$$\Pr_{x\sim D}\left(c(x) \neq h(x)\right) \leq \epsilon \ .$$

We say that $\mathcal{C}$ is statistically efficiently PAC learnable if p is polynomial in $n, 1/\epsilon$, $1/\delta$ and $\mathit{size}(c)$, and computationally efficiently PAC learnable if $\mathcal{A}$ runs in polynomial time in $n, 1/\epsilon$, $1/\delta$ and $\mathit{size}(c)$.

PAC learning is *distribution-free*, in the sense that no assumptions are made about the distribution from which the data comes from. The setting where $\mathcal{C} = \mathcal{H}$ is called *proper learning*, and *improper learning* otherwise.

A.2 Monotone Conjunctions

A conjunction c over $\{0,1\}^n$ can be represented a set of literals $l_1, \ldots, l_k$, where, for $x \in \mathcal{X}_n$, $c(x) = \bigwedge_{i=1}^{k} l_i$. For example, $c(x) = x_1 \wedge \bar{x_2} \wedge x_5$ is a conjunction. Monotone conjunctions are the subclass of conjunctions where negations are not allowed, i.e., all literals are of the form $l_i = x_j$ for some $j \in [n]$.

The standard PAC learning algorithm to learn monotone conjunctions is as follows. We start with the hypothesis $h(x) = \bigwedge_{i\in I_h} x_i$, where $I_h = [n]$. For each example x in the training sample, we remove i from I_h if $c(x) = 1$ and $x_i = 0$.

When one has access to membership queries, one can easily exactly learn monotone conjunctions over the whole input space: we start with the instance where all bits are 1 (which is always a positive example), and we can test whether each variable is in the target conjunction by setting the corresponding bit to 0 and requesting the label.

A.3 Log-Lipschitz Distributions

Definition 31 *A distribution D on $\{0,1\}^n$ is said to be α-log-Lipschitz if for all input points $x, x' \in \{0,1\}^n$, if $d_H(x,x') = 1$, then $|\log(D(x)) - \log(D(x'))| \leq \log(\alpha)$.*

The intuition behind log-Lipschitz distributions is that points that are close to each other must not have frequencies that greatly differ from each other. Note that, by definition, $D(x) > 0$ for all inputs x. Moreover, the uniform distribution is log-Lipschitz with parameter $\alpha = 1$. Another example of log-Lipschitz distributions is the class of product distributions where the probability of drawing a 0 (or equivalently a 1) at index i is in the interval $\left[\frac{1}{1+\alpha}, \frac{\alpha}{1+\alpha}\right]$. Log-Lipschitz distributions have been studied in Awasthi et al. (2013), and its variants in Feldman and Schulman (2012); Koltun and Papadimitriou (2007).

Log-Lipschitz distributions have the following useful properties, which we will often refer to in our proofs.

Lemma 32 *Let D be an α-log-Lipschitz distribution over $\{0,1\}^n$. Then the following hold:*

i. For $b \in \{0,1\}$, $\frac{1}{1+\alpha} \leq \Pr_{x\sim D}(x_i = b) \leq \frac{\alpha}{1+\alpha}$.

ii. For any $S \subseteq [n]$, the marginal distribution $D_{\bar{S}}$ is α-log-Lipschitz, where $D_{\bar{S}}(y) = \sum_{y' \in \{0,1\}^S} D(yy')$.

iii. For any $S \subseteq [n]$ and for any property π_S that only depends on variables x_S, the marginal with respect to $\bar{S}$ of the conditional distribution $(D|\pi_S)_{\bar{S}}$ is α-log-Lipschitz.

iv. For any $S \subseteq [n]$ and $b_S \in \{0,1\}^S$, we have that $\left(\frac{1}{1+\alpha}\right)^{|S|} \leq \Pr_{x\sim D}(x_i = b) \leq \left(\frac{\alpha}{1+\alpha}\right)^{|S|}$.

Proof To prove (i), fix $i \in [n]$ and $b \in \{0,1\}$ and denote by $x^{\oplus i}$ the result of flipping the i-th bit of x. Note that

$$\Pr_{x\sim D}(x_i = b) = \sum_{\substack{z\in\{0,1\}^n:\\ z_i=b}} D(z) = \sum_{\substack{z\in\{0,1\}^n:\\ z_i=b}} \frac{D(z)}{D(z^{\oplus i})} D(z^{\oplus i}) \leq \alpha \sum_{\substack{z\in\{0,1\}^n:\\ z_i=b}} D(z^{\oplus i}) = \alpha \Pr_{x\sim D}(x_i \neq b) \ .$$

The result follows from solving for $\Pr_{x\sim D}(x_i = b)$.

Without loss of generality, let $\bar{S} = \{1, \ldots, k\}$ for some $k \leq n$. Let $x, x' \in \{0,1\}^{\bar{S}}$ with $d_H(x,x') = 1$.

To prove (ii), let $D_{\bar{S}}$ be the marginal distribution. Then,

$$D_{\bar{S}}(x) = \sum_{y\in\{0,1\}^S} D(xy) = \sum_{y\in\{0,1\}^S} \frac{D(xy)}{D(x'y)} D(x'y) \leq \alpha \sum_{y\in\{0,1\}^S} D(x'y) = \alpha D_{\bar{S}}(x') \ .$$

To prove (iii), denote by X_{π_S} the set of points in $\{0,1\}^S$ satisfying property π_S, and by xX_{π_S} the set of inputs of the form xy, where $y \in X_{\pi_S}$. By a slight abuse of notation, let $D(X_{\pi_S})$ be the probability of drawing a point in $\{0,1\}^n$ that satisfies π_S. Then,

$$D(xX_{\pi_S}) = \sum_{y\in X_{\pi_S}} D(xy) = \sum_{y\in X_{\pi_S}} \frac{D(xy)}{D(x'y)} D(x'y) \leq \alpha \sum_{y\in X_{\pi_S}} D(x'y) = \alpha D(x'X_{\pi_S}) \ .$$

We can use the above and show that

$$(D|\pi_S)_{\bar{S}}(x) = \frac{D(xX_{\pi_S})}{D(x'X_{\pi_S})}\frac{D(x'X_{\pi_S})}{D(X_{\pi_S})} \leq \alpha(D|\pi_S)_{\bar{S}}(x') \ .$$

Finally, (iv) is a corollary of (i)—(iii).

■

B. Robust Risk Minimizer for R^C_ρ

Proposition 33 *Under the uniform distribution, for any non-constant concept $c \in$ MON-CONJ, we have that $\mathsf{R}^C_1(c,c) > \mathsf{R}^C_1(c,0)$.*

Proof Let $\mathcal{X} = \{0,1\}^n$ and D be the uniform distribution on $\mathcal{X}$. Let $c(x) = x_1 \wedge \cdots \wedge x_k$ for some $k \in [n]$. Then,

$$\begin{aligned}
\mathsf{R}^C_1(c,c) &= \Pr_{x\sim D}(\exists z \in B_\rho(x) \,.\, c(z) \neq c(x)) \\
&= \Pr_{x\sim D}(c(x) = 1) + \Pr_{x\sim D}(\exists! i \in [k] \,.\, x_i = 0) \\
&= \mathsf{R}^C_1(c,0) + \Pr_{x\sim D}(\exists! i \in [k] \,.\, x_i = 0) \\
&> \mathsf{R}^C_1(c,0) \ .
\end{aligned}$$

■

C. Proofs from Section 3

Proof [of Theorem 5] First, if $\mathcal{C}$ is trivial, we need at most one example to identify the target function.

For the other direction, suppose that $\mathcal{C}$ is non-trivial, and for a given $c \in \mathcal{C}$, denote by $I_c \subseteq [n]$ the index set of relevant variables in the function c.[11] We first start by fixing any learning algorithm and polynomial sample complexity function m. Let $\eta = \frac{1}{2^{\omega(\log n)}}$, $0 < \delta < \frac{1}{2}$, and note that for any constant $a > 0$,

$$\lim_{n\to\infty} n^a \log(1-\eta)^{-1} = 0 \ ,$$

and so any polynomial in n is $o\left((\log(1/(1-\eta)))^{-1}\right)$. Then it is possible to choose n_0 such that for all $n \geq n_0$,

$$m \leq \frac{\log(1/\delta)}{2n\log(1-\eta)^{-1}} \ . \tag{3}$$

Since $\mathcal{C}$ is non-trivial, we can choose concepts $c_1, c_2 \in \mathcal{C}_n$ and points $x, x' \in \{0,1\}^n$ such that c_1 and c_2 agree on x but disagree on x'. This implies that there exists a point $z \in \{0,1\}^n$

11. This means that if $i \in I_c$ there exists $x \in \{0,1\}^n$ such that $c(x^{\oplus i})$, the output of c on flipping the i-th bit of x, differs from $c(x)$.

such that (i) $c_1(z) = c_2(z)$ and (ii) it suffices to change *only one bit* in $I := I_{c_1} \cup I_{c_2}$ to cause c_1 to disagree on z and its perturbation. Let D be a product distribution such that

$$\Pr_{x \sim D}(x_i = z_i) = \begin{cases} 1-\eta & \text{if } i \in I \\ \frac{1}{2} & \text{otherwise} \end{cases} .$$

Draw a sample $S \sim D^m$ and label it according to $c \sim U(c_1, c_2)$. Then,

$$\Pr_{S \sim D^m}(\forall x \in S \quad c_1(x) = c_2(x)) \geq (1-\eta)^{m|I|} . \tag{4}$$

Bounding the RHS below by $\delta > 0$, we get that, as long as

$$m \leq \frac{\log(1/\delta)}{|I| \log(1-\eta)^{-1}} ,$$

Equation 4 holds with probability at least δ. But this is true as Equation 3 holds as well. However, if $x = z$, then it suffices to flip one bit of x to get x' such that $c_1(x') \neq c_2(x')$. Then,

$$\mathsf{R}^E_\rho(c_1, c_2) \geq \Pr_{x \sim D}(x_I = z_I) = (1-\eta)^{|I|} . \tag{5}$$

The constraints on η and the fact that $|I| \leq n$ are sufficient to guarantee that the RHS is $\Omega(1)$. Let $\alpha > 0$ be a constant such that $\mathsf{R}^E_\rho(c_1, c_2) \geq \alpha$.

We can use the same reasoning as in Lemma 6 to argue that, for any $h \in \{0,1\}^{\mathcal{X}}$,

$$\mathsf{R}^E_1(c_1, h) + \mathsf{R}^E_1(c_2, h) \geq \mathsf{R}^E_1(c_1, c_2) .$$

Finally, we can show that

$$\mathop{\mathbb{E}}_{c \sim U(c_1,c_2)} \mathop{\mathbb{E}}_{S \sim D^m} \left[\mathsf{R}^R_1(h, c)\right] \geq \alpha\delta/2,$$

hence there exists a target c with expected robust risk bounded below by a constant.[12] ■

D. Proofs from Section 5

D.1 Proof of Lemma 12

Proof We begin by bounding the probability that c_1 and c_2 agree on an i.i.d. sample of size m:

$$\Pr_{S \sim D^m}(\forall x \in S \cdot c_1(x) = c_2(x) = 0) = \left(1 - \frac{1}{2^l}\right)^{2m} . \tag{6}$$

Bounding the RHS below by $1/2$, we get that, as long as

$$m \leq \frac{\log(2)}{2\log(2^l/(2^l-1))} , \tag{7}$$

12. For a more detailed reasoning, we refer the reader to the proof of Theorem 13, where we bound the expected value $\mathop{\mathbb{E}}_{c,S}\left[\mathsf{R}^E_\rho(\mathcal{A}(S), c)\right]$ of the robust risk of a target chosen at uniformly random and the hypothesis outputted by a learning algorithm $\mathcal{A}$ on a sample S.

Equation 6 holds with probability at least 1/2.

Now, if $l = \omega(\log(n))$, then for a constant $a > 0$,

$$\lim_{n\to\infty} n^a \log\left(\frac{2^l}{2^l - 1}\right) = 0 \ ,$$

and so any polynomial in n is $o\left(\left(\log\left(\frac{2^l}{2^l-1}\right)\right)^{-1}\right)$. ■

D.2 Lemma 34

Lemma 34 *Let $\Phi : \mathcal{X}_n \to \mathcal{X}_d$ be the embedding encoding the truth values of (disjunctive) clauses in a variable-disjoint matching M of size d under an assignment $x \in \mathcal{X}_n$. Let D be an α-log-Lipschitz distribution on $\mathcal{X}_n$ and define D' on $\mathcal{X}_d$ as follows:*

$$D'(y) := \sum_{x\in\Phi^{-1}(y)} D(x) \ ,$$

where $y \in \mathcal{X}_d$. Then D' is α'-log-Lipschitz for $\alpha' = (\alpha + 1)^k - 1$.

Proof Let $y, y' \in \mathcal{X}_d$ be such that $d_H(y, y') = 1$, i.e. y and y' disagree on exactly one clause in M. We want to upper bound the quantity $D(y)/D(y')$ by $\alpha' = (\alpha+1)^k - 1$. To this end, and WLOG, let $y_1 \neq y'_1$ and let the clause K_1 in M where y and y' disagree be a function of the first k bits in $\mathcal{X}_n$. Because M is variable disjoint, and since K_1 is a disjunction of literals, if we fix the bits $x_{k+1}, \ldots, x_n$, then there exists a unique assignment of $x_1, \ldots, x_k$ such that $\Phi(x)_1 = 0$ (where $x = x_1 \ldots x_n$), and thus the remaining $2^k - 1$ are such that K_1 evaluates to 1. Hence, to upper bound $D(y)/D(y')$, we will assume that $y_1 = 1$ and $y'_1 = 0$.

Now, we can partition the preimage $\Phi^{-1}(y)$ into $\{P_{x'}\}_{x'\in\Phi^{-1}(y')}$, where each $x \in P_{x'}$ disagrees with x' on at least one of the first k bits and is the same on the remaining $n - k$ bits. Thus

$$\begin{aligned}
\frac{D'(y)}{D'(y')} &= \frac{\sum_{x'\in\Phi^{-1}(y')} \sum_{x\in P_{x'}} D(x)}{\sum_{x'\in\Phi^{-1}(y')} D(x')} \\
&\leq \frac{\sum_{x'\in\Phi^{-1}(y')} D(x') \sum_{x\in P_{x'}} \alpha^{d_H(x,x')}}{\sum_{x'\in\Phi^{-1}(y')} D(x')} && \text{(by log-Lipschitzness of } D\text{)} \\
&= \frac{\left((\alpha+1)^k - 1\right) \sum_{x'\in\Phi^{-1}(y')} D(x')}{\sum_{x'\in\Phi^{-1}(y')} D(x')} \\
&= (\alpha+1)^k - 1 \ ,
\end{aligned}$$

where we used the fact $(\alpha + 1)^k = \sum_{i=0}^{k} \binom{k}{i}\alpha^i$ for the third step. ■

References

Benny Applebaum, Boaz Barak, and David Xiao. On basing lower-bounds for learning on worst-case assumptions. In *Proceedings of the 49th Annual IEEE symposium on Foundations of computer science*, 2008.

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, 2020.

Pranjal Awasthi, Vitaly Feldman, and Varun Kanade. Learning using local membership queries. In *Conference on Learning Theory*, pages 398–431. PMLR, 2013.

Battista Biggio and Fabio Roli. Wild patterns: Ten years after the rise of adversarial machine learning. *Pattern Recognition*, 84:317–331, 2018. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2018.07.023. URL `https://www.sciencedirect.com/science/article/pii/S0031320318302565`.

Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. Evasion attacks against machine learning at test time. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 387–402. Springer, 2013.

Avrim Blum, Merrick Furst, Michael Kearns, and Richard J Lipton. Cryptographic primitives based on hard learning problems. In *Annual International Cryptology Conference*, pages 278–291. Springer, 1993.

Sébastien Bubeck, Yin Tat Lee, Eric Price, and Ilya Razenshteyn. Adversarial examples from cryptographic pseudo-random generators. *arXiv preprint arXiv:1811.06418*, 2018a.

Sébastien Bubeck, Eric Price, and Ilya Razenshteyn. Adversarial examples from computational constraints. *arXiv preprint arXiv:1805.10204*, 2018b.

Sébastien Bubeck, Yin Tat Lee, Eric Price, and Ilya Razenshteyn. Adversarial examples from computational constraints. In *International Conference on Machine Learning*, pages 831–840. PMLR, 2019.

Nilesh Dalvi, Pedro Domingos, Sumit Sanghai, Deepak Verma, et al. Adversarial classification. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 99–108. ACM, 2004.

Akshay Degwekar, Preetum Nakkiran, and Vinod Vaikuntanathan. Computational limitations in robust classification and win-win results. In *Conference on Learning Theory*, pages 994–1028. PMLR, 2019.

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. *arXiv preprint arXiv:1906.05815*, 2019.

Tommaso Dreossi, Shromona Ghosh, Alberto Sangiovanni-Vincentelli, and Sanjit A Seshia. A formalization of robustness for deep neural networks. *arXiv preprint arXiv:1903.10033*, 2019.

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. *arXiv preprint arXiv:1802.08686*, 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*, pages 637–657, 2015.

Dan Feldman and Leonard J Schulman. Data reduction for weighted and outlier-resistant clustering. In *Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms*, pages 1343–1354. Society for Industrial and Applied Mathematics, 2012.

Zoltán Füredi. Matchings and covers in hypergraphs. *Graphs and Combinatorics*, 4(1): 115–206, 1988.

Justin Gilmer, Luke Metz, Fartash Faghri, Samuel S Schoenholz, Maithra Raghu, Martin Wattenberg, and Ian Goodfellow. Adversarial spheres. *arXiv preprint arXiv:1801.02774*, 2018.

Paul W Goldberg. Some discriminant-based PAC algorithms. *Journal of Machine Learning Research*, 7(Feb):283–306, 2006.

Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions. *Journal of the ACM (JACM)*, 33(4):792–807, 1986.

Pascale Gourdeau, Varun Kanade, Marta Kwiatkowska, and James Worrell. On the hardness of robust classification. In *Advances in Neural Information Processing Systems*, pages 7444–7453, 2019.

Venkatesan Guruswami and Piotr Indyk. Expander-based constructions of efficiently decodable codes. In *Proceedings 42nd IEEE Symposium on Foundations of Computer Science*, pages 658–667. IEEE, 2001.

Vladlen Koltun and Christos H Papadimitriou. Approximately dominating representatives. *Theoretical Computer Science*, 371(3):148–154, 2007.

Daniel Lowd and Christopher Meek. Adversarial learning. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 641–647. ACM, 2005a.

Daniel Lowd and Christopher Meek. Good word attacks on statistical spam filters. In *CEAS*, volume 2005, 2005b.

Saeed Mahloujifar and Mohammad Mahmoody. Can adversarially robust learning leverageComputational 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.

Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2012.

Omar Montasser, Steve Hanneke, and Nathan Srebro. Vc classes are adversarially robustly learnable, but only improperly. In *Conference on Learning Theory*, pages 2512–2530. PMLR, 2019.

Ryan O'Donnell and Rocco A Servedio. Learning monotone decision trees in polynomial time. *SIAM Journal on Computing*, 37(3):827–844, 2007.

Ronald L Rivest. Learning decision lists. *Machine learning*, 2(3):229–246, 1987.

Ali Shafahi, W Ronny Huang, Christoph Studer, Soheil Feizi, and Tom Goldstein. Are adversarial examples inevitable? *arXiv preprint arXiv:1809.02104*, 2018.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2013.

Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. In *International Conference on Learning Representations*, 2019.

Leslie G Valiant. A theory of the learnable. In *Proceedings of the sixteenth annual ACM symposium on Theory of computing*, pages 436–445. ACM, 1984.