On the Hardness of Robust Classification

Pascale Gourdeau, Varun Kanade, Marta Kwiatkowska, and James Worrell

University of Oxford

August 20, 2019

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 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 if the adversary is restricted to perturbing $O(\log n)$ bits, then the class of monotone conjunctions can be robustly learned with respect to a general class of distributions (that includes the uniform distribution). 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 another computational model (e.g. the statistical query model) nor on any hardness assumption other than the existence of a hard learning problem in the PAC framework.

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 [2017], 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 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

---

[1]For an in-depth review and definitions of different types of attacks, the reader may refer to Biggio and Roli [2017], Dreossi et al. [2019].
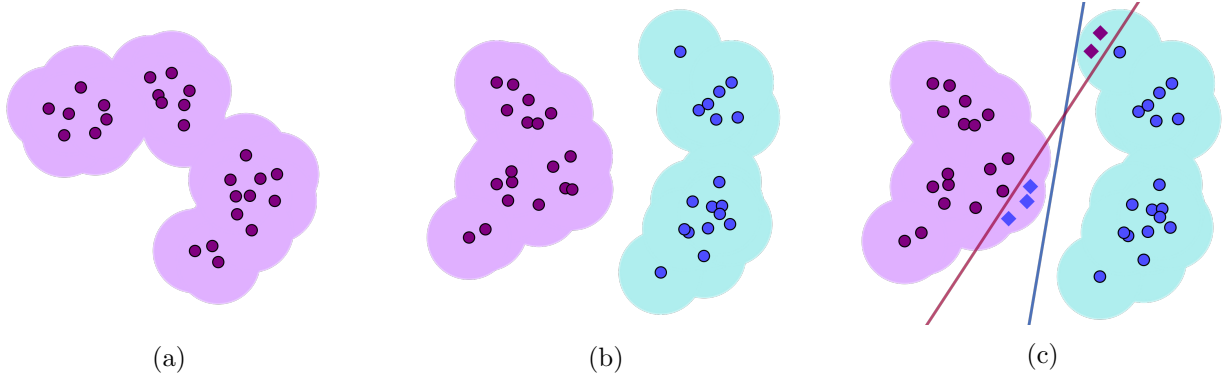
Figure 1: (a) The support of the distribution is such that $\mathsf{R}_\rho^C(h,c) = 0$ can only be achieved if c is constant. (b) The ρ-expansion of the support of the distribution and target c admit hypotheses h such that $\mathsf{R}_\rho^C(h,c) = 0$. (c) An example where R_ρ^C and R_ρ^E differ. The 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 $\mathsf{R}_\rho^E > 0$.

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. 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 $\mathsf{R}_\rho^E(h,c)$ is the probability $\mathbb{P}_{x \sim D}(\exists y \in B_\rho(x) \cdot h(y) \neq c(y))$ 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 $\mathsf{R}_\rho^C(h,c)$ is the probability $\mathbb{P}_{x \sim D}(\exists y \in B_\rho(x) \cdot h(y) \neq c(x))$ 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 concerns 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 $\mathsf{R}_\rho^C(h,c) = 0$ (see

Figure 1). In particular, in most cases $\mathsf{R}_\rho^C(c,c) \neq 0$, 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 on the ball* notion of robust learning, we at least have $\mathsf{R}_\rho^E(c,c) = 0$; 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. We highlight that these results show that a polynomial-size 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 Monotone Conjunctions: Given the impossibility of distribution-free robust learning, we consider robust learning under specific distributions. We consider one of the simplest concept class studied in PAC Learning, the class of *monotone conjunctions*, under the class of log-Lipschitz distributions (which includes the uniform distribution) and show that this class of functions is robustly learnable provided $\rho = O(\log n)$ and is not robustly learnable with polynomial sample complexity for $\rho = \omega(\log n)$. A class of distributions is said to be α-log-Lipschitz if the logarithm of the density function is $\log(\alpha)$-Lipschitz with respect to the Hamming distance. 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 robustly learn with respect to *exact in the ball* loss.

Computational Hardness of 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* using random examples alone. In the case of functions defined on the boolean hypercube, access to a membership query oracle suffices; 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 since in principle it requires enumerating over the reals. Under strong assumptions of *inductive bias* of the target class 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.

Second, we consider the computational complexity of robust learning. Bubeck et al. [2018a] and Degwekar and Vaikuntanathan [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 arguably simpler proof of hardness, based simply on the assumption that there exist concept classes that are hard to PAC 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. Since the existence of one-way functions implies the existence of concept classes that are hard to PAC learn (with or without membership queries), our result is also based on a slightly weaker assumption than Bubeck et al. [2018b][3].

---

[2]We do require 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.

[3]It is believed that the existence of hard to PAC learn concept classes is not sufficient to construct one-way

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.

Closer to the present paper are Diochnos et al. [2018], Mahloujifar and Mahmoody [2018], Mahloujifar et al. [2019], which work the with exact-in-a-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). 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. By contrast we show that a weaker adversary, who can perturb only $\omega(\log n)$ bits, renders it impossible to learn monotone conjunctions with polynomial sample complexity. 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.

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 centre 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*

$$\mathsf{R}_\rho^E(h, c) = \mathop{\mathbb{P}}_{x \sim D} \left(\exists z \in B_\rho(x) : h(z) \neq c(z) \right) .$$

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 analyse 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:

---

functions. Applebaum et al. [2008].

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*

$$\mathsf{R}_\rho^C(h, c) = \mathop{\mathbb{P}}_{x \sim D} \left(\exists z \in B_\rho(x) : h(z) \neq c(x) \right) \ .$$

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 robust risk in the case that $h = c$. (Let us note in passing that the risk functions R_ρ^C and R_ρ^E are in general incomparable. Figure 1c gives an example in which $\mathsf{R}_\rho^C = 0$ and $\mathsf{R}_\rho^E > 0$.) 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 realisable 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. These limitations appear even more stringent when we consider simple concept classes such as parity functions, which are defined for an index set $I \subseteq [n]$ as $f_I(x) = \sum_i x_i + b \mod 2$ for $b \in \{0, 1\}$. This class can be PAC-learned, as well as exactly learned with n membership queries. However, for any point, it suffices to flip one bit of the index set to switch the label, so $\mathsf{R}_\rho^C(f_I, f_I) = 1$ for any $\rho \geq 1$ if $I \neq \emptyset$.

Ultimately, we want the adversary's power to come from creating perturbations that cause the hypothesis and target functions to differ in some regions of the input space. For this reason we favor the exact-in-the-ball definition and henceforth work with that.

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 $poly(\cdot, \cdot, \cdot)$ 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 $\epsilon, \delta > 0$, there exists $m \leq poly(1/\epsilon, 1/\delta, n)$, such that when $\mathcal{A}$ is given access to a sample $S \sim D^m$ it outputs $h : \{0, 1\}^n \to \{0, 1\}$ such that $\mathop{\mathbb{P}}_{S \sim D^m} \left(\mathsf{R}_{\rho(n)}^E(h, c) < \epsilon \right) > 1 - \delta$.*

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 $\mathop{\mathbb{P}}_{x \sim D} (h(x) \neq c(x)) = 0$. We have remarked that exact learnability implies robust learnability; we next 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 $\mathop{\mathbb{P}}_{x \sim D} (x_1 = x_2) = 1$ and $\mathop{\mathbb{P}}_{x \sim D} (x_k = 1) = \frac{1}{2}$ 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 $h \in \{0, 1\}^{\mathcal{X}}$, $\mathsf{R}_1^E(c_1, h) + \mathsf{R}_1^E(c_2, h) \geq \mathsf{R}_1^E(c_1, c_2) = 1$. Then

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

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

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. We refer the reader to Appendix B for further discussion on the relationship between robust and zero-risk learning.

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 a concept class as $\mathcal{C} = \bigcup_{n \geq 1} \mathcal{C}_n$. We say that a class of functions is trivial if $\mathcal{C}_n$ has at most two functions, and that they differ on every point.

Theorem 5. *Any concept class $\mathcal{C}$ 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 $h : \{0, 1\}^n \to \{0, 1\}$*

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

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. $\qquad\square$

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 likely 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 (for example in $\mathbb{R}^n$).

4   Monotone Conjunctions

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. 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.

4.1 Non-Robust Learnability

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 lower bounds the robust risk of two disjoint monotone conjunctions.

Lemma 7. *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 $\rho \geq l/2$, 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 gets larger.*

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

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

$\square$

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 8. *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 9. *MON-CONJ is not efficiently ρ-robustly learnable for $\rho(n) = \omega(\log(n))$.*

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 $poly(\cdot, \cdot, \cdot, \cdot, \cdot)$, and note that, since $size(c)$ and ρ are both at most n, we can simply consider a function $poly(\cdot, \cdot, \cdot)$ 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 8, where $m(n)$ is the fixed sample complexity function.Then Equation (8) 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 7. 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 $m = poly(1/\epsilon, 1/\delta, n)$. By Lemma 8, 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 $S \sim D^m$:

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

Then, by Lemmas 8 and 6,

$$\underset{c,S}{\mathbb{E}}\left[\mathsf{R}_\rho^E(\mathcal{A}(S),c)\right] \geq \underset{c,S}{\mathbb{P}}(\mathcal{E})\,\underset{c,S}{\mathbb{E}}\left[\mathsf{R}_\rho^E(\mathcal{A}(S),c)\mid\mathcal{E}\right]$$

$$> \frac{1}{2}\left(\underset{c,S}{\mathbb{P}}(\mathcal{E}_{c_1})\,\underset{S}{\mathbb{E}}\left[\mathsf{R}_\rho^E(\mathcal{A}(S),c)\mid\mathcal{E}\cap\mathcal{E}_{c_1}\right] + \underset{c,S}{\mathbb{P}}(\mathcal{E}_{c_2})\,\underset{S}{\mathbb{E}}\left[\mathsf{R}_\rho^E(\mathcal{A}(S),c)\mid\mathcal{E}\cap\mathcal{E}_{c_2}\right]\right)$$

$$= \frac{1}{4}\,\underset{S}{\mathbb{E}}\left[\mathsf{R}_\rho^E(\mathcal{A}(S),c_1) + \mathsf{R}_\rho^E(\mathcal{A}(S),c_2)\mid\mathcal{E}\right]$$

$$\geq \frac{1}{4}\,\underset{S}{\mathbb{E}}\left[\mathsf{R}_\rho^E(c_2,c_1)\right]$$

$$> 0.1\ .$$

$\square$

4.2 Robust Learnability Against a Logarithmically-Bounded Adversary

The argument showing the non-robust learnability of MON-CONJ under the uniform distribution in the previous section cannot be carried through if the conjunction lengths are logarithmic in the input dimension, or if the robustness parameter is small compared to that target conjunction's length. In both cases, we show that it is possible to efficiently robustly learn these conjunctions 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 10. *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 monotone conjunctions is ρ-robustly learnable with respect to $\mathcal{D}$ for robustness function $\rho(n) = O(\log n)$.*

The proof can be found in Appendix D. This combined with Theorem 10 shows that $\rho(n) = \log(n)$ is essentially the threshold for efficient robust learnability of the class MON-CONJ .

5 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 $\mathcal{D}'$. This is essentially a version of the main result of Bubeck et al. [2018b], which used the constant-in-the-ball definition of robust risk. 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 a hard learning problem in the PAC framework and is valid under *both* Definitions 1 and 2 of robust risk.

Construction of $\mathcal{C}'$. 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 and distribution pairs $\{(c',D')\}_{D'\in\mathcal{D}'_{c'},c'\in\mathcal{C}'}$, where $\mathcal{C}' = \{\mathcal{C}'_{(k,n)}\}_{k,n\in\mathbb{N}}$ on $\mathcal{X}'_{k,n} = \{0,1\}^{(2k+1)n+1}$ as follows. Let maj_k :

$\mathcal{X}'_{k,n} \to \mathcal{X}_n$ be the function that returns the majority vote on each subsequent block of k bits, and ignores the last bit. We define $\mathcal{C}'_{(k,n)} = \{c \circ \mathrm{maj}_{2k+1} \mid c \in \mathcal{C}_n\}$. Let $\varphi_k : \mathcal{X}_n \to \mathcal{X}'_{k,n}$ be defined as

$$\varphi_k(x) := \underbrace{x_1 \ldots x_1 x_2 \ldots x_{d-1} x_d \ldots x_d}_{2k+1 \text{ copies of each } x_i} c(x) , \quad \varphi_k(S) := \{\varphi_k(x_i) \mid x_i \in S\} ,$$

for $x = x_1 x_2 \ldots x_d \in \mathcal{X}$ and $S \subseteq \mathcal{X}$. For a concept $c \in \mathcal{C}_n$, each $D \in \mathcal{D}_n$ induces a distribution $D' \in \mathcal{D}'_{c'}$, where $c' = c \circ \mathrm{maj}_{2k+1}$ and $D'(z) = D(x)$ if $z = \varphi_k(x)$, and $D'(z) = 0$ otherwise.

  As shown below, this set up allows us to see that any algorithm for learning $\mathcal{C}_n$ with respect to $\mathcal{D}_n$ yields an algorithm for learning the pairs $\{(c', D')\}_{D' \in \mathcal{D}'_{c'}, c' \in \mathcal{C}'}$. However, any *robust* learning algorithm cannot solely rely on the last bit of the input, as it could be flipped by an adversary. Then, this algorithm can be used to PAC-learn $\mathcal{C}_n$. This establishes the equivalence of the computational difficulty between PAC-learning $\mathcal{C}_n$ with respect to $\mathcal{D}'_{(k,n)}$ and robustly learning $\{(c', D')\}_{D' \in \mathcal{D}'_{c'}, c' \in \mathcal{C}'_{(k,n)}}$. As mentioned earlier, we can still efficiently PAC-learn the pairs $\{(c', D')\}_{D' \in \mathcal{D}'_{c'}, c' \in \mathcal{C}'}$ simply by always outputting a hypothesis that returns the last bit of the input.

Theorem 11. *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 concept class $\mathcal{C}'_{(k,n)}$ and a family of distributions $\mathcal{D}'_{(k,n)}$ over $\{0,1\}^{(2k+1)n+1}$ such that efficient k-robust learnability of the concept-distribution pairs $\{(c', D')\}_{D' \in \mathcal{D}'_{c'}, c' \in \mathcal{C}'_{(k,n)}}$ and either of the robust risk functions R_k^C or R_k^E implies efficient PAC-learnability of $\mathcal{C}_n$ with respect to $\mathcal{D}_n$.*

  Before proving the above result, let us first prove the following proposition.

Proposition 12. *The concept-distribution pairs $\{(c', D')\}_{D' \in \mathcal{D}'_{c'}, c' \in \mathcal{C}'_{(k,n)}}$ can be k-robustly learned using $O\left(\frac{1}{\epsilon}\left(\log|\mathcal{C}_n| + \log\frac{1}{\delta}\right)\right)$ examples.*

Proof. First 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 $h'(x')$. Thus, we also have k-robustly learned $\mathcal{C}'_{(k,n)}$. □

Remark 13. The sample complexity in Proposition 12 is independent of k, and so the construction of the class $\mathcal{C}'$ on $\mathcal{X}'$ allows the adversary to modify $\frac{1}{2n}$ fraction of the bits. There are ways to make the adversary more powerful and keep the sample complexity unchanged. Indeed, the fraction of the bits the adversary can flip can be increased by using error correction codes. For example, BCH codes Bose and Ray-Chaudhuri [1960], Hocquenghem [1959] would allow us to obtain an input space $\mathcal{X}'$ of dimension $n + k \log n$ where the adversary can flip $\frac{k}{n+k\log n}$ bits.

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

Proof of Theorem 11. Given $\mathcal{C}_n$ and $\mathcal{D}$, let $\mathcal{C}'_{(k,n)}$ and $\{\mathcal{D}'_{c'}\}_{c' \in \mathcal{C}'_{(k,n)}}$ be constructed as above. Suppose that it is hard to PAC-learn $\mathcal{C}_n$ with respect to the distribution family $\mathcal{D}_n$. Suppose that we are given an algorithm $\mathcal{A}'$ to k-robustly learn $\{(c', D')\}_{D' \in \mathcal{D}'_{c'}, c' \in \mathcal{C}'_{(k,n)}}$ and a sample complexity m.

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

Since $\mathcal{A}'$ k-robustly learns the concept-distribution pairs $\{(c', D')\}_{D' \in \mathcal{D}'_{c'}, c' \in \mathcal{C}'_{(k,n)}}$, with probability at least $1 - \delta$ over S', for any $x \sim D$, we have that h' will be wrong on $\varphi_k(x)$ (where the last bit is random) with probability at most ϵ. So by outputting $h = h' \circ \varphi_k$, we have an algorithm to PAC-learn $\mathcal{C}_n$ with respect to the distribution family $\mathcal{D}_n$. $\qquad\square$

6   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. We have moreover given a tight characterization of the strength of an adversary to prevent robust learning of monotone conjunctions under certain distributional assumptions. An interesting avenue for future work is to see whether this result can be generalised to other classes of functions. Finally, we have provided a simpler proof of the previously established result of the computational hardness of robust learning.

In the 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. It is possible that requiring a classifier to be correct near a point is asking too much, and that, on the other hand, we can only solve "easy problems" with strong distributional assumptions in the case where we required our classifier to be constant near a point. Nevertheless, we note that we would perhaps benefit from studying robust learning in different learning models, for example where one has access to membership queries.

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.

Pranjal Awasthi, Vitaly Feldman, and Varun Kanade. Learning using local membership queries. In *COLT*, volume 30, pages 1–34, 2013.

Battista Biggio and Fabio Roli. Wild patterns: Ten years after the rise of adversarial machine learning. *arXiv preprint arXiv:1712.03141*, 2017.

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.

Raj Chandra Bose and Dwijendra K Ray-Chaudhuri. On a class of error correcting binary group codes. *Information and control*, 3(1):68–79, 1960.

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.

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 and Vinod Vaikuntanathan. Computational limitations in robust classification and win-win results. *arXiv preprint arXiv:1902.01086*, 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.

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.

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.

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

Alexis Hocquenghem. Codes correcteurs d'erreurs. *Chiffres*, 2(2):147–56, 1959.

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 leverage computational hardness? *arXiv preprint arXiv:1810.01407*, 2018.

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.

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.

Appendix

A Learning Theory Basics

A.1 The PAC framework

We study the problem of robust classification. This is a generalization of standard classification tasks, which are defined on an input space $\mathcal{X}_n$ of dimension n and finite output space $\mathcal{Y}$. Common examples of input spaces are $\{0,1\}^n$, $[0,1]^n$, and $\mathbb{R}^n$. We focus on *binary classification* in the *realizable setting*, where $\mathcal{Y} = \{0,1\}$, and we get access to a sample $S = \{(x_i, y_i)\}_{i=1}^m$ where the x_i's are drawn i.i.d. from an unknown underlying distribution D, and there exists $c : \mathcal{X} \to \mathcal{Y}$ such that $y_i = c(x_i)$, namely, there exists a *target concept* that has labeled the sample. In the PAC framework Valiant [1984], our goal is to find a function h that approximates c with high probability over the training sample. This means we are allowing a small chance of having a sample that is not representative of the distribution. As we require our confidence to increase, we require more data. PAC learning is formally defined for *concept classes* $\mathcal{C}_n \subseteq \{0,1\}^{\mathcal{X}_n}$ as follows.

Definition 14 (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 $p(\cdot, \cdot, \cdot)$ 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 $m \geq p(n, 1/\epsilon, 1/\delta)$ examples drawn i.i.d. from D and labeled with c, $\mathcal{A}$ outputs $h \in \mathcal{H}$ such that with probability at least $1 - \delta$,*

$$\mathbb{P}_{x \sim D} (c(x) \neq h(x)) \leq \epsilon \ .$$

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

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 S, 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.

We refer the reader to Mohri et al. [2012] for an in-depth introduction to machine learning theory.

A.3   Log-Lipschitz Distributions

Definition 15. *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 16. *Let D be an α-log-Lipschitz distribution over $\{0,1\}^n$. Then the following hold:*

1. *For $b \in \{0,1\}$, $\frac{1}{1+\alpha} \leq \mathop{\mathbb{P}}_{x \sim D}(x_i = b) \leq \frac{\alpha}{1+\alpha}$.*

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

3. *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.*

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

Proof. To prove (1), 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

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

The result follows from solving for $\mathop{\mathbb{P}}_{x \sim D}(x_i = b)$.

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 (2), 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 (3), 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,

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

14

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, (4) is a corollary of (1)–(3).

$\square$

B Discussion on the Relationship between Robust and Zero-Risk Learning

We saw that, for both robust risks R_ρ^C and R_ρ^E, zero-risk learning does not necessarily imply robust learning. Moreover, as shown in Section 3, efficient distribution-free robust learning is not possible even in the realizable setting. What can be said if we have access to a robust learning algorithm for a specific distribution on the boolean hypercube? We will show that distribution-dependent robust learning implies zero-risk learning for both robust risk definitions, under certain conditions on the measure of balls in the support of the distribution. Let us start with Definition 1, where we require the hypothesis to be exact in the ρ-balls around a point.

Proposition 17. *For any probability measure μ on $\{0,1\}^n$, robustness parameter ρ and concepts h, c, there exists $\epsilon > 0$ such that if $\mathsf{R}_\rho^E(h, c) < \epsilon$ then $h(x) = c(x)$ for any $x \in \mathcal{X}$ such that $\mu(B_\rho(x)) > 0$. In particular, one has that h and c agree on the support of μ.*

Proof. Suppose there exists $x^* \in \mathcal{X}$ with $\mu(B_\rho(x^*)) > 0$ such that $h(x^*) \neq c(x^*)$. Then for any $z \in B_\rho(x^*)$, we have that $\mathsf{R}_\rho^E(h, c, z)$, the robust risk of h with respect to c at point z, is 1. Let $\tilde{\mathcal{X}} := \{x \in \mathcal{X} : \mu(B_\rho(x)) > 0\}$, and $\epsilon = \min_{x \in \tilde{\mathcal{X}}} \mu(B_\rho(x))$. We have that

$$\mathsf{R}_\rho^E(h, c) \geq \sum_{z \in B_\rho(x^*)} \mu(\{z\})\ell_\rho^R(h, c, z) = \mu(B_\rho(x^*)) \geq \epsilon \ .$$

$\square$

Corollary 18. *For any fixed distribution D, robust learning with respect to D implies zero-risk learning with respect to D for any robustness parameter as long as ϵ in Proposition 17 satisfies $\epsilon^{-1} = poly(n)$.*

Proof. Fix a distribution $D \in \mathcal{D}$ on $\mathcal{X}$. Suppose that we have a ρ-robust learning algorithm $\mathcal{A}_{\mathcal{F}}^R(D)$ for $\mathcal{F}$, namely for all $\epsilon, \delta, \rho > 0$, for all $c \in \mathcal{F}$, if $\mathcal{A}_{\mathcal{F}}^R(D)$ has access to a sample S of size $m \geq \text{poly}(\frac{1}{\epsilon}, \frac{1}{\delta}, \text{size}(c), n)$, it returns $f \in \mathcal{F}$ such that

$$\mathbb{P}_{S \sim D^m}\left(\ell_\rho^R(f, c) < \epsilon\right) \geq 1 - \delta \ . \tag{1}$$

By Proposition 17, we can choose ϵ such that $\mathsf{R}_\rho^E(h, c) < \epsilon$ implies that $h(x) = c(x)$ for any $x \in \mathcal{X}$ such that $\mu(B_\rho(x)) > 0$. Note that this ϵ depends on D, ρ and n. So we have that

$$\mathbb{P}_{x \sim D}\left(f(x) \neq c(x)\right) = 0 \ , \tag{2}$$

with probability at least $1 - \delta$ over the training sample S, whose size remains polynomial in $\frac{1}{\delta}$ and n by the proposition assumptions.

$\square$

Remark 19. The assumption on ϵ in Corollary 18 is necessary to use the robust learning algorithm as a black box: in Section 4.2, we work under a well-behaved class of distributions that includes the uniform distribution and show that, for long enough monotone conjunctions and small enough robustness parameter (with respect to the conjunction length), efficient robust learning is possible. However, we cannot exactly learn these monotone conjunctions. In the uniform distribution setting, the ρ-balls all have the same probability mass and ϵ^{-1} is essentially superpolynomial in n.

To show the same result for R_ρ^C, where the hypothesis is constant in a ball, we can use the exact same reasoning as in Corollary 18, except that we need to show the analogue of Proposition 17 for this setting.

Proposition 20. *For any probability measure μ on $\{0,1\}^n$ and for any concepts h, c, there exists $\epsilon > 0$ such that if $\mathsf{R}_\rho^C(h, c) < \epsilon$ then h and c agree on the support of μ.*

Proof. Fix h, c, D and let $\epsilon = \min_{x \in \mathrm{supp}(\mu)} \mu(\{x\})$. Suppose there exists $x^* \in \mathrm{supp}(\mu)$ and $z \in B_\rho(x^*)$ such that $c(x^*) \neq h(z)$. Then

$$\mathsf{R}_\rho^C(h, c) = \mathbb{P}_{x \sim \mu} (\exists z \in B_\rho(x) \, . \, c(x) \neq h(z)) \geq \epsilon \ .$$

$\square$

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. 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 $z \in \{0, 1\}^n$ 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 such that

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

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

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

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

$$m \leq \frac{\log(1/\delta)}{|I| \log(1-\eta)^{-1}} \quad , \tag{5}$$

(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,

$$\mathsf{R}_\rho^E(c_1, c_2) \geq \mathbb{P}_{x \sim D}(x_I = z_I) = (1-\eta)^{|I|} \quad . \tag{6}$$

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 $\mathsf{R}_\rho^E(c_1, c_2) \geq \alpha$.

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

$$\mathsf{R}_1^E(c_1, h) + \mathsf{R}_1^E(c_2, h) \geq \mathsf{R}_1^E(c_1, c_2) \quad .$$

Finally, we can show that

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

hence there exists a target c with expected robust risk bounded below by a constant[4]. $\qquad\square$

D Proofs from Section 4

D.1 Proof of Lemma 8

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

$$\mathbb{P}_{S \sim D^m}(\forall x \in S \cdot c_1(x) = c_2(x) = 0) = \left(1 - \frac{1}{2^l}\right)^{2m} \quad . \tag{7}$$

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

$$m \leq \frac{\log(2)}{2 \log(2^l/(2^l - 1))} \quad , \tag{8}$$

(7) holds with probability at least $1/2$.

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

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

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

---

[4]For a more detailed reasoning, we refer the reader to the proof of Theorem 9, where we bound the expected value $\mathbb{E}_{c,S}\left[\mathsf{R}_\rho^E(\mathcal{A}(S), c)\right]$ 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.

D.2 Proof of Theorem 10

Proof. We show that the algorithm $\mathcal{A}$ for PAC-learning monotone conjunctions (see Mohri et al. [2012], chapter 2) is a robust learner for an appropriate choice of sample size. We start with the hypothesis $h(x) = \bigwedge_{i \in I_h} x_i$, where $I_h = [n]$. For each example x in S, we remove i from I_h if $c(x) = 1$ and $x_i = 0$.

Let $\mathcal{D}$ be a class of α-log-Lipschitz distributions. Let $n \in \mathbb{N}$ and $D \in \mathcal{D}_n$. Suppose moreover that the target concept c is a conjunction of l variables. Fix $\varepsilon, \delta > 0$. Let $\eta = \frac{1}{1+\alpha}$, and note that by Lemma 16, for any $S \subseteq [n]$ and $b_S \in \{0,1\}^S$, we have that $\eta^{|S|} \leq \mathbb{P}_{x \sim D}(x_i = b) \leq (1-\eta)^{|S|}$.

Claim 1. If $m \geq \left\lceil \frac{\log n - \log \delta}{\eta^{l+1}} \right\rceil$ then given a sample $S \sim D^m$, algorithm $\mathcal{A}$ outputs c with probability at least $1 - \delta$.

Proof of Claim 1. Fix $i \in \{1, \ldots, n\}$. Algorithm $\mathcal{A}$ eliminates i from the output hypothesis just in case there exists $x \in S$ with $x_i = 0$ and $c(x) = 1$. Now we have $\mathbb{P}_{x \sim D}(x_i = 0 \wedge c(x) = 1) \geq \eta^{l+1}$ and hence

$$\mathbb{P}_{S \sim D}(\forall x \in S \cdot i \text{ remains in } I_h) \leq (1 - \eta^{l+1})^m \leq e^{-m\eta^{l+1}} = \frac{\delta}{n} .$$

The claim now follows from union bound over $i \in \{1, \ldots, n\}$.

Claim 2. If $l \geq \frac{8}{\eta^2} \log(\frac{1}{\varepsilon})$ and $\rho \leq \frac{\eta l}{2}$ then $\mathbb{P}_{x \sim D}(\exists z \in B_\rho(x) \cdot c(z) = 1) \leq \varepsilon$.

Proof of Claim 2. Define a random variable $Y = \sum_{i \in I_c} \mathbb{I}(x_i = 1)$. We simulate Y by the following process. Let $X_1, \ldots, X_l$ be random variables taking value in $\{0,1\}$, and which may be dependent. Let D_i be the marginal distribution on X_i conditioned on $X_1, \ldots, X_{i-1}$. This distribution is also α-log-Lipschitz by Lemma 16, and hence,

$$\mathbb{P}_{X_i \sim D_i}(X_i = 1) \leq 1 - \eta . \tag{9}$$

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

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

The sequence $Z_1, \ldots, Z_l$ is a supermartingale with respect to $X_1, \ldots, X_l$:

$$\mathbb{E}\left[Z_{k+1} \mid X_1, \ldots, X_k\right] = \mathbb{E}\left[Z_k + X'_{k+1} - (1-\eta) \mid X'_1, \ldots, X'_k\right]$$
$$= Z_k + \mathbb{P}\left(X'_{k+1} = 1 \mid X'_1, \ldots, X'_k\right) - (1-\eta)$$
$$\leq Z_k . \qquad \text{(by (9))}$$

Now, note that all Z_k's satisfy $|Z_{k+1} - Z_k| \leq 1$, and that $Z_l = Y - l(1 - \eta)$. We can thus apply the

Azuma-Hoeffding (A.H.) Inequality to get

$$\mathbb{P}\left(Y \geq l - \rho\right) \leq \mathbb{P}\left(Y \geq l(1-\eta) + \sqrt{2\ln(1/\varepsilon)l}\right)$$

$$= \mathbb{P}\left(Z_l - Z_0 \geq \sqrt{2\ln(1/\varepsilon)l}\right)$$

$$\leq \exp\left(-\frac{\sqrt{2\ln(1/\varepsilon)l}^2}{2l}\right) \qquad\qquad\text{(A.H.)}$$

$$= \varepsilon \ ,$$

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

$$l - \rho = (1-\eta)l + \frac{\eta l}{2} + \frac{\eta l}{2} - \rho$$

$$\geq (1-\eta)l + \frac{\eta l}{2} \qquad\qquad\text{(since } \rho \leq \frac{\eta l}{2})$$

$$\geq (1-\eta)l + \sqrt{2\log(1/\varepsilon)l} \ . \qquad\qquad\text{(since } l \geq \frac{8}{\eta^2}\log(\frac{1}{\varepsilon}))$$

This completes the proof of Claim 2.

We now combine Claims 1 and 2 to prove the theorem. Define $l_0 := \max(\frac{2}{\eta}\log n, \frac{8}{\eta^2}\log(\frac{1}{\varepsilon}))$. Define $m := \left\lceil \frac{\log n - \log \delta}{\eta^{l_0+1}} \right\rceil$. Note that m is polynomial in n, δ, ε.

Let h denote the output of algorithm $\mathcal{A}$ given a sample $S \sim D^m$. We consider two cases. If $l \leq l_0$ then, by Claim 1, $h = c$ (and hence the robust risk is 0) with probability at least $1 - \delta$. If $l_0 \leq l$ then, since $\rho = \log n$, we have $l \geq \frac{8}{\eta^2}\log(\frac{1}{\varepsilon})$ and $\rho \leq \frac{\eta l}{2}$ and so we can apply Claim 2. By Claim 2 we have

$$\mathsf{R}_\rho^E(h,c) \leq \mathbb{P}_{x\sim D}\left(\exists z \in B_\rho(x) \cdot c(z) = 1\right) \leq \varepsilon$$

$\square$