On Optimal Learning Under Targeted Data Poisoning

Anonymous Author(s)
Affiliation
Address
email

Abstract

Consider the task of learning a hypothesis class $\mathcal{H}$ in the presence of an adversary that can replace up to an η fraction of the examples in the training set with arbitrary adversarial examples. The adversary aims to fail the learner on a particular target test point x which is *known* to the adversary but not to the learner. In this work we aim to characterize the smallest achievable error $\varepsilon = \varepsilon(\eta)$ by the learner in the presence of such an adversary in both realizable and agnostic settings. We fully achieve this in the realizable setting, proving that $\varepsilon = \Theta(\mathtt{VC}(\mathcal{H}) \cdot \eta)$, where $\mathtt{VC}(\mathcal{H})$ is the VC dimension of $\mathcal{H}$. Remarkably, we show that the upper bound can be attained by a deterministic learner. In the agnostic setting we reveal a more elaborate landscape: we devise a deterministic learner with a multiplicative regret guarantee of $\varepsilon \leq C \cdot \mathtt{OPT} + O(\mathtt{VC}(\mathcal{H}) \cdot \eta)$, where $C > 1$ is a universal numerical constant. We complement this by showing that for any deterministic learner there is an attack which worsens its error to at least $2 \cdot \mathtt{OPT}$. This implies that a multiplicative deterioration in the regret is unavoidable in this case. Finally, the algorithms we develop for achieving the optimal rates are inherently improper. Nevertheless, we show that for a variety of natural concept classes, such as linear classifiers, it is possible to retain the dependence $\varepsilon = \Theta_{\mathcal{H}}(\eta)$ by a proper algorithm in the realizable setting. Here $\Theta_{\mathcal{H}}$ conceals a polynomial dependence on $\mathtt{VC}(\mathcal{H})$.

1 Introduction

A basic goal in machine learning is to develop a predicting model from labeled examples (i.e., training data) that can reliably generalize to unseen examples (i.e., test data). In its simplest form, namely, binary classification, a learner Lrn is given a training set $\mathcal{S} = \{(x_1, y_1), \ldots, (x_n.y_n)\}$, usually assumed to be i.i.d. samples drawn from an unknown distribution D of labeled examples where x_i's are the domain instances (or data points) and $y_i \in \{+1, -1\}$ are the labels. The aim is to produce a mapping $h = \mathsf{Lrn}(\mathcal{S})$ that predicts the labels of fresh samples $(x, y) \sim D$ as accurately as possible, i.e., to minimize the population loss $L_D(h) = \Pr_{(x,y)\sim D}[h(x) \neq y]$. This classical setting has been extensively studied in the last half a century. This accumulated work resulted in fundamental mathematical characterizations regarding the nature of learnability when the training samples are truly i.i.d without any tampering by an adversary [Shalev-Shwartz and Ben-David, 2014]. The goal of this paper is to offer a similar characterization in the presence of an adversary who can tamper with a subset of the training data.

With the emergence of sensitive machine learning applications, it is critical to ensure the trustworthy of such predictive models in the non-ideal scenarios. In this paper, we consider *robust* learnability when the training examples can be altered by an adversary whose goal is to make sure that a target test point will be predicted incorrectly. For instance, consider an adversary who aims to fool a self-driving car to speed up once it observes a stop sign. If such an adversary can somehow influence the training sets used for training the decision rules, she has all the reasons to strategically change them with the specific goal of misleading the self-driving car. As another example, consider a loan applicant

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

who wants to make sure that her loan will be granted. If he can can somehow change the training set used by the bank, he might be able to make his application approved. Note that the training set is the lens through which a learning algorithm obtains information about the underlying learning process. Therefore, once we allow the training examples to be tampered with by an adversary, even slightly, unexpected outcomes may take place. To quantify the robustness of learning algorithms, in this paper, we show how much the outcome of a learning algorithm for a particular target test can be trusted once the training set is being altered.

(PAC) learning under targeted poisoning. More formally, we consider an adversary Adv that is allowed to *replace* η fraction of the training examples $\mathcal{S}$, resulting to a tampered training examples $\mathcal{S}'$ given to the learning algorithm Lrn. Note that even though the training examples $\mathcal{S}$ are all drawn i.i.d. from a distribution D, the tampered training examples $\mathcal{S}'$ do not enjoy this property anymore. Such attackers are also called *poisoning* adversaries [Barreno et al., 2006], and variants of them are previously studied under the name of *malicious* noise [Valiant, 1985, Kearns and Li, 1993] or *nasty* noise [Bshouty et al., 2002]. More specifically, we study poisoning settings in which the adversarial perturbation of the original sample S *can also depend on the final test instance* x. Due to adversary's knowledge of the target test x, such poisoning attacks are sometimes referred to as *targeted* poisoning attacks [Barreno et al., 2006]. Even without any manipulation to the training set, it is too much to ask the learning algorithm to predict correctly all the time while given only a finite number of samples to learn from. In the same vein, we can only hope to design a robust learning algorithm that is correct with high probability over the selection of $(x, y) \sim D$, even if the adversary knows the test instance (x, y) before manipulating the training set $\mathcal{S}$ to $\mathcal{S}'$. Gao et al. [2021], building on ideas from Levine and Feizi [2020], proved that PAC learnability under targeted poisoning attacks is achievable only when $\eta = o(1)$. In other words, when the adversary can only change a *sublinear* $o(n)$ number of n examples, then the optimal learner can achieve error $o(1)$ that goes to zero when the number of examples n goes to infinity.

1.1 Our Results

The prior work leaves several key questions open on the exact parameters of learnability under instance-targeted poisoning. Most importantly, the Gao et al. [2021] does not quantify the error rate when the adversary's budget is $\eta = \Omega(1)$ (e.g., if the adversary can corrupt $n/100$ of the examples). Secondly, Gao et al. [2021] only assume the realizable setting as it is crucial for their results that all the "sub-models" trained using the bagging technique will have error that goes to *zero*. Hence, the question of finding optimal learning rates is left open both for both realizable and agnostic settings. Finally, as the developed robust algorithms are all based on "bagging" they are inherently improper learning technique.

In this work, we make progress on all the directions above and achieve optimal error rates (up to constant factors) for general η, both for the realizable and agnostic settings. We further study the proper nature of the obtained algorithms and give the first proper learning methods that are robust against targeted poisoning attacks for natural hypothesis classes such as linear classifiers.

More precisely, we give a characterization of the optimal error rate of learning under targeted poisoning attacks with budget $\eta \cdot n$ as follows.

Realizable setting. we show that the optimal error[1] is $\Theta(\eta \cdot d)$ where d is the VC dimension of the hypothesis set $\mathcal{H}$. To prove this, we first present an upper bound, showing that a (deterministic) learner can guarantee the error to be at most $O(\eta d)$ under any targeted poisoning attacks of budget ηn. We then also show a matching lower bound (up to a constant factor) as follows. For any *nontrivial* hypothesis class of VC dimension d, we show how to design a distribution D over the examples such that no matter how the learning proceeds, there always exists an adversary of budget ηn that can increase the error (under the targeted attack) to $\Omega(\eta d)$. Our lower bound above holds even if the learning algorithm uses *private randomness* that is not known to the adversary. This model is referred to as the "weak" learning model (under targeted poisoning attacks) in the work of Gao et al. [2021]. Our positive result, however, is deterministic, and so can be seen as satisfying the *strong* learning guarantee, in which the adversary's perturbations to the training set is allowed to depend on the learner's randomness (which in our case does not exist).

[1] Optimal error ε here means that there is a learner that guarantees error $\leq \varepsilon$, and for every learner there is an instance-targeted adversary that increases the error to $\geq \varepsilon$.

Agnostic setting. We also extend our result above to the agnostic setting in which no hypothesis function $h \in \mathcal{H}$ might completely fit to the training data before the attack. In this setting, we show that a similar method can lead to error $O(\mathsf{OPT}) + \delta$ where OPT is the minimum test error of any hypothesis $h \in \mathcal{H}$, and δ only depends on η and the VC dimension of $\mathcal{H}$. A natural question is whether one can achieve adversarial error $\mathsf{OPT} + \delta$, which is the standard way "agnostic learning" is usually defined. We show that this is in fact *not possible*, at least for deterministic learners, by presenting a negative result. In particular we show that for any deterministic learner Lrn, there is an extremely simple hypothesis class (just consisting of two functions) and an input distribution such that the learner is forced to have adversarial error $\geq 2\mathsf{OPT}$. This negative result uses tools from the computational concentration of products [Talagrand, 1995] and the continuity of polynomials.

Proper learning. The deterministic algorithm witnessing the above upper bound is inherently improper which might be a disadvantage in terms of interpretability or test-time computational complexity. In contrast, in (the non-adversarial) PAC setting proper algorithms are known to achieve near optimal learning rates (up to log factors). We therefore explore the cost of proper learning under targeted-poisoning attacks. We show that in many natural settings, such as half spaces, it is indeed possible to properly learn hypothesis classes under instance targeted poisoning attacks. For such proper learning methods, we again use deterministic learning algorithms with error rate $O(d^3\eta)$, whenever the given class is half spaces over $\mathbb{R}^d$. At a technical level, we achieve this result by relying on the *projection number* of the hypothesis set, which is also closely related to the star number of the hypothesis set Bousquet et al. [2020], Kane et al. [2019], Braverman et al. [2019]. In particular, when the projection number of the hypothesis set is small enough, one can project any majority-based hypothesis with high confidence back to the hypothesis set without introducing much error.

1.2 Relation to Certification and Stability

Certification. Robustness to targeted poisoning provides the guarantee that most of the test instances have a stable prediction when the database is changed by a bounded amount in Hamming distance. A stronger guarantee is to *certify* this robustness on a point by point basis. In other words, a certifying predictor provides a bound k along with the prediction label y, such that any sample S' with Hamming distance at most k would lead to the same prediction label y. The certification bound k does not have to be either 0 or a constant k, and it could, in general, depend on the test instance x. So, in a sense, a certifying learner is useful if it provides large certification bounds k for more test instances (Note that it is always possible to provide $k = 0$, since no changes in the training set will not lead to a change in the prediction). Our results extend to certifying predictors that provide $k \approx \eta n$ robustness for $\approx 1 - \varepsilon$ of the test instances.

Connection to stability. We also present a new perspective on targeted poisoning attacks by showing how they can be seen as natural forms of algorithmic stability [Bousquet and Elisseeff, 2002, Rakhlin et al., 2005]. In particular, we show that one can study the adversarial robustness (around the *true* label) to targeted poisoning by decoupling the (pure) stability aspect (which does not depend on the true labels) from the (non-adversarial) risk. We refer to the former as the *prediction stability*. Roughly speaking, prediction stability requires that the model's prediction on x does not change even if the adversary changes the training set withing its budget ηn. Note that here we do not care whether the model's output on x is the correct label or not, and hence is a pure measure of robustness of the predictions. In contrast, the more usual way that stability is studied in learning theory is as follows. For a data sample S of size n, for any *fixed* $i \in [n]$, and for most points x sampled from $(x, y) \sim D$ (by ignoring y), the prediction of the model trained on S and tested on x is likely not changed if one substitutes the i-th sample in S with a *fresh* choice. Prediction stability is a strengthening of this condition in two ways: (1) the choice of what coordinate in S to change can adversarially depend on the test instance, (2) the adversary is allowed to change *more* than one (i.e., up to $\eta \cdot n$) examples.

1.3 Related Work

Poisoning attacks have a history in theoretical learning and are studied under various noise models [Valiant, 1985, Kearns and Li, 1993, Sloan, 1995, Bshouty et al., 2002]. However, these works focus on the *non-targeted* setting in which the adversary does *not* know the target instance. More recently, starting with Diakonikolas et al. [2016], Lai et al. [2016] (also see the survey [Diakonikolas and Kane, 2019]) the *computational* aspects of efficient learning under (non-targeted) poisoning is

studied. In contrast, our work focuses on targeted poisoning, and we study the learning rates *information theoretically* regardless of learner's computing power. The work of Steinhardt et al. [2017] further studied the *certification* of the overall (non-targeted) error. More recently, such (non-targeted) poisoning attacks are combined with *test-time* attacks and are studied under the name of *backdoor* attacks [Gu et al., 2017, Ji et al., 2017].

Most relevant to our setting are the recent works of Gao et al. [2021], Blum et al. [2021] where the general problem of learning (and more quantitative variant of learning error rate) under *instance-targeted* poisoning was formally defined and studied. In particular Blum et al. [2021] studied learnability under instance-targeted poisoning where the adversary can add an *unbounded* number of clean-label examples to the training set. Gao et al. Gao et al. [2021] also showed that when the adversary's corruption is only an $o(1)$ fraction of the training set, PAC learning is possible (if it is possible without the attack). In a concurrent work, Balcan et al. Balcan et al. [2022] study the problem of certifying the prediction of the models to be the *true* label even under instance targeted. In this work, however, our methods can be used to obtain certification of the stability of the model (as done in Gao et al. [2021]), while we also control the overall error (under the targeted attack).

In a closely related line of work, theoretical ideas (such as randomized smoothing) were used to *empirically* study (even certification of) robustness against instance-targeted poisoning. Rosenfeld et al. Rosenfeld et al. [2020] initiated this line of work and showed that randomized smoothing can provide robustness against label-flipping attacks. Subsequently, Levine and Feizi [2020] used deterministic methods that further allowed addition/removal attacks (not just replacing ones), and Chen et al. [2020], Weber et al. [2020], Jia et al. [2020] further developed randomized bagging/subsampling for the goal of resisting targeted poisoning attacks.

Finally, we comment that other theoretical works have also studied instance-targeted poisoning attacks [Mahloujifar and Mahmoody, 2017, Etesami et al., 2020]. These works show how to *amplify* error for specific test instances, say from 0.01 error to 0.5, through targeted poisoning. In particular, these works do not talk about the *fraction* of the test population that is vulnerable to targeted poisoning. The work of Shafahi et al. [2018] studied the power of such attacks empirically.

2 Definitions and Notation

Notation and basic learning theory definitions. We consider the setting of binary classification. Let $\mathcal{X}$ denote the input domain and $\mathcal{Y} = \{0, 1\}$ denote the label-set. A pair $(x, y) \in \mathcal{X} \times \mathcal{Y}$ is called an *example*. A sequence $S = (x_1, y_1), \ldots, (x_n, y_n) \in (\mathcal{X} \times \mathcal{Y})^n$ of n examples is a *sample* of size n. The i'th example in S is denoted by S_i.

A function $h\colon \mathcal{X} \to \mathcal{Y}$ is called an hypothesis or a concept. A set of hypotheses $\mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$ is called an *hypothesis class*, or a *concept class*. We denote the VC-dimension of a concept class $\mathcal{H}$ by $d = d(\mathcal{H})$.

For a set Z, let $Z^* = \cup_n Z^n$ denote the set of all finite sequences with elements from Z. A *learning rule* or *learning algorithm* or *learner* $\mathsf{Lrn}\colon (\mathcal{X} \times \mathcal{Y})^* \to \mathcal{X}^{\mathcal{Y}}$ is a deterministic[2] mapping which takes an input sample $S \in (\mathcal{X} \times \mathcal{Y})^*$ and maps it to a hypothesis $\mathsf{Lrn}(S) = h \in \mathcal{X}^{\mathcal{Y}}$. If it is guaranteed that $\mathsf{Lrn}(S) \in \mathcal{H}$ for all input samples S then Lrn is said to be *proper*; otherwise, it is *improper*.

Let D be a distribution over examples, and let h be an hypothesis. The *population loss* of h with respect to D is defined by $L_D(h) = \Pr_{(x,y)\sim D}[h(x) \neq y] = \mathbb{E}_{(x,y)\sim D}[1[h(x) \neq y]]$. A distribution D is said to be *realizable* by $\mathcal{H}$ if $\inf_{h\in\mathcal{H}} L_D(h) = 0$. Similarly, for a sample S, let $L_S(h) = \frac{1}{|S|}\sum_{i=1}^{n} 1[h(x_i) \neq y_i]$ denote the *empirical error* of h with respect to S, and call a sample realizable by a class $\mathcal{H}$ if there exists $h \in \mathcal{H}$ such that $L_S(h) = 0$. The expected loss (also called risk) of a learning algorithm Lrn w.r.t a distribution D and sample size n is defined by

$$\varepsilon_n(\mathsf{Lrn}|D) := \Pr_{S\sim D^n,(x,y)\sim D}[\mathsf{Lrn}(S)(x) \neq y]\,.$$

The function $n \mapsto \varepsilon_n(\mathsf{Lrn}|D)$ is called the *learning curve*, or *learning rate* of Lrn w.r.t D.

For a real number r, let $\lfloor r \rceil$ denote the nearest integer to r. In case of ties, when $r = k + 1/2$ for some $k \in \mathbb{Z}$, then define $\lfloor r \rceil = k + 1$. For any finite multiset $\mathcal{H}' \subset \mathcal{H}$, denote by $\mathsf{Maj}(\mathcal{H}')$ the

[2] In Appendix B, we extend the definition in a way that captures also a family of randomized learners.

function defined for all $x \in \mathcal{X}$ by

$$\mathsf{Maj}(\mathcal{H}')(x) = \left\lfloor \frac{1}{|\mathcal{H}'|} \sum_{h' \in \mathcal{H}'} h'(x) \right\rfloor .$$

Adversarial risk and prediction stability. Before we introduce the definition of Adversarial risk, we define *Hamming distance* between samples, which is a natural way to quantify distance between samples of equal size.

Definition 2.1 (Hamming distance between samples)**.** Fix $n \in \mathbb{N}$ and let $S, S' \in (\mathcal{X} \times \mathcal{Y})^n$. We define the *Hamming distance* between S and S', denoted by $\mathsf{d_H}(S, S')$ as follows:

$$\mathsf{d_H}(S, S') = \sum_{i=1}^{n} 1[S_i \neq S'_i].$$

Note that the Hamming distance is defined only for samples of equal sizes. If $\mathsf{d_H}(S, S') \leq \eta \cdot n$, we say that S, S' are η-close. For any sample S, let

$$B_\eta(S) := \left\{ S' : \mathsf{d_H}(S, S') \leq \eta \cdot n \right\}.$$

Definition 2.2 (η-adversarial risk)**.** Let $\eta \in (0, 1)$ be the adversary's budget, let Lrn be a learning rule, and let D be a distribution over examples. The *η-adversarial risk* of Lrn w.r.t D and sample size n is defined by

$$\varepsilon_n(\mathsf{Lrn}|D, \eta) := \Pr_{S \sim D^n, (x,y) \sim D} \left[\exists S' \in B_\eta(S) : \mathsf{Lrn}(S')(x) \neq y \right].$$

Note that in the above definition, the set S' is chosen in a way enforces the learner to make a mistake on a targeted data point x. In other words, given a training set S and a target point x, we search for a tampered training set S', withing the budget of the adversary, with the hope of forcing the learning algorithm to make a mistake.

Discussion on potential role of randomness. In Definition 2.2 above we define adversarial risk for *deterministic* learning rules and deterministic models. When the learning and testing process is allowed to use randomness, then the notion of adversarial risk can be extended to the randomized setting in several different ways, depending on whether the adversary can (see and) depend its changes also on the used randomness. If the learner's output model is deterministic and only the learner itself uses randomness r_{Lrn}, then the work of Gao et al. [2021], presented two extensions to Definition 2.2 based on adversary's power to see the randomness (called the strong attack model) or not (called the weak attack model). Here, we observe that when *both* the learner and the output model can be randomized, and if the learner is *improper*, then the whole process of training and testing can be delegated to the "testing" time, and hence it is enough to study randomized predictors. In Definition B.1 present a definition for this general case, in which the adversary *does not* see the randomness. We then prove our result about the *limitations* of the learning rate (Theorem 3.3) in this model. Note that since this is the *weakest* attack model (since the learner and the model can use randomness unknown to the adversary) proving the impossibility using this definition is in fact *stronger*. Furthermore, when it comes to proving positive result, we present a *deterministic* learners in all of our results, which achieves the stronger result (against stronger attackers who can see the randomness) simply because the randomness does not exist.

Decoupling adversarial risk into stability and risk. It is convenient and illustrative to decouple robust learnability to two properties: prediction stability and small expected loss. As we explained briefly earlier, prediction stability is a new notion that is quite different from the usual stability studied in learning theory. Here, the prediction of the learning algorithm is stable against adversarial changes to the training examples. We should also highlight the fact that being prediction stable does not mean that the algorithm is correct on a target point x. It simply means that the prediction does not change when the training set is changed by an adversary. The way we ensure prediction stability is by taking the majority vote over sufficiently many classifiers trained on disjoint samples.

Definition 2.3 (Prediction stability)**.** Let $n \in \mathbb{N}$, $\sigma, \eta \in (0, 1)$. Let Lrn be a learning rule and D be a distribution over examples. We say that the learning rule Lrn is *(n, σ, η)-prediction stable with respect to D* if the following holds

$$\lambda_n(\mathsf{Lrn}|D, \eta) = \Pr_{S \sim D^n, x \sim D_x} \left[\exists S' \in B_\eta(S) : \mathsf{Lrn}(S')(x) \neq \mathsf{Lrn}(S)(x) \right] \leq \sigma.$$

where D_x is the marginal distribution induced by D on the domain $\mathcal{X}$.

Of course, prediction stability alone does not guarantee robust learning as the outputs of a prediction stable algorithm may not correspond to the true labels. At the minimum, the learning algorithm should be a good learner, meaning that at least in the absence of the adversary it should have a small expected loss. In the following theorem, we show that robust learning results from prediction stability and small population risk. In other words, an algorithm whose population risk is small and is not very sensitive to small changes to the training set is indeed a robust learner.

Observation 2.4 (Prediction stability + small error = robust learning)**.** *Let* Lrn *be any learner and* D *any distribution. Then,*

$$\max\{\varepsilon_n(\mathsf{Lrn}|D), \lambda_n(\mathsf{Lrn}|D,\eta)\} \leq \varepsilon_n(\mathsf{Lrn}|D,\eta) \leq \lambda_n(\mathsf{Lrn}|D,\eta) + \varepsilon_n(\mathsf{Lrn}|D).$$

In other words, if Lrn is an (n,σ,η)-prediction stable with respect to D whose expected population loss is $\varepsilon_n(\mathsf{Lrn}|D) \leq \varepsilon$. Then Lrn learns D with an adversarial expected loss $\sigma + \epsilon$. Conversely, if $\varepsilon_n(\mathsf{Lrn}|D,\eta) \leq \varepsilon$ then Lrn is (n,ε,η)-prediction stable with respect to D and its expected population loss is also $\varepsilon_n(\mathsf{Lrn}|D) \leq \varepsilon$.

3 Realizable Setting

The following theorems provide a characterization of the optimal adversarial risk, which is tight up to multiplicative constant factors.

Theorem 3.1 (Realizable case - positive result)**.** *There exists a constant* $c_1 > 0$ *so that the following holds. Let* $\mathcal{H}$ *be a hypothesis class with VC dimension* d *and let* $\eta \in (0,1)$*. Then there exists a learner* Lrn *having* η*-adversarial risk*

$$\varepsilon_n(\mathsf{Lrn}|D,\eta) \leq c_1\eta d$$

for any distribution D *realizable by* $\mathcal{H}$ *and for any sample size* $n \geq 1/\eta$*.*

We prove all the theorems in Appendix A. The positive result is attained by our STABLE PARTITION AND VOTE (or SPV, for short) meta-algorithm, described in figure 1. The meta-algorithm is based on the idea of partitioning and then voting used in [Gao et al., 2021], but with a more refined and precise analysis. We elaborate on this when proving Theorem 3.1. The partition and vote technique works as follows. First, partition the input sample to slices of a carefully chosen size. Then, train a given learner (which is called the *input learner* of SPV) on each slice, and finally let the trained learners vote to determine the output label. The size of each slice trades-off, in a way, expected loss and prediction-stability: if it is too small, the given learner will perform poorly on each slice. On the other hand, if it is relatively large, then there might be too few slices than what is required to attain prediction-stability: A large fraction of the slices might be attacked, and consequently too many learners might give poisoned results.

We are interested in both reasonable learnability and prediction-stability to attain near-optimal adversarial risk, and reflect this interest in the size we choose for each slice, which is $\Theta(1/\eta)$. Note that choosing each slice to be of size $\Theta(1/\eta)$ conceals an assumption that the sample size is $n = \Omega(1/\eta)$. Indeed, throughout the paper we assume that $n \geq 1/\eta$, since otherwise the discussion on learning under poisoning attacks is reduced to the standard learning scenario: it holds that $n < 1/\eta$ if and only if $\eta n < 1$, which means that the adversary's budget is none.

To state the complementing impossibility result, we need the following definition of *non-trivial concept classes* [Bshouty et al., 2002].

Definition 3.2 (Non-trivial concept classes)**.** Let $\mathcal{H}$ be a concept class over a domain $\mathcal{X}$. We say that $\mathcal{H}$ is *non-trivial*, if there are $x_1, x_2 \in \mathcal{X}$ and $h_1, h_2 \in \mathcal{H}$ so that $h_1(x_1) = h_2(x_1)$ and $h_1(x_2) \neq h_2(x_2)$.

Theorem 3.3 (Realizable case - impossibility result)**.** *There exists a constant* $c_2 > 0$ *so that the following holds. Let* $\mathcal{H}$ *be a non-trivial hypothesis class with VC dimension* d *and let* $\eta \in (0,1)$*. Then, there exists a distribution* D *realizable by* $\mathcal{H}$*, so that every learner* Lrn *has* η*-adversarial risk*

$$\varepsilon_n(\mathsf{Lrn}|D,\eta) \geq \min\{c_2\eta d, 1/100\}$$

for any sample size $n \geq 1/\eta$*.*

We prove Theorem 3.3 in Appendix B. In fact, the theorem not only applies to all deterministic learners, but also to a variety of randomized learners captured by a generalization of the definition of η-adversarial risk presented in Appendix B.

SPV: STABLE PARTITION AND VOTE

Input: Stability parameter $\eta \in (0,1)$, a learning algorithm Lrn and an input sample $S \sim D^n$ where $n \geq 1/\eta$.
Output: A classifier $h\colon \mathcal{X} \to \mathcal{Y}$.

1. Partition S into $\lceil 7\eta n \rceil$ consecutive slices such that all first $t = \lfloor 7\eta n \rfloor$ slices are of size at least $\frac{1}{7\eta}$. Denote the i'th part by $S^{(i)}$.
2. For all $i \in [t]$, run the learning algorithm Lrn on $S^{(i)}$ to obtain a hypothesis $h_i = \mathsf{Lrn}(S^{(i)})$.
3. Return the hypothesis h defined as follows for all $x \in \mathcal{X}$:

$$h(x) = \mathsf{Maj}\left(\{h_1, \ldots, h_t\}\right)(x).$$

Figure 1: SPV - A meta algorithm implementing a stable version of the input learning algorithm Lrn.

3.1 Certification

Besides of Prediction-stability, another useful property our SPV meta-algorithm has is the ability to efficiently calculate and output a *certificate* for its predictions. A certificate is a function $\eta : \mathcal{X} \to [0,1]$, outputted by a learner in addition to its prediction, stating that the learner will not change its prediction on the test point x as long as it is given as an input sample some S' which is $\eta(x)$-close to the given input sample S. If one ignores computational considerations, outputting optimal certificates is always possible:

Definition 3.4 (Optimal Certificate)**.** Let Lrn be any learning rule, let S be an input sample. Define the *optimal certificate* $\eta^\star(\cdot) = \eta^\star(\cdot|S)$ of Lrn on input sample S as follows. For any point $x \in \mathcal{X}$, denote $\mathsf{Lrn}(S)(x) = y$. The optimal certificate $\eta^\star(x|S)$ is equal to $\frac{k-1}{n}$ where k the smallest integer such that there exists a sample S' with hamming distance k from S such that $\mathsf{Lrn}(S')(x) \neq y$.

In other words, if S is a sample that was corrupted by an adversary with budget η such that $\eta < \eta^\star(x|S)$ then the output label $\mathsf{Lrn}(S)(x)$ is equal to the label that would have been outputted if the learner was trained with the uncorrupted sample.

The issue with the optimal certificate $\eta^\star(x)$ is that it can be impossible to compute as it requires to iterate over the potentially infinite space of all samples S' of hamming distance at most $n \cdot \eta(x)$ from the input sample S. In contrast, our SPV learner can *efficiently* calculate a non-trivial lower bound on $\eta^\star$ which therefore also serves as a certificate. The key property which enables this is the fact that its output hypothesis is the majority vote of base learners, each trained on a *disjoint* subsample. This is summarized in the following proposition:

Proposition 3.5. *Consider a learner whose output hypothesis is given by a majority vote of t learners $L_1, \ldots, L_t$ that are trained on t disjoint subsamples $S_1, \ldots, S_t$ of the input sample S. Define*

$$\eta(x|S) = \frac{1}{n} \cdot \Big(\frac{\sum_{i\in[t]} 1[h_i(x) = y] - \sum_{i\in[t]} 1[y_i \neq y]}{2} - 1 \Big),$$

where h_i is the output hypothesis of L_i, y is the output label of the majority vote of the L_i's, and n is the size of the input sample S. Then, $\eta(x|S) \leq \eta^\star(x|S)$.

Proof. Notice that $n \cdot \eta(x|S) + 1$ is equal to the minimal number of h_i's whose prediction on x must be flipped in order to enforce that

$$|\{i : h_i(x) = y\}| \leq |\{i : h_i(x) \neq y\}|.$$

Therefore, at least one example in each of the S_i such that $h_i(x) = y$ must be replaced in order to change the prediction of $\mathsf{Lrn}(S)$ on x. In particular, if only $n \cdot \eta(x|S)$ examples are replaced than the prediction of $\mathsf{Lrn}(S)$ on x remains the same. This implies that $\eta^\star(x|S) \geq \eta(x|S)$ as stated. $\square$

In light of Proposition 3.5, our SPV learner can efficiently compute and output a certificate $\eta(x)$ which is proportional to η (where η is the stability parameter given to SPV), with probability proportional to the expected loss of the input learner given to SPV when executed on a sample of size $\left\lceil \frac{1}{7\eta} \right\rceil$.

PSPV: PROPER STABLE PARTITION AND VOTE

Input: Stability parameter $\eta \in (0,1)$, a proper learning algorithm Lrn_p and an input sample $S \sim D^n$ where $n \geq 1/\eta$.
Output: A classifier $h \in \mathcal{H}$.

1. Partition S into $\lceil 5k_p\eta n \rceil$ consecutive slices such that all first $t = \lfloor 5k_p\eta n \rfloor$ slices are of size at least $\frac{1}{5k_p\eta}$. Denote the i'th part by $S^{(i)}$.
2. For all $i \in [t]$, train Lrn_p on $S^{(i)}$ to obtain a hypothesis $h_i = \mathsf{Lrn}_p(S^{(i)})$.
3. Return $h \in \mathcal{H}$ such that
$$h(x) = \mathsf{Maj}\left(\{h_1, \ldots, h_t\}\right)(x)$$
holds for all $x \in \mathcal{X}_{\{h_1,\ldots,h_t\},2k_p}$.

Figure 2: PSPV - A meta-algorithm that implements a stable version of the input proper learning algorithm Lrn_p and maintains properness.

3.2 A Proper Variant of SPV

We indeed design a proper variation of SPV. For the class of linear classifiers, this variation performs well, when fed with the SVM learner. The main result, proved in Appendix C, is the following.

Theorem 3.6. *There exists a constant $c > 0$ so that the following holds. Let $\mathcal{H}$ be the class of halfspaces over $\mathbb{R}^d$ for some $d \geq 1$, and let $\eta \in (0,1)$. Then, there exists a proper learner* Lrn *having η-adversarial risk*
$$\varepsilon_n(\mathsf{Lrn}|D,\eta) \leq c\eta d^3$$
for any distribution D realizable by $\mathcal{H}$ and for any sample size $n \geq 1/\eta$.

To derive the above theorem, we use a mixture of the idea used in the improper SPV, together with the technique of *projecting* a majority vote back to the concept class, which was introduced by Bousquet et al. [2020] and implemented in PSPV using the *projection number* k_p introduced in the same work.

Definition 3.7 (Projection Number)**.** Let $\mathcal{H}$ be a concept class. For any $\ell \geq 2$ and for any multiset $\mathcal{H}' \subset \mathcal{H}$ define the set $\mathcal{X}_{\mathcal{H}',\ell}$ to be the set of all $x \in \mathcal{X}$, for which the number of hypotheses in $\mathcal{H}'$ that disagree with $\mathsf{Maj}(\mathcal{H}')(x)$ is less than $|\mathcal{H}'|/\ell$. The Projection Number of the class $\mathcal{H}$, denoted $k_p = k_p(\mathcal{H})$, is defined to be the smallest ℓ so that for any finite multiset $\mathcal{H}' \subset \mathcal{H}$, there exist $h \in \mathcal{H}$ such that $h(x) = \mathsf{Maj}(\mathcal{H}')(x)$ for all $x \in \mathcal{X}_{\mathcal{H}',\ell}$. If no such ℓ exists then $k_p = \infty$.

4 Agnostic Setting

In this section, we extend the results on robust learnability to the agnostic case. First, by a simple generalization of the positive result for the realizable case, we provide a robust semi-agnostic learner. That is, our learner has adversarial loss depending linearly on $\mathsf{OPT} = \mathsf{OPT}(\mathcal{H}, D) := \min_{h\in\mathcal{H}} L_D(h)$. While semi-agnostic learning is considered not ideal in many cases, we complement our positive result by showing that semi-agnostic learning is unavoidable when the goal is to design a robust and deterministic (as ours) learner for the agnostic setting.

4.1 A Semi-agnostic Learner

Before stating our result about semi-agnostically learning under instance-targeted poisoning, we first discuss how achieving regret $O(d(\mathsf{OPT} + \eta)$ is possible by reduction to the realizable setting.

Reduction to the realizable setting. Suppose a learner is given a training set $\mathcal{S}'$ of size n that comes with ηn rel placements by the adversary from an original set $\mathcal{S}$. Moreover, suppose S is sampled from a distribution D such where $h \in \mathcal{H}$ has OPT error on D. This means that, roughly OPT fraction of S does *not* match with h. Therefore, one can see $\mathcal{S}'$ as first sampled from D (without noise) followed by $\approx (\eta + \mathsf{OPT}) \cdot n$ replacement corruptions. This way, one can employ a learner

that can tolerate $\eta' = \eta + \mathsf{OPT}$ fraction of adversarial corruptions in the realizable setting and obtain total adversarial error $O(d(\mathsf{OPT} + \eta)$.

The above discussion raises the natural question whether a learner can achieve adversarial risk $O(\mathsf{OPT} + d\eta)$ or even (ideally) $\mathsf{OPT} + O(d\eta)$, which is the typical type of error bound in agnostic settings where the goal is to get an additive vanishing regret.

The following theorem, which we prove in Appendix D states the positive result.

Theorem 4.1 (Positive result for the agnostic case)**.** *There exist constants c_1, c_2 so that the following holds. Let $\mathcal{H}$ be a hypothesis class with VC dimension d and let $\eta \in (0, 1)$. Then, there exists a learner* Lrn *having η-adversarial risk*

$$\varepsilon_n(\mathsf{Lrn}|D, \eta) \leq c_2 \cdot \mathsf{OPT} + c_1 \cdot d \cdot \eta$$

for any distribution D over examples and for any sample size $n \geq 1/\eta$.

4.2 Ruling Out Agnostic Learning

Note that Corollary D.2 only proves the existence of a *semi*-agnostic learner under targeted poisoning. A more desirable goal would be to obtain (standard) *agnostic* learners whose error under the attack is simply $\mathsf{OPT} + \psi$ where ψ is a vanishing (additive) error term when $\eta \to 0$. Here we will prove that at least when it comes to *deterministic* learners, such a goal is out of reach, and the best we can hope for is $2\mathsf{OPT}$ plus additive terms that depend on η and the VC dimension. This partly explains why we only achieve a semi-agnostic learner.

The following theorem, which we prove in Appendix E shows that in Theorem 4.1, the constant c_1 needs to be at least 2, and so the standard way agnostic learners bound their regret is not possible for instance-targeted poisoning.

Theorem 4.2 (Impossibility of agnostic learning)**.** *Let $\eta' \in (0, 1), n \in \mathbb{N}$. For any hypothesis class $\mathcal{H}$ that has at least two hypotheses and any deterministic learner, there is a distribution D over (two) examples and $\eta = \eta' + \widetilde{O}(1/\sqrt{n})$ such that* Lrn *has η-adversarial risk*

$$\varepsilon_n(\mathsf{Lrn}|D, \eta) \geq 2\mathsf{OPT} + \Omega(\eta') - O(1/n).$$

5 Conclusion and Open questions

In this work, we studied the optimal rate of learning for binary classification problems under instance-targeted poisoning. We showed that in the realizable setting the error rate can be characterized up to a constant factor and is proportional both to adversary's budget and the VC dimension of the class. In the agnostic setting, we proved a perhaps surprising lower bound that standard agnostic learning (with additive regret compared to the optimal error in the no-attack setting) is impossible for deterministic learners, and also complemented this with a positive result using a *semi-agnostic* learner. We also showed how to make our learners proper in a variety of interesting settings.

Our work leaves a few interesting directions for future research.

- **Finding the exact constant in the realizable case.** Our results in the realizable case characterize the optimal adversarial risk up to a constant multiplicative factor in the sense that there exist constants c_1, c_2 so that achieving η-adversarial risk of $c_1\eta d$ is possible for any hypothesis class with VC-dimension d, whereas obtaining η-adversarial risk of $c_2\eta d$ can't be achieved for any hypothesis class with VC-dimension d. However, there is a large gap between c_1, c_2. Can we close or shrink this gap?
- **Finding the correct multiplicative factor in the agnostic case.** Our results show that in the agnostic case, there must be a constant $C \geq 2$ so that the best adversarial risk attainable is $C \cdot \mathsf{OPT}$. What is the value of C?
- **Characterizing proper robust learning.** In the proper and realizable case, our stable learner for linear classifiers depends on d^3, while our lower bound depends linearly on d, as in the general improper case. It remains open to identify the correct dependence on d.
- **Characterizing the role of randomness.** Our impossibility result for the agnostic learning (Theorem 4.2) only applies to deterministic learners. It remains open to either effectively use randomness during the learning (known or unknown to the adversary) and obtain an agnostic learner, or to extend the negative result to cover such randomized learners as well.

References

Maria-Florina Balcan, Avrim Blum, Steve Hanneke, and Dravyansh Sharma. Robustly-reliable learners under poisoning attacks. *arXiv preprint arXiv:2203.04160*, 2022.

Marco Barreno, Blaine Nelson, Russell Sears, Anthony D Joseph, and J Doug Tygar. Can machine learning be secure? In *Proceedings of the 2006 ACM Symposium on Information, computer and communications security*, pages 16–25. ACM, 2006.

Avrim Blum, Steve Hanneke, Jian Qian, and Han Shao. obust learning under clean-label attack. In *Conference on Learning Theory*, 2021.

Olivier Bousquet and André Elisseeff. Stability and generalization. *The Journal of Machine Learning Research*, 2:499–526, 2002.

Olivier Bousquet, Steve Hanneke, Shay Moran, and Nikita Zhivotovskiy. Proper learning, helly number, and an optimal svm bound. In *Conference on Learning Theory*, pages 582–609. PMLR, 2020.

Mark Braverman, Gillat Kol, Shay Moran, and Raghuvansh R Saxena. Convex set disjointness, distributed learning of halfspaces, and lp feasibility. *arXiv preprint arXiv:1909.03547*, 2019.

Nader H Bshouty, Nadav Eiron, and Eyal Kushilevitz. Pac learning with nasty noise. *Theoretical Computer Science*, 288(2):255–275, 2002.

Ruoxin Chen, Jie Li, Chentao Wu, Bin Sheng, and Ping Li. A framework of randomized selection based certified defenses against data poisoning attacks, 2020.

Ilias Diakonikolas and Daniel M. Kane. Recent advances in algorithmic high-dimensional robust statistics, 2019.

Ilias Diakonikolas, Gautam Kamath, Daniel M Kane, Jerry Li, Ankur Moitra, and Alistair Stewart. Robust estimators in high dimensions without the computational intractability. In *Foundations of Computer Science (FOCS), 2016 IEEE 57th Annual Symposium on*, pages 655–664. IEEE, 2016.

Omid Etesami, Saeed Mahloujifar, and Mohammad Mahmoody. Computational concentration of measure: Optimal bounds, reductions, and more. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 345–363. SIAM, 2020.

Ji Gao, Amin Karbasi, and Mohammad Mahmoody. Learning and certification under instance-targeted poisoning. In *Uncertainty in Artificial Intelligence*, pages 2135–2145. PMLR, 2021.

Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*, 2017.

David Haussler, Nick Littlestone, and Manfred K Warmuth. Predicting $\{0, 1\}$-functions on randomly drawn points. *Information and Computation*, 115(2):248–292, 1994.

Yujie Ji, Xinyang Zhang, and Ting Wang. Backdoor attacks against learning systems. In *2017 IEEE Conference on Communications and Network Security (CNS)*, pages 1–9. IEEE, 2017.

Jinyuan Jia, Xiaoyu Cao, and Neil Zhenqiang Gong. Intrinsic certified robustness of bagging against data poisoning attacks. *arXiv preprint arXiv:2008.04495*, 2020.

Daniel Kane, Roi Livni, Shay Moran, and Amir Yehudayoff. On communication complexity of classification problems. In *Conference on Learning Theory*, pages 1903–1943. PMLR, 2019.

Michael Kearns and Ming Li. Learning in the presence of malicious errors. *SIAM Journal on Computing*, 22(4):807–837, 1993.

Kevin A Lai, Anup B Rao, and Santosh Vempala. Agnostic estimation of mean and covariance. In *Foundations of Computer Science (FOCS), 2016 IEEE 57th Annual Symposium on*, pages 665–674. IEEE, 2016.

Alexander Levine and Soheil Feizi. Deep partition aggregation: Provable defenses against general poisoning attacks. In *International Conference on Learning Representations*, 2020.

Philip M Long. The complexity of learning according to two models of a drifting environment. *Machine Learning*, 37(3):337–354, 1999.

Saeed Mahloujifar and Mohammad Mahmoody. Blockwise p-tampering attacks on cryptographic primitives, extractors, and learners. In *Theory of Cryptography Conference*, pages 245–279. Springer, 2017.

Alexander Rakhlin, Sayan Mukherjee, and Tomaso Poggio. Stability results in learning theory. *Analysis and Applications*, 3(04):397–417, 2005.

Elan Rosenfeld, Ezra Winston, Pradeep Ravikumar, and Zico Kolter. Certified robustness to label-flipping attacks via randomized smoothing. In *International Conference on Machine Learning*, pages 8230–8241. PMLR, 2020.

Ali Shafahi, W Ronny Huang, Mahyar Najibi, Octavian Suciu, Christoph Studer, Tudor Dumitras, and Tom Goldstein. Poison frogs! targeted clean-label poisoning attacks on neural networks. *Advances in neural information processing systems*, 31, 2018.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

Robert H. Sloan. Four Types of Noise in Data for PAC Learning. *Information Processing Letters*, 54 (3):157–162, 1995.

Jacob Steinhardt, Pang Wei Koh, and Percy Liang. Certified defenses for data poisoning attacks. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 3520–3532, 2017.

Michel Talagrand. Concentration of measure and isoperimetric inequalities in product spaces. *Publications Mathématiques de l'Institut des Hautes Etudes Scientifiques*, 81(1):73–205, 1995.

Leslie G Valiant. Learning disjunction of conjunctions. In *IJCAI*, pages 560–566, 1985.

Vladimir Vapnik and Alexey Chervonenkis. Theory of pattern recognition. 1974.

Maurice Weber, Xiaojun Xu, Bojan Karlas, Ce Zhang, and Bo Li. Rab: Provable robustness against backdoor attacks. *arXiv preprint arXiv:2003.08904*, 2020.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [No]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [N/A]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [N/A]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [N/A]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]