Generalization Properties of Adversarial Training for ℓ_0-Bounded Adversarial Attacks

Payam Delgosha[1], Hamed Hassani[2], and Ramtin Pedarsani[3]

2023 IEEE Information Theory Workshop (ITW) | 979-8-3503-0149-6/23/$31.00 ©2023 IEEE | DOI: 10.1109/ITW55543.2023.10161648

***Abstract*—We have widely observed that neural networks are vulnerable to small additive perturbations to the input causing misclassification. In this paper, we focus on the ℓ_0-bounded adversarial attacks, and aim to theoretically characterize the performance of adversarial training for an important class of *truncated* classifiers. Such classifiers are shown to have strong performance empirically, as well as theoretically in the Gaussian mixture model, in the ℓ_0-adversarial setting. The main contribution of this paper is to prove a novel generalization bound for the binary classification setting with ℓ_0-bounded adversarial perturbation that is distribution-independent. Deriving a generalization bound in this setting has two main challenges: (i) the truncated inner product which is highly non-linear; and (ii) maximization over the ℓ_0 ball due to adversarial training is non-convex and highly non-smooth. To tackle these challenges, we develop new coding techniques for bounding the combinatorial dimension of the truncated hypothesis class.**

I. Introduction

It is well-known that machine learning models are susceptible to adversarial attacks that can cause classification error. These attacks are typically in the form of a small norm-bounded perturbation to the input data that are carefully designed to incur misclassification–e.g. they can be an additive ℓ_p-bounded perturbation for some $p \geq 0$ [1], [2], [3], [4].

There is an extensive body of prior work studying adversarial machine learning, most of which have focused on ℓ_2 and ℓ_∞ attacks [5], [6], [7], [8], [9]. To train models that are more robust against such attacks, adversarial training is the state-of-the-art defense method. However, the success of the current adversarial training methods is mainly based on empirical evaluations [10]. It is therefore imperative to study the fundamental limits of robust machine learning under different classification settings and attack models.

In this paper, we focus on the case of ℓ_0-bounded attacks that has been less investigated so far. In such attacks, given an ℓ_0 budget k, an adversary can change k entries of the input vector in an arbitrary fashion – i.e. the adversarial perturbations belong to the ℓ_0-ball of radius k. In contrast with ℓ_p-balls ($p \geq 1$), the ℓ_0-ball is non-convex and non-smooth. Moreover, the ℓ_0-ball contains inherent discrete (combinatorial) structures that can be exploited by both the learner and the adversary. As a result, the ℓ_0-adversarial setting bears various challenges that are absent in common ℓ_p-adversarial settings. In this regard, it has recently been shown that any piece-wise linear classifier, e.g. a feed-forward neural network with ReLu activations, completely fails in the ℓ_0 setting [11].

Perturbing only a few components of the data or signal has many real-world applications including natural language processing [12], malware detection [13], and physical attacks in object detection [14]. There have been several prior works on ℓ_0-adversarial attacks including white-box attacks that are gradient-based, e.g. [4], [15], [16], and black-box attacks based on zeroth-order optimization, e.g. [17], [18]. Defense strategies against ℓ_0-bounded attacks have also been proposed, e.g. defenses based on randomized ablation [19] and defensive distillation [20]. None of the above works have studied the fundamental limits of the ℓ_0-adversarial setting theoretically.

Recently, [21] proposed a classification algorithm called `FilTrun` and showed that it is robust against ℓ_0 adversarial attacks in a Gaussian mixture setting. Specifically, they show that asymptotically as the data dimension gets large, no other classification algorithm can do better than `FilTrun` in the presence of adversarial attacks. Their algorithm consists of two component, namely truncation and filteration. Although truncation can be efficiently implemented, filteration is computationally expensive. Later, [22] proposed that employing truncation in a neural network architecture together with adversarial training results in a classification algorithm which is robust against ℓ_0 attacks. They proved this for the Gaussian mixture setting in an asymptotic scenario as the data dimension goes to infinity. Furthermore, they demonstrated the effectiveness of their proposed method against ℓ_0 adversarial attacks through experiments.

In the previous theoretical results in ℓ_0–bounded adversarial attacks, it is assumed that the data distribution is in the form of a Gaussian mixture with known parameters, and the focus is on showing the asymptotic optimality of the proposed architecture as the data dimension goes to infinity. In practical supervised learning scenarios, we usually have indirect access to the distribution through i.i.d. training data samples. In this setting, adversarial training is a natural method for learning model parameters that are robust against adversarial attacks, as shown empirically in the prior work.

Motivated by the theoretical and empirical success of truncation against ℓ_0 adversarial attacks, in this paper we study its generalization properties. Generalization properties of adversarial training have been studied for other adversarial settings (for instance [23], [24], [25], [26], [27], [28], [29], [30], [31], and [32]) mainly involving $\ell_p, p \geq 1$. There are challenges inherent in the ℓ_0 setting which make the standard techniques inapplicable. In this paper, we discuss these challenges and

We thank the NSF grants CNS-2003035, CIF-1909320, and CIF-2236484.
[1]Uni. of Illinois at Urbana-Champaign, `delgosha@illinois.edu`
[2]Uni. of Pennsylvania, `hassani@seas.upenn.edu`
[3]Uni. of California Santa Barbara, `ramtin@ucsb.edu`

develop novel techniques to address this problem. We believe that the proposed mathematical techniques in this work are of independent interest and potentially have applications in other generalization settings which are combinatorial in nature, such as neural network architectures equipped with truncation components for robustness purposes.

Summary of Contributions. Our main contributions are as follows:

- We consider a binary classification setting in the presence of an ℓ_0 adversary with truncated linear classifiers as our hypothesis class. We prove a generalization bound in this setting that is distribution-independent, i.e. it holds for any distribution on the data (see Theorem 1 in Section III).
- We observe that due to the complex and combinatorial nature of our problem, the classical techniques for bounding the combinatorial dimension and the VC dimension are not applicable to our setting (see the discussion in Section III). To this end, we introduce novel techniques that may be generalized to problems involving non-linear and combinatorial operations.
- Specifically, there are two key challenges in bounding the combinatorial dimension in our setting: (a) the truncated inner product which is highly non-linear, and (b) the inner maximization over the ℓ_0 ball due to adversarial training, which is challenging to work with as it is non-convex and highly non-smooth. It is worth mentioning that as [25] has shown, it is possible that the original hypothesis class (truncated inner products in our case) has a finite VC dimension, but the corresponding adversarial setting is only PAC learnable with an improper learning rule. Therefore, it is crucial in our work to resolve the two challenges individually and show that the VC dimension is finite even in the adversarial setting proving proper robust PAC learnability.
- We tackle the first challenge by employing a novel *coding* technique, which encodes the sign of the truncated inner product by a finite number of conventional inner products. This enables us to bound the growth function using the known bounds on the VC dimension of conventional inner product (see Proposition 1 in Section III). We tackle the second challenge by decomposing our loss function into two terms, one which does not involve maximization over the ℓ_0, and one which involves studying the range of the truncated inner product over the ℓ_0 ball (see the discussion in Section III-B and Propositions 2).

In Section II, we formulate the problem and give an overview of prior results, in Section III, we give the main results and highlights the proof techniques, and in Section IV we conclude the paper.

II. Problem Formulation

We consider the binary classification problem where the true label is denoted by $y \in \{\pm 1\}$, and the feature vector has dimension d and is denoted by $\boldsymbol{x} \in \mathbb{R}^d$. We denote the joint distribution of $(\boldsymbol{x}, y)$ by $\mathcal{D}$. A classifier is a function $\mathcal{C} : \mathbb{R}^d \to \{\pm 1\}$ which predicts the label from the input. We consider the 0-1 loss $\ell(\mathcal{C}; \boldsymbol{x}, y) := \mathbb{1}[\mathcal{C}(\boldsymbol{x}) \neq y]$. We study classification under ℓ_0 perturbations; i.e. the adversary can perturb the input $\boldsymbol{x}$ to $\boldsymbol{x}' \in \mathbb{R}^d$ where the ℓ_0 distance between the two vectors defined as

$$\|\boldsymbol{x} - \boldsymbol{x}'\|_0 := \sum_{i=1}^{d} \mathbb{1}[x_i \neq x_i'],$$

is bounded. In other words, the adversary can modify the input $\boldsymbol{x}$ to any other vector $\boldsymbol{x}'$ within the ℓ_0 ball of radius k around $\boldsymbol{x}$ defined as

$$\mathcal{B}_0(\boldsymbol{x}, k) := \{\boldsymbol{x}' \in \mathbb{R}^d : \|\boldsymbol{x} - \boldsymbol{x}'\|_0 \leq k\}.$$

Here, k is the *budget* of the adversary, and is effectively the number of input coordinates that the adversary is allowed to change. The robust classification error (or robust error for short) of a classifier $\mathcal{C}$ when the adversary has ℓ_0 budget k is defined as

$$\mathcal{L}_{\mathcal{D}}(\mathcal{C}, k) := \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[\widetilde{\ell}_k(\mathcal{C}; \boldsymbol{x}, y)\right], \tag{1}$$

where

$$\widetilde{\ell}_k(\mathcal{C}; \boldsymbol{x}, y) := \max_{\boldsymbol{x}' \in \mathcal{B}_0(\boldsymbol{x},k)} \ell(\mathcal{C}; \boldsymbol{x}', y). \tag{2}$$

Here, $(\boldsymbol{x}, y) \sim \mathcal{D}$ means that the feature vector-label pair $(\boldsymbol{x}, y)$ has distribution $\mathcal{D}$, and the maximum represents the adversary which can perturb the input vector $\boldsymbol{x}$ arbitrarily within the ℓ_0 ball $\mathcal{B}_0(\boldsymbol{x}, k)$.

Overview of Prior Results. The authors of [21] study the above problem in the setting of the Gaussian mixture model. More precisely, they assume that $y \sim \text{Unif}\{\pm 1\}$, and conditioned on y, $\boldsymbol{x} \sim \mathcal{N}(y\boldsymbol{\mu}, \Sigma)$ is normally distributed with mean $y\boldsymbol{\mu}$ and covariance matrix Σ. Here, $\boldsymbol{\mu} \in \mathbb{R}^d$, and Σ is a positive-definite matrix. They study this problem in an asymptotic fashion as the dimension d goes to infinity. They propose an algorithm called `FilTrun` and prove that it is asymptotically optimal when Σ is diagonal. Here, asymptotic optimality means that asymptotically as the dimension d goes to infinity, the robust error of `FilTrun` gets close to the optimal robust error, defined as the minimum of the robust error over all possible classifiers. `FilTrun` makes use of two components, namely Filteration and Truncation.

- **Filteration** refers to a preprocessing phase, where upon receiving the perturbed data vector $\boldsymbol{x}'$, we remove certain coordinates, or effectively set them to zero. The purpose of filteration is to remove the *non-robust* coordinates. Let us denote the output of the filteration phase by $\widetilde{\boldsymbol{x}}'$.
- **Truncation** refers to applying the *truncated inner product* of an appropriate weight vector $\boldsymbol{w}$ by the output of the filteration phase $\widetilde{\boldsymbol{x}}'$. More precisely, the weight vector $\boldsymbol{w}$ is chosen appropriately based on the distribution parameters $\boldsymbol{\mu}$ and Σ, and the classification output is computed based as the sign of the truncated inner product $\langle \boldsymbol{w}, \widetilde{\boldsymbol{x}}' \rangle_k$ defined as follows. Let $\boldsymbol{u} := \boldsymbol{w} \odot \widetilde{\boldsymbol{x}}'$ be the coordinate-wise product of $\boldsymbol{w}$ and $\widetilde{\boldsymbol{x}}'$, and let $u_{(1)} \leq u_{(2)} \leq \cdots \leq u_{(d)}$ be the values in $\boldsymbol{u}$ after sorting. Then, $\langle \boldsymbol{w}, \widetilde{\boldsymbol{x}}' \rangle_k := \sum_{i=k+1}^{d-k} u_{(i)}$.

Effectively, $\langle \boldsymbol{w}, \widetilde{\boldsymbol{x}}' \rangle_k$ is the summation of the values in the coordinate-wise product of $\boldsymbol{w}$ and $\widetilde{\boldsymbol{x}}'$ after removing the k largest and the k smallest values. Note that when $k = 0$, this reduces to the usual inner product, and removing the top and bottom k values effectively removes the *outliers* in the input, which are possibly caused by the adversary.

Although truncation can be implemented in a computationally efficient way, filteration turns out to be computationally expensive. The authors in [22] have shown that in the Gaussian mixture setting with diagonal covariance matrix, optimizing for the weight vector $\boldsymbol{w}$ in the class of truncated classifiers of the form $\mathcal{C}_{\boldsymbol{w}}^{(k)}(\boldsymbol{x}') := \operatorname{sgn}(\langle \boldsymbol{w}, \boldsymbol{x}' \rangle_k)$ results in an asymptotically optimal classifier as the data dimension goes to infinity. Motivated by this, they propose a neural network architecture where the inner products in the first layer are replaced by truncated inner products. Furthermore, they show through several experiments that in practice when we do not have access to the distribution parameters, adversarial training as a proxy for optimizing the model parameters results in an efficient and robust classifier.

Adversarial Training for Parameter Tuning. In the theoretical analysis in above mentioned works, it is assumed that the distribution $\mathcal{D}$ is in the form of a Gaussian mixture with known parameters, and therefore we can optimize for the model parameters within the proposed architecture. In practice, we usually do not have access to the distribution $\mathcal{D}$. Instead, we usually have i.i.d. training data samples $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)$ distributed according to $\mathcal{D}$. We stick to the usual setting in the adversarial attacks framework, where the training data is clean, while the test data is perturbed by the adversary, and the objective is the robust error at the test time. In this setting, adversarial training is a natural choice for finding the model parameters. Motivated by the prior work described above, we consider the hypothesis class of truncated linear classifiers of the form $\mathcal{C}_{\boldsymbol{w}}^{(k)} : \boldsymbol{x} \mapsto \operatorname{sgn}(\langle \boldsymbol{w}, \boldsymbol{x} \rangle_k)$, parametrized by $\boldsymbol{w} \in \mathbb{R}^d$, where for $\alpha \in \mathbb{R}$, $\operatorname{sgn}(\alpha) := +1$ when $\alpha \geq 0$, and $\operatorname{sgn}(\alpha) := -1$ when $\alpha < 0$. Also, motivated by the prior work mentioned above, we set k to be equal to the adversary's budget. Furthermore, note that we are comparing $\langle \boldsymbol{w}, \boldsymbol{x} \rangle_k$ with zero. This is without loss of generality, since we may assume that there is a coordinate in $\boldsymbol{x}$ with constant value 1. Since we focus on this hypothesis class, we use the shorthand notation $\mathcal{L}_{\mathcal{D}}(\boldsymbol{w}, k)$ for the robust error of the classifier $\mathcal{C}_{\boldsymbol{w}}^{(k)}$, i.e. for $\boldsymbol{w} \in \mathbb{R}^d$, we define $\mathcal{L}_{\mathcal{D}}(\boldsymbol{w}, k) := \mathcal{L}_{\mathcal{D}}(\mathcal{C}_{\boldsymbol{w}}^{(k)}, k)$.

Adversarial training in this scenario translates to choosing the hypothesis parameter $\boldsymbol{w} \in \mathbb{R}^d$ by minimizing the adversarial empirical loss

$$\widehat{\mathcal{L}}_n(\boldsymbol{w}, k) := \frac{1}{n} \sum_{i=1}^{n} \widetilde{\ell}_k(\mathcal{C}_{\boldsymbol{w}}^{(k)}; \boldsymbol{x}_i, y_i). \tag{3}$$

Recall that $\widetilde{\ell}_k(\mathcal{C}_{\boldsymbol{w}}^{(k)}; \boldsymbol{x}_i, y_i)$ which was defined in (2) is the maximum zero-one loss over the ℓ_0 ball around the ith data sample. Effectively, we assume that we have access to a perfect adversary during the training phase which allows us to have access to $\widetilde{\ell}_k(\mathcal{C}_{\boldsymbol{w}}^{(k)}; \boldsymbol{x}_i, y_i)$. Let

$$\widehat{\boldsymbol{w}}_n \in \arg\min_{\boldsymbol{w} \in \mathbb{R}^d} \widehat{\mathcal{L}}_n(\boldsymbol{w}, k), \tag{4}$$

be the hypothesis parameter vector which is obtained by optimizing the above adversarial empirical loss over the training dataset. In this paper, we analyze the generalization properties of the adversarial training for the above hypothesis class of linear truncated classifiers. More precisely, our main question is whether the robust error corresponding to $\widehat{\boldsymbol{w}}_n$ (which is obtained by employing adversarial training) converges to the robust error of the best truncated linear classifier in our hypothesis class. More precisely, if

$$\boldsymbol{w}^* \in \arg\min_{\boldsymbol{w} \in \mathbb{R}^d} \mathcal{L}_{\mathcal{D}}(\mathcal{C}_{\boldsymbol{w}}^{(k)}, k), \tag{5}$$

correspond to the best classifier in our hypothesis class, can we show that as the number of samples n goes to infinity, $\mathcal{L}_{\mathcal{D}}(\widehat{\boldsymbol{w}}_n, k)$ converges to $\mathcal{L}_{\mathcal{D}}(\boldsymbol{w}^*, k)$? This question is formalized in the following definition.

Definition 1 (robust PAC learnability)**.** *We say that a hypothesis class $\mathcal{H}$ is robust PAC learnable with respect to an ℓ_0 adversary with budget k, if there exists a learning algorithm $\mathcal{A}$ such that for any $\epsilon, \delta > 0$, and for any distribution $\mathcal{D}$, $\mathcal{A}$ maps i.i.d. data samples $\mathcal{S} = ((\boldsymbol{x}_i, y_i), 1 \leq i \leq n)$ to $\mathcal{A}(\mathcal{S}) \in \mathcal{H}$, such that if $n > m(\epsilon, \delta)$, with probability at least $1 - \delta$, we have*

$$\mathcal{L}_{\mathcal{D}}(\mathcal{A}(\mathcal{S}), k) \leq \inf_{h \in \mathcal{H}} \mathcal{L}_{\mathcal{D}}(h, k) + \epsilon.$$

Notation. $[n]$ denotes the set $\{1, \ldots, n\}$. We denote vectors with boldface notation. Given a vector $\boldsymbol{u} = (u_1, \ldots, u_d) \in \mathbb{R}^d$, we denote by $u_{(1)} \leq \cdots \leq u_{(d)}$ the vector containing elements in $\boldsymbol{u}$ in a non-decreasing order. Given $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^d$, $\boldsymbol{u} \odot \boldsymbol{w} \in \mathbb{R}^d$ is defined to be the element-wise product of $\boldsymbol{u}$ and $\boldsymbol{v}$, i.e. its ith coordinate is $u_i v_i$. For vectors $\boldsymbol{w}, \boldsymbol{x} \in \mathbb{R}^d$ and integer $k < d/2$, with $\boldsymbol{u} := \boldsymbol{w} \odot \boldsymbol{x}$, the k–truncated inner product $\langle \boldsymbol{w}, \boldsymbol{x} \rangle_k$ between $\boldsymbol{w}$ and $\boldsymbol{x}$ is defined by $\sum_{i=k+1}^{d-k} u_{(i)}$.

III. Main Results

Our main result is to show that the class of truncated inner products in our adversarial setting is robustly PAC learnable as formalized in Definition 1. This is a direct consequence of the following Theorem 1. In the remaining of this section, we explain the ideas and main steps of the proof.

Theorem 1. *For any joint distribution $\mathcal{D}$ on the label $y \in \{\pm 1\}$ and feature-vector $\boldsymbol{x} \in \mathbb{R}^d$, and any adversarial budget $0 < k < d/2$, for $n > d + 1$, if $\widehat{\boldsymbol{w}}_n$ denotes the model parameters obtained from adversarial training as in (4), with probability at least $1 - \delta$, we have*

$$\mathcal{L}_{\mathcal{D}}(\widehat{\boldsymbol{w}}_n, k) \leq \mathcal{L}_{\mathcal{D}}(\boldsymbol{w}^*, k) + c\sqrt{\frac{d\left[\binom{d}{2k} + \binom{d}{2}\right] \log \frac{en}{d}}{n}} + 5\sqrt{\frac{2 \log \frac{8}{\delta}}{n}},$$

where c is a universal constant.

In order to obtain this bound, it suffices to bound the Rademacher complexity of our hypothesis class composed

with the loss $\widetilde{\ell}$ defined in (2) and use standard bounds such as [33, Theorem 26.5]. Let $\widetilde{\mathcal{T}}_{d,k} \subset \{-1,+1\}^{\mathbb{R}^d \times \{\pm 1\}}$ be the class of functions $\widetilde{T}_{\boldsymbol{w},k}$ parametrized by $\boldsymbol{w} \in \mathbb{R}^d$ obtained by applying the loss $\widetilde{\ell}$ to the truncated linear classifier $\mathcal{C}_{\boldsymbol{w}}^{(k)}$, i.e.

$$\widetilde{T}_{\boldsymbol{w},k}(\boldsymbol{x},y) := \widetilde{\ell}_k(\mathcal{C}_{\boldsymbol{w}}^{(k)};\boldsymbol{x},y) = \max_{\boldsymbol{x}' \in \mathcal{B}_0(\boldsymbol{x},k)} \mathbb{1}\left[y \neq \operatorname{sgn}(\langle \boldsymbol{w},\boldsymbol{x}'\rangle_k)\right]. \tag{6}$$

Note that since the loss $\widetilde{\ell}$ is the maximum of a zero-one loss, the range of the functions in $\widetilde{\mathcal{T}}_{d,k}$ is indeed $\{-1,+1\}$. To simplify the notation, with $\mathcal{Z} := \mathbb{R}^d \times \{\pm 1\}$, we denote the feature vector-label pair $(\boldsymbol{x},y)$ by $\boldsymbol{z} \in \mathcal{Z}$. In general, the Rademacher complexity of a function class $\mathcal{F} \subset [-1,1]^{\mathcal{Z}}$ is defined to be

$$R_n(\mathcal{F}) := \mathbb{E}\left[\sup_{f\in\mathcal{F}} \left|\frac{1}{n}\sum_{i=1}^{n} \epsilon_i f(\boldsymbol{z}_i)\right|\right], \tag{7}$$

where expectation is taken with respect to i.i.d. Rademacher random variables $\epsilon_i \in \{\pm 1\}$ and i.i.d. samples $\boldsymbol{z}_i = (\boldsymbol{x}_i, y_i)$ with law $\mathcal{D}$. In the classification setting, where the function class is of the form $\mathcal{F} \subset \{\pm 1\}^{\mathcal{Z}}$, we have from Massart lemma (see, for instance, [33, Lemma 26.8]), that

$$R_n(\mathcal{F}) \leq \sqrt{\frac{2\log(\Pi_{\mathcal{F}}(n))}{n}}, \tag{8}$$

where

$$\Pi_{\mathcal{F}}(n) := \max\{|\{(f(z_1),\ldots,f(z_n) : f \in \mathcal{F}\}| : z_i \in \mathcal{Z}, 1 \leq i \leq n\}.$$

is called the *growth function* of $\mathcal{F}$. Motivated by this, our strategy is to find a bound for $\Pi_{\widetilde{\mathcal{T}}_{d,k}}(n)$ which is polynomial in n.

Note that from (6), there are two challenges for bounding the combinatorial dimension of the functions $\widetilde{T}_{\boldsymbol{w},k}$: (a) the truncated inner product $\langle \boldsymbol{w}, x\rangle_k$, and (b) the maximization over the ℓ_0 ball $\mathcal{B}_0(\boldsymbol{x},k)$. These two components bring fundamental challenges beyond those present in the usual machine learning scenarios where we deal with the usual inner product and ℓ_p norms for $p \geq 1$. More precisely,

1) The truncated inner product $\langle \boldsymbol{w},\boldsymbol{x}\rangle_k$ is not linear, i.e. $\langle \boldsymbol{w},\boldsymbol{x}_1+\boldsymbol{x}_2\rangle_k$ is not necessarily equal to $\langle \boldsymbol{w},\boldsymbol{x}_1\rangle_k + \langle \boldsymbol{w},\boldsymbol{x}_2\rangle_k$.
2) The ℓ_0 ball $\mathcal{B}_0(\boldsymbol{x},k)$ is unbounded, non-convex, and non-smooth. Due to this, maximization over the ball is not tractable, unlike the case of ℓ_p balls for $p \geq 1$ (for instance [24] in the ℓ_∞ setting).

In Sections III-A and III-B below, we discuss the above two challenges. In order to focus on these two challenges individually and to convey the main ideas, we first study the function class corresponding to truncated inner products without maximization over the ℓ_0 ball. More precisely, let $\mathcal{T}_{d,k} \subset \{-1,1\}^{\mathbb{R}^d}$ be the class of truncated inner product functions of the form $T_{\boldsymbol{w},k} : \boldsymbol{x} \mapsto \operatorname{sgn}(\langle \boldsymbol{w},\boldsymbol{x}\rangle_k)$, i.e. $\mathcal{T}_{d,k} := \{T_{\boldsymbol{w},k} : \boldsymbol{w} \in \mathbb{R}^d\}$. In Section III-A below, we study the growth function $\Pi_{\mathcal{T}_{d,k}}(n)$ of this function class. Then, in Section III-B, we bring the maximization over the ℓ_0 ball into our discussion and study the growth function $\Pi_{\widetilde{\mathcal{T}}_{d,k}}(n)$. Note that in fact, $\mathcal{T}_{d,k}$ is our hypothesis class, and $\widetilde{\mathcal{T}}_{d,k}$ is the composition of our hypothesis class with the maximized 0-1 loss $\widetilde{\ell}$.

A. Bounds on $\Pi_{\mathcal{T}_{d,k}}(n)$

Our main idea to bound the growth function $\Pi_{\mathcal{T}_{d,k}}(n)$ is to encode the truncated inner product in terms of a finite number of conventional inner products. Note that $\langle \boldsymbol{w},\boldsymbol{x}\rangle_k$ is the sum of $d-2k$ coordinates in $\boldsymbol{w}\odot\boldsymbol{x}$. Therefore, if we know exactly which coordinates survive after truncation, we can form the zero-one vector $\boldsymbol{\alpha}$ where α_i is one if the ith coordinate of $\boldsymbol{w}\odot\boldsymbol{x}$ survives after truncation, and is zero otherwise. Then, it is easy to see that $\langle \boldsymbol{w},\boldsymbol{x}\rangle_k = \langle \boldsymbol{w},\boldsymbol{x}\odot\boldsymbol{\alpha}\rangle$, where the right hand side is the conventional inner product (no truncation). However, the problem is that the vector $\boldsymbol{\alpha}$ is not known beforehand, and it depends on the values in $\boldsymbol{w}\odot\boldsymbol{x}$. But if we know the ordering of $\boldsymbol{w}\odot\boldsymbol{x}$, we can form the appropriate $\boldsymbol{\alpha}$ by selecting the $d-2k$ intermediate values. In order to address this, observe that the ordering of values in $\boldsymbol{w}\odot\boldsymbol{x}$ can be determined by knowing the sign of all $\binom{d}{2}$ pairwise terms of the form $w_ix_i - w_jx_j$ for $1 \leq i < j \leq d$. But this can be in fact written as $w_ix_i - w_jx_j = \langle \boldsymbol{w},\boldsymbol{x}\odot\boldsymbol{\beta}\rangle$, where $\boldsymbol{\beta} \in \mathbb{R}^d$ is the vector whose ith coordinates is $+1$, $j$the coordinate is -1, and other coordinates are zero. This discussion motivates the following lemma

Lemma 1. *Given $\boldsymbol{w},\boldsymbol{x} \in \mathbb{R}^d$, $\operatorname{sgn}(\langle \boldsymbol{w},\boldsymbol{x}\rangle_k)$ can be determined by knowing $\operatorname{sgn}(\langle \boldsymbol{w},\boldsymbol{x}\odot\boldsymbol{\alpha}_i\rangle)$ for $1 \leq i \leq \binom{d}{2k}$, and $\operatorname{sgn}(\langle \boldsymbol{w},\boldsymbol{x}\odot\boldsymbol{\beta}_j\rangle)$ for $1 \leq j \leq \binom{d}{2}$. Here, $\boldsymbol{\alpha}_i$'s are the indicators of all the $\binom{d}{2k}$ subsets of size $d-2k$, and $\boldsymbol{\beta}_j$'s are the vectors corresponding to all the $\binom{d}{2}$ pairs as in the above discussion.*

Figure 1 illustrates Lemma 1 through an example. In fact, Lemma 1 suggests that $T_{w,k}(\boldsymbol{x}) = \operatorname{sgn}(\langle \boldsymbol{w},\boldsymbol{x}\rangle_k)$ can be "coded" in terms of the signs of $\binom{d}{2k}+\binom{d}{2}$ conventional inner products. Therefore, given $\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n \in \mathbb{R}^d$, and $\boldsymbol{w} \in \mathbb{R}^d$, we can form the ± 1 matrix with size $n \times (\binom{d}{2k}+\binom{d}{2})$ whose entry is row i and column j is $\operatorname{sgn}(\langle \boldsymbol{w},\boldsymbol{x}_i\odot\boldsymbol{\alpha}_j\rangle)$ if $1 \leq j \leq \binom{d}{2k}$, and is $\operatorname{sgn}(\langle \boldsymbol{w},\boldsymbol{x}_i\odot\boldsymbol{\beta}_{j-\binom{d}{2k}}\rangle)$ if $\binom{d}{2k} < j$. This implies that the growth function $\Pi_{\mathcal{T}_{d,k}}(n)$ is bounded by the number of configurations of this matrix as $\boldsymbol{w}$ ranges in $\mathbb{R}^d$. Since all the entries in this matrix are formed by conventional inner products, classical VC dimension results yield the following.

Proposition 1. *$\Pi_{\mathcal{T}_{d,k}}(n)$ is bounded by a degree d polynomial in n, whose coefficients depend on d and k.*

B. Bounds on $\Pi_{\widetilde{\mathcal{T}}_{d,k}}(n)$

Now, we extend the ideas from Section III-A to bring the maximization over the ℓ_0 ball into play and bound the growth function of the function class $\widetilde{\mathcal{T}}_{d,k}$. Observe that given a function $\widetilde{T}_{\boldsymbol{w},k}(.) \in \widetilde{\mathcal{T}}_{d,k}$, we may write

$$\begin{aligned}\widetilde{T}_{\boldsymbol{w},k}(\boldsymbol{x},y) &= \mathbb{1}\left[\exists \boldsymbol{x}' \in \mathcal{B}_0(\boldsymbol{x},k) : y \neq \operatorname{sgn}(\langle \boldsymbol{w},\boldsymbol{x}'\rangle_k)\right] \\ &= \mathbb{1}\left[\operatorname{sgn}(\langle \boldsymbol{w},\boldsymbol{x}\rangle_k) \neq y\right] \\ &\quad \vee \mathbb{1}[\exists \boldsymbol{x}' \in \mathcal{B}_0(\boldsymbol{x},k) :\end{aligned}$$

i	$\boldsymbol{\alpha}_i$	$\boldsymbol{\alpha}_i \odot \boldsymbol{x}$	$\text{sgn}(\langle \boldsymbol{w}, \boldsymbol{x} \odot \boldsymbol{\alpha}_i \rangle)$
1	$(1,1,0,0)$	$(1,-1,0,0)$	-1
2	$(1,0,1,0)$	$(1,0,2,0)$	-1
3	$(1,0,0,1)$	$(1,0,0,-3)$	-1
4	$(0,1,1,0)$	$(0,-1,2,0)$	$+1$
5	$(0,1,0,1)$	$(0,-1,0,-3)$	$+1$
6	$(0,0,1,1)$	$(0,0,2,-3)$	-1

i	$\boldsymbol{\beta}_i$	$\boldsymbol{\beta}_i \odot \boldsymbol{x}$	$\text{sgn}(\langle \boldsymbol{w}, \boldsymbol{x} \odot \boldsymbol{\beta}_i \rangle)$	conclusion
1	$(1,-1,0,0)$	$(1,1,0,0)$	-1	$w_1x_1 < w_2x_2$
2	$(1,0,-1,0)$	$(1,0,-2,0)$	-1	$w_1x_1 < w_3x_3$
3	$(1,0,0,-1)$	$(1,0,0,3)$	-1	$w_1x_1 < w_4x_4$
4	$(0,1,-1,0)$	$(0,-1,-2,0)$	$+1$	$w_2x_2 \geq w_3x_3$
5	$(0,1,0,-1)$	$(0,-1,0,3)$	$+1$	$w_2x_2 \geq w_4x_4$
6	$(0,0,1,-1)$	$(0,0,2,3)$	$+1$	$w_3x_3 \geq w_4x_4$

Fig. 1: Illustration of Lemma 1 for $d = 4$, $k = 1$, $\boldsymbol{x} = (1, -1, 2, -3)$, and $\boldsymbol{w} = (-5, -4, -1, 1)$. From $\text{sgn}(\langle \boldsymbol{w}, \boldsymbol{x} \odot \boldsymbol{\beta}_j \rangle)$ for $1 \leq j \leq 6$ on the right, we realize that $w_1x_1 \leq w_4x_4 \leq w_3x_3 \leq w_2x_2$. This means that $\langle \boldsymbol{w}, \boldsymbol{x} \rangle_k = w_3x_3 + w_4x_4 = \langle \boldsymbol{w}, \boldsymbol{x} \odot \boldsymbol{\alpha}_6 \rangle$ whose sign can be read from the highlighted row on the left table.

$$\text{sgn}(\langle \boldsymbol{w}, \boldsymbol{x}' \rangle_k) \neq \text{sgn}(\langle \boldsymbol{w}, \boldsymbol{x} \rangle_k)], \quad (9)$$

where $\vee$ denotes the logical OR. The first term is very similar to what we discussed in Section III-A. Let us focus on the second term, which we denote by $I_1(\boldsymbol{w}, \boldsymbol{x})$. Equivalently, we may write

$$I_1(\boldsymbol{w}, \boldsymbol{x}) = \mathbb{1}\Bigg[\text{sgn}\left(\inf_{\boldsymbol{x}' \in \mathcal{B}_0(\boldsymbol{x}, k)} \langle \boldsymbol{w}, \boldsymbol{x}' \rangle_k\right) \neq \text{sgn}\left(\sup_{\boldsymbol{x}' \in \mathcal{B}_0(\boldsymbol{x}, k)} \langle \boldsymbol{w}, \boldsymbol{x}' \rangle_k\right)\Bigg],$$

where we let $\text{sgn}(\infty) := +1$ and $\text{sgn}(-\infty) := -1$. This motivates studying the maximum and minimum values of the truncated inner product over the ℓ_0 ball. It is useful to define a notation for this purpose. Given a vector $\boldsymbol{u} \in \mathbb{R}^d$, the truncated sum of $\boldsymbol{u}$ is defined as $\mathsf{TSum}_k(\boldsymbol{u}) := \sum_{i=k+1}^{d-k} u_{(i)}$. Recall that $u_{(i)}$ denotes the ith smallest value in $\boldsymbol{u}$. Observe that $\langle \boldsymbol{w}, \boldsymbol{x} \rangle_k = \mathsf{TSum}_k(\boldsymbol{w} \odot \boldsymbol{x})$. On the other hand, we have

$$\{\boldsymbol{w} \odot \boldsymbol{x}' : \boldsymbol{x}' \in \mathcal{B}_0(\boldsymbol{x}, k)\} \subset \mathcal{B}_0(\boldsymbol{w} \odot \boldsymbol{x}, k).$$

Note that if $\boldsymbol{w}$ has some zero coordinates, the inclusion is strict, since $\boldsymbol{w} \odot \boldsymbol{x}'$ is always zero in the zero coordinates of $\boldsymbol{w}$. However, if $w_i \neq 0$ for all $i \in [d]$, the two sets are in fact equal. We focus on the case where all the coordinates of $\boldsymbol{w}$ are nonzero.This means that

$$I_1(\boldsymbol{w}, \boldsymbol{x}) = \mathbb{1}\Bigg[\text{sgn}\left(\inf_{\boldsymbol{u}' \in \mathcal{B}_0(\boldsymbol{u}, k)} \mathsf{TSum}_k(\boldsymbol{u}')\right) \neq \text{sgn}\left(\sup_{\boldsymbol{u}' \in \mathcal{B}_0(\boldsymbol{u}, k)} \mathsf{TSum}_k(\boldsymbol{u}')\right)\Bigg]. \quad (10)$$

It turns out that the maximum and minimum of the truncated sum can be explicitly found, as is stated in the following Lemma 2.

Lemma 2. *For $\boldsymbol{u} \in \mathbb{R}^d$, we have*

$$\min\{\mathsf{TSum}_k(\boldsymbol{u}') : \boldsymbol{u}' \in \mathcal{B}_0(\boldsymbol{u}, k)\} = u_{(1)} + \cdots + u_{(d-2k)}$$
$$\max\{\mathsf{TSum}_k(\boldsymbol{u}') : \boldsymbol{u}' \in \mathcal{B}_0(\boldsymbol{u}, k)\} = u_{(2k+1)} + \cdots + u_{(d)}.$$

Fig. 2 illustrates the intuitive reasoning behind this lemma. Using Lemma 2 in (10), we realize that

$$I_1(\boldsymbol{w}, \boldsymbol{x}) = \mathbb{1}[\text{sgn}(u_{(1)} + \cdots + u_{(d-2k)}) \neq \text{sgn}(u_{(2k+1)} + \ldots u_{(d)})],$$

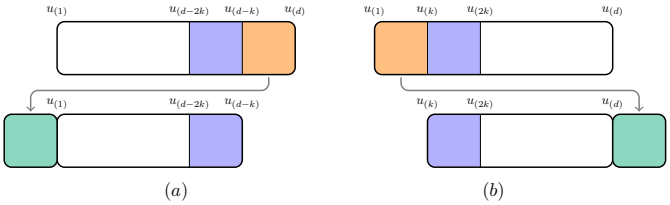

Fig. 2: (a) Sorted elements in $\boldsymbol{u}$ are illustrated on top, and $\boldsymbol{u}' \in \mathcal{B}_0(\boldsymbol{u}, k)$ on the bottom. To minimize $\mathsf{TSum}_k(\boldsymbol{u}')$, we need to make the top k elements in $\boldsymbol{u}$ (orange block) smaller than $u_{(1)}$ (green block). After truncating the green and blue blocks in $\boldsymbol{u}'$, we get $\mathsf{TSum}_k(\boldsymbol{u}') = u_{(1)} + \cdots + u_{(d-2k)}$. (b) similarly, $u_{(2k+1)} + \cdots + u_{(d)}$ is the maximum.

where $\boldsymbol{u} := \boldsymbol{w} \odot \boldsymbol{x}$. Using this in (9),

$$\widetilde{T}_{\boldsymbol{w},k}(\boldsymbol{x}, y) = \mathbb{1}\left[\text{sgn}(\mathsf{TSum}_k(\boldsymbol{u})) \neq y\right] \vee \mathbb{1}\left[\text{sgn}\left(\sum_{i=1}^{d-2k} u_{(i)}\right) \neq \text{sgn}\left(\sum_{i=2k+1}^{d} u_{(i)}\right)\right],$$

where $\boldsymbol{u} = \boldsymbol{w} \odot \boldsymbol{x}$. Note that all the three sign terms depend on the summation of some $d - 2k$ coordinates in $\boldsymbol{u} = \boldsymbol{w} \odot \boldsymbol{x}$ after sorting the elements in $\boldsymbol{u}$. Therefore, a coding technique similar to the one we used in Lemma 1 results in the following.

Proposition 2. *The growth function $\Pi_{\widetilde{\mathcal{T}}_{d,k}}(n)$ is bounded by a degree d polynomial in n, whose coefficients depend on d and k.*

Using the bound of Proposition 2 in (8), we get an upper bound of order $\sqrt{\log n / n}$ for $R_n(\widetilde{\mathcal{T}}_{d,k})$, which yields the bound of our Theorem 1.

IV. Conclusion

In this paper, we proved a distribution-independent generalization bound for the binary classification setting with ℓ_0–bounded adversarial perturbation. We saw that deriving such generalization bound is challenging, in particular due to (i) the nonlinearity of the truncated inner product, and (ii) non-smoothness and non-convexity of the ℓ_0 ball. We tackled these challenges by introducing a novel technique which enables us to bound the growth function of our hypothesis class.

REFERENCES

[1] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Šrndić, P. Laskov, G. Giacinto, and F. Roli, "Evasion attacks against machine learning at test time," in *Joint European conference on machine learning and knowledge discovery in databases*. Springer, 2013, pp. 387–402.

[2] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *International Conference on Learning Representations, 2014, Banff, AB, Canada, April 14-16*, 2014. [Online]. Available: http://arxiv.org/abs/1312.6199

[3] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[4] N. Carlini and D. A. Wagner, "Towards evaluating the robustness of neural networks," in *2017 IEEE Symposium on Security and Privacy, San Jose, CA, USA, May 22-26,*, 2017, pp. 39–57.

[5] A. Athalye, N. Carlini, and D. A. Wagner, "Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples," in *Proceedings of the 35th International Conference on Machine Learning, ICML, Stockholm, Sweden, July 10-15*, 2018, pp. 274–283. [Online]. Available: http://proceedings.mlr.press/v80/athalye18a.html

[6] R. Bhattacharjee and K. Chaudhuri, "Consistent non-parametric methods for adaptive robustness," *arXiv preprint arXiv:2102.09086*, 2021.

[7] R. Bhattacharjee, S. Jha, and K. Chaudhuri, "Sample complexity of adversarially robust linear classification on separated data," *arXiv preprint arXiv:2012.10794*, 2020.

[8] E. Wong and Z. Kolter, "Provable defenses against adversarial examples via the convex outer adversarial polytope," in *International Conference on Machine Learning*. PMLR, 2018, pp. 5286–5295.

[9] A. Raghunathan, J. Steinhardt, and P. Liang, "Certified defenses against adversarial examples," *arXiv preprint arXiv:1801.09344*, 2018.

[10] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. [Online]. Available: https://openreview.net/forum?id=rJzIBfZAb

[11] A. Shamir, I. Safran, E. Ronen, and O. Dunkelman, "A simple explanation for the existence of adversarial examples with small hamming distance," *arXiv preprint arXiv:1901.10861*, 2019.

[12] D. Jin, Z. Jin, J. T. Zhou, and P. Szolovits, "Is bert really robust? natural language attack on text classification and entailment," *arXiv preprint arXiv:1907.11932*, vol. 2, 2019.

[13] K. Grosse, N. Papernot, P. Manoharan, M. Backes, and P. McDaniel, "Adversarial perturbations against deep neural networks for malware classification," *arXiv preprint arXiv:1606.04435*, 2016.

[14] J. Li, F. Schmidt, and Z. Kolter, "Adversarial camera stickers: A physical camera-based attack on deep learning systems," in *International Conference on Machine Learning*. PMLR, 2019, pp. 3896–3904.

[15] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *2016 IEEE European symposium on security and privacy (EuroS&P)*. IEEE, 2016, pp. 372–387.

[16] A. Modas, S.-M. Moosavi-Dezfooli, and P. Frossard, "Sparsefool: a few pixels make a big difference," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9087–9096.

[17] L. Schott, J. Rauber, M. Bethge, and W. Brendel, "Towards the first adversarially robust neural network model on mnist," *arXiv preprint arXiv:1805.09190*, 2018.

[18] F. Croce, M. Andriushchenko, N. D. Singh, N. Flammarion, and M. Hein, "Sparse-rs: a versatile framework for query-efficient sparse black-box adversarial attacks," *arXiv preprint arXiv:2006.12834*, 2020.

[19] A. Levine and S. Feizi, "Robustness certificates for sparse adversarial attacks by randomized ablation." in *AAAI*, 2020, pp. 4585–4593.

[20] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," in *2016 IEEE symposium on security and privacy (SP)*. IEEE, 2016, pp. 582–597.

[21] P. Delgosha, H. Hassani, and R. Pedarsani, "Robust classification under ℓ_0 attack for the gaussian mixture model," *arXiv preprint arXiv:2104.02189*, 2021.

[22] M. Beliaev, P. Delgosha, H. Hassani, and R. Pedarsani, "Efficient and robust classification for sparse attacks," *arXiv preprint arXiv:2201.09369*, 2022.

[23] L. Schmidt, S. Santurkar, D. Tsipras, K. Talwar, and A. Madry, "Adversarially robust generalization requires more data," *Advances in neural information processing systems*, vol. 31, 2018.

[24] D. Yin, R. Kannan, and P. Bartlett, "Rademacher complexity for adversarially robust generalization," in *International conference on machine learning*. PMLR, 2019, pp. 7085–7094.

[25] O. Montasser, S. Hanneke, and N. Srebro, "Vc classes are adversarially robustly learnable, but only improperly," in *Conference on Learning Theory*. PMLR, 2019, pp. 2512–2530.

[26] A. Raghunathan, S. M. Xie, F. Yang, J. C. Duchi, and P. Liang, "Adversarial training can hurt generalization," 2019.

[27] I. Attias, A. Kontorovich, and Y. Mansour, "Improved generalization bounds for robust learning," in *Algorithmic Learning Theory*. PMLR, 2019, pp. 162–183.

[28] U. Feige, Y. Mansour, and R. Schapire, "Learning and inference in the presence of corrupted inputs," in *Conference on Learning Theory*. PMLR, 2015, pp. 637–657.

[29] P. Awasthi, N. Frank, and M. Mohri, "On the rademacher complexity of linear hypothesis sets," *arXiv preprint arXiv:2007.11045*, 2020.

[30] J. Khim and P.-L. Loh, "Adversarial risk bounds via function transformation," *arXiv preprint arXiv:1810.09519*, 2018.

[31] A. Najafi, S.-i. Maeda, M. Koyama, and T. Miyato, "Robustness to adversarial perturbations in learning from incomplete data," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[32] Y. Xing, Q. Song, and G. Cheng, "On the generalization properties of adversarial training," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2021, pp. 505–513.

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