Robust Online Learning

Sajad Ashkezari SAJAD.ASHKEZARI@UWATERLOO.CA
University of Waterloo

Editors: Matus Telgarsky and Jonathan Ullman

Abstract

We study the problem of learning robust classifiers where the classifier will receive a perturbed input. Unlike robust PAC learning studied in prior work, here the clean data and its label are also adversarially chosen. We formulate this setting as an online learning problem and consider both the realizable and agnostic learnability of hypothesis classes. We define a new dimension of classes and show it controls the mistake bounds in the realizable setting and the regret bounds in the agnostic setting. In contrast to the dimension that characterizes learnability in the PAC setting, our dimension is rather simple and resembles the Littlestone dimension. We generalize our dimension to multiclass hypothesis classes and prove similar results in the realizable case. Finally, we study the case where the learner does not know the set of allowed perturbations for each point and only has some prior on them.

Keywords: Robustness, Online learning, Littlestone dimension

1. Introduction

In this paper we study the online learning of robust predictors, i.e., predictors whose prediction remains correct even if an adversary perturbs their input. It has been shown that even predictors that have high accuracy on clean data can drastically fail on slightly perturbed inputs, which could be indistinguishable from the clean input for humans (Goodfellow et al., 2014). Learning robust neural networks has been widely studied and is an ongoing research direction (Chakraborty et al., 2021).

Our goal is to formulate and study this problem through the theoretic online learning framework (Littlestone, 1988). Informally, we consider the following interactive learning game between an adversary and a learner. At each round the adversary reveals a perturbed input to the learner who will then predict a label on this input. The adversary then reveals the clean input and its true label. The goal of the learner is to minimize the number of mistakes it makes. We formally define this problem in Definition 1. We will then define a notion of robust online learnability of a class and answer the following questions:

- What is the optimal number of mistakes achievable in the robust online learning problem?
- Is there a combinatorial measure of complexity of a class that controls this optimal value?

Related work

Robust learnability has been extensively studied in the learning theory literature. However, most prior work focus on robust PAC learnability where unlike our setup, the clean data comes from a distribution and is then manipulated. It has been shown that VC classes are robustly PAC learnable, but finite VC is not a necessary condition (Montasser et al., 2019). It was later shown that the learnability is characterized by a new dimension that depends on the *global one-inclusion graph* of

a hypothesis class (Montasser et al., 2022). Other works have also studied robust PAC learnability when the perturbation set that the adversary is allowed to perturb a point into is unknown to the learner or belongs to a class of such sets (Montasser et al., 2021; Lechner et al., 2023). Adversarially robust PAC regression has been studied by Attias and Hanneke (2023).

The online learning framework was introduced in the seminal work of Littlestone Littlestone (1988), which was later extended to agnostic online learning as well Ben-David et al. (2009). This framework has also been studied for multiclass hypothesis classes Daniely et al. (2015); Hanneke et al. (2023). To the best of our knowledge, this is the first work to study robust online learning in this framework.

Contributions

We make the following contributions:

- We formulate the robust online learning problem through a well-known theoretical framework.
- We define a new dimension of a binary class $\mathcal{H}$, $\mathrm{L}_{\mathcal{U}}(\mathcal{H})$, and show that it characterizes robust online learnability. We show the optimal mistake bound in the realizable case is exactly equal to this dimension. In the agnostic case, we show the optimal expected regret up to logarithmic factors is $\tilde{\mathcal{O}}(\sqrt{\mathrm{L}_{\mathcal{U}}(\mathcal{H})T})$.
- We extend our dimension for multiclass hypothesis classes and show similar results for realizable robust online learning.
- We also study the setting where the learning algorithm does not exactly know the perturbations that the adversary is allowed to make, but has prior knowledge that it belongs to a finite family $\mathcal{G}$ of perturbation functions. We prove upper bounds for this setting that depend logarithmically on the cardinality of $\mathcal{G}$.

2. Setup and Problem Formulation

We use $\mathcal{X}$ and $\mathcal{Y}$ to denote our instance space and label space, respectively. Here, we focus on binary label space, i.e., $\mathcal{Y} = \{0, 1\}$. A hypothesis is a mapping $h : \mathcal{X} \to \mathcal{Y}$ from instance space to label space. A hypothesis class, $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$, is a set of such mappings. A learning algorithm $\mathcal{A} : (\mathcal{X} \times \mathcal{Y})^* \to \mathcal{Y}^{\mathcal{X}}$ gets a finite number of instance-label pairs and outputs a hypothesis. We let $\mathcal{U} : \mathcal{X} \to 2^{\mathcal{X}}$ be any function that maps instances to the set of allowed perturbations. That is, for each $x \in \mathcal{X}$, the adversary can perturb it only to $z \in \mathcal{U}(x)$. Unless stated otherwise, we assume $\mathcal{U}$ is fixed and known to the learner (we study uncertain perturbation sets in Section 6). The adversarial loss of a hypothesis h on (x, y) is defined as follows:

$$l_{\mathcal{U}}(h, (x, y)) = \sup_{z \in \mathcal{U}(x)} \mathbf{1}[h(z) \neq y], \tag{1}$$

where $\mathbf{1}[A] = 1$ if A is true, and $\mathbf{1}[A] = 0$ otherwise.
We are now ready to define our learning problem.

Definition 1 (Robust Online Learning) *Robust online learning is an iterative game between an adversary and a learner such that:*

At each round $t = 1, 2, \cdots$*:*

(a) The adversary selects Z_t *and reveals it to the learner.*

(b) The learner predicts $\hat{Y}_t \in \mathcal{Y}$*.*

(c) The adversary selects $X_t \in \mathcal{X}$ *with* $Z_t \in \mathcal{U}(X_t)$ *and* $Y_t \in \mathcal{Y}$ *and reveals* (X_t, Y_t) *to the learner.*

(d) The learner incurs a loss of $\mathbf{1}[\hat{Y}_t \neq Y_t]$*.*

Remark 2 *In Definition 1, we assume the adversary knows how the learner will predict on each point and the goal of the adversary is to maximize the number of mistakes. So we can equivalently reformulate the problem such that at each round the learner first picks a hypothesis* $h_t : \mathcal{X} \to \mathcal{Y}$ *and then the adversary picks and reveals* $(\mathcal{X}_t, \mathcal{Y}_t)$*. Finally, the learner incurs a loss of* $l_{\mathcal{U}}(h, (X_t, Y_t))$*.*

Consider a sequence $X_1, Y_1, \cdots, X_T, Y_T$ for $T \leq \infty$. For any finite $t \leq T$, define $S_{<t} = X_1, Y_1, \cdots, X_{t-1}, Y_{t-1}$. We respectively define the mistake bound and the regret of a learner $\mathcal{A}$ with respect to $\mathcal{H}$ on a sequence S as follows:

$$\mathrm{M}(\mathcal{A}, S) = \sum_{t=1}^{T} \mathbf{1}[\mathcal{A}(S_{<t})(Z_t) \neq Y_t] \tag{2}$$

$$\mathrm{R}_{\mathcal{H}}(\mathcal{A}, S) = \sum_{t=1}^{T} \mathbf{1}[\mathcal{A}(S_{<t})(Z_t) \neq Y_t] - \inf_{h \in \mathcal{H}} \sum_{t=1}^{T} \mathbf{1}[h(Z_t) \neq Y_t] \tag{3}$$

Similar to the classic online learning, we can define robust online learning with respect to a hypothesis class. We say a sequence $X_1, Y_1, \cdots, X_T, Y_T$ is $\mathcal{U}$-robust realizable with respect to $\mathcal{H}$ if $\inf_{h \in \mathcal{H}} \sum_{t=1}^{T} l_{\mathcal{U}}(h, (X_t, Y_t)) = 0$. Let $\mathrm{RE}(\mathcal{H})$ denote the set of all realizable sequences w.r.t. $\mathcal{H}$.

Definition 3 (Realizable Robust Online Learnability) *We say that* $\mathcal{H}$ *is realizable robust online learnable with optimal mistake bound* $\mathbf{M}^* < \infty$*, if the following holds:*

$$\inf_{\mathcal{A}} \sup_{S \in \mathrm{RE}(\mathcal{H})} \mathrm{M}(\mathcal{A}, S) = \mathbf{M}^* \tag{4}$$

We can also define a learnability task without the realizability assumption, which is known as agnostic learning. However, instead of bounding the number of mistakes, we are interested in bounds on the regret. Furthermore, in this setup, we assume the number of rounds is some finite T known to the learner in advance.

Definition 4 (Agnostic Robust Online Learnability) *We say that* $\mathcal{H}$ *is agnostic robust online learnable with optimal regret* $\mathbf{R}^*_T < \infty$ *for any* $T < \infty$ *if the following holds:*

$$\inf_{\mathcal{A}} \sup_{S : |S| = T} \mathrm{R}_{\mathcal{H}}(\mathcal{A}, S) = \mathbf{R}^*_T \tag{5}$$

3. Optimal mistake bounds for realizable robust online learning

In this section we derive optimal mistake bounds for realizable robust online learning of a class $\mathcal{H}$ with respect to a combinatorial parameter of it. To do this, we first introduce an easier version of the robust online learning problem, which makes the definition of our dimension more intuitive.

3.1. Orientation Game

For any given Z_t, there could be many candidates $x \in \mathcal{X}$ such that $Z_t \in \mathcal{U}(x)$. So the decision of the learner will potentially need to depend on all these points and how the class behaves on them. What if we are given only two candidates X_t^0 and X_t^1 such that $Z_t \in X_t^0 \cap X_t^1$ and only need to make a decision between these two? We will show that unlike our original problem, finding a learner for this problem can be simpler. We first formalize this problem.

Definition 5 (Orientation Game) *Orientation game is an iterative game between an adversary and a learner such that:*

At each round $t = 1, 2, \cdots$:

- *(a) The adversary selects $X_t^0, X_t^1 \in \mathcal{X}$ with $\mathcal{U}(X_t^0) \cap \mathcal{U}(X_t^1) \neq \emptyset$ and reveals them to the learner.*
- *(b) The learner predicts $\hat{Y}_t \in \{0, 1\}$.*
- *(c) The adversary selects $Y_t \in \mathcal{Y}$ and reveals $(X_t^{Y_t}, Y_t)$ to the learner.*
- *(d) The learner incurs a loss of $\mathbf{1}[\hat{Y}_t \neq Y_t]$.*

We say that a sequence $S = (X_1^0, X_1^1), Y_1, (X_2^0, X_2^1), Y_2, \cdots, (X_T^0, X_T^1), Y_T$ is realizable by $\mathcal{H}$ if the sequence $X_1^{Y_1}, Y_1, \cdots, X_T^{Y_T}, Y_T$ is realizable by $\mathcal{H}$. We also define $\mathcal{X}_{\mathcal{U}}^2 := \{(x_1, x_2) \in \mathcal{X}^2 : \mathcal{U}(x_1) \cap \mathcal{U}(x_2) \neq \emptyset\}$. With these definitions, we are now ready to define our dimension.

Definition 6 ($\mathcal{U}$-adversarial Littlestone tree) *A $\mathcal{U}$-adversarial Littlestone tree is a full binary tree of depth $d \in \mathbb{N}$ whose internal nodes are labeled by $\mathcal{X}_{\mathcal{U}}^2$ and the two outgoing edges from each node are labeled by 0 and 1.*

In other words, the tree can be represented as the following collection:

$$\{(x_{\mathbf{u}}^0, x_{\mathbf{u}}^1) \in \mathcal{X}_{\mathcal{U}}^2 : \mathbf{u} \in \{0,1\}^k, 0 \leq k < d\}$$

We say that a tree is shattered by $\mathcal{H}$ if each path emanating from the root is realizable by $\mathcal{H}$. That is, for each $\mathbf{u} = (u_1, \cdots, u_d) \in \{0,1\}^d$ and for each $0 \leq k < d$, there exists $h \in \mathcal{H}$ such that for all $0 \leq i \leq k$, $h(z) = u_{i+1}$ for all $z \in \mathcal{U}(x_{\mathbf{u}_{\leq i}}^{u_{i+1}})$, i.e., $l_{\mathcal{U}}(h, (x_{\mathbf{u}_{\leq i}}^{u_{i+1}}, u_{i+1})) = 0$, where $\mathbf{u}_{\leq i} = (u_1, \cdots, u_i)$. Figure 1 illustrates an example of such a tree of depth 2.

Definition 7 ($\mathcal{U}$-adversarial Littlestone Dimension) *The $\mathcal{U}$-adversarial Littlestone dimension of a class $\mathcal{H}$ is the maximum depth of a tree that it shatters. We denote this dimension by $\mathrm{L}_{\mathcal{U}}(\mathcal{H})$. Furthermore, we say $\mathrm{L}_{\mathcal{U}}(\mathcal{H}) = \infty$ if $\mathcal{H}$ shatters trees of arbitrary large depth.*

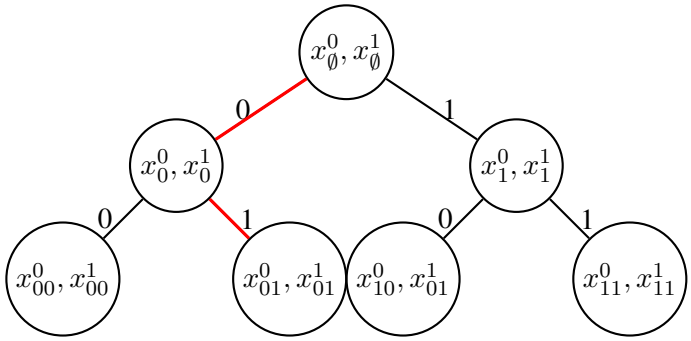

Figure 1: A $\mathcal{U}-$adversarial tree of depth 2. For each $k \leq 2$ and $\mathbf{u} \in \{0,1\}^k$ we have $\mathcal{U}(x^0_{\mathbf{u}}) \cap \mathcal{U}(x^1_{\mathbf{u}}) \neq \emptyset$. The tree is shattered $\mathcal{H}$ if each of its root-to-leaf paths are realizable by $\mathcal{H}$. For example, there must exist $h_{01} \in \mathcal{H}$ such that $h_{01}(z) = 0$ for all $z \in \mathcal{U}(x^0_\emptyset)$ and $h_{01}(z) = 1$ for all $z \in \mathcal{U}(x^1_0)$.

Algorithm 1 Standard Optimal Algorithm for Orientation Game ($\mathrm{SOA}_{\mathrm{OG}}$)

Input: Hypothesis class $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$, perturbation set $\mathcal{U}$
Initialize $\mathcal{V}_0 \leftarrow \mathcal{H}$
for $t \geq 1$ **do**
  Receive (X^0_t, X^1_t)
  for $y \in \mathcal{Y}$ **do**
    Define $\mathcal{V}^y_t = \{h \in \mathcal{V}_{t-1} : \forall z \in \mathcal{U}(X^y_t), h(z) = y\}$
  end
  Predict $\hat{Y}_t$ for some$= \hat{Y}_t \in \arg\max_{y \in \mathcal{Y}} \mathrm{L}_{\mathcal{U}}(\mathcal{V}^y_t)$
  Receive Y_t
  Let $\mathcal{V}_t = \mathcal{V}^{Y_t}_t$
end

Remark 8 *A quick sanity check would be to see if our definition coincides with the classic Littlestone dimension when the adversary is not allowed to perturb the instances, i.e., $\mathcal{U}(x) = \{x\}$ for all $x \in \mathcal{X}$. In this case, $\mathcal{X}^2_{\mathcal{U}} = \{(x,x) : x \in \mathcal{X}\}$ as $\mathcal{U}(x_1) \cap \mathcal{U}(x_2) = \emptyset$ for all $x_1 \neq x_2$. Thus, we can simply denote each node by an instance of $\mathcal{X}$ and the definition becomes the same as the classic definition.*

Similar to the realizable robust online learning, we can also define realizable version of the orientation game and the optimal mistake. We denote the optimal mistake bound of this problem with $\mathbf{M}^*_{OG}$. Then we get the following results on the optimal mistake.

Theorem 9 *A hypothesis class $\mathcal{H}$ with finite $\mathrm{L}_{\mathcal{U}}(\mathcal{H})$ is realizable learnable in the orientation game with optimal mistake bound $\mathbf{M}^*_{OG} = \mathrm{L}_{\mathcal{U}}(\mathcal{H})$. In particular, the algorithm $\mathrm{SOA}_{\mathrm{OG}}$ achieves the optimal mistake bound by ensuring that each mistake reduces the $L_{\mathcal{U}}$ dimension of the version space by at least one.*

Proof *We first show that $\mathbf{M}^*_{OG} \geq L := \mathrm{L}_{\mathcal{U}}(\mathcal{H})$. Fix any learning algorithm. The adversary plays according to a shattered tree of depth L, $\{(x^0_{\mathbf{u}}, x^1_{\mathbf{u}}) \in \mathcal{X}^2_{\mathcal{U}} : \mathbf{u} \in \{0,1\}^k, 0 \leq k < L\}$. It starts*

by picking the root of the tree, $(x^0_\emptyset, x^1_\emptyset)$, and for any prediction $\hat{y}$ of the algorithm, it chooses it's label to be $1-\hat{y}$ and then continue the process for the child in the tree connect to the current root and continuing the same process. By the definition of shattering, the adversary can continue this process for at least L steps while maintaining realizability. Since we can force L mistake on any learner, $\mathbf{M}^*_{OG} \geq L$.

We now show $\mathrm{SOA}_{\mathrm{OG}}$ makes at most L mistakes on any realizable sequence, which implies $\mathbf{M}^*_{OG} \leq L$, and thus, $\mathbf{M}^*_{OG} = L$. Consider any round t at which the algorithm makes a mistake. We claim $\mathrm{L}_{\mathcal{U}}(\mathcal{V}_{t+1}) < \mathrm{L}_{\mathcal{U}}(\mathcal{V}_t) =: L_t$. Assume otherwise. Then it must be that $\mathrm{L}_{\mathcal{U}}(\mathcal{V}_{t+1}) = \mathrm{L}_{\mathcal{U}}(\mathcal{V}_t)$ as $\mathcal{V}_{t+1} \subseteq \mathcal{V}_t$ and thus $\mathrm{L}_{\mathcal{U}}(\mathcal{V}_{t+1}) \leq \mathrm{L}_{\mathcal{U}}(\mathcal{V}_t)$. By definition of $\hat{Y}_t$, $\mathrm{L}_{\mathcal{U}}(\mathcal{V}_t) \geq \mathrm{L}_{\mathcal{U}}(\mathcal{V}_t^{\hat{Y}_t}) \geq \mathrm{L}_{\mathcal{U}}(\mathcal{V}_t^{Y_t}) = \mathrm{L}_{\mathcal{U}}(\mathcal{V}_{t+1}) = \mathrm{L}_{\mathcal{U}}(\mathcal{V}_t)$. Let T_0 and T_1 be trees of depth L_t shattered by $\mathcal{V}^0_t$ and $\mathcal{V}^1_t$, respectively. Create a new tree whose root is (X^0_t, X^1_t) and whose left and right subtrees are T_0 and T_1, respectively. They by definition of $\mathcal{V}^y_t$, this tree is shattered by $\mathcal{V}_t$, and thus $\mathrm{L}_{\mathcal{U}}(\mathcal{V}_t) \geq L_t + 1$ which is a contradiction and thus the claim holds. Thus, each time the algorithm makes a mistake, the dimension of $\mathcal{V}_t$ decreases by at least 1, since the dimension is nonnegative, the algorithm makes at most L mistakes. ■

3.2. From orientations to learners

Here, we show how we can convert a learner in the orientation game to a learner for the robust online learning problem. Suppose the learner receives an input z and wants to predict which x with $z \in \mathcal{U}(x)$ the adversary will choose. For a candidate x^y representing label y, the learner checks the decision of the orientation between this candidate and each candidate representing the opposite label. If the orientation is always towards x^y, then the learner predicts y. The learner does this process for all y and all x^y. If the learner makes a mistake on X_t, Y_t at some round t, then it must be that the orientation between X_t and some candidate from the other label was wrong. Thus, we can upper bound the number of mistakes made in the online learning problem by the number of mistakes made in the orientation game, which we can upper bound by the results in the previous section. We present our learner in Algorithm 2 where we define $\mathcal{V}^{\mathcal{U}}_{x,y} := \{h :\in \mathcal{V} : \forall z \in \mathcal{U}(x), h(z) = y\}$ be the set of hypotheses in $\mathcal{V}$ consistent on (x, y). We note that this idea has also been used by Montasser et al. (2022) for converting orientations of their global one-inclusion graph to PAC learners.

We are now ready to state our main result in this section.

Theorem 10 *For a hypothesis class $\mathcal{H}$ with $\mathrm{L}_{\mathcal{U}}(\mathcal{H}) = L < \infty$, the optimal mistake bound in the realizable robust online learning satisfies $\mathbf{M}^* = L$.*

Proof *We prove the theorem first by showing $\mathbf{M}^* \geq L$ and then $\mathbf{M}^* \leq L$.*

To prove the lower bound consider the following strategy by the adversary. Pick a tree of depth L that is shattered by the class. Start from the root $(X^0_\emptyset, X^1_\emptyset)$ and pick any $Z_1 \in X^0_\emptyset \cap X^1_\emptyset$. For any prediction $\hat{Y}_1$ of the learner, pick $Y_1 = 1 - \hat{Y}_1$ and $X_1 = X^{Y_t}_\emptyset$. Go down along the edge labeled by Y_t and continue the same process for the child there. By definition of the tree and $\mathrm{L}_{\mathcal{U}}$, the sequence picked by the adversary is realizable. Thus, we the adversary can force at least L mistakes. Since the learner was arbitrary, $\mathbf{M}^* \geq L$.

We now proceed to prove the upper bound. We follow the prediction strategy outlined in Algorithm 2. Consider a round t where the learner makes a mistake and consider $\xi_\tau = ((x_0, x_1), Y_t)$ as defined in the algorithm. Then by definition of f, $\eta(\xi_1, \cdots, \xi_{\tau-1}, (x_0, x_1)) = \hat{Y}_t \neq Y_t$. This means the orientation learner makes a mistake on all its inputs in a game defined by the sequence

Algorithm 2 Robust Online Learning Strategy $\hat{Y}_t$

Input: Hypothesis class $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$, perturbation set $\mathcal{U}$, orientation learner $\eta : (\mathcal{X}_{\mathcal{U}}^2 \times \mathcal{Y})^* \times \mathcal{X}^2 \to \{0,1\}$, test point z

Initialize $\mathcal{V} \leftarrow \mathcal{H}$, $f(.,.) \leftarrow \eta((.,.))$, $\tau \leftarrow 1$

for $t \geq 1$ **do**
- Receive Z_t
- **for** $y \in \mathcal{Y}$ **do**
  - Define $P_y := \{x \in \mathcal{X} : Z_t \in \mathcal{U}(x) \wedge \mathcal{V}_{x,y}^{\mathcal{U}} \neq \emptyset\}$
- **end**
- **if** $\exists y \in \mathcal{Y}, \exists x_y \in P_y, \forall x_{1-y} \in P_{1-y} : f(x_0, x_1) = y$ **then**
  - $\hat{Y}_t \leftarrow y$
- **end**
- **else**
  - $\hat{Y}_t \leftarrow 1$
- **end**
- Receive (X_t, Y_t)
- **if** $\hat{Y}_t \neq Y_t$ **then**
  - Let $X_\tau^{Y_t} \leftarrow X_t$
  - Let $X_\tau^{1-Y_t}$ be an arbitrary $x \in P_{1-Y_t}$ such that $f(X_\tau^0, X_\tau^1) = 1 - Y_t$
  - Let $\xi_\tau \leftarrow ((X_\tau^0, X_\tau^1), Y_t)$
  - Update $f(.,.) \leftarrow \eta(\xi_1, \ldots, \xi_\tau, (.,.))$
  - Update $\tau \leftarrow \tau + 1$
- **end**
- $\mathcal{V} \leftarrow \mathcal{V}_{X_t, Y_t}^{\mathcal{U}}$

end

$\xi_1, \cdots, \xi_\tau$. *Thus, the number of mistakes that the learner makes is at most the number of mistakes that the orientation learner makes. However, by Theorem 9, we know there is a orientation learner, i.e.,* $\mathrm{SOA}_{\mathrm{OG}}$*, that makes at most* L *mistakes. Thus,* $\mathbf{M}^* \leq L$*. The only thing we need to address is that* x_{1-Y_t} *exists. We know* $x_{Y_t} = X_t \in P_{Y_t}$*, thus it must be the case* $\exists x_{1-Y_t}$ *such that* $f(x_0, x_1) = 1 - Y_t$ *as otherwise the algorithm would have predicted* Y_t *by the definition of* $\hat{Y}_t$*.* ■

4. Multiclass robust online learning

In the previous section, we only studied binary classes. In this section, we state similar results for a general label space $\mathcal{Y}$ potentially with an infinite size (e.g., $\mathcal{Y} = \mathbb{N}$). We use similar techniques to show our results. To do so, we define a multiclass orientation game where at each step t, in addition to presenting (x_t^0, x_t^1), the adversary also presents two labels $y_t^0 \neq y_t^1$.

Definition 11 (Multiclass Orientation Game) *Orientation game is an iterative game between an adversary and a learner such that:*

At each round $t = 1, 2, \cdots$*:*

(a) The adversary selects $(X_t^0, X_t^1) \in \mathcal{X}_{\mathcal{U}}^2$ *and* $Y_t^0, Y_t^1 \in \mathcal{Y}$ *s.t.* $Y_t^0 \neq Y_t^1$ *and reveals them to the learner.*

(b) The learner predicts $\hat{Y}_t \in \mathcal{Y}$.

(c) The adversary selects $i_t \in \{0,1\}$ *and reveals* $(X_t^{i_t}, Y_t^{i_t})$ *to the learner.*

(d) The learner incurs a loss of $\mathbf{1}[\hat{Y}_t \neq Y_t^{i_t}]$.

We say the problem is realizable if the sequence $X_1^{i_1}, Y_1^{i_1}, \cdots, X_t^{i_t}, Y_t^{i_t}$ is realizable for each finite t. We now define a multiclass version of our dimension and again show that it characterizes learnability.

A multiclass $\mathcal{U}$-adversarial Littlestone tree is a collection of nodes as follows:

$$\{(x_{\mathbf{u}}^0, x_{\mathbf{u}}^1, y_{\mathbf{u}}^0, y_{\mathbf{u}}^1) \in \mathcal{X}_{\mathcal{U}}^2 \times \mathcal{Y}_{\neq}^2 : \mathbf{u} \in \{0,1\}^k, 0 \leq k < d\},$$

where $\mathcal{Y}_{\neq}^2 := \{(y_1, y_2) \in \mathcal{Y}^2 : y_1 \neq y_2\}$. We say such a tree is shattered by $\mathcal{H}$ if for each $\mathbf{u} \in \{0,1\}^d$ and for each $0 \leq k < d$, there exists $h \in \mathcal{H}$ such that for all $0 \leq i \leq k$, $h(z) = y_{\mathbf{u}_{\leq i}}^{u_{i+1}}$ for all $z \in \mathcal{U}(x_{\mathbf{u}_{\leq i}}^{u_{i+1}})$, i.e., $l_{\mathcal{U}}(h, (x_{\mathbf{u}_{\leq i}}^{u_{i+1}}, y_{\mathbf{u}_{\leq i}}^{u_{i+1}})) = 0$. The multiclass $\mathcal{U}$-adversarial Littlestone dimension of $\mathcal{H}$ is similarly defined as the maximum depth of a tree that is shattered by $\mathcal{H}$. Here we abuse notation and also denote this dimension by $\mathrm{L}_{\mathcal{U}}(\mathcal{H})$.

We now present our results for the multiclass learning problems. The proofs for these results are similar to those in section 3. Thus, we only give sketch of the proof and outline where they would differ from the binary case.

Theorem 12 *For any* $\mathcal{H}$ *with finite* $L = \mathrm{L}_{\mathcal{U}}(\mathcal{H})$ *the optimal mistake bound achievable in the realizable multiclass orientation game equals* L.

Proof *[proof sketch] The proof for lower bound is again by an adversary that plays according a shattered tree and for each prediction* $\hat{Y}_t = Y_t^y$ *of the learner set the true label to* Y_t^{1-y} *and follow the respective edge. The definition of the dimension then would ensure the adversary can continue for* $\mathrm{L}_{\mathcal{U}}(\mathcal{H})$ *rounds.*

The upper bound is achieved by an adapted version of $\mathrm{SOA}_{\mathrm{OG}}$ *that predicts* $\hat{Y}_t = Y_t^y$ *for* $y \in \arg\max_{y \in \{0,1\}} \mathrm{L}_{\mathcal{U}}(\mathcal{V}_t^{X_t^y, Y_t^y})$ *where* $\mathcal{V}_t^{X_t^y, Y_t^y}$ *is the set of all hypotheses in the current version space that robustly label* X_t^y *with* Y_t^y*. Again, the idea is to ensure every time the learner makes a mistake, the dimension of the version space reduces and since it will remain nonnegative, the number of times it decreases and thus the number of mistakes is bounded by* $\mathrm{L}_{\mathcal{U}}(\mathcal{H})$. ■

Theorem 13 *Any* $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ *with* $\mathrm{L}_{\mathcal{U}}(\mathcal{H}) = L < \infty$ *is realizable robust online learnable with optimal mistake bound* L.

Proof *The lower bound again simply follows from the definition of the dimension. The upper bound can be achieved by a modified version of the learner for the binary case. Here the learner predicts* $\hat{Y}_t = y$ *if*

$$\exists y \in \mathcal{Y}, \exists x_y \in P_y, \forall y' \in \mathcal{Y} \backslash \{y\} \forall x_{y'} \in P_{y'} : f(x_y, x_{y'}, y, y') = y.$$

That is, the learner compares each candidate label against all other labels in the label space. Then by the same logic as the binary case, if the algorithm makes a mistake, then it must be that there is a $\xi = (x, x', y, y')$ *that the orientation learner has made a mistake on. Thus, the number of mistakes is bounded by the number of mistakes an orientation learner makes which we can upper bound by* $\mathrm{L}_{\mathcal{U}}(\mathcal{H})$ *by Theorem 12.* ■

5. Regret bounds for agnostic robust online learning

So far, the adversary was restricted to choose realizable sequences. Here we consider a problem where the adversary does not have any restriction on its sequence. In this case, we are interested in finding an algorithm whose regret w.r.t. the best hypothesis in the class is sublinear. For this problem, we must allow the learner to have some private randomization by the same arguments made for classic online learning (Shalev-Shwartz and Ben-David, 2014). Then we have the following result on the achievable expected regret (Theorem 15). The proof of this result is by the general technique of agnostic to realizable reduction (Ben-David et al., 2009). The idea of this technique is to create a set of "experts" such that for each input sequence and for each hypothesis in the class, there is an expert that predicts exactly the same as the hypothesis. Thus, it is enough to achieve small regret w.r.t. the best expert, which can be done using the algorithm of "prediction with expert advice" (Cesa-Bianchi and Lugosi, 2006, Chapter 2). Here we only focus on binary classes, however, we believe similar results can be proven by adapting the methods used for the non-robust online learning for multiclass classes (Daniely et al., 2015; Hanneke et al., 2023).

Lemma 14 *Consider any hypothesis class $\mathcal{H}$ with $\mathrm{L}_{\mathcal{U}}(\mathcal{H}) < \infty$. For any $h \in \mathcal{H}$, $\tau \leq T$, $X_1, \cdots, X_\tau \in \mathcal{X}$, and for any $Z_1, \cdots, Z_\tau$, there exists L and $i_1, \cdots, i_L \in \mathbb{N}$, such that the expert $Y_t(i_1, \cdots, i_L)$ defined in Algorithm 3 predicts the same as h on Z_t for all $t \leq \tau$.*

Proof Assume we run the realizable online learner $\hat{Y}_t$ on the sequence $Z_1, \cdots Z_\tau$ with the feedback $X_1, h(Z_1), \cdots, X_\tau, h(Z_\tau)$. We let L be the number of mistakes made by the learner. By Theorem 10, $L \leq \mathrm{L}_{\mathcal{U}}(\mathcal{H})$. Furthermore, we define $i_1, \cdots, i_L$ to be the rounds at which the learner makes a mistake. Now when we run $\hat{Y}_t(i_1, \cdots, i_L)$ on $Z_1, \cdots, Z_t$, we will always predict according to h because on rounds that the realizable learner makes a mistake we are flipping its prediction. Note that by keeping $\hat{Y}_t = Y_t = h(Z_t)$, the update rules (e.g., the version space) in $\hat{Y}_t(i_1, \cdots, i_L)$ is the same as the realizable learner (a more formal proof is by a simple induction). ■

Theorem 15 *For any binary hypothesis class $\mathcal{H}$ with finite $\mathrm{L}_{\mathcal{U}}(\mathcal{H})$, there exists a learning algorithm that achieves expected regret $\mathcal{O}(\sqrt{\mathrm{L}_{\mathcal{U}}(\mathcal{H}) T \log(T)})$ in the agnostic robust online learning problem.*

Proof *Consider all possible experts for $L \in \{0, \cdots, \mathrm{L}_{\mathcal{U}}(\mathcal{H})\}$ and all indices $i_1, \cdots, i_L \in [T]$. Thus, the total number of experts is $N = \sum_{L=0}^{\mathrm{L}_{\mathcal{U}}(\mathcal{H})} \binom{T}{L} \leq T^{\mathrm{L}_{\mathcal{U}}}$. By Lemma 14, for any sequence chosen by the adversary, for any $h \in \mathcal{H}$, there exists an expert whose behavior, and thus whose number of mistakes, is the same as h. This is specifically true for the hypothesis that achieves the optimal number of mistakes on the sequence. Thus, the regret of a learner w.r.t. the best expert is larger than its regret w.r.t. the best hypothesis in the class. Thus, it suffices to bound the former. To achieve this goal we can use the algorithm for the problem of learning with expert advice which achieves the expected regret of $\mathcal{O}(\sqrt{\log(N) T}) = \mathcal{O}(\sqrt{\mathrm{L}_{\mathcal{U}}(\mathcal{H}) T \log(T)})$ (see Appendix A for a short introduction to prediction with expert advice).* ■

We also have the following lower bound on the regret of any learner. We omit the proof here as it is a simple generalization of the proof for the classic agnostic online learning (Ben-David et al., 2009, Lemma 14).

Theorem 16 *For any binary hypothesis class $\mathcal{H}$, no learner can achieve (expected) regret smaller than $\Omega(\sqrt{\mathrm{L}_{\mathcal{U}}(\mathcal{H}) T})$ in the agnostic robust online learning.*

Algorithm 3 Expert $\hat{Y}_t(i_1, \cdots, i_L)$

Input: Hypothesis class $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$, perturbation set $\mathcal{U}$, orientation learner $\eta : (\mathcal{X}_{\mathcal{U}}^2 \times \mathcal{Y})^* \times \mathcal{X}^2 \to \{0, 1\}$, indices $i_1, \cdots, i_L \in [T]$

Initialize $\mathcal{V} \leftarrow \mathcal{H}$, $f(.,.) \leftarrow \eta((.,.))$, $\tau \leftarrow 1$

for $t \geq 1$ **do**
- Receive Z_t
- **for** $y \in \mathcal{Y}$ **do**
  - Define $P_y := \{x \in \mathcal{X} : Z_t \in \mathcal{U}(x) \wedge \mathcal{V}_{x,y}^{\mathcal{U}} \neq \emptyset\}$
- **end**
- **if** $\exists y \in \mathcal{Y}, \exists x_y \in P_y, \forall x_{1-y} \in P_{1-y} : f(x_0, x_1) = y$ **then**
  - $\tilde{Y}_t \leftarrow y$
- **end**
- **else**
  - $\tilde{Y}_t \leftarrow 1$
- **end**
- **if** $t \in \{i_1, \cdots, i_L\}$ **then**
  - $\hat{Y}_t \leftarrow 1 - \tilde{Y}_t$
- **end**
- **else**
  - $\hat{Y}_t \leftarrow \tilde{Y}_t$
- **end**
- Receive X_t
- **if** $t \in \{i_1, \cdots, i_L\}$ **then**
  - Let $X_\tau^{\hat{Y}_t} \leftarrow X_t$
  - Let $X_\tau^{1-\hat{Y}_t}$ be an arbitrary $x \in P_{1-\hat{Y}_t}$ such that $f(X_\tau^0, X_\tau^1) = 1 - \hat{Y}_t$
  - Let $\xi_\tau \leftarrow ((X_\tau^0, X_\tau^1), \hat{Y}_t)$
  - Update $f(.,.) \leftarrow \eta(\xi_1, \ldots, \xi_\tau, (.,.))$
  - Update $\tau \leftarrow \tau + 1$
- **end**
- $\mathcal{V} \leftarrow \mathcal{V}_{X_t, \hat{Y}_t}^{\mathcal{U}}$

end

6. Robust online learning with uncertain perturbation sets

In previous sections we assumed the learner knows $\mathcal{U}$. In this section we weaken this assumption so that the learner does not fully know $\mathcal{U}$ but has some prior knowledge about it. Formally, we assume the true perturbation function $\mathcal{U}^*$ belongs to some **finite** $\mathcal{G} \subseteq (2^{\mathcal{X}})^{\mathcal{X}}$, which is known to the learner. Here we study this scenario for both realizable and agnostic learning of binary classes in finite number of rounds T. The overall idea is to run the algorithms developed in the previous sections for all possible perturbation functions in $\mathcal{G}$ and then consider them as experts that we can use to make predictions. We are now ready to state our results.

Theorem 17 *For any hypothesis class $\mathcal{H}$ with $L^* := \max_{\mathcal{U}\in\mathcal{G}} \mathrm{L}_{\mathcal{U}}(\mathcal{H}) < \infty$ and for any input sequence that is $\mathcal{U}^*-$robustly realizable by $\mathcal{H}$ for some $\mathcal{U}^* \in \mathcal{G}$, the optimal expected number of mistakes is upper bounded by $L^* + \mathcal{O}(\sqrt{L^* \log(|\mathcal{G}|)} + \log(|\mathcal{G}|))$.*

Proof *For each $\mathcal{U} \in \mathcal{G}$, define expert $\mathcal{A}_{\mathcal{U}}$ that predicts according to $\mathrm{SOA}_{\mathrm{OG}}$ assuming $\mathcal{U}$ is the true perturbation function. It is possible that for some of the functions, the version space becomes empty. In that case, the expert will predict 1. By Theorem 10, we know $\mathcal{A}_{\mathcal{U}^*}$ will make at most $\mathrm{L}_{\mathcal{U}^*}(\mathcal{H}) \leq L^*$ mistakes. The results follow by using the algorithm for prediction with advice with $N = |\mathcal{G}|$ experts guaranteed by Theorem 21.* ■

The mistake bound of Theorem 17 could be too loose in cases where $L_{\mathcal{U}^*}(\mathcal{H})$ is much smaller than $L^* = \max_{\mathcal{U}\in\mathcal{G}} \mathrm{L}_{\mathcal{U}}(\mathcal{H})$. In fact, L^* could be infinite. Can we still get a bound on the number of mistakes? Our next result answers this question positively.

Theorem 18 *For any $\mathcal{H}$ and for any sequence that is $\mathcal{U}^*-$robustly realizable by $\mathcal{H}$ for some $\mathcal{U}^* \in \mathcal{G}$, the optimal number of mistakes is upper bounded by $(\mathrm{L}_{\mathcal{U}^*}(\mathcal{H}) + 1)\log(|\mathcal{G}|)$.*

Proof *Consider the experts $\mathcal{A}_{\mathcal{U}}$ as defined in the proof of Theorem 17. The learner operates in multiple phases. In each phase the learner starts with the set of all experts and predicts with the majority label at each round. After each round the learner removes the experts that made a mistake. Each phase continues until the set of experts becomes empty, after which the next phase starts with all experts. In each phase, the learner makes at most $\log(|\mathcal{G}|)$ mistakes because each mistake reduces the number of experts by half. The expert $\mathcal{A}_{\mathcal{U}^*}$ makes at most $\mathrm{L}_{\mathcal{U}^*}$ mistakes, thus the number of phases is at most $\mathrm{L}_{\mathcal{U}^*}(\mathcal{H}) + 1$ since in that phase there will be an expert that does not make a mistake and will not get removed. Thus, the total number of mistakes is bounded by $(\mathrm{L}_{\mathcal{U}^*}(\mathcal{H}) + 1)\log(|\mathcal{G}|)$.* ■

We can also prove regret bounds in the agnostic setting similar to Theorem 17.

Theorem 19 *Any class $\mathcal{H}$ with $L^* := \max_{\mathcal{U}\in\mathcal{G}} \mathrm{L}_{\mathcal{U}}(\mathcal{H}) < \infty$ can be agnostically $\mathcal{U}^*-$robustly learned with expected regret bounded by $\mathcal{O}(\sqrt{T(L^* \log(T) + \log(|\mathcal{G}|))})$.*

Proof *For each $\mathcal{U} \in \mathcal{G}$ and for each $L \leq L^*$ and $i_1, \cdots, i_L \leq L$, define expert $\mathcal{A}_{\mathcal{U}}(i_1, \cdots, i_L)$ that predicts according to the strategy $\hat{Y}_t(i_1, \cdots, i_L)$ assuming the perturbation function is $\mathcal{U}$. Again, if the version space become empty the expert will predict 1. By Lemma 14, we know for input sequence and for any $h \in \mathcal{H}$, there exists $L \leq \mathrm{L}_{\mathcal{U}^*}(\mathcal{H}) \leq L^*$ and $i_1, \cdots, i_L$ such that $\mathcal{A}_{\mathcal{U}^*}(i_1, \cdots, i_L)$ predicts the same as h on that sequence. Thus, the regret of any learner with respect to these experts is bigger than the regret w.r.t. the best hypothesis in the class. Finally, the results follows by Theorem 20 with $N \leq T^L|\mathcal{G}|$.* ■

7. Conclusion

In this work, we initiated the study of robust online learning. We formulate this learning problem similar to the classic online learning framework of Littlestone (1988). We also introduce a new dimension and show it characterize robust online learnability and proved mistake and regret bounds that are controlled by our dimension. Unlike the dimension that characterize robust PAC learnability

(Montasser et al., 2022), our dimension is simple and resembles the Littlestone dimension. We also studied a more general case where the learner only knows that the perturbation function belongs to a finite class. We prove upper bounds for both realizable and agnostic case that depend logarithmically on the cardinality of the class.

We conclude the paper with some questions for future work.

- Here we assumed the learner either exactly knows $\mathcal{U}$ or knows it belongs to a finite class. What if the class that contains $\mathcal{U}$ is infinite, but has some structure? What about the case where we do not know anything about $\mathcal{U}$ but have access to some oracle? See for example Montasser et al. (2021) and Lechner et al. (2023) who studied this question in the PAC setting.
- In our setup, the learner receives the clean input X_t. Can we remove or weaken this assumption and still be able to learn?
- We also assume full feedback about the true label. What characterizes learnability in case of partial feedback (bandit setting)? See for example Daniely and Helbertal (2013), Raman et al. (2024a), and Raman et al. (2024b).
- Our lower and upper bound had a multiplicative difference of $\sqrt{\log(T)}$. Is it possible to close this gap? The same question has been answered for the classic online learning (Alon et al., 2021).
- Here we focused on classification tasks. Can we extend our results to regression task? Robust PAC regression has been studied by Attias and Hanneke (2023).

References

Noga Alon, Omri Ben-Eliezer, Yuval Dagan, Shay Moran, Moni Naor, and Eylon Yogev. Adversarial laws of large numbers and optimal regret in online classification. In *Proceedings of the 53rd annual ACM SIGACT symposium on theory of computing*, 2021.

Idan Attias and Steve Hanneke. Adversarially robust PAC learnability of real-valued functions. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*. PMLR, 2023.

Shai Ben-David, Dávid Pál, and Shai Shalev-Shwartz. Agnostic online learning. In *COLT*, 2009.

Nicolo Cesa-Bianchi and Gabor Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.

Anirban Chakraborty, Manaar Alam, Vishal Dey, Anupam Chattopadhyay, and Debdeep Mukhopadhyay. A survey on adversarial attacks and defences. *CAAI Transactions on Intelligence Technology*, 2021.

Amit Daniely and Tom Helbertal. The price of bandit information in multiclass online classification. In *Conference on Learning Theory*. PMLR, 2013.

Amit Daniely, Sivan Sabato, Shai Ben-David, and Shai Shalev-Shwartz. Multiclass learnability and the erm principle. *J. Mach. Learn. Res.*, 2015.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Steve Hanneke, Shay Moran, Vinod Raman, Unique Subedi, and Ambuj Tewari. Multiclass online learning and uniform convergence. In *The Thirty Sixth Annual Conference on Learning Theory*. PMLR, 2023.

Tosca Lechner, Vinayak Pathak, and Ruth Urner. Adversarially robust learning with uncertain perturbation sets. *Advances in Neural Information Processing Systems*, 2023.

Nick Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine learning*, 1988.

Omar Montasser, Steve Hanneke, and Nathan Srebro. Vc classes are adversarially robustly learnable, but only improperly. In Alina Beygelzimer and Daniel Hsu, editors, *Proceedings of the Thirty-Second Conference on Learning Theory*. PMLR, 2019.

Omar Montasser, Steve Hanneke, and Nathan Srebro. Adversarially robust learning with unknown perturbation sets. In *Conference on Learning Theory*. PMLR, 2021.

Omar Montasser, Steve Hanneke, and Nati Srebro. Adversarially robust learning: A generic minimax optimal learner and characterization. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems NeurIPS*, 2022.

Ananth Raman, Vinod Raman, Unique Subedi, Idan Mehalel, and Ambuj Tewari. Multiclass online learnability under bandit feedback. In *International Conference on Algorithmic Learning Theory*. PMLR, 2024a.

Vinod Raman, Unique Subedi, Ananth Raman, and Ambuj Tewari. Apple tasting: Combinatorial dimensions and minimax rates. In *The Thirty Seventh Annual Conference on Learning Theory*. PMLR, 2024b.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.

Appendix A. Prediction with expert advice

Prediction with expert advice is an iterative interaction between a learner and a (potentially adversarial) environment. Here the learner has access to N experts who will each at every round make a binary prediction. The learner the use this information to make a prediction of its own. The learner then incurs a loss based on the true label and will also receive the loss incurred by each expert. The goal of the learner is to minimize the total loss it incurs compared to the total loss that the best expert incurs in the hindsight.

Theorem 20 (Cesa-Bianchi and Lugosi (2006), Theorem 2.2) *Let N be the number of experts and let T be the number of rounds. Let $l_t^i \in [0,1]$ be the loss that the expert i incurs at round t. Then there is a random algorithm whose losses l_t, satisfy:*

$$\mathbb{E}[\sum_{t=1}^{T} l_t] - \min_{i\in[N]} \sum_{t=1}^{T} l_t^i \leq \mathcal{O}(\sqrt{T\log(N)})$$

Theorem 21 (Cesa-Bianchi and Lugosi (2006), Corollary 2.4) *Consider the setting of Theorem 20. If we further assume that $\min_{i\in[N]} \sum_{t=1}^{T} l_t^i \leq L^*$, then there is a random algorithm whose losses l_t, satisfy:*

$$\mathbb{E}[\sum_{t=1}^{T} l_t] \leq L^* + \mathcal{O}(\sqrt{L^*\log(N)} + \log(N))$$

Note that the algorithms in the above theorems are the same algorithm with different values for a parameter. We refer the readers to Cesa-Bianchi and Lugosi (2006) for the algorithm and the proof as we only use the algorithm as a black box. However, what's crucial here is that this guarantee holds for any set of experts. Specifically, the experts do not need to be fixed and could adapt to the adversary, which is the case for us as the experts we create use the clean inputs in the previous rounds.