Technical Report
TTIC-TR-2008-2

October 2008

Agnostic Online learnability

Shai Shalev-Shwartz
Toyota Technological Institute—Chicago
shai@tti-c.org

ABSTRACT

We study a fundamental question. What classes of hypotheses are learnable in the online learning model? The analogous question in the PAC learning model [12] was addressed by Vapnik and others [13], who showed that the VC dimension characterizes the learnability of a hypothesis class. In his influential work, Littlestone [9] studied the online learnability of hypothesis classes, but only in the realizable case, namely, assuming that there exists a hypothesis in the class that perfectly explains the entire data. In this paper we study the online learnability in the agnostic case, namely, no hypothesis perfectly predicts the entire data, and our goal is to minimize regret. We first present an impossibility result, discovered by Cover in the context of universal prediction of individual sequences, which implies that even a class whose Littlestone's dimension is only 1, is not learnable in the agnostic online learning model. We then overcome the impossibility result by allowing randomized predictions, and show that in this case Littlestone's dimension does capture the learnability of hypotheses classes in the agnostic online learning model.

1 Introduction

The goal of this paper is to give a combinatorial characterization of hypothesis classes that are learnable in the online model. In the online learning model, a learning algorithm observes instances in a sequential manner. After each observation, the algorithm attempts to predict the label associated with the instance. For example, the instance can be a vector of barometric features and the learner should predict if it's going to rain tomorrow. Once the algorithm has made a prediction, it is told whether the prediction was correct (e.g. it was rainy today) and then uses this information to improve its prediction mechanism. The goal of the learning algorithm is simply to make few mistakes.

In the online learning model discussed in this paper, we make no statistical assumptions regarding the origin of the sequence of examples. We allow the sequence to be deterministic, stochastic, or even adversarially adaptive to our own behavior (as happens to be the case, e.g., in spam email filtering). Clearly, learning is hopeless if there is no correlation between past and present examples. Classic statistical theory of sequential prediction therefore enforces strong assumptions on the statistical properties of the input sequence (for example, it must form a stationary stochastic process). Since we make no statistical assumptions, we must capture the correlation between past and present examples in a different way.

One way to do this is to assume that all labels in the sequence of examples are determined by a fixed, yet unknown, hypothesis. The learner knows that the hypothesis is taken from a predefined hypothesis class and his task reduces to pinpointing the target hypothesis that generates the labels. We say that a hypothesis class is online learnable if there exists an online algorithm that makes a finite number of prediction mistakes before pinpointing the target hypothesis. More precisely, the class is online learnable if there exists an online algorithm which makes at most $M < \infty$ mistakes on any realizable sequence (i.e. a sequence in which the labels are determined by some hypothesis from the class). A natural question is therefore which hypothesis classes are online learnable. Littlestone [9] studied this question and provided a powerful result. He defined a combinatorial measure, which we call Littlestone's dimension, that characterizes online learnability.

Despite the elegance of Littlestone's theory, it received little attention by online learning researchers. This might be contributed to the fact that the realizable assumption is rather strong. In recent years, much attention has been given to the unrealizable case, in which we assume nothing about the "true" labeling function. Following the PAC learning literature [8, 7], we call this model "agnostic online learning". In PAC learning, the VC dimension characterizes the learnability of hypothesis classes both in the realizable and agnostic cases.

In this paper we study online learnability in the agnostic case. Since in this

case even the optimal hypothesis in the hypothesis class might make many prediction mistakes, we cannot hope to have a finite bound on the number of mistakes the learner makes. Instead, we analyze the performance of the learner using the notion of *regret*. The learner's regret is the difference between the learner's number of mistakes and the number of mistakes of the optimal hypothesis. This is termed 'regret' since it measures how 'sorry' the learner is, in retrospect, not to have followed the predictions of the optimal hypothesis. We first present a negative result, showing that even a trivial hypothesis class, whose Littlestone's dimension is 1, is not agnostic online learnable. Then, we show that this negative result can be circumvented by allowing the learner to make randomized predictions. In this case, we prove that Littlestone's dimension characterizes online learnability in the agnostic case as well. Our proof is constructive – we present an online algorithm whose expected regret vanishes, and the rate of convergence depends on the Littlestone's dimension.

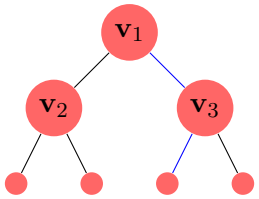
Notation We denote by $\mathcal{X}$ the set of instances and by $\mathbf{x}_t$ the instance the algorithm receives on round t. The associated label is denoted by $y_t \in \{-1, +1\}$. We use $\mathcal{H}$ to denote a hypothesis class, namely, each $h \in \mathcal{H}$ is a mapping from $\mathcal{X}$ to $\{+1, -1\}$. For a predicate π we denote the indicator function by $\mathbb{1}[\![\pi]\!]$. For example, if $h(\mathbf{x}_t) \neq y_t$ then $\mathbb{1}[\![h(\mathbf{x}_t) \neq y_t]\!] = 1$ and otherwise $\mathbb{1}[\![h(\mathbf{x}_t) \neq y_t]\!] = 0$. The set of integers $\{1, \ldots, n\}$ is denoted by $[n]$.

2 The Realizable Case

In this section we study online learnability in the realizable case and formally define Littlestone's dimension. The content of this section is adopted from [9].

Recall that in the realizable case we assume that there exists $h \in \mathcal{H}$ such that for all t, $y_t = h(\mathbf{x}_t)$. However, we make no further statistical assumptions regarding the origin of the instances or the choice of h. The online algorithm should be able to compete even against sequence of examples that are adversarially adaptive to its own behavior. We therefore analyze the performance of the online algorithm in a worst case manner.

Definition 1 (Mistake bound) *Let $\mathcal{H}$ be a hypothesis class of functions from $\mathcal{X}$ to $\{\pm 1\}$ and A be an online algorithm. We say that M is a mistake bound for A, if for any hypothesis $h^\star \in \mathcal{H}$ and any sequence of examples $(\mathbf{x}_1, h^\star(\mathbf{x}_1)), \ldots, (\mathbf{x}_n, h^\star(\mathbf{x}_n))$, the online algorithm A does not make more than M prediction mistakes when running on the sequence of examples.*



	h_1	h_2	h_3	h_4
$\mathbf{v}_1$	-1	-1	1	1
$\mathbf{v}_2$	-1	1	?	?
$\mathbf{v}_3$	?	?	-1	1

Figure 1: An illustration of a shattered tree of depth 2. The blue path corresponds to the sequence of examples $((\mathbf{v}_1, 1), (\mathbf{v}_3, -1))$, which can also be written as $((\mathbf{v}_1, h_3(\mathbf{v}_1)), (\mathbf{v}_3, h_3(\mathbf{v}_3)))$. The tree is shattered by $\mathcal{H} = \{h_1, h_2, h_3, h_4\}$, where the predictions of each hypothesis in $\mathcal{H}$ on the instances $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ is given in the table (a question mark means that $h_j(\mathbf{v}_i)$ can be either 1 or -1).

We say that a hypothesis class is learnable in the online learning model if there exists an online learning algorithm with a finite mistake bound, no matter how long the sequence of examples is.

Next, we turn to describe a combinatorial measure of hypothesis classes that characterizes the best possible achievable mistake bound. This measure was proposed by Littleston and we therefore refer to it as $\mathrm{Ldim}(\mathcal{H})$.

To motivate the definition of Ldim it is convenient to view the online learning process as a game between two players: the learner vs. the environment. On round t of the game, the environment picks an instance $\mathbf{x}_t$, the learner predicts a label $\hat{y}_t \in \{+1, -1\}$, and finally the environment outputs the true label, $y_t \in \{+1, -1\}$. Suppose that the environment wants to make the learner err on the first M rounds of the game. Then, it must output $y_t = -\hat{y}_t$, and the only question is how to choose the instances $\mathbf{x}_t$ in such a way that ensures that for some $h \in \mathcal{H}$ we have $y_t = h(\mathbf{x}_t)$ for all $t \in [M]$.

It makes sense to assume that the environment should pick $\mathbf{x}_t$ based on the previous predictions of the learner, $\hat{y}_1, \ldots, \hat{y}_{t-1}$. Since in our case we have $y_t = -\hat{y}_t$ we can also say that $\mathbf{x}_t$ is a function of $y_1, \ldots, y_{t-1}$. We can represent this dependence using a complete binary tree of depth M (we define the depth of the tree as the number of edges in a path from the root to a leaf). We have $2^{M+1} - 1$ nodes in such a tree, and we attach an instance to each node. Let $\mathbf{v}_1, \ldots, \mathbf{v}_{2^{M+1}-1}$ be these instances. We start from the root of the tree, and set $\mathbf{x}_1 = \mathbf{v}_1$. At round t, we set $\mathbf{x}_t = \mathbf{v}_{i_t}$ where i_t is the current node. At the end of round t, we go to the left child of i_t if $y_t = -1$ or to the right child if $y_t = 1$. That is, $i_{t+1} = 2i_t + \frac{y_t+1}{2}$. Unraveling the recursion we obtain $i_t = 2^{t-1} + \sum_{j=1}^{t-1} \frac{y_j+1}{2} 2^{t-1-j}$.

The above strategy for the environment succeeds only if for any $(y_1, \ldots, y_M)$

there exists $h \in \mathcal{H}$ such that $y_t = h(\mathbf{x}_t)$ for all $t \in [M]$. This leads to the following definition.

Definition 2 (Shattered tree) *A shattered tree of depth d is a sequence of instances $\mathbf{v}_1, \ldots, \mathbf{v}_{2^d-1}$ in $\mathcal{X}$ such that for all labeling $(y_1, \ldots, y_d) \in \{\pm 1\}^d$ there exists $h \in \mathcal{H}$ such that for all $t \in [d]$ we have $h(\mathbf{v}_{i_t}) = y_t$ where $i_t = 2^{t-1} + \sum_{j=1}^{t-1} \frac{y_j+1}{2} 2^{t-1-j}$.*

An illustration of a shattered tree of depth 2 is given in Figure 1.

Definition 3 (Littlestone's dimension (Ldim)) *$\mathrm{Ldim}(\mathcal{H})$ is the maximal integer M such that there exist a shattered tree of depth M.*

The definition of Ldim and the discussion above immediately imply the following:

Lemma 1 *If $\mathrm{Ldim}(\mathcal{H}) = M$ then no algorithm can have a mistake bound strictly smaller than M.*

Proof Let $\mathbf{v}_1, \ldots, \mathbf{v}_{2^M-1}$ be a shattered tree (such a tree must exist since $\mathrm{Ldim}(\mathcal{H}) = M$). If the environment sets $\mathbf{x}_t = \mathbf{v}_{i_t}$ and $y_t = -\hat{y}_t$ for all $t \in [M]$, then the learner makes M mistakes while the definition of a shattered tree implies that there exists a hypothesis $h \in \mathcal{H}$ such that $y_t = h(\mathbf{x}_t)$ for all t. ■

We have shown that $\mathrm{Ldim}(\mathcal{H})$ lower bounds the mistake bound of any algorithm. Interestingly, there is a standard algorithm whose mistake bound matches this lower bound. Thus, $\mathrm{Ldim}(\mathcal{H})$ is the exact characterization of the learnability of $\mathcal{H}$ in the online model.

Algorithm 1 Standard Optimal Algorithm (SOA)

INPUT: A hypothesis class $\mathcal{H}$
INITIALIZE: $V_1 = \mathcal{H}$
FOR $t = 1, 2, \ldots$
  Receive $\mathbf{x}_t$
  For $r \in \{\pm 1\}$ let $V_t^{(r)} = \{h \in V_t : h(\mathbf{x}_t) = r\}$
  Predict $\hat{y}_t = \arg\max_r \mathrm{Ldim}(V_t^{(r)})$
  Receive true answer y_t
  IF $y_t \neq \hat{y}_t$
    Update $V_{t+1} = V_t^{(y_t)}$
  ELSE
    Update $V_{t+1} = V_t$

The following lemma formally establishes the optimality of SOA.

Lemma 2 *The SOA algorithm enjoys the mistake bound* $M = \mathrm{Ldim}(\mathcal{H})$.

Proof It suffices to prove that whenever the algorithm makes a prediction mistake we have $\mathrm{Ldim}(V_{t+1}) \leq \mathrm{Ldim}(V_t) - 1$. We prove this claim by assuming the contrary, that is, $\mathrm{Ldim}(V_{t+1}) = \mathrm{Ldim}(V_t)$. If this holds true, then the definition of $\hat{y}_t$ implies that $\mathrm{Ldim}(V_t^{(r)}) = \mathrm{Ldim}(V_t)$ for both $r = 1$ and $r = -1$. But, in this case we can construct a shattered tree of depth $\mathrm{Ldim}(V_t) + 1$ for the class V_t, which leads to the desired contradiction. ■

Combining Lemma 2 and Lemma 1 we obtain:

Corollary 1 *Let $\mathcal{H}$ be a hypothesis class. Then, SOA (Algorithm 1) enjoys the mistake bound* $\mathrm{Ldim}(\mathcal{H})$ *and no other algorithm can have a mistake bound strictly smaller than* $\mathrm{Ldim}(\mathcal{H})$.

3 Agnostic Online Learnability

In the previous section we have shown that Littlestone's dimension exactly characterizes online learnability in the realizable case. However, the realizable assumption is rather strong. In this section we consider the (more realistic) agnostic case. In the agnostic case, our goal is to minimize the difference between the learner's number of mistakes and the number of mistakes of the optimal hypothesis in $\mathcal{H}$. This is termed 'regret' since it measures how 'sorry' the learner is, in retrospect, not to have followed the predictions of the optimal hypothesis. We say that a class $\mathcal{H}$ is agnostic online learnable if there exists an online algorithm that has a regret bound of $o(n)$, where n is the length of the sequence of examples. In other words, a regret bound of $o(n)$ means that the difference between the error rate of the learner and the error rate of the optimal hypothesis in $\mathcal{H}$ converges to 0 when $n \to \infty$.

As before, we are interested in a combinatorial measure that determines the learnability of hypothesis classes in the agnostic case. A natural candidate is the Littlestone's dimension. We start this section with a negative result, showing a class with $\mathrm{Ldim}(\mathcal{H}) = 1$ that has a non-vanishing regret. Next, we slightly change our model, by allowing the learner to randomized his predictions and analyze the expected regret. As a warm-up, we show that finite hypotheses classes are learnable, if randomized predictions are allowed. Finally, we present our main result, showing that any class with $\mathrm{Ldim}(\mathcal{H}) < \infty$ is agnostic online learnable, if randomized predictions are allowed. In particular, we present a constructive online algorithm that enjoys the expected regret bound of $\mathrm{Ldim}(\mathcal{H}) + \sqrt{\mathrm{Ldim}(\mathcal{H})\, n\, \log(n)}$.

3.1 Cover's impossibility result

We present a negative result, showing a hypothesis class with $\mathrm{Ldim}(\mathcal{H}) = 1$ that is not agnostic online learnable. This impossibility result was given in [3].

Theorem 1 (Cover's impossibility result) *There exists a hypothesis class $\mathcal{H}$ with $\mathrm{Ldim}(\mathcal{H}) = 1$, such that for any online algorithm, there exists a sequence of examples $(x_1, y_1), \ldots, (x_n, y_n)$ on which the online algorithm makes n mistakes, while the optimal hypothesis in $\mathcal{H}$ makes at most $n/2$ mistakes. Namely, the regret is at least $n/2$.*

Proof Consider the class of two constant functions $\mathcal{H} = \{h_1(\mathbf{x}) = 1, h_2(\mathbf{x}) = -1\}$. It is easy to verify that $\mathrm{Ldim}(\mathcal{H}) = 1$. For any sequence of n examples, the number of prediction mistakes of the constant prediction sign $(\sum_{t=1}^{n} y_t)$ is at most $n/2$. Therefore, on any sequence of n examples, the optimal hypothesis in $\mathcal{H}$ makes at most $n/2$ mistakes. Let A be an arbitrary online algorithm. We present A the following sequence of examples. On round t we present an arbitrary $\mathbf{x}_t$, receive the prediction of the algorithm $\hat{y}_t$, and then provide the output $y_t = -\hat{y}_t$. Therefore, the number of mistakes the algorithm makes on a sequence of length n is exactly n. This concludes our proof. ■

3.2 Randomized Predictions

The negative result given in the previous subsection shows that even trivial hypotheses classes are not agnostic online learnable. Early results in game theory [11, 1, 6] and information theory [4, 5] sidestepped Cover's impossibility result by allowing the learner to randomized his predictions.

Formally, on round t the learner defines a probability $\hat{p}_t \in [0, 1]$ over $\{+1, -1\}$, and the prediction $\hat{y}_t$ is a random variable with $\mathbb{P}[\hat{y}_t = 1] = \hat{p}_t$. We analyze the expected regret of the algorithm and define:

Definition 4 (Expected regret) *The expected regret of an online algorithm on a sequence $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$ with respect to a hypothesis set $\mathcal{H}$ is:*

$$\sum_{t=1}^{n} \mathbb{E}[\mathbb{1}[\![\hat{y}_t \neq y_t]\!]] - \min_{h \in \mathcal{H}} \sum_{t=1}^{n} \mathbb{1}[\![h(\mathbf{x}_t) \neq y_t]\!] \ ,$$

where

$$\mathbb{E}[\mathbb{1}[\![\hat{y}_t \neq y_t]\!]] = \begin{cases} 1 - \hat{p}_t & \text{if } y_t = 1 \\ \hat{p}_t & \text{if } y_t = -1 \end{cases} \quad = \quad \frac{1 + y_t}{2} - y_t \hat{p}_t \ .$$

As before, we say that $\mathcal{H}$ is agnostic online learnable with randomized predictions if there exists an algorithm with a sub-linear expected regret.

Next, we show how to circumvent Cover's impossibility results by allowing the online learner to make randomized predictions.

3.3 Finite Hypothesis Classes and Experts Algorithms

Let $\mathcal{H}$ be a finite hypothesis classes and denote $\mathcal{H} = \{h_1, \ldots, h_d\}$. We can think on the hypotheses in $\mathcal{H}$ as "experts", and the goal of the online learning algorithm is to track the optimal expert.

One way to do this is by using the weighted majority algorithm [10]. The version of the algorithm we give here, as well as the regret bound, is based on [2].

Algorithm 2 Learning with Expert Advice

INPUT: Number of experts d ; Number of rounds n

INITIALIZE: $\eta = \sqrt{2\ln(d)/n}$; $\forall i \in [d],\ M_i^0 = 0$

FOR $t = 1, 2, \ldots, n$

Receive experts advice $(f_1^t, \ldots, f_d^t) \in \{\pm 1\}^d$

Define $w_i^{t-1} = \exp(-\eta M_i^{t-1}) / (\sum_j \exp(-\eta M_j^{t-1}))$

Define $\hat{p}_t = \sum_{i:f_i^t=1} w_i^{t-1}$

Predict $\hat{y}_t = 1$ with probability $\hat{p}_t$

Receive true answer y_t

Update: $M_i^t = M_i^{t-1} + \mathbb{1}[\![f_i^t \neq y_t]\!]$

The algorithm maintains the number of prediction mistakes each expert made so far, M_i^{t-1}, and assign a probability weight to each expert accordingly. Then, the learner sets $\hat{p}_t$ to be the total mass of the experts which predict 1. The definition of $\hat{y}_t$ clearly implies that

$$\mathbb{E}[\mathbb{1}[\![\hat{y}_t \neq y_t]\!]] = \sum_{i=1}^{d} w_i^{t-1} \mathbb{1}[\![f_i^t \neq y_t]\!] \,. \tag{1}$$

That is, the probability to make a mistake equals to the expected error of experts, where expectation is with respect to the probability vector $\mathbf{w}^t$.

The following theorem analyzes the expected regret of the algorithm.

Theorem 2 *Algorithm 2 satisfies*

$$\sum_{t=1}^{n} \mathbb{E}[\mathbb{1}[\![\hat{y}_t \neq y_t]\!]] - \min_{i \in [d]} \sum_{t=1}^{n} \mathbb{1}[\![f_i^t \neq y_t]\!] \;\leq\; \sqrt{0.5\, \ln(d)\, n}\,.$$

Proof Define $Z_t = \sum_i e^{-\eta M_i^t}$. We have,

$$\ln \frac{Z_t}{Z_{t-1}} = \ln \frac{\sum_i e^{-\eta M_i^{t-1}} e^{-\eta \mathbb{1}[\![f_i^t \neq y_t]\!]}}{Z_{t-1}} = \ln \sum_i w_i^{t-1} e^{-\eta \mathbb{1}[\![f_i^t \neq y_t]\!]}\,.$$

Note that $\mathbf{w}^t$ is a probability vector and $\mathbb{1}[\![f_i^t \neq y_t]\!] \in [0,1]$. Therefore, we can apply Hoeffding's inequality (see for example [2], Lemma 2.2) on the right-hand side of the above to get

$$\ln \frac{Z_t}{Z_{t-1}} \;\leq\; -\eta \sum_i w_i^{t-1} \mathbb{1}[\![f_i^t \neq y_t]\!] + \frac{\eta^2}{8} \;=\; -\eta \mathbb{E}[\mathbb{1}[\![\hat{y}_t \neq y_t]\!]] + \frac{\eta^2}{8}\,,$$

where the last equality follows from Eq. (1). Summing the above inequality over t we get

$$\ln(Z_n) - \ln(Z_0) = \sum_{t=1}^{n} \ln \frac{Z_t}{Z_{t-1}} \leq -\eta \sum_{t=1}^{n} \mathbb{E}[\mathbb{1}[\![\hat{y}_t \neq y_t]\!]] + \frac{n\,\eta^2}{8}\,. \qquad (2)$$

Next, we note that $\ln(Z_0) = \ln(d)$ and that

$$\ln Z_n = \ln \left(\sum_i e^{-\eta M_i^n} \right) \geq \ln \left(\max_i e^{-\eta M_i^n} \right) = -\eta \min_i M_i^n\,.$$

Combining the above with Eq. (2) and rearranging terms we obtain that

$$\sum_t \mathbb{E}[\mathbb{1}[\![\hat{y}_t \neq y_t]\!]] - \min_i M_i^n \;\leq\; \frac{\ln(d)}{\eta} + \frac{\eta\, n}{8}\,.$$

Setting $\eta = \sqrt{8 \ln(d)/n}$ and rearranging terms we conclude our proof. ■

Based on the above, we can easily define an agnostic online learning algorithm for finite hypothesis classes as follows.

Algorithm 3 Agnostic online learning of a finite hypothesis class

INPUT: Finite class $\mathcal{H} = \{h_1, \ldots, h_d\}$; Number of rounds n

LOOP: Run Algorithm 2, where on round t, set $f_i^t = h_i(\mathbf{x}_t)$

As a direct corollary of Theorem 2 we obtain:

Corollary 2 *Let $\mathcal{H}$ be a finite hypothesis class and let $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$ be an arbitrary sequence of examples. Then, Algorithm 3 satisfies*

$$\sum_{t=1}^{n} \mathbb{E}[\mathbb{1}[\![\hat{y}_t \neq y_t]\!]] - \min_{h \in \mathcal{H}} \sum_{t=1}^{n} \mathbb{1}[\![h(\mathbf{x}_t) \neq y_t]\!] \;\leq\; \sqrt{0.5 \, \ln(|\mathcal{H}|) \, n} \, .$$

In particular, the above implies that $\mathcal{H}$ is agnostic online learnable.

The next step is to characterize the learnability of infinite size classes.

3.4 Agnostic Online Learnability

In this section we present our main result. We describe an algorithm for agnostic online learnability that achieves an expected regret bound of $\mathrm{Ldim}(\mathcal{H}) + \sqrt{0.5 \, \mathrm{Ldim}(\mathcal{H}) \, n \, \log(n)}$. The immediate corollary is that Littlestone's dimension characterizes the learnability of a hypothesis class also in the agnostic case, as long as we allow randomized predictions.

As in the case of finite hypothesis classes, the main idea is to construct a set of experts and then to use the Learning-with-experts-advice algorithm. Recall that the expected regret of the Learning-with-experts-advice algorithm is $O(\sqrt{\log(d) \, n})$, where d is the number of experts. Therefore, unless $|\mathcal{H}|$ is finite, we cannot use each $h \in \mathcal{H}$ as an expert. The challenge is therefore how to define a set of experts that on one hand is not excessively large while on the other hand contains an expert that gives accurate predictions.

The basic idea is to simulate each expert by running an instance of the SOA algorithm (Algorithm 1) on a small sub-sequence of examples. Formally, let L be an integer such that $L \leq \mathrm{Ldim}(\mathcal{H})$. For each sub-sequence $1 \leq i_1 < i_2 < \ldots < i_L \leq n$, we define an expert as follows.

Algorithm 4 $\mathrm{Expert}(i_1, \ldots, i_L)$

INPUT: A hypothesis class $\mathcal{H}$;

Indices $i_1 < i_2 < \ldots < i_L$

INITIALIZE: $V_1 = \mathcal{H}$

FOR $t = 1, 2, \ldots, n$

Receive $\mathbf{x}_t$

For $r \in \{\pm 1\}$ let $V_t^{(r)} = \{h \in V_t : h(\mathbf{x}_t) = r\}$

Predict $\hat{y}_t = \mathrm{argmax}_r \, \mathrm{Ldim}(V_t^{(r)})$

Receive true answer y_t

IF $y_t \neq \hat{y}_t$ **and** $t \in \{i_1, \ldots, i_L\}$

Update $V_{t+1} = V_t^{(y_t)}$

ELSE

Update $V_{t+1} = V_t$

The following key lemma shows that there exists an expert whose performance is almost optimal.

Lemma 3 *Let $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$ be a sequence of examples and let $\mathcal{H}$ be a hypothesis class with $\mathrm{Ldim}(\mathcal{H}) < \infty$. There exists $L \leq \mathrm{Ldim}(\mathcal{H})$ and a sub-sequence $1 \leq i_1 < \ldots < i_L \leq n$, such that $\mathrm{Expert}(i_1, \ldots, i_L)$ makes at most*

$$L + \min_{h \in \mathcal{H}} \sum_{t=1}^{n} \mathbb{1}[\![h(\mathbf{x}_t) \neq y_t]\!]$$

mistakes on the sequence of examples.

Proof To simplify our notation, let $\mathcal{M}(h)$ be the number of mistakes a hypothesis h makes on the sequence of examples. Let $h^\star \in \mathcal{H}$ be an optimal hypothesis, that is $\mathcal{M}(h^\star) = \min_h \mathcal{M}(h)$. Let $j_1, \ldots, j_k$ be the set of rounds on which $h^\star$ does not err. Thus, $k = n - \mathcal{M}(h^\star)$. The sequence of examples $(x_{j_1}, y_{j_1}), \ldots, (x_{j_k}, y_{j_k})$ is realizable for $\mathcal{H}$ (since $h^\star \in \mathcal{H}$ never err on this sequence). Therefore, if we run SOA (Algorithm 1) on this sequence we have at most $\mathrm{Ldim}(\mathcal{H})$ mistakes. Choose $i_1, \ldots, i_L$ to be a sub-sequence of $j_1, \ldots, j_k$ that contains the indices of examples on which SOA errs, and note that $L \leq \mathrm{Ldim}(\mathcal{H})$. Since the predictions of SOA on $j_1, \ldots, j_k$ are exactly the same as the predictions of $\mathrm{Expert}(i_1, \ldots, i_L)$ on $j_1, \ldots, j_k$ we get that the total number of prediction

mistakes of $\mathrm{Expert}(i_1, \ldots, i_L)$ on the entire sequence is at most $L + \mathcal{M}(h^\star)$. ■

Our agnostic online learning algorithm is now an application of the Learning-with-experts-advice algorithm.

Algorithm 5 Agnostic Online Learning Algorithm

INPUT: A hypothesis class $\mathcal{H}$ with $\mathrm{Ldim}(\mathcal{H}) < \infty$; Horizon n

INITIALIZE:

For each $L \leq \mathrm{Ldim}(\mathcal{H})$

For each sub-sequence $i_1, \ldots, i_L$

Construct an expert from $(i_1, \ldots, i_L)$ as in Figure 4

LOOP: Run Algorithm 2 with the set of constructed experts

To analyze Algorithm 5 we combine Lemma 3 with the upper bound on the number of experts,

$$d = \sum_{L=0}^{\mathrm{Ldim}(\mathcal{H})} \binom{n}{L} \leq n^{\mathrm{Ldim}(\mathcal{H})}$$

to obtain the following:

Theorem 3 *Let $\mathcal{H}$ be a hypothesis class with $\mathrm{Ldim}(\mathcal{H}) < \infty$. If we run Algorithm 5 on any sequence $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$, we obtain the expected regret bound,*

$$\sum_{t=1}^{n} \mathbb{E}[\mathbb{1}[\![\hat{y}_t \neq y_t]\!]] \;\leq\; \min_{h \in \mathcal{H}} \sum_{t=1}^{n} \mathbb{1}[\![h(\mathbf{x}_t) \neq y_t]\!] + \mathrm{Ldim}(\mathcal{H}) + \sqrt{0.5\,\mathrm{Ldim}(\mathcal{H})\, n\, \ln(n)}\ .$$

In particular, the above implies that a class $\mathcal{H}$ that has a finite Littlestone dimension is agnostic online learnable.

References

[1] D. Blackwell. An analog of the minimax theorem for vector payoffs. *Pacific Journal of Mathematics*, 6(1):1–8, Spring 1956.

[2] N. Cesa-Bianchi and G. Lugosi. *Prediction, learning, and games*. Cambridge University Press, 2006.

[3] Thomas M. Cover. Behavior of sequential predictors of binary sequences. *Trans. 4th Prague Conf. Information Theory Statistical Decision Functions, Random Processes*, 1965.

[4] Thomas M. Cover and Aaron Shenhar. Compound Bayes predictors for sequences with apparent Markov structure. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-7(6):421–424, June 1977.

[5] M. Feder, N. Merhav, and M. Gutman. Universal prediction of individual sequences. *IEEE Transactions on Information Theory*, 38:1258–1270, 1992.

[6] J. Hannan. Approximation to Bayes risk in repeated play. In M. Dresher, A. W. Tucker, and P. Wolfe, editors, *Contributions to the Theory of Games*, volume III, pages 97–139. Princeton University Press, 1957.

[7] David Haussler. Probably approximately correct learning. In *Proceedings of the 8th National Conference on Artificial Intelligence*, pages 1101–1108. Morgan Kaufmann, 1990.

[8] Michael J. Kearns, Robert E. Schapire, and Linda M. Sellie. Toward efficient agnostic learning. *Machine Learning*, 17:115–141, 1994.

[9] N. Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, 2:285–318, 1988.

[10] N. Littlestone and Manfred K. Warmuth. The weighted majority algorithm. *Information and Computation*, 108:212–261, 1994.

[11] H. Robbins. Asymptotically subminimax solutions of compound statistical decision problems. In *Proceedings of the 2nd Berkeley symposium on mathematical statistics and probability*, pages 131–148, 1951.

[12] L. G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, November 1984.

[13] V. N. Vapnik. *Statistical Learning Theory*. Wiley, 1998.